

# Winning Strategies for the Synchronization Game on Subclasses of Finite Automata\*

Henning Fernau

Universität Trier, Fachbereich IV, Informatikwissenschaften, Trier, Germany  
fernau@uni-trier.de

Carolina Haase

haasec@uni-trier.de

Stefan Hoffmann

hoffmanns.tcs@gmail.com

Mikhail Volkov

Institute of Natural Sciences and Mathematics, Ural Federal University, Ekaterinburg, Russia  
m.v.volkov@urfu.ru

We exhibit a winning strategy for Synchronizer in the synchronization game on every synchronizing automaton in whose transition monoid the regular  $\mathcal{D}$ -classes form subsemigroups.

## 1 Introduction

A complete deterministic finite automaton (DFA) is a pair  $(Q, \Sigma)$  of two finite sets equipped with a map  $Q \times \Sigma \rightarrow Q$  whose image at  $(q, a) \in Q \times \Sigma$  is denoted by  $q \cdot a$ . We call  $Q$  the *state set* and  $\Sigma$  the *input alphabet*. Elements of  $Q$  and  $\Sigma$  are referred to as *states* and, respectively, *letters*, and for a state  $q \in Q$  and a letter  $a \in \Sigma$ , we refer to  $q \cdot a$  as the result of the *action of  $a$  at  $q \in Q$* . The action of letters in  $\Sigma$  naturally extends to the action of words over  $\Sigma$ : if  $w = a_1 a_2 \cdots a_n$  with  $a_1, a_2, \dots, a_n \in \Sigma$ , then  $q \cdot w := (\dots((q \cdot a_1) \cdot a_2) \dots) \cdot a_n$ .

A DFA  $(Q, \Sigma)$  is called *synchronizing* if there exists a word  $w$  over  $\Sigma$  whose action brings the DFA to one particular state no matter at which state  $w$  is applied:  $q \cdot w = q' \cdot w$  for all  $q, q' \in Q$ . Any word  $w$  with this property is said to be a *reset word* for the automaton.

Synchronizing automata serve as transparent and natural models of error-resistant systems in many applications (coding theory, robotics, testing of reactive systems) and reveal interesting connections with symbolic dynamics, substitution systems, and other parts of mathematics. We refer the reader to chapter [12] of the ‘Handbook of Automata Theory’ and survey [19] for an introduction to the area and an overview of its state-of-the-art.

The fourth-named author initiated viewing synchronizing automata through the lens of game theory; the motivation for this came from a game-theoretical approach to software testing suggested in [4]. In a synchronization game on a DFA  $\mathcal{A}$ , two players, Alice (Synchronizer) and Bob (Desynchronizer), take turns choosing letters from the input alphabet of  $\mathcal{A}$ . Alice who wants to synchronize  $\mathcal{A}$  wins when the sequence of chosen letters forms a reset word. Bob aims to prevent synchronization or, if synchronization is unavoidable, to delay it as long as possible. Provided that both players play optimally, the outcome of such a game depends on the automaton only. This raises the problem of classifying synchronizing automata into those on which Alice and, respectively, Bob have a winning strategy. DFAs on which Alice can ensure win are of interest because they are more amenable to synchronization, in a sense. For brevity, we call such DFAs *A-automata*.

---

\*Mikhail Volkov was supported by the Ministry of Science and Higher Education of the Russian Federation, project FEUZ-2023-2022.

A few initial results on synchronization games were obtained in [8]. In particular, [8, Theorem 4] provides an algorithm that, given a DFA  $\mathcal{A}$  with  $n$  states and  $k$  input letters, decides who has a winning strategy in the synchronization game on  $\mathcal{A}$  in  $O(n^2k)$  time. Thus, for any individual DFA, one can determine whether it is an A-automaton. Here, however, we are interested in general conditions ensuring that all synchronizing DFAs of a certain type are A-automata. One such condition was mentioned in [8]: Alice always wins on definite DFAs introduced in [14].

In [7], the first three authors of the present note showed that within two further families of automata considered in the literature—weakly acyclic DFAs and commutative DFAs—every synchronizing DFA is an A-automaton. Here we continue this line of research by designing a winning strategy for Alice that applies to synchronizing automata from yet another family of DFAs. Automata in this family are distinguished by a structure feature of their transition monoids: the regular  $\mathfrak{D}$ -classes in these monoids form subsemigroups. The set  $\mathbf{DS}$  of all finite semigroups with regular  $\mathfrak{D}$ -classes being subsemigroups plays a distinguished role in the algebraic theory of regular languages; see [1, Chapter 8]. Therefore, DFAs with transition monoids in  $\mathbf{DS}$  often show up in the literature; see, e.g., [2, 3]. As they seem to have no specific name so far, we coin them DS-automata. Thus, our main result says that Alice can win the synchronization game on every synchronizing DS-automaton. Since the family of DS-automata is extensive and encompasses all the families above (definite, weakly acyclic, and commutative DFAs), this provides a vast generalization of the mentioned results from [8, 7].

Our approach is algebraic as it exploits the structural properties of transition monoids. We have collected all necessary prerequisites from semigroup theory in Section 2 to make the note self-contained, to a reasonable extent. Section 3 presents Alice’s winning strategy in synchronization games on synchronizing DS-automata. In Section 4, we relate our result to previously found facts about winning strategies for synchronization games and state two open questions.

## 2 Preliminaries

### 2.1 Transition Monoids and Synchronization

For a DFA  $\mathcal{A} = (Q, \Sigma)$ , the map  $\tau_a: Q \rightarrow Q$  defined by the rule  $q \mapsto q \cdot a$  is a transformation on the set  $Q$ .

**Definition 1.** The *transition monoid* of a DFA  $\mathcal{A} = (Q, \Sigma)$  is the submonoid of the monoid of all transformations on the set  $Q$  generated by the set  $\{\tau_a \mid a \in \Sigma\}$ .

We denote the transition monoid of a DFA  $\mathcal{A} = (Q, \Sigma)$  by  $T(\mathcal{A})$ . It is easy to see that any product  $\tau_{a_1} \tau_{a_2} \cdots \tau_{a_n}$  is nothing but the transformation  $\tau_w$  defined by the rule  $q \mapsto q \cdot w$  where  $w$  stands for the word  $a_1 a_2 \cdots a_n$ . Thus, the transition monoid  $T(\mathcal{A})$  can alternatively be defined as the monoid of all transformations on the set  $Q$  caused by the action of words over  $\Sigma$ .

If  $\mathcal{A} = (Q, \Sigma)$  is a synchronizing DFA and  $w$  is a reset word for  $\mathcal{A}$ , then the transformation  $\tau_w$  is a constant map on  $Q$ , that is,  $Q\tau_w = \{q\}$  for a certain  $q \in Q$ . Thus, the transition monoid of a synchronizing automaton always contains a constant transformation. Conversely, if  $\zeta \in T(\mathcal{A})$  is a constant transformation, then any word  $w$  with  $\tau_w = \zeta$  is a reset word for  $\mathcal{A}$  and so  $\mathcal{A}$  is synchronizing. We see that synchronization is actually a property of the transition monoid of an automaton rather than the automaton itself: for DFAs  $\mathcal{A} = (Q, \Sigma)$  and  $\mathcal{A}' = (Q, \Sigma')$  with the same state set but different input alphabets, the equality  $T(\mathcal{A}) = T(\mathcal{A}')$  guarantees that  $\mathcal{A}'$  is synchronizing if and only if so is  $\mathcal{A}$ .

We can say a bit more about transition monoids of synchronizing automata, but for this, we first need to recall some concepts of semigroup theory. A nonempty subset  $I$  of a semigroup  $S$  is called an *ideal* in  $S$  if, for all  $s \in S$  and  $i \in I$ , both products  $si$  and  $is$  lie in  $I$ . If  $I$  and  $J$  are two ideals in  $S$ , their intersection

$I \cap J$  contains any product  $ij$  with  $i \in I, j \in J$ . So  $I \cap J$  is nonempty, and then it is easy to verify that  $I \cap J$  is again an ideal in  $S$ . Therefore, each finite semigroup  $S$  has a least ideal called the *kernel* of  $S$  and denoted  $\text{Ker} S$ .

The following observation is folklore but we provide a proof for the sake of completeness.

**Lemma 1.** *A DFA is synchronizing if and only if the kernel of its transition monoid consists of constant transformations.*

*Proof.* The ‘if’ part readily follows from the already mentioned fact that if the transition monoid of a DFA contains a constant transformation, then the DFA is synchronizing.

For the ‘only if’ part, let  $\mathcal{A}$  be a synchronizing DFA; then the set  $C$  of all constant transformations in the transition monoid  $T(\mathcal{A})$  is nonempty. For any  $\tau \in T(\mathcal{A})$  and any  $\zeta \in C$ , we have

$$\tau\zeta = \zeta. \quad (1)$$

Equality (1) implies that the set  $C$  is contained in every ideal of the monoid  $T(\mathcal{A})$ , in particular, in its kernel  $\text{Ker} T(\mathcal{A})$ . Further, for every  $\tau \in T(\mathcal{A})$  and every  $\zeta \in C$ , the product  $\zeta\tau$  is a constant transformation: if  $Q\zeta = \{q\}$ , then  $Q\zeta\tau = \{q\tau\}$ . Together with (1), this observation implies that  $C$  forms an ideal in  $T(\mathcal{A})$ . Since  $\text{Ker} T(\mathcal{A})$  contains  $C$ , we have  $\text{Ker} T(\mathcal{A}) = C$  by the definition of the kernel.  $\square$

## 2.2 Structure of Semigroups in DS

Green [9] defined five important relations on every semigroup  $S$ , collectively referred to as *Green’s relations*, of which we need the following three:

$$a \mathfrak{R} b \iff \text{either } a = b \text{ or } a = bs \text{ and } b = at \text{ for some } s, t \in S;$$

$$a \mathfrak{L} b \iff \text{either } a = b \text{ or } a = sb \text{ and } b = ta \text{ for some } s, t \in S;$$

$$a \mathfrak{D} b \iff a \mathfrak{R} c \text{ and } c \mathfrak{L} b \text{ for some } c \in S.$$

The relations  $\mathfrak{R}$  and  $\mathfrak{L}$  are obviously equivalencies. The definition of  $\mathfrak{D}$  means that  $\mathfrak{D}$  is the product of  $\mathfrak{R}$  and  $\mathfrak{L}$  as binary relations. As observed in [9],  $\mathfrak{D}$  is also the product of  $\mathfrak{L}$  and  $\mathfrak{R}$ , and this implies that  $\mathfrak{D}$  is the least equivalence containing both  $\mathfrak{R}$  and  $\mathfrak{L}$ .

An element  $a$  of a semigroup  $S$  is said to be *regular* if  $asa = a$  for some  $s \in S$ . A  $\mathfrak{D}$ -class  $D$  is called *regular* if it contains a regular element. (In this case, every element of  $D$  is known to be regular; see [9, Theorem 6].) We denote by **DS** the set of all finite semigroups  $S$  such that the regular  $\mathfrak{D}$ -classes of  $S$  are subsemigroups in  $S$ .

The structure of semigroups **DS** is well understood in terms of their decompositions into some basic blocks. As we will use this structural result, we recall the notions involved.

A *semilattice* is a semigroup satisfying the laws of commutativity  $xy = yx$  and idempotency  $x^2 = x$ .

**Definition 2.** Let  $Y$  be a semilattice and  $\{S_y\}_{y \in Y}$  a family of disjoint semigroups indexed by the elements of  $Y$ . A semigroup  $S$  is said to be a *semilattice of semigroups*  $S_y, y \in Y$ , if:

$$(S1) \quad S = \bigcup_{y \in Y} S_y;$$

$$(S2) \quad \text{each } S_y \text{ is a subsemigroup in } S;$$

$$(S3) \quad \text{for every } y, z \in Y \text{ and every } s \in S_y, t \in S_z, \text{ the product } st \text{ belongs to } S_{yz}.$$

We say that a semigroup  $S$  is  $m$ -nilpotent over its kernel ( $m$  being a positive integer) if every product of  $m$  elements of  $S$  belongs to  $\text{Ker} S$ . We call a semigroup nilpotent over its kernel if it is  $m$ -nilpotent over its kernel for some  $m$ . (To a semigroupist, finite semigroups nilpotent over their kernels are familiar as finite Archimedean semigroups.)

The following is a specialization of the equivalence (4c)  $\Leftrightarrow$  (1b) in [18, Theorem 3] to finite semigroups<sup>1</sup>.

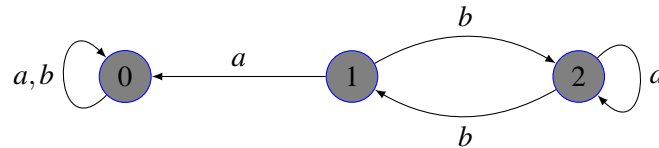
**Lemma 2.** *Every semigroup in DS is a semilattice of semigroups nilpotent over their kernels.*

### 3 Winning Strategy in Synchronization Games on DS-Automata

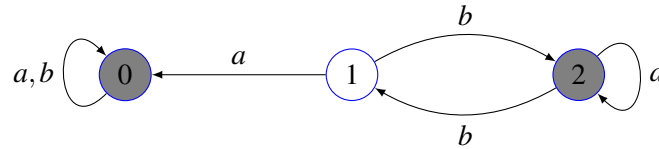
We start with a visual yet rigorous description of the synchronization game under consideration. In this game, two players, Alice and Bob, play on a fixed DFA  $\mathcal{A} = (Q, \Sigma)$ . At the start, each state in  $Q$  holds a token. During the game, some tokens can be removed according to the rules specified in the next paragraph. Alice wins if only one token remains, while Bob wins if he can keep at least two tokens unremoved for an indefinite amount of time.

Alice moves first, and then players alternate moves. The player whose turn is to move proceeds by selecting a letter  $a \in \Sigma$ . Then, for each state  $q \in Q$  that held a token before the move, the token advances to the state  $q \cdot a$ . (In the standard graphical representation of  $\mathcal{A}$  as the labelled digraph with  $Q$  as the vertex set and the labelled edges of the form  $q \xrightarrow{a} q \cdot a$ , one can visualize the move as follows: all tokens simultaneously slide along the edges labelled  $a$ .) If several tokens arrive at the same state after this, all of them but one are removed so that when the move is completed, each state holds at most one token.

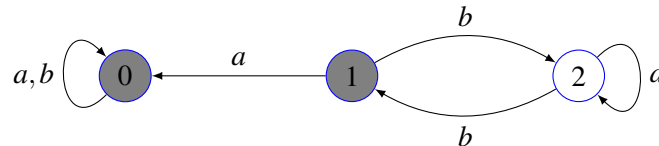
To illustrate, let us look at how the synchronization game might play out on the following DFA in which, initially, each state holds a token (shown in gray).



If Alice chooses the letter  $a$  on her first move, the tokens in states 0 and 2 remain due to the loops at these states. The token from state 1 moves to 0 and then is removed because the state 0 is ‘occupied’. Hence, the position after Alice’s first moves looks as follows:

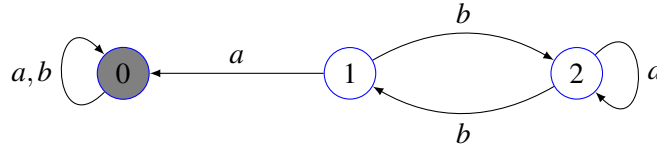


If Bob replies by choosing the letter  $b$ , the token in state 0 remains while the token from state 2 moves to 1. Here is the position after Bob’s reply:



<sup>1</sup>Theorem 3 in [18] deals with semigroups in which every element has a power that belongs to a subgroup. Every finite semigroup has this property.

Now choosing  $a$ , Alice wins because after the token from state 1 moves to 0, it is removed, and we get the position with only one token:



Notice that Alice won the game described above only because of Bob's unfortunate reply. In fact, Bob has a winning strategy in the synchronization game on this DFA: if he repeats Alice's moves, that is, chooses the same letter Alice chose on her previous move, he can maintain two tokens unremoved forever. Hence, there are simple synchronizing DFAs that are not A-automata.

Recall that a DS-automaton is a DFA whose transition monoid lies in the set **DS** of all finite semigroups with regular  $\mathcal{D}$ -classes being subsemigroups. The following is the main result of this note.

**Theorem 3.** *Alice has a winning strategy on every synchronizing DS-automaton.*

*Proof.* Take an arbitrary synchronizing DS-automaton  $\mathcal{A} = (Q, \Sigma)$ . We denote the transition monoid  $T(\mathcal{A})$  by  $S$  to lighten the notation. Since  $S \in \mathbf{DS}$ , by Lemma 2 there is a semilattice  $Y$  such that  $S$  is a semilattice of semigroups  $S_y$ ,  $y \in Y$ , where each semigroup  $S_y$  is nilpotent over its kernel.

The relation  $\leq$  defined by  $x \leq y \iff xy = x$  is known (and easy to see) to be a partial order on every semilattice, and so on  $Y$ . Due to the laws of commutativity and idempotency, the inequalities  $xy \leq x$  and  $xy \leq y$  hold for all  $x, y \in Y$ . Since the semilattice  $Y$  is finite, it has a least element with respect to this order; denote it by  $z$ . Then  $yz = zy = z$  for every  $y \in Y$  whence the semigroup  $S_z$  is an ideal in  $S$  by item (S3) in Definition 2. Therefore  $S_z$  contains the kernel  $\text{Ker} S$  of  $S$ , and therefore,  $\text{Ker} S_z \subseteq \text{Ker} S$ . (In fact, it is easy to show that the equality  $\text{Ker} S_z = \text{Ker} S$  holds, but it is not needed for the present proof.)

Fix a positive integer  $m$  such that the semigroup  $S_z$  is  $m$ -nilpotent over its kernel  $\text{Ker} S_z$ . We show that Alice can win in the synchronization game on  $\mathcal{A}$ , using the following  $m$ -round strategy. Denote by  $a_i^k$  and  $b_i^k$  the  $i$ -th letters chosen in the  $k$ -th round by Alice and Bob, respectively. In each round, Alice chooses the first letter  $a_1^k$  at random. Then, after each reply of Bob, she checks whether the word  $u_i^k := a_1^k b_1^k \cdots a_i^k b_i^k$  causes a transformation in  $S_z$ . If yes, then Alice starts the next round. If no, the transformation caused by  $u_i^k$  lies in some subsemigroup  $S_y$  with  $y \neq z$  (recall that  $S = \bigcup_{y \in Y} S_y$  by item (S1) in Definition 2). For each letter  $a \in \Sigma$ , denote by  $y(a)$  the element of the semilattice  $Y$  such that the transformation  $\tau_a$  lies in the subsemigroup  $S_{y(a)}$ . Take any transformation  $\tau \in S_z$ . By Definition 1,  $\tau = \tau_{a_1} \tau_{a_2} \cdots \tau_{a_n}$  for some  $a_1, a_2, \dots, a_n \in \Sigma$  whence by item (S3) in Definition 2,  $z = y(a_1)y(a_2) \cdots y(a_n)$ . If  $y \leq y(a_i)$  for all  $i = 1, 2, \dots, n$ , then  $y \leq y(a_1)y(a_2) \cdots y(a_n) = z$ , and this would contradict the choice of  $z$  as the least element with respect to  $\leq$  and the assumption  $y \neq z$ . Thus, there must be a letter  $a \in \Sigma$  such that  $y \not\leq y(a)$ , and Alice chooses any such letter  $a$  as  $a_{i+1}^k$ .

By the construction, if  $S_y \neq S_z$ , then the index  $x$  of the subsemigroup  $S_x$  containing the transformation caused by the word  $u_{i+1}^k$  is strictly less than  $y$  in the partially ordered set  $(Y; \leq)$ . Hence, for each  $k$ , the number  $\ell_k$  of pairs of moves in the  $k$ -th round does not exceed the maximum length of strictly decreasing chains in  $(Y; \leq)$ , and the transformation caused by the word  $u_{\ell_k}^k$  lies in the subsemigroup  $S_z$ . Then the transformation  $\tau_w$  caused by the word  $w := u_{\ell_1}^1 u_{\ell_2}^2 \cdots u_{\ell_m}^m$  is a product of  $m$  elements of  $S_z$  and so  $\tau_w$  belongs to  $\text{Ker} S_z$  as the semigroup  $S_z$  is  $m$ -nilpotent over its kernel. Since  $\mathcal{A}$  is a synchronizing automaton, the kernel  $\text{Ker} S$  of its transition monoid consists of constant transformations by Lemma 1. From the inclusion  $\text{Ker} S_z \subseteq \text{Ker} S$  registered above, we conclude that  $\tau_w$  is a constant transformation, and so  $w$  is a reset word for  $\mathcal{A}$ .  $\square$

## 4 Relations to Earlier Results and Future Work

### 4.1 Corollaries

In the introduction, we mentioned a few previously known families of A-automata. Now we show that all these families consist of DS-automata so their winning strategies are subsumed by that of Theorem 3.

**Definite automata.** This DFA family was introduced by some of the pioneers of automata theory back in 1963 [14]. In [14], the term ‘automaton’ meant a recognizer, that is, a DFA with a designated initial state and a distinguished set of final states. However, DFAs without initial and final states as defined in the present note also appeared in [14] but under the name ‘transition tables’. The following is [14, Definition 13] stated in our terminology and notation.

**Definition 3.** A DFA  $(Q, \Sigma)$  is *weakly  $k$ -definite* if for every word  $w$  of length at least  $k$  over  $\Sigma$ ,  $q \cdot w = q' \cdot w$  for all  $q, q' \in Q$ . A DFA is  *$k$ -definite* if it is weakly  $k$ -definite but not weakly  $(k - 1)$ -definite. A DFA is *definite* if it is  $k$ -definite for some  $k$ .

By Definition 3, every definite DFA is synchronizing and any word of length at least  $k$  is a reset word for every  $k$ -definite DFA. Therefore, Alice wins on every definite automaton by selecting her moves at random.

The transition monoid of a definite DFA is nilpotent over its kernel. This fact is implicitly contained in [14] and in the explicit form, it is a part of [20, Theorem 3]. Comparing it with Lemma 2, we see that definite automata constitute a special subfamily of DS-automata. Moreover, for definite automata, Alice’s winning strategy from Theorem 3 specializes exactly to the random choice of moves. In fact, a DFA is definite if and only if Alice can win by pure random choices of moves.

**Commutative automata.** A DFA is said *commutative* if its transition monoid is commutative, that is, satisfy the law  $xy = yx$ . Synchronizing commutative automata were considered in [15, 16, 10]. A simple winning strategy for Alice in synchronization games on such automata was suggested in [7, Theorem 5.2]: Alice must just choose letters spelling a reset word. Hence, as in the previous case, Alice can completely ignore the moves of Bob.

Obviously, on any commutative semigroup, Green’s relations  $\mathfrak{R}$  and  $\mathfrak{L}$  coincide with each other, and hence, with the relation  $\mathfrak{D}$ . If an element  $a$  of a commutative semigroup  $S$  is regular, then  $a = a^2s$  for some  $s \in S$  whence  $a \mathfrak{R} a^2$ . By [9, Theorem 7], this ensures that the  $\mathfrak{D}$ -class containing  $a$  is a subsemigroup (and even a subgroup) of  $S$ . Thus, in all commutative semigroups, regular  $\mathfrak{D}$ -classes are subsemigroups. Therefore, commutative DFAs are DS-automata, and Theorem 3 applies to commutative synchronizing DFAs.

**Weakly acyclic automata.** Let  $\mathcal{A} = (Q, \Sigma)$  be a DFA and  $p, q \in Q$ . We say that  $q$  is *reachable from  $p$  in  $\mathcal{A}$*  if either  $p = q$  or there exists a word  $w$  over  $\Sigma$  such that  $q = p \cdot w$ . The reachability relation of any DFA is reflexive and transitive. A DFA is called *weakly acyclic* if its reachability relation is a partial order.

Various properties of synchronizing weakly acyclic DFAs were considered in [17, 11]. A winning strategy for Alice in synchronization games on such automata was suggested in [7, Theorem 2.3].

By [5, Proposition 6.2] the transition monoid  $T(\mathcal{A})$  of any weakly acyclic DFA  $\mathcal{A}$  is  $\mathfrak{R}$ -trivial, which means that Green’s relation  $\mathfrak{R}$  on  $T(\mathcal{A})$  coincides with the equality relation. It is well known that regular  $\mathfrak{D}$ -classes are subsemigroups in any  $\mathfrak{R}$ -trivial semigroup, but we failed to locate a source where

this fact was formulated such that it would be convenient to refer to it. Therefore we state it as a lemma and provide a proof.

**Lemma 4.** *In every  $\mathfrak{R}$ -trivial semigroup, regular  $\mathfrak{D}$ -classes are subsemigroups.*

*Proof.* Let  $S$  be an  $\mathfrak{R}$ -trivial semigroup and let  $D$  be its regular  $\mathfrak{D}$ -class. Every element  $b \in D$  is regular, so that  $b = btb$  for some  $t \in S$ . Then  $b \mathfrak{R} bt$  whence  $b = bt$  since  $S$  is  $\mathfrak{R}$ -trivial. We have  $b^2 = btb = (bt)t = bt = b$ . Now if  $a \in D$ , then  $a \mathfrak{L} b$  since  $\mathfrak{D} = \mathfrak{L}$  in every  $\mathfrak{R}$ -trivial semigroup. By the definition of the relation  $\mathfrak{L}$ , we have either  $a = b$  or  $a = sb$  for some  $s \in S$ . If  $a = b$ , then  $ab = b^2 = b = a$ . If  $a = sb$ , multiplying through on the right by  $b$ , we get  $ab = sb^2 = sb = a$ . Thus,  $ab = a \in D$  for arbitrary  $a, b \in D$ , whence  $D$  is a subsemigroup.  $\square$

Thus, weakly acyclic DFAs are DS-automata, and Theorem 3 applies again.

## 4.2 Open Questions

Theorem 3 generalizes several earlier results on A-automata. Can it be further generalized? This is an interesting question, but it requires specification.

We mentioned in Section 2.1 that synchronization is a property of transition monoids. This is not true for the property of being an A-automaton. Moreover, for every synchronizing DFA  $\mathcal{A} = (Q, \Sigma)$ , there exists an A-automaton  $\mathcal{A}' = (Q, \Sigma')$  such that  $T(\mathcal{A}) = T(\mathcal{A}')$ . To see this, let  $\Sigma' := \Sigma \cup \{c\}$  where the action of the added letter  $c$  coincides with the action of a fixed reset word for  $\mathcal{A}$ . The transformations caused by the letters in  $\Sigma'$  generate the same submonoid of the monoid of all transformations on the set  $Q$  as do the transformations caused by the letters in  $\Sigma$ . Still, Alice instantly wins the synchronization game on  $\mathcal{A}'$  by choosing the letter  $c$  on her first move.

Thus, one cannot hope for a characterization of A-automata in terms of transition monoids. One can, however, try to find new sets  $\mathbf{P}$  of finite semigroups such that  $\mathbf{DS} \subsetneq \mathbf{P}$  and every synchronizing DFA whose transition monoid lies in  $\mathbf{P}$  is an A-automaton.

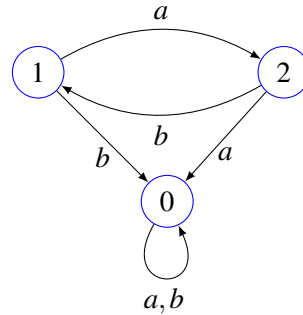
It is easy to verify that the property of being an A-automaton is inherited by subautomata, homomorphic images, and finite direct products. This suggests looking for sets  $\mathbf{P}$  closed under corresponding operations with semigroups. A set of finite semigroups closed under forming finite direct products and taking subsemigroups and homomorphic images is called a *pseudovariety*; the set  $\mathbf{DS}$  is an example of a pseudovariety. Using the notion of a pseudovariety, we can specify a possible direction towards generalizing Theorem 3 as follows:

**Question 1.** *Is there a pseudovariety  $\mathbf{P}$  of finite semigroups that strictly contains  $\mathbf{DS}$  while all synchronizing DFAs with transition monoids in  $\mathbf{P}$  are A-automata?*

The 5-element *Brandt semigroup*  $B_2$  consists of the following five  $2 \times 2$ -matrices, multiplied according to the usual rule:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The monoid  $B_2^1$  obtained by adding the identity  $2 \times 2$ -matrix to  $B_2$  is the syntactic monoid of the language  $(ab)^*$ , and hence, the transition monoid of the minimal automaton  $\mathcal{M}$  of this language (shown below).



It is known that every pseudovariety of finite semigroups not contained in **DS** must include the semigroup  $B_2$  (this fact occurs as Exercise 8.1.6 in [1]; the solution to this exercise follows from the proof of [13, Theorem 3]). Thus, if Bob had a winning strategy on the automaton  $\mathcal{M}$ , then the answer to Question 1 would be ‘No’, and moreover, **DS** would be the largest pseudovariety **P** with the property that Alice can win the synchronization game on every synchronizing DFA whose transition monoid lies in **P**. However, it is easy to see that Alice wins on  $\mathcal{M}$ : she can start with choosing  $a$ ; if Bob replies with  $a$ , he loses, and if he replies with  $b$ , Alice wins by choosing  $b$ .

The fact that  $\mathcal{M}$  is an A-automaton, along with other examples of A-automata beyond the family of DS-automata, indicates that Question 1 might have an affirmative answer. We have a candidate pseudovariety to witness such an answer but its definition requires more structure theory of semigroups than assumed here.

Another question of interest concerns the speed of synchronization for A-automata. When we mentioned in the introduction that A-automata seem more amenable to synchronization, we meant that they tend to have short reset words. Indeed, in all examples we know, an A-automaton with  $n$  states admits a reset word of length at most  $n - 1$ . For DFAs in the range of Theorem 3, that is, synchronizing DS-automata, this was established in [3, Theorem 2.6]. The DFA  $\mathcal{M}$  with 3 states is reset by the words  $a^2$  and  $b^2$  of length 2 and hence provides another example. These observations lead to the next question.

**Question 2.** *Is each A-automaton with  $n$  states reset by a word of length  $n - 1$ ?*

Recall that for each  $n > 2$ , there exist synchronizing DFAs with  $n$  states whose shortest reset words have length  $(n - 1)^2$ ; see [6, Lemma 1]. It can be verified that none of these ‘slowly synchronizing’ DFAs are A-automata.

**Acknowledgement.** The authors are grateful to the anonymous referees for their attention and remarks.

## References

- [1] Almeida, Jorge: Finite Semigroups and Universal Algebra. Series in Algebra, vol. 3. World Scientific, Singapore (1994)
- [2] Almeida, Jorge, Margolis, Stuart, Steinberg, Benjamin, Volkov, Mikhail: Representation theory of finite semigroups, semigroup radicals and formal language theory, Transactions of the American Mathematical Society **361**:3, 1429–1461 (2009). <https://doi.org/10.1090/S0002-9947-08-04712-0>
- [3] Almeida, Jorge, Steinberg, Benjamin: Matrix mortality and the Černý–Pin conjecture. In: Diekert, Volker, Nowotka, Dirk (eds.), Developments in Language Theory, 13th International Conference (DLT 2009), Lecture Notes in Computer Science, vol. 5583, pp. 67–80. Springer, Berlin (2009). [https://doi.org/10.1007/978-3-642-02737-6\\_5](https://doi.org/10.1007/978-3-642-02737-6_5)



- [4] Blass, Andreas, Gurevich, Yuri, Nachmanson, Lev, Veanes, Margus: Play to test. In: Grieskamp, Wolfgang, Weise, Carsten (eds.), *Formal Approaches to Software Testing, 5th International Workshop (FATES 2005)*, Lecture Notes in Computer Science, vol. 3997, pp. 32–46. Springer, Berlin, Heidelberg (2006). [https://doi.org/10.1007/11759744\\_3](https://doi.org/10.1007/11759744_3)
- [5] Brzozowski, Janusz A., Fich, Faith E.: Languages of  $\mathcal{R}$ -trivial monoids. *Journal of Computer and System Sciences* **20**:1, 32–49 (1980). [https://doi.org/10.1016/0022-0000\(80\)90003-3](https://doi.org/10.1016/0022-0000(80)90003-3)
- [6] Černý, Jan: Poznámka k homogénnym experimentom s konečnými automatami. *Matematicko-fyzikálny Časopis Slovenskej Akadémie Vied* **14**(3): 208–216 (1964) (in Slovak; English translation: A note on homogeneous experiments with finite automata. *Journal of Automata, Languages, and Combinatorics* **24**:2-4, 123–132 (2019). <https://doi.org/10.25596/jalc-2019-123>)
- [7] Fernau, Henning, Haase, Carolina, Hoffmann, Stefan: The synchronization game on subclasses of automata. In: Fraigniaud, Pierre, Uno, Yushi (eds.), *11th International Conference on Fun with Algorithms (FUN 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 226, pp. 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl (2022). <https://doi.org/10.4230/LIPIcs.FUN.2022.14>
- [8] Fominykh, Fedor M., Martyugin, Pavel V., Volkov, Mikhail V.: P(l)aying for synchronization, *International Journal of Foundations of Computer Science* **24**:6, 765–780 (2013). <https://doi.org/10.1142/S0129054113400170>
- [9] Green, James Alexander: On the structure of semigroups. *Annals of Mathematics, Second Series* **54**:1, 163–172 (1951). <https://doi.org/10.2307/1969317>
- [10] Hoffmann, Stefan: Constrained synchronization and commutativity. *Theoretical Computer Science* **890**, 147–170 (2021). <https://doi.org/10.1016/j.tcs.2021.08.030>
- [11] Hoffmann, Stefan: Constrained synchronization and subset synchronization problems for weakly acyclic automata, in: Moreira, Nelma, Reis, Rogério (eds.), *Developments in Language Theory, 25th International Conference (DLT 2021)*, Lecture Notes in Computer Science, vol. 12811, pp. 204–216. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-81508-0\\_17](https://doi.org/10.1007/978-3-030-81508-0_17)
- [12] Kari, Jarkko, Volkov, Mikhail: Černý’s conjecture and the road colouring problem. In: Pin, Jean-Éric (ed.), *Handbook of Automata Theory*, vol. I, pp. 525–565. EMS Publishing House, Zürich (2021). <https://doi.org/10.4171/AUTOMATA-1/15>
- [13] Margolis, Stuart W. On M-varieties generated by power monoids. *Semigroup Forum* **22**, 339–353 (1981). <https://doi.org/10.1007/BF02572813>
- [14] Perles, Micha, Rabin, Michael O., Shamir, Eliahu: The theory of definite automata. *Transactions on Electronic Computers* **12**, 233–243 (1963). <https://doi.org/10.1109/PGEC.1963.263534>
- [15] Rystsov, Igor: Exact linear bound for the length of reset words in commutative automata. *Publicationes Mathematicae Debrecen* **48**:3-4, 405–409 (1996)
- [16] Rystsov, Igor: Reset words for commutative and solvable automata. *Theoretical Computer Science* **172**:1–2, 273–279 (1997). [https://doi.org/10.1016/S0304-3975\(96\)00136-3](https://doi.org/10.1016/S0304-3975(96)00136-3)
- [17] Ryzhikov, Andrew: Synchronization problems in automata without non-trivial cycles. *Theoretical Computer Science* **787**, 77–88 (2019). <https://doi.org/10.1016/j.tcs.2018.12.026>
- [18] Shevrin, Lev N.: On the theory of epigroups. I. *Russian Academy of Sciences. Sbornik. Mathematics* **82**:2, 485–512 (1995). <https://doi.org/10.1070/SM1995v082n02ABEH003577>
- [19] Volkov, Mikhail V.: Synchronization of finite automata. *Russian Mathematical Surveys* **77**:5, 819–891 (2022). <https://doi.org/10.4213/rm10005e>
- [20] Zalcstein, Yechezkel: Locally testable languages. *Journal of Computer and System Sciences* **6**, 151–167 (1972). [https://doi.org/10.1016/S0022-0000\(72\)80020-5](https://doi.org/10.1016/S0022-0000(72)80020-5)