Primitive Automata that are Synchronizing

Igor Rystsov **□ 0**

National Technical University of Ukraine, Kiev, Ukraine

Marek Szykuła ⊠ •

University of Wrocław, Faculty of Mathematics and Computer Science, Wrocław, Poland

Abstract

A deterministic finite (semi)automaton is primitive if its transition monoid (semigroup) acting on the set of states has no non-trivial congruences. It is synchronizing if it contains a constant map (transformation). In analogy to synchronizing groups, we study the possibility of characterizing automata that are synchronizing if primitive. We prove that the implication holds for several classes of automata. In particular, we show it for automata whose every letter induce either a permutation or a semiconstant transformation (an idempotent with one point of contraction) unless all letters are of the first type. We propose and discuss two conjectures about possible more general characterizations.

2012 ACM Subject Classification Theory of computation \rightarrow Formal languages and automata theory; Mathematics of computing \rightarrow Combinatorics

Keywords and phrases Černý conjecture, permutation group, primitive, primitivity, reset word, simple idempotent, synchronizing automaton, synchronizing word, transition monoid, transformation semigroup

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding This work was supported in part by the National Science Centre, Poland under project number 2021/41/B/ST6/03691.

1 Introduction

We consider deterministic finite semiautomata (called shortly *automata*) and the properties of their transition monoids. An automaton is *synchronizing* if it admits a *reset* word, which is a word such that after reading it, the automaton is left in one known state, regardless of the initial state. In the transition monoid of the automaton (or the semigroup acting on the set of states), this corresponds to the existence of a constant map, which is induced by a reset word.

The theory of synchronizing automata is most famous due to the Černý conjecture, which says that every synchronizing automaton with n states admits a reset word of length at most $(n-1)^2$ [13]. This longstanding open problem from 1969 motivated researchers to develop a vast number of results. The currently best general upper bound is cubic in n [23, 26, 29]. Most of the research was collected in the recent survey [33]; see also the older ones [19, 32].

Here, we consider the possibilities of relating the primitivity of the transition monoid with its synchronizability. Both these properties often appear together in the literature.

We define the primitivity in analogue to that in the context of permutation groups, which means that there is no non-trivial congruence on the set of states preserved by the action of letters. This is often useful since we can then construct a smaller *quotient* automaton, where a state represents a class in the original one. So such congruences are used to derive results, in particular, upper bounds on the length of the shortest reset words in particular cases [7, 15, 31]. There are also some bounds stated for primitive synchronizing automata [1] (primitive automata are called there *simple*) and other results relating the primitivity and the synchronizability of automata [16, 17].

A very related problem concerning permutation groups was the subject of extensive

research [2, 3, 4]. The problem was to characterize when a primitive permutation group, after adding one non-permutational transformation, results in a synchronizing semigroup. The transformations of this property have been successfully characterized. However, the whole study concerns only the case when the group is primitive, which is a strong condition from the automata point of view. In many cases, the group contained in the transition monoid of an automaton is not only non-primitive but non-transitive or even trivial (just if the automaton has no permutational letters). This question is naturally generalized to semigroups, where the non-permutational transformations also contribute to the primitiveness of the transition monoid.

Hence, our specific research question is the following: Under what additional condition(s) every primitive automaton is synchronizing?

Additionally, if, in some cases, primitivity implies synchronizability, then sometimes we could relax necessary conditions, e.g., where an automaton needs to be both primitive and synchronizing [25], or we could obtain a synchronizing automaton when needed by ensuring its primitivity instead of other properties, e.g., we could think about auxiliary constructions that need to be synchronizing such as the *induced automata* [7, 9].

1.1 Our contribution

We propose criteria that presumably are sufficient to imply the synchronizability from the primitivity of an automaton. We discuss two variants of the conjecture (weak and strong) referring to the shape of non-permutational transformations induced by the letters. We show that they cannot be (much) relaxed and provide experimental support.

Based on the literature results, we show that the implication holds in several cases. In particular, we prove that it holds for automata with permutational and semiconstant letters, which are a generalization of automata with simple idempotents [18, 24]. This class is a restricted case covered by our conjecture in the strong variant.

Note: The weak variant of our conjecture in a stronger form has been recently solved by Mikhail Volkov [34], together with several new results concerning our problem.

2 Preliminaries

An automaton \mathscr{A} is a triple (Q, Σ, δ) , where Q is a finite set of n elements called states, Σ is a finite non-empty set of letters, and $\delta \colon Q \times \Sigma$ is a totally-defined transition function. The transition function is naturally extended to words (finite sequences of elements from Σ) in Σ^* . Given a word $w \in \Sigma^*$, let $\delta(w)$ denote its induced transformation, which is a function (transformation, map) $\delta(w) \colon Q \to Q$ defined by $\delta(w)(q) = \delta(q, w)$.

For a set T of maps $Q \to Q$, the transformation monoid (Q, M) generated by T and acting on Q is denoted by $\langle T \rangle$, where M is the set of all transformations that can be obtained from maps from T by composition. For an automaton $\mathscr{A} = (Q, \Sigma, \delta)$, its associated transition monoid is $\langle \{\delta(a) \mid a \in \Sigma\} \rangle = (Q, M)$, where M is the set of all maps $\delta(w)$ for every word $w \in \Sigma^*$. In the literature, there are many names for a transformation monoid, e.g., transformation semigroup, operand [14], polygon [5]; and it is also an algebra with unary operations [27].

We use the right-hand side convention and denote by (q)f the *image* of the state $q \in Q$ under the action of $f: Q \to Q$. For a subset of states $S \subseteq Q$, the *image* of the map f we denote by $(S)f = \{(q)f \mid q \in S\}$. The cardinality |(Q)f| is the rank of f. The deficiency (or co-rank) of f is n - |(Q)f|. Maps of rank n (equivalently, of deficiency 0) are permutations (bijections on Q).

A transformation monoid (Q, M) is transitive (or strongly connected) if for every two states $s, t \in Q$, there is a transformation $f \in M$ such that (s)f = t. If a transformation monoid contains a map of rank 1, then the monoid is called *synchronizing*. A pair of different states $s, t \in Q$ are called *compressible* if there is a transformation $f \in M$ such that $|(\{s,t\})f| = 1$; then f compresses the pair. It is well known that a transformation monoid is synchronizing if and only if every pair of states is compressible [13].

We transfer the above terminology from the transition monoid to the automaton and from transformations to words and letters that induce them. Thus, a synchronizing automaton is one that admits a word of rank 1. Such a word is also a *reset* word.

2.1 Relations and Primitivity

Let (Q, M) be a transformation monoid. Then the action of M on Q naturally continues on the square $Q \times Q$ by components:

$$(s,t)f = ((s)f,(t)f), \text{ for each } f \in M, s,t \in Q.$$

For a binary relation $\rho \subseteq Q \times Q$, we put:

$$(\rho)M = \{(s,t)f \mid (s,t) \in \rho, f \in M\}.$$

A relation ρ is called *invariant* for (Q, M) if $(\rho)M = \rho$. If ρ is additionally an equivalence relation, then it is a *congruence* of (Q, M). The equivalence classes of this congruence are called *blocks*.

A relation ρ is trivial if it is the identity relation (diagonal) $\triangle_Q = \{(q,q) \mid q \in Q\}$ or the total relation (square) $\nabla_Q = Q \times Q$.

▶ **Definition 1.** A transformation monoid is primitive if it does not have any non-trivial congruence. An automaton is primitive if its transition monoid is primitive.

Letters of deficiency 0, whose induced transformation is a permutation are called permutational, and these transformations generate a permutation group acting on Q and contained in the transition monoid.

3 The Conjecture on Primitivity implying Synchronizability

Every automaton with a letter of certain deficiency and a primitive group generated by the permutational letters is synchronising [2]. This holds for deficiencies 1, 2, n-3, and n-4, and also for other deficiencies if the letter's map additionally has some properties. Hence, it is natural to use deficiency as an additional condition for our question.

Our best guess for a general conjecture is the following:

▶ Conjecture 2 (Weak variant). Every primitive automaton with permutational letters and letters of deficiency 1 is synchronizing unless all letters are permutational.

The class of these automata contains, in particular, *almost-group* automata [8], which have only one non-permutational letter of deficiency 1, and it is known that at random they are synchronizing with high probability.

First, we note that the conjecture does not hold in the reverse direction. There also exist almost-group automata that are synchronizing and non-primitive.

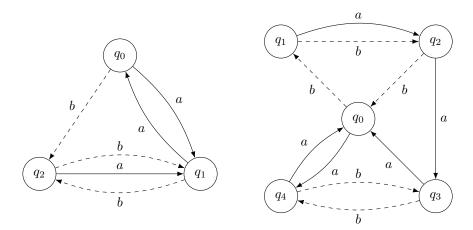


Figure 1 Two non-primitive automata that are synchronizing. **Left:** Both letters have deficiency 1. **Right:** Letter *a* has deficiency 1 and letter *b* is permutational.

▶ Example 3. Both automata from Figure 1 are strongly connected, non-primitive, and synchronizing, though they have only permutational letters and letters of deficiency 1. The automaton on the right is almost-group.

Proof. The automata are synchronizing, since the words ab and a^2ba^3 are reset respectively for the automaton on the left and on the right. The automata are non-primitive, since the equivalence relation with classes $\{q_0, q_1\}$, $\{q_2\}$ is a non-trivial congruence for the automaton on the left and the equivalence relation with classes $\{q_0\}$, $\{q_1\}$, $\{q_2\}$, $\{q_3, q_4\}$ is a non-trivial congruence for the automaton on the right.

3.1 Known Cases

By some results from the literature, we know that the conjecture holds in certain cases.

A sink state $q_0 \in Q$ is such that $(q_0)M = \{q_0\}$. An automaton is 0-transitive if it has a unique sink q_0 and (q)M = Q for all $q \in Q \setminus \{q_0\}$; in other words, there is a sink reachable from every state and the other states form one strongly connected component.

▶ Proposition 4 ([28, Proposition 5.1]). For $n \ge 3$, a primitive automaton (Q, Σ, δ) is either strongly connected or 0-transitive.

It follows that it would be enough to show the conjecture for strongly connected automata, as 0-transitive automata are synchronized in q_0 .

An automaton is called *aperiodic* if its transition monoid does not have any non-trivial subgroups; in other words, in there is no transformation that acts cyclically on some subset of states of size ≥ 2 .

From the following statement, we get that a primitive aperiodic automaton is synchronizing:

▶ Proposition 5 ([30, rephrased Lemma 6]). An aperiodic automaton (Q, Σ, δ) is synchronizing if and only if it has a state $q \in Q$ that is reachable from all the states, i.e., $q \in (p)M$ for all $p \in Q$.

Another case is automata with a prime number of states that also contains a full cyclic permutation in their transition monoid. It follows by an old result of Pin [22], stating that a

circular automaton, i.e., with a letter that induces one cycle on all the states, with a prime number of states is synchronizing whenever it has any non-permutational letter.

▶ Corollary 6 (By [22]). An automaton with a prime number of states n that also contains a full cyclic permutation in its transition monoid is primitive and it is synchronizing if and only if it has any non-permutational letter.

Proof. The permutation group contained in the automaton's transition monoid is transitive, and it is known that every transitive group of a prime degree is primitive [3]. The statement that the automaton is synchronizing if it contains any non-permutational letter has been proved in [22].

3.2 Strong Variant

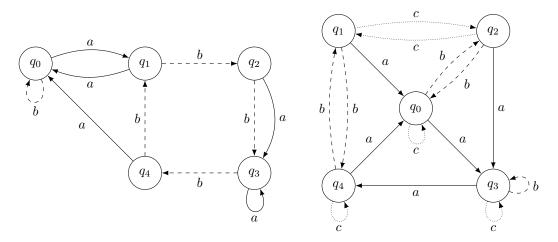


Figure 2 Two primitive automata that are not synchronizing. Letter a has deficiency 2, and letters b and c are permutational.

We note that restricting deficiency is essential. Examples of primitive non-synchronizing automata can be easily found already if we admit letters of deficiency 2.

Example 7. Both automata from Figure 2 are primitive and non-synchronizing.

Proof. Consider the automaton on the left. Observe that every pair of states can be mapped to $\{q_0, q_1\}$. If there is a non-trivial congruence on the set of states, so some two states q_i , q_j are in one block, then so q_0 , q_1 are in one block. But then also q_0 and each q_i for $i = 1, \ldots, 4$ are in one block by the action of b, thus the congruence is trivial. Also, the automaton is non-synchronizing, because the pairs $\{q_0, q_i\}$ for $i = 1, \ldots, 4$, $\{q_1, q_3\}$, and $\{q_2, q_4\}$ are not compressible.

Consider the automaton on the right. Observe that every pair of states can be mapped to $\{q_0, q_3\}$, and this pair can be mapped to each $\{q_i, q_3\}$ for $i \in \{0, 1, 2, 4\}$. Thus, as before, the automaton is primitive. Also, the pairs $\{q_i, q_3\}$ for $i \in \{0, 1, 2, 4\}$, $\{q_0, q_4\}$, and $\{q_1, q_2\}$ are not compressible, thus the automaton is non-synchronizing.

However, we can still slightly relax the condition and propose a stronger conjecture. For this, we refer to the notion of components and cycles of a transformation, which are defined on the digraph D(t) in which from every state q there is exactly one outgoing arc (q, (q)t).

23:6 Primitive Automata that are Synchronizing

Then, a component of a transformation t is a weakly connected component of D(t), which always contains exactly one cycle, which can be a loop. The strongly connected components of D(t) are precisely its cycles.

▶ Conjecture 8 (Strong variant). Every primitive automaton where each letter:

- 1. is permutational,
- **2.** has deficiency 1, or
- 3. its transformation has one component with a cycle of size ≤ 2 and the other components being cycles,

is synchronizing, unless all letters are permutational.

Both automata from Figure 1 satisfy the conditions of Conjecture 8 except that they are non-primitive; in particular, their letter a meets condition (3). On the other hand, the examples from Figure 2 show that we cannot strengthen the conjecture more: These automata do not satisfy the conditions due to the transformation of the letter a; in the left automaton, it has two components that are not cycles, and in the right automaton, the cycle in the unique component has size 3.

3.3 Experimental Support

Alphabet size	Number of states n							
$ \Sigma $	≤ 4	5	6	7	8	9	10	11
2	√	√	√	✓	✓	✓	√	✓*
3	✓	\checkmark	\checkmark	\checkmark				
4	✓	\checkmark	\checkmark					
5	✓	\checkmark						
6	✓	\checkmark						
≥ 7	✓							

 \checkmark – Both Conjecture 2 and Conjecture 8 verified. \checkmark * – Only Conjecture 2 verified.

Table 1 The range where our conjectures have been verified.

By the automata enumeration method of [21, 20], we have verified both conjectures for automata with a small number of states and a small alphabet. Table 1 summarizes the cases. The computation of the binary case with n = 11 (only the weak variant) took about 81 hours of one CPU core.

We also tried further strengthening of Conjecture 8 and then it was easy to find many counterexamples already for $n \leq 5$ (for instance, as in Figure 2).

4 Primitive Automata with Semiconstant Letters

A map f is semiconstant if there is a subset $S \subseteq Q$ such that $(S)f = \{r\}$, for some state $r \in Q$, and (q)f = q for every $q \in Q \setminus S$. We denote it by $(S \to r)$. The special cases of a semiconstant map are:

- 1. the *identity* map, which can be denoted by $(\emptyset \to r)$ for any $r \in Q$;
- **2.** constant maps, which can be denoted by $(Q \to r)$ for $r \in Q$;
- **3.** unitary (or simple idempotent) maps, denoted by $(\{q\} \to r)$, where $q \neq r$. We denote it simpler by $(q \to r)$.

We transfer this terminology to letters that induce these transformations.

An automaton $\mathscr{A}=(Q,\Sigma,\delta)$ where every letter is permutational or semiconstant is a PSc-automaton. If additionally, every letter of the automaton is permutational or unitary, then it is a PU-automaton. In the literature, the latter was often called an automaton with simple idempotents [18, 24], whereas the names unitary and semiconstant appear in [11]. The later terminology is adopted as it offers and distinguish both types of letters that we consider here.

In this section, we prove that every primitive PSc-automaton is synchronizing. We first show this for PU-automata and then generalize to the wider class.

4.1 Closures of Relations

Before we consider PU-automata, we state some basic properties of relations that will be used.

Let (Q,M) be a transformation monoid. It is easy to see that the union and the intersection of invariant relations are also invariant (for (Q,M)). Since the identity relation $\Delta_Q = \{(q,q) \mid q \in Q\}$ and the inverse relation $\rho^{-1} = \{(q,p) \mid (p,q) \in \rho\}$ of an invariant ρ relation are also invariant, we get the following statement:

▶ Proposition 9. If ρ is an invariant relation for (Q, M), then its symmetric closure $\Delta_Q \cup \rho \cup \rho^{-1}$ is also invariant for (Q, M).

For a relation ρ , we assign the oriented graph $\Gamma(\rho) = (Q, \rho)$, called shortly the *orgraph* of ρ . For each pair of different states $(s,t) \in \rho$, there is an *arc* (directed edge) in the orgraph from s to t, and for $(s,s) \in \rho$, there is a loop. When considering orgraphs, we use the usual notions from graph theory such as a path, a simple path, a route (walk), and a cycle.

Denote by ρ^* the transitive closure of ρ – this is the reachability relation of the orgraph $\Gamma(\rho)$, i.e., $(s,t) \in \rho^*$ if and only if there is a (possibly empty) path from s to t in $\Gamma(\rho)$.

▶ **Proposition 10.** If ρ is an invariant relation for (Q, M), then also ρ^* is invariant.

Proof. Let $(s,t) \in \rho^*$ and let $P = (s = s_1, \ldots, s_k = t)$ be a path in the orgraph $\Gamma(\rho)$ such that $(s_i, s_{i+1}) \in \rho$ for all $1 \le i < k$. Consider a map $f \in M$ and the image $(P)f = ((s_1)f, \ldots, (s_k)f)$. Then for all $1 \le i < k$, $((s_i)f, (s_{i+1})f) \in \rho$ since ρ is invariant. Thus, (P)f is a route in $\Gamma(\rho)$, which implies that $((s)f, (t)f) \in \rho^*$.

Denote by $\epsilon[\rho]$ the equivalence closure of ρ , i.e., the smallest equivalence relation containing ρ . This can be obtained by taking the reflexive, symmetric, and transitive closures:

$$\epsilon[\rho] = (\Delta_O \cup \rho \cup \rho^{-1})^*.$$

The equivalence classes of $\epsilon[\rho]$ are the weakly connected components of the orgraph $\Gamma(\rho)$. From Propositions 9 and 10, we get:

- ▶ Corollary 11. If ρ is an invariant relation of (Q, M), then its equivalence closure $\epsilon[\rho]$ is a congruence of (Q, M).
- ▶ Proposition 12. For a binary relation ρ , we have: $\epsilon[\rho] = \epsilon[\rho^*]$.

Proof. The direction \subseteq follows from the trivial inclusion $\rho \subseteq \rho^*$, and the other direction \supseteq follows from $\rho^* \subseteq \epsilon[\rho] = (\triangle_Q \cup \rho \cup \rho^{-1})^*$ by definition.

The following will be a useful property of relations:

- ▶ **Definition 13.** A binary relation ρ is cyclic if its transitive closure is symmetric: $\rho^* = (\rho^*)^{-1}$.
- ▶ Proposition 14. A binary relation ρ is cyclic if and only if one of the following equivalent conditions are satisfied:
- **1.** $\rho^{-1} \subseteq \rho^*$.
- **2.** $\Gamma(\rho)$ is a union of (not necessarily disjoint) cycles.
- **3.** Every weakly connected component of $\Gamma(\rho)$ is strongly connected.

Proof. If ρ is cyclic, then $\rho^{-1} \subseteq (\rho^*)^{-1} = \rho^*$ (1).

If for each arc $(s,t) \in \rho$, we have $(t,s) \in \rho^*$ (1), so there is a simple path P from s to t in $\Gamma(\rho)$, then the path (s,t)P is a cycle containing (s,t) (2).

From (2), every s and t from one weakly connected component are in a cycle, thus they are in the same strongly connected component (3).

From (3), for every $(s,t) \in \rho^*$, there is also a path from t to s, thus we have $(t,s) \in \rho^*$, and vice versa, thus the relation is cyclic.

▶ Corollary 15. The transitive closure of a cyclic relation ρ is an equivalence relation:

$$\rho^* = \epsilon[\rho].$$

Proof. By Proposition 12, we have $\rho^* \subseteq \epsilon[\rho^*] = \epsilon[\rho]$. Consider the other direction. The identity relation Δ_Q and ρ are trivially contained in ρ^* . From Proposition 14(1), we also have $\rho^{-1} \subseteq \rho^*$. Thus, $\epsilon[\rho] \subseteq \rho^*$, and the equality follows.

4.2 Transition Semigroups of Automata with Permutational and Unitary Letters

Let $\mathscr{A} = (Q, \Sigma, \delta)$ be a strongly connected PU-automaton, and let P and U be respectively the set of the permutations and the set of unitary transformations induced by its letters. The transition monoid of this automaton is $(Q, \langle P \cup U \rangle)$.

Consider the submonoid $G = (Q, \langle P \rangle)$, which is a permutation group on Q. Note that the reachability relation in G is an equivalence relation, i.e., each weakly connected component is strongly connected, which is a class in this relation called an *orbit*. We denote by (s)G the orbit of G that contains $s \in Q$: $(s)G = \{(s)g \mid g \in G\}$.

The action of the permutation group G continues on the square $Q \times Q$. The (strongly) connected components of G on $Q \times Q$ are called *orbitals*. Denote by (s,t)G the orbital of G that contains (s,t), i.e., $(s,t)G = \{(s,t)g \mid g \in G\}$. Note that each orbital is an invariant relation of G. Two pairs of states are G-equivalent if they belong to the same orbital of G.

We define the unitary relation $\delta(U)$ as follows:

$$\delta(U) = \bigcup_{(s \to t) \in U} (s, t).$$

The group closure of the idempotent relation is defined by the formula:

$$\pi = (\delta(U))G = \bigcup_{(s,t)\in\delta(U)} (s,t)G. \tag{1}$$

Since π is a union of orbitals of G, it is an invariant relation for G.

A pair of states $(s,t) \in \pi$ is called *internal* if both states are in the same orbit of G. The pair is *external* if s and t are in different orbits of the group. We split π into disjoint π_{int} containing the internal pairs and π_{ext} containing the external pairs.

▶ **Lemma 16.** The relations π_{int} and π_{ext} are invariant for G.

Proof. Since π is invariant, for every $(s,t) \in \pi_{\text{int}}$, we have $((s)g,(t)g) \in \pi$. If $(s,t) \in \pi_{\text{int}}$, thus s and t are in the same orbit of G, then we know that also (s)g and (t)g are in the same orbit, hence we have an internal arc: $((s)g,(t)g) \in \pi_{\text{int}}$. Similarly, if $(s,t) \in \pi_{\text{ext}}$, thus s and t are in different orbits of G, then (s)g and (t)g are in different orbits, hence we have an external arc: $((s)g,(t)g) \in \pi_{\text{ext}}$.

The following statement in other terms has been used in the theory of permutation groups [6].

▶ **Lemma 17.** The relation π_{int} is cyclic.

Proof. Consider the orgraph $\Gamma(\pi_{\rm int}) = (Q, \pi_{\rm int})$ and an arc $(s,t) \in \pi_{\rm int}$. Since s and t are in the same orbit of G, there is a permutation $g \in G$ such that (s)g = t. Denote by $k \geq 1$ the minimum number such that $(t)g^k = s$, and consider the sequence of pairs $C = ((s,t)g^i : i \leq i \leq k)$, where g^0 is the identity map (the unit of G). As $\pi_{\rm int}$ is invariant for G, all the arcs from C are in $\Gamma(\pi_{\rm int})$ thus C is a cycle in $\Gamma(\pi_{\rm int})$. It follows that every arc from $\Gamma(\pi_{\rm int})$ is in a cycle, thus $\Gamma(\pi_{\rm int})$ is a union of cycles and by Proposition 14, $\pi_{\rm int}$ is cyclic.

Consider the orgraph $\Gamma(\pi_{\text{ext}})$ and let $(s,t) \in \pi_{\text{ext}}$. Then the pair (s,t) is located between the orbits (s)G and (t)G of the group G. Consider the orbital (s,t)G of the pair (s,t) and its orbital orgraph $\Gamma(s,t) = ((s)G \cup (t)(G), (s,t)G)$. It follows from formula (1) and Lemma 16 that $\Gamma(s,t)$ is a subgraph of $\Gamma(\pi_{\text{ext}})$.

▶ Lemma 18. For an external pair $(s,t) \in \pi_{\text{ext}}$, the orbital orgraph $\Gamma(s,t)$ is a bipartite graph with parts (s)G and (t)G such that from each state in (s)G goes out an arc to a state in the orbit (t)G.

Proof. For each state $s' \in (s)G$, we can find a permutation $g \in G$ such that (s',t') = (s,t)g is the outgoing arc from s' to a state $t' \in (t)G$.

▶ **Lemma 19.** For every $s, t \in Q$, there is a path in the orgraph $\Gamma(\pi_{\text{ext}})$ from s to a state in the orbit (t)G.

Proof. Since we assume that \mathscr{A} is strongly connected, there is a path of transitions e_1, \ldots, e_k from the state s to a state in (t)G, i.e., for each $e_i = (s_i, t_i)$ there is either a permutational letter inducing $g \in P$ such that $(s_i)g = t_i$ or the unitary transition $(s_i \to t_i) \in U$. In the former case, e_i is an internal arc from π_{int} , and in the latter case, it can be either an internal or an external arc from π .

We remove all internal arcs, obtaining a sequence of external arcs e_{i_1}, \ldots, e_{i_m} only, which is a subsequence of e_1, \ldots, e_k . If this is the empty sequence (m=0), then $s \in (t)G$, so the lemma holds. Otherwise, for the first arc $e_{i_1} = (s_{i_1}, t_{i_1})$, we have s_{i_1} in the orbit (s)G and t_{i_1} in another orbit. By Lemma 18 applied for e_{i_1} , we can find a pair $(s, s_1) \in \pi_{\text{ext}}$ that is G-equivalent to e_{i_1} , thus where s_1 is a state from $(t_{i_1})G$. There may be several such arcs, and we can choose any. Then for e_{i_2} , we choose an arc $(s_1, s_2) \in \pi_{\text{ext}}$ in the same way, and so on till e_{i_m} . As the result, we obtain a route in the orgraph $\Gamma(\pi_{\text{ext}})$ that starts from s, follows the same orbits as the path e_1, \ldots, e_k , and ends in a state from (t)G.

▶ Lemma 20. The relation π_{ext} is cyclic.

Proof. Consider the orgraph $\Gamma(\pi_{\text{ext}}) = (Q, \pi_{\text{ext}})$ and an arc $(s_1, t_1) \in \pi_{\text{ext}}$. We are going to show that $(s_1, t_1) \in \pi_{\text{ext}}$ is in some cycle in $\Gamma(\pi_{\text{ext}})$, thus the lemma follows by Proposition 14.

By Lemma 19, there is a path P_1 in the orgraph $\Gamma(\pi_{\text{ext}})$ from t_1 to some state $s_2 \in (s_1)G$. If $s_1 = s_2$, then we have found the cycle $(s_1, t_1)P_1$, which contains (s_1, t_1) . Otherwise, we choose a state $t_2 \in (t_1)G$ such that the arc $(s_2, t_2) \in \pi_{\text{ext}}$ is G-equivalent to (s_1, t_1) , and we again choose a path P_2 from t_2 to some state $s_3 \in (s_1)G$. We continue this process, which stops when the found path P_k ends in a state $s_{k+1} \in (s_1)G$ that has been encountered before as an s_i for some $i \leq k$. As the size of $(s_1)G$ is finite, this finally happens.

Now, $(s_i, t_i)P_i \dots (s_k, t_k)P_k$ is a closed route containing $(s_i, t_i) \in \pi_{\text{ext}}$. Let X be the path separated from the route $P_i \dots (s_k, t_k)P_k$ by removing all cycles so that every state occurs at most once. Then $C = (s_i, t_i)X$ is a cycle containing the external arc (s_i, t_i) , which is G-equivalent to (s_1, t_1) . Let $g \in G$ be such that $(s_1, t_1) = (s_i, t_i)g$. Then the cycle C' = (C)g is contained in $\Gamma(\pi_{\text{ext}})$ (as π_{ext} is invariant for G) and contains (s_1, t_1) .

▶ Corollary 21. The relation $\pi = \pi_{int} \cup \pi_{ext}$ is cyclic.

We recall the known characterization of synchronizing automata with permutational and unitary letters. In the literature, the orgraph $\Gamma(\pi)$ is often called the Rystsov graph of an automaton [10, 12, 18].

▶ **Theorem 22** ([24, Corollary 2 and Theorem 1]). A strongly connected PU-automaton is synchronizing if and only if the orgraph $\Gamma(\pi)$ is strongly connected.

We have now all ingredients for the main result of this section.

▶ Theorem 23. A primitive PU-automaton with at least one unitary letter is synchronizing.

Proof. By Proposition 4, it is enough to consider a strongly connected primitive U-automaton. Suppose that such an automaton is not synchronizing, thus by Theorem 22, the orgraph $\Gamma(\pi)$ is not strongly connected. Then, for the transitive closure π^* of π , we obtain the strict inclusion: $\pi^* \subsetneq Q \times Q$. Since the automaton has at least one unitary letter, the relation π is non-empty and thus $\Delta \subsetneq \pi^*$. Since π is cyclic (Corollary 21), from Corollary 15, we get $\pi^* = \epsilon[\pi]$ so it is an equivalence relation. As π is invariant for G (Lemma 16), from Corollary 11, the equivalence closure $\epsilon[\pi]$ is a congruence of the automaton's transition monoid $(Q, \langle P \cup U \rangle)$. Thus, $\epsilon[\pi]$ is a non-trivial congruence of $(Q, \langle P \cup U \rangle)$, so the automaton is non-primitive.

4.3 Generalization to Semiconstant Letters

The generalization follows by a decomposition of semiconstant transformations into unitary transformations. If \mathscr{A} is an PSc-automaton, then by $\mathscr{A}^{\operatorname{Sc}\to U}$ we denote the PU-automaton obtained from \mathscr{A} in the following way: every letter that induces a semiconstant transformation $(\{s_1,\ldots,s_k\}\to r)$, where the states s_i are pairwise distinct and different from r, is replaced with k fresh letters inducing the unitary transformations $(s_1\to r),\ldots,(s_k\to r)$, respectively.

- ▶ Lemma 24. Let \(\alpha \) be a PSc-automaton. Then:
- 1. If \mathscr{A} is primitive then also $\mathscr{A}^{\operatorname{Sc} \to \operatorname{U}}$ is primitive.
- **2.** \mathscr{A} is synchronizing if and only if $\mathscr{A}^{\operatorname{Sc} \to \operatorname{U}}$ is synchronizing.

Proof. A semiconstant transformation $(\{s_1, \ldots, s_k\} \to r)$ can be generated as the composition of the k unitary transformations that replace it:

$$(s_1 \rightarrow r) \cdot \ldots \cdot (s_k \rightarrow r) = (\{s_1, \ldots, s_k\} \rightarrow r).$$

Hence, the transition monoid of \mathscr{A} is a submonoid of $\mathscr{A}^{Sc \to U}$. This implies (1) and the \Rightarrow direction of (2).

For the \Leftarrow direction of (2), denote $\mathscr{A} = (Q, \Sigma_{\mathscr{A}}, \delta_{\mathscr{A}})$ and $\mathscr{B} = (Q, \Sigma_{\mathscr{B}}, \delta_{\mathscr{B}}) = \mathscr{A}^{\operatorname{Sc} \to \operatorname{U}}$. We show by induction on k that every pair of states $\{p,q\}$ which is compressible in \mathscr{B} with a word of length k is compressible in \mathscr{A} . For k=0 it is trivial. Assume the statement for k and consider a pair $\{p,q\}$ which is compressible in \mathscr{B} with a word w=au of length k+1, where $a \in \Sigma_{\mathscr{B}}$ and $u \in \Sigma_{\mathscr{B}}^*$.

If a is permutational, then let $\tau(a) = a$, and the following equation holds trivially:

$$\{\delta_{\mathscr{B}}(p,a), \delta_{\mathscr{B}}(q,a)\} = \{\delta_{\mathscr{A}}(p,\tau(a)), \delta_{\mathscr{A}}(q,\tau(a))\}$$
(2)

Then the statement follows from the induction hypothesis since u of length k compresses the pair from (2) in \mathcal{B} .

Otherwise, a induces a unitary transformation $(s \to r)$. If $s \notin \{p, q\}$, then $\delta_{\mathscr{B}}(\{p, q\}, a) = \{p, q\}$, so again u of length k compresses the pair. Without loss of generality, suppose s = p. Then let $\tau(a)$ be a semiconstant letter that induces $(S \to r)$, where $s = p \in S$. If $q \in S \cup \{r\}$, then $\tau(a)$ compresses $\{p, q\}$, so we are done. Otherwise, $q \notin S \cup \{r\}$, thus (2) holds again.

It follows that if \mathcal{B} is synchronizing, thus all pairs of states are compressible, then also they are in \mathcal{A} , so \mathcal{A} is synchronizing.

▶ **Theorem 25.** A primitive PSc-automaton is synchronizing, unless all its letters are permutational.

Proof. If \mathscr{A} is a primitive PSc-automaton, then by Lemma 24(1), $\mathscr{A}^{\text{Sc}\to\text{U}}$ is primitive, so by Theorem 23 $\mathscr{A}^{\text{Sc}\to\text{U}}$ is synchronizing. Then by Lemma 24(2), \mathscr{A} is also synchronizing.

- References -

- J. Almeida and E. Rodaro. Semisimple Synchronizing Automata and the Wedderburn-Artin Theory. International Journal of Foundations of Computer Science, 27(02):127-145, 2016.
- 2 J. Araújo, W. Bentz, P. J. Cameron, G. Royle, and A. Schaefer. Primitive groups, graph endomorphisms and synchronization. *Proceedings of the London Mathematical Society*, 113(6):829–867, 2016.
- 3 J. Araújo, P. J. Cameron, and B. Steinberg. Between primitive and 2-transitive: Synchronization and its friends. *EMS Surv. Math. Sci.*, 4(2):101–184, 2017.
- 4 F. Arnold and S. Steinberg. Synchronizing groups and automata. *Theoretical Computer Science*, 359(1):101–110, 2006.
- V. A. Artamonov, V. N. Saliy, L. A. Skornyakov, L. N. Shevrin, and E. G. Shulgeifer. General Algebra, volume 2. Nauka, Moscow, 1991. In Russian.
- 6 E. Bannai, E. Bannai, T. Ito, and R. Tanaka. Algebraic Combinatorics. De Gruyter, 2021.
- 7 M. V. Berlinkov, R. Ferens, A. Ryzhikov, and M. Szykuła. Synchronizing Strongly Connected Partial DFAs. In *STACS*, volume 187 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl, 2021.
- 8 M. V. Berlinkov and C. Nicaud. Synchronizing Almost-Group Automata. *International Journal of Foundations of Computer Science*, 31(08):1091–1112, 2020.
- 9 M. V. Berlinkov and M. Szykuła. Algebraic synchronization criterion and computing reset words. *Information Sciences*, 369:718–730, 2016.
- E. A. Bondar and M. V. Volkov. A Characterization of Completely Reachable Automata. In Mizuho Hoshi and Shinnosuke Seki, editors, DLT, pages 145–155. Springer, 2018.
- J. Brzozowski and M. Szykuła. Large Aperiodic Semigroups. International Journal of Foundations of Computer Science, 26(07):913–931, 2015.
- D. Casas and M. V. Volkov. Binary completely reachable automata. In Armando Castañeda and Francisco Rodríguez-Henríquez, editors, LATIN 2022: Theoretical Informatics, pages 345–358. Springer, 2022. Full version at https://arxiv.org/abs/2205.09404.

- J. Černý. Poznámka k homogénnym experimentom s konečnými automatami. Matematicko-fyzikálny Časopis Slovenskej Akadémie Vied, 14(3):208–216, 1964. In Slovak.
- 14 A. H. Clifford and G. B. Preston. The Algebraic Theory of Semigroups, volume 2. American Mathematical Society, 1967.
- 15 M. Grech and A. Kisielewicz. The Černý conjecture for automata respecting intervals of a directed graph. *Discrete Mathematics & Theoretical Computer Science*, Vol. 15 no. 3, 2013.
- S. Hoffmann. Completely Reachable Automata, Primitive Groups and the State Complexity of the Set of Synchronizing Words. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, Language and Automata Theory and Applications, LNCS, pages 305–317. Springer, 2021.
- 17 S. Hoffmann. Sync-Maximal Permutation Groups Equal Primitive Permutation Groups. In Yo-Sub Han and Sang-Ki Ko, editors, *Descriptional Complexity of Formal Systems*, pages 38–50. Springer, 2021.
- S. Hoffmann. Constrained Synchronization for Monotonic and Solvable Automata and Automata with Simple Idempotents. In Pascal Caron and Ludovic Mignot, editors, *Implementation and Application of Automata*, LNCS, pages 225–237. Springer, 2022.
- J. Kari and M. V. Volkov. Černý conjecture and the road colouring problem. In *Handbook of automata*, volume 1, pages 525–565. European Mathematical Society Publishing House, 2021.
- A. Kisielewicz, J. Kowalski, and M. Szykuła. Experiments with Synchronizing Automata. In Implementation and Application of Automata, volume 9705 of LNCS, pages 176–188. Springer, 2016.
- A. Kisielewicz and M. Szykuła. Generating Small Automata and the Černý Conjecture. In Implementation and Application of Automata, volume 7982 of LNCS, pages 340–348. Springer, 2013.
- J.-E. Pin. Sur un cas particulier de la conjecture de Černý. In ICALP, volume 62 of LNCS, pages 345–352. Springer, 1978.
- J.-E. Pin. On two combinatorial problems arising from automata theory. In Proceedings of the International Colloquium on Graph Theory and Combinatorics, volume 75 of North-Holland Mathematics Studies, pages 535–548, 1983.
- 24 I. K. Rystsov. Estimation of the length of reset words for automata with simple idempotents. Cybern. Syst. Anal. 36, pages 339–344, 2000.
- 25 I. K. Rystsov. Primitive and Irreducible Automata. Cybernetics and Systems Analysis, 51:506-513, 2015.
- Y. Shitov. An Improvement to a Recent Upper Bound for Synchronizing Words of Finite Automata. *Journal of Automata, Languages and Combinatorics*, 24(2–4):367–373, 2019.
- D. M. Smirnov. Correspondence between regular varieties of unary algebras and semigroups. Algebra and Logic, 17(4):310–315, 1978.
- 28 B. Steinberg. A Theory of Transformation Monoids: Combinatorics and Representation Theory. *Electron. J. Comb.*, 17, 2010.
- 29 M. Szykuła. Improving the Upper Bound on the Length of the Shortest Reset Word. In STACS 2018, LIPIcs, pages 56:1–56:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.
- 30 A. N. Trahtman. The Černý conjecture for aperiodic automata. Discrete Mathematics & Theoretical Computer Science, Vol. 9 no. 2, 2007.
- 31 M. V. Volkov. Synchronizing Automata Preserving a Chain of Partial Orders. In Jan Holub and Jan Žďárek, editors, *Implementation and Application of Automata*, pages 27–37. Springer, 2007.
- M. V. Volkov. Synchronizing automata and the Černý conjecture. In Language and Automata Theory and Applications, volume 5196 of LNCS, pages 11–27. Springer, 2008.
- 33 M. V. Volkov. Synchronization of finite automata. Uspekhi Matematicheskikh Nauk, 77:53–130, 2022. in Russian.

34 M. V. Volkov. Synchronization of primitive automata. https://arxiv.org/abs/2306.13317, 2023.