Reset Complexity and Completely Reachable Automata with Simple Idempotents

Stefan Hoffmann(✉)

Informatikwissenschaften, FB IV, Universität Trier, Trier, Germany
hoffmanns@informatik.uni-trier.de

Abstract. Every regular ideal language is the set of synchronizing words of some automaton. The reset complexity of a regular ideal language is the size of such an automaton with the minimal number of states. The state complexity is the size of a minimal automaton recognizing a regular language in the usual sense. There exist regular ideal languages whose state complexity is exponentially larger than its reset complexity. We call an automaton sync-maximal, if the reset complexity of the ideal language induced by its set of synchronizing words equals the number of states of the automaton and the gap between the reset complexity and the state complexity of this language is maximal possible. An automaton is completely reachable, if we can map the whole state set to any non-empty subset of states (for synchronizing automata, it is only required that the whole state set can be mapped to a singleton set). We first state a general structural result for sync-maximal automata. This shows that sync-maximal automata are closely related to completely reachable automata. We then investigate automata with simple idempotents and show that for these automata complete reachability and sync-maximality are equivalent. Lastly, we find that for automata with simple idempotents over a binary alphabet, subset reachability problems that are PSPACE-complete in general are solvable in polynomial time.

Keywords: finite automata · synchronization · set of synchronizing words · automata with simple idempotents · completely reachable automata · sync-maximal automata

1 Introduction

Let Σ be a *finite set of symbols* and Σ^* be the *free monoid* with neutral element ε (called the *empty word*). *Languages* are subsets of Σ^*. A language $I \subseteq \Sigma^*$ is an *ideal language* if $x, y \in \Sigma^*$ and $u \in I$ imply $xuy \in I$. A *(semi-)automaton* is a triple $\mathcal{A} = (Q, \Sigma, \delta)$ where Q is *finite set of states* and $\delta : Q \times \Sigma \to Q$ a (totally defined) *transition function*. Here, we simplify our notation by not mentioning δ explicitly, i.e., we write $q.a$ when applying the transition function δ to the state $q \in Q$ and letter $a \in \Sigma$ and we write an automaton as a 2-tuple $\mathcal{A} = (Q, \Sigma)$.

Published by Springer Nature Switzerland AG 2022
Y.-S. Han and G. Vaszil (Eds.): DCFS 2022, LNCS 13439, pp. 85–99, 2022.
https://doi.org/10.1007/978-3-031-13257-5_7

The transition function is extended to a function $Q \times \Sigma^* \to Q$ (denoted by the dot notation as well) by setting $q.\varepsilon = q$ and $q.ua = (q.u).a$ for $u \in \Sigma^*$, $a \in \Sigma$ and $q \in Q$. Furthermore, we extend it to subsets $S \subseteq Q$ by setting $S.u = \{q.u \mid q \in S\}$ for $u \in \Sigma^*$.

A language $L \subseteq \Sigma^*$ is a *regular language* if there exists an automaton $\mathcal{A} = (Q, \Sigma)$ and $q_0 \in Q$ and $F \subseteq Q$ such that $L = \{u \in \Sigma^* : q_0.u \in F\}$. We say the automaton $\mathcal{A}$ *accepts* (or *recognizes*) the language L. For a regular language $L \subseteq \Sigma^*$ we denote by $\mathrm{sc}(L) = \min\{|Q| : \mathcal{A} = (Q, \Sigma) \text{ accepts } L\}$ the *state complexity* of L.

Set $\mathrm{Syn}(\mathcal{A}) = \{u \in \Sigma^* \mid |Q.u| = 1\}$, the *set of synchronizing words* of $\mathcal{A}$ (which is an ideal language). If $\mathrm{Syn}(\mathcal{A}) \neq \varnothing$, the automaton is called *synchronizing*. For every n-state automaton $\mathcal{A}$ we have $\mathrm{sc}(\mathrm{Syn}(\mathcal{A})) \leqslant 2^n - n$ [15]. We call $\mathcal{A}$ *sync-maximal* if $\mathrm{sc}(\mathrm{Syn}(\mathcal{A}))$ equals $2^n - n$. If $I \subseteq \Sigma^*$ is a regular ideal language, then it is precisely the set of synchronizing words of the automaton with the least number of states accepting it [9,15]. The *reset complexity* of I is

$$\mathrm{rc}(I) = \min\{|Q| : \mathcal{A} = (Q, \Sigma) \text{ with } I = \mathrm{Syn}(\mathcal{A})\},$$

i.e., the least number of states such that I is realized as the set of synchronizing word of an automaton. We have $\mathrm{rc}(I) \leqslant \mathrm{sc}(I)$. There are known families of ideal languages I_n associated to synchronizing automata such that $\mathrm{sc}(I_n) = 2^n - n$ but $\mathrm{rc}(I_n) = n$ [15]. For example, the set of synchronizing words of the *Černý family* of automata $\mathcal{C}_n = (Q_n, \{a, b\})$ defined by Černý [6] for $n > 1$ by

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

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

Hence, describing regular ideal languages as the set of synchronizing words can be exponentially more succinct than the usual notion of acceptance by automata.

For $L \subseteq \Sigma^*$, let $||L|| = \min\{|u| \mid u \in L\}$ be the length of a shortest word in L. In combinatorial automata theory (and also in related applications, see [19] for a survey in the context of model-based testing), the question on the length of shortest synchronizing words arises. The *Černý conjecture* states that for an n-state automaton $\mathcal{A}$ we have $||\mathrm{Syn}(\mathcal{A})|| \leqslant (n-1)^2$. For the n-state automata $\mathcal{C}_n$ from Fig. 1 we have $||\mathrm{Syn}(\mathcal{C}_n)|| = (n-1)^2$. The best general upper bound proven so far is cubic in the number of states [21]. For more information and further references, see the recent survey [22]. Investigating the length of shortest words in regular ideal languages yields a natural approach to the Černý conjecture, more precisely it is equivalent to the statement that $\mathrm{rc}(I) \geqslant \sqrt{||I||} + 1$ for every regular ideal language $I \subseteq \Sigma^*$.

In fact, the connection between regular ideal languages and synchronizing automata is even deeper. It can be shown that every regular ideal language equals the set of synchronizing word of a strongly connected automaton [16].

Automata with simple idempotents have been introduced in [18] and it has been shown that a shortest synchronizing word has at most quadratic length for these automata.

Overview and Contribution. In Sect. 2 we introduce automata with simple idempotents and further notation that was not already introduced in the introduction. Then we make the conditions for sync-maximality more precise. Example 1 gives an automaton that is completely reachable, but not sync-maximal.

Section 3 discusses completely reachable automata.

In Sect. 4 we determine the structure of sync-maximal automata. Our results show that both notions are closely connected, as every sync-maximal automaton contains a completely reachable subautomaton. In Example 2 we give automata that are sync-maximal but not completely reachable.

Then in Sect. 5 we investigate automata with simple idempotents. Hence, this chapter is concerned with the reset complexity of ideal languages induced by automata with simple idempotents. However, we do not mention ideal languages anymore, but rather express our result directly with automata (ideal languages were introduced in the introduction to give a broader context of our results). We show that an automaton with simple idempotents that is completely reachable is sync-maximal. Note that Example 2 and Example 3 give automata with simple idempotents that are sync-maximal, but not completely reachable. However, for strongly connected automata with simple idempotents, it follows that complete reachability and sync-maximality are equivalent.

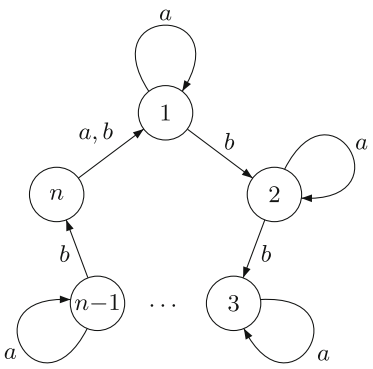

Fig. 1. The automaton $\mathcal{C}_n$

2 Some More Notation and Preliminary Results

Let $f : X \to Y$ be a function. Here, function application is written on the right, i.e., xf or $(x)f$ denotes the function f applied to x. The same applies to the extension to subsets, i.e., if $S \subseteq X$, then $(S)f$ and Sf denote the set $\{xf \mid x \in S\}$. In this respect, if $g : Y \to Z$ is another function, the *function*

composition fg is the function $x(fg) = (xf)g$, i.e. f is applied first. This "right action notation" deviates from the more usual "left action notation" $f(x)$ used in formal language theory. We chose this notation as, in our opinion, it makes certain algebraic manipulations in Sect. 5 easier to read, as it conforms better with the way function composition is defined and read from left to right. This notation[1] is actually quite common in more algebraic approaches, for example, in [1].

For $n \geqslant 0$, we set $[0] = \varnothing$ and $[n] = \{1, \ldots, n\}$ if $n > 0$. If $f : [n] \to [n]$ is a permutation, i.e., a bijective mapping, then $f^{-1} : [n] \to [n]$ denotes the *inverse mapping* with $xff^{-1} = xf^{-1}f = x$ for all $x \in [n]$.

For algebraic notions as semigroup, monoid, generating set etc., we refer to the literature, e.g., the textbook [14]. By $\mathfrak{T}_n$ we denote the *transformation monoid of all mappings on a finite set* of cardinality n.

Sets with precisely k elements are called *k-sets*, and 1-sets are also called *singleton sets*.

For $f : X \to Y$ with X, Y finite, the *defect* is $|X \setminus \{xf \mid x \in X\}|$.

A mapping $f : X \to X$ on a finite set X is a *simple idempotent (mapping)* if it has defect one and for each $x \in X$ we have $xff = xf$. Note that a simple idempotent mapping is completely specified by the two points $x \in X$ and xf with $xf \neq x$. In Sect. 5 we need the following "cancellation property" of simple idempotent mappings.

Lemma 1. *Let $f : [n] \to [n]$ be a simple idempotent mapping and $A, B \subseteq [n]$ with $|A| = |B|$. If $Af = Bf$ and there exist $x \in A$, $y \in B$ such that $xf \neq x$ and $yf \neq y$, then $A = B$ and $x = y$.*

Remark 1. Let $f : \{1, 2\} \to \{1, 2\}$ with $(1)f = (2)f = 2$. Then $\{1\}f = \{1, 2\}f$ and $\{1\}f = \{2\}f$. The former equation shows that the assumption $|A| = |B|$ is necessary in Lemma 1, the latter equation shows the assumption that the element that is moved by f is contained in both sets is necessary.

Let $\mathcal{A} = (Q, \Sigma)$ be an automaton. The *defect* of a letter $a \in \Sigma$ is the defect of the induced function $q \mapsto q.a$ for $q \in Q$. A *simple idempotent letter* is a letter that induces a simple idempotent mapping on the states, and a *permutational letter* is a letter that induces a permutation on the state set. The automaton $\mathcal{A}$ is an *automaton with simple idempotents* if every letter is either a permutational letter or a simple idempotent letter. A subset $S \subseteq Q$ defines (or induces) a *subautomaton* if $s.a \in S$ for each $s \in S$ and $a \in \Sigma$. In this case, the set S together with the transition function restricted to S in the arguments and the image gives a totally defined function, i.e., we can regard it as an automaton on its own.

The *transformation monoid of the automaton* $\mathcal{A}$ is the monoid generated by the mappings $q \mapsto q.a$ for $q \in Q$ induced by each letter $a \in \Sigma$ on the state set.

[1] Note that we actually mix both notations, as we write certain operators (which are never composed here), for example $\mathrm{Syn}(\mathcal{A})$, in the other convention. But this is also done in the literature, for example [1], and should pose no problem.

A state $t \in Q$ is *reachable* from another state $s \in Q$ if there exists $u \in \Sigma^*$ such $s.u = t$. A *strongly connected component* is a maximal subset of states such that for every two states in this subset are reachable from each other. The automaton $\mathcal{A}$ is *strongly connected* if Q is a strongly connected component.

An automaton $\mathcal{A}$ is minimal [13] for $\{u \in \Sigma^* \mid q_0.u \in F\}$ with $q_0 \in Q$ and $F \subseteq Q$ if and only if every state is reachable from q_0 and every pair of distinct state $q, q' \in Q$ is *distinguishable*, which means that there exists $u \in \Sigma^*$ such that precisely one of the two states $q.u$ and $q'.u$ is final, i.e., the following holds true: $q.u \in F$ if and only if $q'.u \notin F$.

The *power automaton* is $\mathcal{P}_\mathcal{A} = (\{S \mid \varnothing \neq S \subseteq Q\}, \Sigma)$ where the transition function of $\mathcal{P}_\mathcal{A}$ is the transition function of $\mathcal{A}$ extended to subsets. The automaton $\mathcal{A}$ is *completely reachable*, if every non-empty subset is reachable from the state Q in the power automaton of $\mathcal{A}$. Setting $F = \{\{q\} \mid q \in Q\}$, as $\mathrm{Syn}(\mathcal{A}) = \{u \in \Sigma^* \mid Q.u \in F\}$, the power automaton accepts $\mathrm{Syn}(\mathcal{A})$. The states in F can be merged into a single state to get another automaton accepting $\mathrm{Syn}(\mathcal{A})$. Hence, $\mathrm{sc}(\mathrm{Syn}(\mathcal{A})) \leqslant 2^n - n$.

Translating the condition of minimality of an automaton to the specific language $\mathrm{Syn}(\mathcal{A})$ and the power automaton, we find that the sync-maximality of $\mathcal{A}$ is equivalent to the following two conditions:

1. Every non-empty subset with at least two states is reachable in $\mathcal{P}_\mathcal{A}$ and at least one singleton subset is reachable in $\mathcal{P}_\mathcal{A}$.
2. For any two non-empty and distinct subsets $S, T \subseteq Q$ with $\min\{|S|, |T|\} \geqslant 2$ there exists a word $u \in \Sigma^*$ such that precisely one of the two subsets $T.u$ and $S.u$ is a singleton subsets, i.e., both subset are distinguishable (in $\mathcal{P}_\mathcal{A}$).

In [10, Lemma 3.1] (and a little more general in [12, Theorem 7]) it was shown that distinguishability of states in the power automaton with respect to $\mathrm{Syn}(\mathcal{A})$ can be simplified by only considering 2-sets.

Proposition 2. *Let $\mathcal{A} = (Q, \Sigma)$. Then all states in the power automaton $\mathcal{P}_\mathcal{A}$ are distinguishable if and only if all 2-sets are distinguishable in $\mathcal{P}_\mathcal{A}$.*

Example 1. The automaton (see Fig. 2) with the state set $Q = \{1, 2, 3\}$, input letters $a_{[1]}, a_{[2]}, a_{[3]}, a_{[1,2]}$ and transition function given by

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

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

is taken from [3, Example 2] as an example of a completely reachable automaton. However, it is not sync-maximal as the two 2-sets $\{1, 3\}$ and $\{2, 3\}$ are not distinguishable.

Next, we introduce two decision problems that have been investigated in [2,3,17]. Subsets as in the latter problem were called *totally extensible* in [2],

and in [17] the problem was called the *global inclusion problem for non-initial automata*.

Definition 3. Reachable Subset
Input: $\mathcal{A} = (Q, \Sigma)$ *and* $\varnothing \neq S \subseteq Q$.
Question: *Exists* $w \in \Sigma^*$ *with* $Q.w = S$*?*

Definition 4. Sync-Into-Subset
Input: $\mathcal{A} = (Q, \Sigma)$ *and* $S \subseteq Q$.
Question: *Exists* $w \in \Sigma^*$ *with* $Q.w \subseteq S$*?*

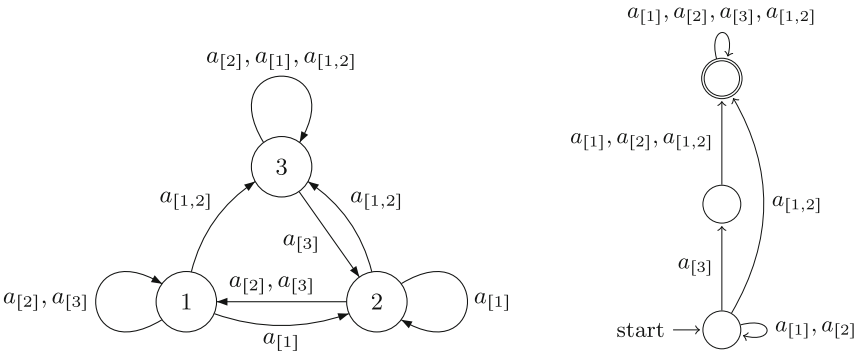

Fig. 2. Left: A completely reachable automaton (taken from [3, Example 2]) that is not sync-maximal. Right: A minimal automaton for the set of synchronizing words.

Lastly, we mention the following easy facts [10] that will be used without special mentioning.

Lemma 5. *A strongly connected sync-maximal automaton is completely reachable. A completely reachable automaton is strongly connected.*

3 Completely Reachable Automata

The notion of a completely reachable automaton was introduced in [3], based on a sufficient condition for the reachability of all subsets in circular automata [7]. This sufficient condition was generalized in this very first work [3], and later extended to a characterization with more general constructions in [4]. An extended version of [3,4] (and with further results) is under submission and a preliminary version available on arXiv [5]. Complete reachability has been used in [10] to characterize primitive permutation groups.

Given a finite automaton and two subsets of states, it is complete for deterministic polynomial space to check if there exists a word mapping one subset onto the other, see [3,17]. This implies that complete reachability can be checked in non-deterministic polynomial space by checking if a given automaton is not completely reachable in the following way: non-deterministically guess a non-empty subset and check if it is reachable from the whole state set, if not, then the automaton is not completely reachable. By Savitch's theorem [20] this problem is solvable in deterministic polynomial space as well, which implies that

complete reachability is decidable in deterministic polynomial space. However, the precise computational complexity of deciding complete reachability is an open problem [3–5].

With Proposition 2, which yields a polynomial time procedure to check distinguishability of all non-empty subset [10, Corollary 3.2], we get the next result.

Proposition 6. *Deciding if a given automaton is sync-maximal can be done in polynomial space.*

In [8] a completely reachable automaton with letters of defect one was given for which the sufficient condition from [3] is not fulfilled, which was meant to be a counter-example to a conjecture from [3]. However, with another result from [8, Theorem 20] it can be deduced that complete reachability is decidable in polynomial time for automata with simple idempotents.

Theorem 7. *For automata $\mathcal{A} = (Q, \Sigma)$ with simple idempotents it is decidable in polynomial time if they are completely reachable.*

Proof (sketch). In [3] a sufficient (graph-theoretical) condition for complete reachability was stated, and in [8] it was shown that this sufficient condition can be checked in polynomial time. Furthermore [8, Theorem 20] states that the following implies the mentioned sufficient condition: For every proper non-empty $S \subseteq Q$ there exists $w \in \Sigma^*$ with $S = Q.w$ and $w_1, w_2 \in \Sigma^*$ with $w = w_1 w_2$ such that $|Q.w| + 1 = |Q.w_1|$ and w_2 has defect one. Now, it can be shown that this is fulfilled for completely reachable automata with simple idempotents. Combining these facts yields the claim. □

4 The Structure of Sync-Maximal Automata

Here, we determine the structure of sync-maximal automata. We show that they are either completely reachable or consist of precisely two strongly connected components, where one contains only a single "dangling state" and the other component forms a completely reachable subautomaton.

Theorem 8. *Let $\mathcal{A} = (\Sigma, Q)$ be an n-state semi-automaton with $n \geqslant 3$. If $\mathcal{A}$ is sync-maximal, i.e., the smallest recognizing automaton for* $\mathrm{Syn}(\mathcal{A})$ *has $2^n - n$ states, then either $\mathcal{A}$ is completely reachable or all of the the following statements hold true:*

1. $|\Sigma| \geqslant 3$,
2. *we have two strongly connected components $\{q\}$, $q \in Q$, and $S = Q \setminus \{q\}$,*
3. *there exists $a \in \Sigma$ with*[2] *$q.a \in S$ and such that $q'.a \neq q.a$ for at least one state $q' \in S$,*
4. *there exists $b \in \Sigma$ having defect one and $c \in \Sigma \setminus \{b\}$ with $q.b = q.c = q$,*
5. *if $|\Sigma| = 3$ and $n \geqslant 4$, then the letter b cyclically permutes S.*

[2] Observe that as $\{q\}$ and S are strongly connected components, the condition $q.a \in S$ implies that the state q is not reachable from any state in S and so $Q.u \neq \{q\}$ for all $u \in \Sigma^*$.

Proof. As $\mathrm{Syn}(\mathcal{A}) \neq \varnothing$, there must exist a state $s \in Q$ and a synchronizing word $w \in \Sigma^*$ such that $Q.w = \{s\}$. Let $S \subseteq Q$ be the strongly connected component containing s. As $q.w = s$ for every $q \in Q$, this strongly connected component is uniquely determined for any choice of a state s such that there exists a word $w \in \Sigma^*$ with $Q.w = \{s\}$. Furthermore, it has the property that, once entered, we cannot leave S, i.e., $S.u \subseteq S$ for all $u \in \Sigma^*$. However, this implies $S \cap Q.u \neq \varnothing$ for every $u \in \Sigma^*$. Hence, no non-empty subset of $Q \backslash S$ is reachable. As by assumption every non-empty subset with at least two elements is reachable, we find $|S| = |Q|$ or $|S| = |Q| - 1$.

In the first case, $\mathcal{A}$ is strongly connected. This implies that if at least one singleton subset is reachable, then all singleton subsets are reachable and so $\mathcal{A}$ is completely reachable.

In the second case, we can write $Q = S \cup \{q\}$ with $q \notin S$. Note that in this case $\mathcal{A}$ is not completely reachable, as $\{q\}$ is not reachable. As $Q.w \in S$, there exists at least one letter mapping q into S.

Let $s, t \in S$ be two arbitrary distinct states. Consider the states $\{q, s\}$ and $\{q, t\}$ in the power automaton. They must be distinguishable, i.e., there must exist a word $u \in \Sigma^*$ mapping precisely one, say $\{q, s\}$, to a singleton set but not the other. Then $S.u \subseteq S$ and we can write $u = u'au''$ with $u', u'' \in \Sigma^*$ and $a \in \Sigma$ such that $q.ua \in S$ and $q.u = q$ and so $q.a \in S$. We must have $|\{q, t\}.ua| = 2$, which implies $t.a \neq q.a$.

Now, suppose $n \geqslant 3$. Then we must have at least two distinct letters $b, c \in \Sigma$ such that $q.b = q.c = q$. To see this, consider a non-empty subset $T \subseteq S$ with $|T| = n - 2$. The subset $T \cup \{q\}$ must be reachable. This is only possible if there exists a letter $b \in \Sigma$ with $q.b = q$ (recall $q \notin S.u$ for each $u \in \Sigma^*$) having defect one, for if every letter fixing q permutes the states or has defect at least two, then no subset of the form $T \cup \{q\}$ with $|T| = n - 2$ is reachable.

Next, consider a subset $T' \cup \{q\}$ with $|T'| = n - 2$ such that $T' \cup \{q\} \neq Q.b$. We cannot use the letter a to reach the subset $T' \cup \{q\}$ as $q \notin Q.a$. Let $i \geqslant 2$. Then $|Q.b^i| \leqslant n - 1$ and $|Q.b^i| = n - 1$ implies, as $Q.b^i \subseteq Q.b$, that $Q.bb = Q.b$. So, there must exist a third letter $c \in \Sigma \backslash \{a, b\}$ with $q.c = q$ to reach the subset $T' \cup \{q\}$.

Lastly, suppose $\Sigma = \{a, b, c\}$. Then a is the only letter such that $q.a \in S$. Furthermore, for each $q' \in S$ the subset $\{q\} \cup S \backslash \{q'\}$ must be reachable. If $Q.ub$ contains q and has size $n - 1$, then, as $Q.ub \subseteq Q.b$, we must have $Q.ub = Q.b$. If $|Q.c| \leqslant n$ and $n \geqslant 3$, then $Q.c$ must be a subset of size $n - 1$ to reach another subset not equal to $Q.b$ of size $n - 1$. However, as shown before for the letter b, if $Q.cc$ has size $n - 1$, then $Q.cc = Q.c$. So, if $n \geqslant 4$ there exists a subset of size $n - 1$ that is not reachable and hence in this case we must have $|Q.c| = n$. Putting all the arguments together, every subset of size $n - 1$ containing q must be reachable by a word of the form bc^i for some $i \geqslant 0$. Let $q' \in S$ such that $Q.b = Q \setminus \{q'\}$ and choose $q'' \in S$. Then there exists $i \geqslant 0$ such that $Q.bc^i = Q \setminus \{q''\}$. As c permutes the states, this implies $q'.c = q''$. However, a single permutation that maps all states in S onto each other is only possible if this permutation induces a single cycle on these states and we conclude that c cyclically permutes the states in S. □

In case a sync-maximal automaton is not completely reachable, then we can show that one strongly connected component forms a completely reachable subautomaton.

Theorem 9. *Let $\mathcal{A} = (Q, \Sigma)$ be a sync-maximal automaton that is not completely reachable. Suppose $q \in Q$ is the "dangling state" that exists according to Theorem 8. Then the states in $Q \setminus \{q\}$ form a strongly connected, completely reachable and sync-maximal subautomaton.*

We can use sync-maximal and completely reachable automata to construct sync-maximal automata that are not completely reachable by adding a dangling state, as done in the next example.

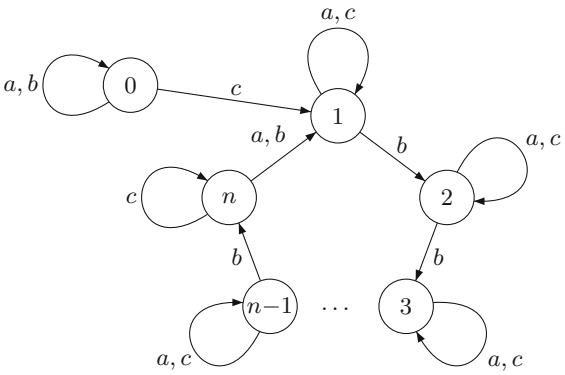

Fig. 3. The automaton $\mathcal{A}_n$ from Example 2

Example 2. We derive from the Černý family a new family of automata by adjoining an additional state and a new letter (see Fig. 3) that give sync-maximal, but not completely reachable (as they are not strongly connected) automata. Let $\mathcal{A}_n = (\{0, 1, \ldots, n\}, \{a, b, c\})$ with

$$i.a = \begin{cases} i & \text{if } i < n, \\ 1 & \text{if } i = n; \end{cases} \quad i.b = \begin{cases} 0 & \text{if } i = 0, \\ i+1 & \text{if } 0 < i < n, \\ 1 & \text{if } i = n. \end{cases} \quad i.c = \begin{cases} 1 & \text{if } i = 0, \\ i & \text{if } i \neq 0. \end{cases}$$

Observe that the automaton induced on the states $\{1, \ldots, n\}$ and by the letters $\{a, b\}$ is precisely the Černy automaton $\mathcal{C}_n$. The Černy automata are completely reachable and sync-maximal (which is implied by results from [11], but also by Theorem 12 of the present work). Hence, all non-empty subsets of $\{1, \ldots, n\}$ are distinguishable. If $S, T \subseteq \{0, 1, \ldots, n\}$ are non-empty and distinct, and at least one, say S, contains the state 0, we can distinguish them the following way: (1) If $0 \notin T$, then choosing any word from $\{a, b\}^*$ that maps T to a singleton distinguishes S and T, as it maps S to a 2-set. (2) If $0 \in T$, then write $S = \{0, s\}$

and $T = \{0, t\}$. By assumption $s \neq t$. There exists $m > 0$ such that $s.b^m = 1$. Then $t.b^m \neq 1$. Hence $S.b^m c = \{1\}$ and $T.b^m a = \{1, t.b^m\}$ is a 2-set. So, S and T are distinguishable.

5 Automata with Simple Idempotents

A strongly connected sync-maximal automaton is also completely reachable [10, Lemma 3.3]. Here, we show that if an automaton with simple idempotents is completely reachable, then it is sync-maximal. Hence, for strongly connected automata with simple idempotents, complete reachability and sync-maximality are equivalent.

Let $f : [n] \to [n]$ be a mapping and $g : [n] \to [n]$ be a permutation, then the *conjugate of* f *by* g, written f^g, is the mapping given by $f^g = g^{-1}fg$. Note that if f is a simple idempotent mapping with $a \neq b$ and $af = b$, then f^g is a simple idempotent mapping ag to bg.

Crucial for our result are the following two Lemmata 10 and 11 formulated in the language of transformation semigroups. The first lemma says that for reachability of subsets, it is sufficient to consider products of simple idempotents.

Lemma 10. *Let $T \leqslant \mathcal{T}_n$ be a transformation semigroup generated by permutations and simple idempotents and $S \subseteq [n]$. If there exists $t \in T$ such that $S = ([n])t$, then there exists a product $t' \in T$ of conjugates of the generators that are simple idempotents by permutations in T such that $S = ([n])t'$. In particular, t' is a product of simple idempotents from T.*

Proof. Write $t = g_1 f_1 g_2 f_2 g_3 \cdots f_{n-1} g_n f_n g_{n+1}$ where the f_i are the simple idempotents from the generating set and the g_i are permutations generated by the permutations in the generating set. Then (this relation was already observed in [1])

$$\begin{aligned} t = g_1 f_1 g_2 f_2 g_3 \cdots f_{n-1} g_n g_{n+1} f_n^{g_{n+1}} &= g_1 f_1 g_2 f_2 g_3 \cdots g_{n-1} g_n g_{n+1} f_{n-1}^{g_n g_{n+1}} f_n^{g_{n+1}} \\ &= \ldots = g_1 g_2 \cdots g_{n+1} f_1^{g_2 g_3 \cdots g_n g_{n+1}} f_2^{g_3 \cdots g_n g_{n+1}} \cdots f_{n-1}^{g_n g_{n+1}} f_n^{g_{n+1}}. \end{aligned}$$

Now, as $g_1 g_2 \cdots g_{n+1}$ is a permutation, we have $([n])(g_1 g_2 \cdots g_{n+1}) = [n]$. Hence, if we set $t' = f_1^{g_2 g_3 \cdots g_n g_{n+1}} f_2^{g_3 \cdots g_n g_{n+1}} \cdots f_{n-1}^{g_n g_{n+1}} f_n^{g_{n+1}}$ we have $S = ([n])t'$. Each conjugate f_i^g where g is a permutation is simple idempotent. Observe that the number of simple idempotent in the resulting product of t' is the same as the number of simple idempotents used in t. □

Next, we give a sufficient condition for the distinguishability of 2-sets.

Lemma 11. *Let $T \leqslant \mathcal{T}_n$ be a transformation semigroup generated by permutations and simple idempotents and containing a constant map. Then for every two distinct 2-sets there exists an element in T mapping precisely one 2-set to a singleton but not the other.*

Proof. Let $\{a,b\},\{c,d\} \subseteq [n]$ be distinct 2-sets. By assumption and Lemma 10, there exists a product of simple idempotents $f \in T$ such that $|\{a,b\}f| = 1$ or $|\{c,d\}f| = 1$. Choose the element f that is expressible as a shortest possible product of simple idempotents, i.e. $f = f_1 \cdots f_m$ with m is minimal, the f_i are simple idempotent mappings and $|\{a,b\}f| = 1$ or $|\{c,d\}f| = 1$.

Assume $|(\{a,b\})f| = |(\{c,d\})f| = 1$.

The function f_m has defect one. Hence the two elements mapped to a single element are unique. By the choice of f as a minimal product and as f_m is applied at the end, we can conclude that $|\{a,b\}(f_1 \cdots f_{m-1})| = |\{c,d\}(f_1 \cdots f_{m-1})| = 2$ and $\{a,b\}(f_1 \cdots f_{m-1}) = \{c,d\}(f_1 \cdots f_{m-1})$.

Let $i \in \{1,\ldots,m-1\}$. Set $h_i = f_1 \cdots f_i$ and let h_0 denote the identity transformation. Assume $\{a,b\}h_i = \{a,b\}h_{i+1}$, then $|\{a,b\}(h_i f_{i+2} \cdots f_m)| = 1$ and f can be written as a product of $m-1$ simple idempotents, contradicting the minimal length of the product. Similarly, we must have $\{c,d\}h_i \neq \{c,d\}h_{i+1}$. Hence, for every $i \in \{0,1,\ldots,m-1\}$ we have

$$\{a,b\}h_i \neq \{a,b\}h_{i+1} \text{ and } \{c,d\}h_i \neq \{c,d\}h_{i+1}. \tag{1}$$

Now, for $i \in \{1,\ldots,m-1\}$, suppose $\{a,b\}h_i = \{c,d\}h_i$. Set $A = \{a,b\}h_{i-1}$ and $B = \{c,d\}h_{i-1}$. By Eq. (1), we have $A \neq Af$ and $B \neq Bf$. So there exist $x \in A$, $y \in B$ such that $xf_i \neq x$ and $yf_i \neq y$. By Lemma 1 we can conclude $A = B$. So, inductively, as $\{a,b\}h_{m-1} = \{c,d\}h_{m-1}$, we find $\{a,b\} = \{c,d\}$. However, this contradicts our assumption that both 2-sets are distinct and so we cannot have that both are mapped to a singleton. □

If we consider the transformation monoid of a given automaton with simple idempotents, Proposition 2 and Lemma 11 directly give the main result of this section.

Theorem 12. *Let $\mathcal{A} = (Q,\Sigma)$ be an automaton with simple idempotents. Then if $\mathcal{A}$ is completely reachable, then it is sync-maximal.*

As every strongly connected sync-maximal automaton is completely reachable, we get the next corollary.

Corollary 13. *Let $\mathcal{A} = (Q,\Sigma)$ be a strongly connected automaton with simple idempotents. Then $\mathcal{A}$ is completely reachable if and only if it is sync-maximal.*

By Theorem 8, every sync-maximal automaton over a binary alphabet is completely reachable, which implies that the automaton is strongly connected. This yields the next corollary.

Corollary 14. *Let $\mathcal{A}$ be an automaton with simple idempotents over a binary alphabet. Then $\mathcal{A}$ is completely reachable if and only if it is sync-maximal.*

The equivalence between complete reachability and sync-maximality holds only for strongly connected automata with simple idempotents and for automata over a binary alphabet. Example 2 gives automata with simple idempotents over a ternary alphabet that are sync-maximal but not completely reachable. Next, we give a different example.

Example 3. Here, we give further examples an automata with simple idempotents that are sync-maximal but not completely reachable. Let the automaton $\mathcal{A} = (\{0, 1, \ldots, n\}, \{a, b, c, d\})$ be such that

$$i.a = \begin{cases} 1 & \text{if } i = 0, \\ i & \text{if } i > 0; \end{cases} \qquad i.b = \begin{cases} 1 & \text{if } i = 2, \\ i & \text{if } i \neq 1; \end{cases}$$

$$i.c = \begin{cases} 1 & \text{if } i = 2, \\ 2 & \text{if } i = 1, \\ i & \text{if } i \notin \{1, 2\}; \end{cases} \qquad i.d = \begin{cases} i+1 & \text{if } i \notin \{0, n\}, \\ 0 & \text{if } i = 0, \\ 1 & \text{if } i = n. \end{cases}$$

Then $\mathcal{A}$ is an automaton with simple idempotents. As $\mathcal{A}$ is not strongly connected, it is not completely reachable. However, it is sync-maximal. This follows as the letters d and c generate the full symmetric group on $\{1, 2, \ldots, n\}$. Let $\{q_1, q_2\}, \{p_1, p_2\}$ be two distinct 2-sets. If $\{q_1, q_2\}, \{p_1, p_2\} \subseteq \{1, 2, \ldots, n\}$, then there exists a word $u \in \{c, d\}^*$ such that $\{q_1, q_2\}.u = \{1, 2\}$ and $\{p_1, p_2\}.u \neq \{1, 2\}$ and the word ub maps $\{1, 2\}$ to $\{2\}$ and $\{p_1, p_2\}$ to another 2-set. Otherwise, at least one subset contains 0, say, without loss of generality, $0 \in \{q_1, q_2\}$ and $q_1 = 0$. Let $u \in \{c, d\}^*$ be a word such that $q_2.u = 1$. Then $\{q_1, q_2\}.ua = \{1\}$ as $\{q_1, q_2\}.u = \{0, 1\}$. Furthermore, $\{p_1, p_2\}.ua$ is a 2-set. If $0 \notin \{p_1, p_2\}$, this is clear as ua permutes the states $\{1, \ldots, n\}$. If $0 \in \{p_1, p_2\}$, then $q_2 \notin \{p_1, p_2\}$, which implies $\{0, 1\} = \{p_1, p_2\}.u \neq \{q_1, q_2\}.u$ and hence $|\{q_1, q_2\}.u| = 2$.

Example 4. Here, we give an infinite family of synchronizing automata with simple idempotents over a binary alphabet that are neither sync-maximal nor completely reachable. Let $\mathcal{A} = (\{0, 1, \ldots, n\}, \{a, b\})$ with

$$i.a = \begin{cases} n & \text{if } i = n-1, \\ i & \text{if } i \neq n-1; \end{cases} \qquad i.b = \begin{cases} n & \text{if } i = n, \\ i+1 \bmod n & \text{if } i \in \{0, \ldots, n-1\}; \end{cases}$$

The word $a(ba)^{n-2}$ synchronizes $\mathcal{A}$. However, $\mathcal{A}$ is not completely reachable as it is not strongly connected, which implies, by Theorem 8, as it is over a binary alphabet, that it is not sync-maximal.

Lastly, we state that the problems Sync-Into-Subset and Reachable Subset are solvable in polynomial time for automata with simple idempotents over a binary alphabet. This is based on the following lemma about the structure of the reachable subsets in automata with simple idempotents.

Lemma 15. *Let $\Sigma = \{a, b\}$ and $\mathcal{A} = (Q, \Sigma)$ be an automaton with $Q = \{0, 1, \ldots, n-1\}$ and $q.b = (q+1) \bmod n$ and $a \in \Sigma$ be a simple idempotent letter with a state $q \in Q$ such that $\delta(q, a) \neq q$. Let $d > 0$ be the greatest common divisor of n and the number[3] $0 < r \leqslant n$ with $q.a = q.b^r$. Then for $S \subseteq Q$ we have $Q.u = S$ for some $u \in \Sigma^*$ if and only if $S = A_0 \cup \ldots \cup A_{d-1}$ for non-empty subsets $A_i \subseteq Q$ such that $s \in A_i$ implies $s \equiv i \pmod{d}$.*

[3] The case $r = n$ is a borderline case as it essentially implies that a acts as the identity. However, the statement entails it with $S = Q$ being the only reachable subset.

Proposition 16. *For automata with simple idempotents over a binary alphabet, the problems* Sync-Into-Subset *and* Reachable Subset *are solvable in polynomial time.*

6 Conclusion

We have introduced the sync-maximal automata and determined their structure. They are closely connected to completely reachable automata in the sense that they are either completely reachable or contain a completely reachable subautomaton. Furthermore, in a sync-maximal automaton all subsets with at least two states are reachable.

A natural question is how sync-maximality relates to the length of shortest synchronizing words. Intuitively, it means the set of synchronizing words is a "complicated" set, and one might expect that this might yield lower bounds on shortest possible paths in the power automaton. However, we can clearly construct automata with very short synchronizing words that are sync-maximal by adding to an existing automaton that is sync-maximal a single letter that maps everything to a single state, as the property of sync-maximality is retained when adding letters. But such a construction feels rather artificial, and a natural question is then what happens if we do not have arbitrary many letters at hand or the letters have to fulfill a certain property (like being idempotent or only having a certain defect). What can we say about lower bounds for shortest synchronizing words for automata over a binary alphabet, or only having the least number of letters of a certain type yielding a sync-maximal automaton on the given state set? For an upper bound, note that for completely reachable automata over a binary alphabet, the Černý conjecture has been confirmed in [5]. As sync-maximal automata over a binary alphabet are completely reachable by Theorem 8, sync-maximal automata over a binary alphabet also fulfill Černý's conjecture.

Furthermore, we have shown that a completely reachable automaton with simple idempotents must be sync-maximal. Hence, for strongly connected automata over simple idempotents being completely reachable is equivalent to sync-maximality. It is known that as soon as the transformation monoid of an automaton contains a primitive permutation group, it is both completely reachable and sync-maximal [10]. But what properties on the letters do we need to retain this equivalence (or simply that complete reachability already implies sync-maximality) that are more general than being either a permutation or a simple idempotent? In our method of proof, we used the idempotency (i.e., $q.aa = q.a$ for each state $q \in Q$) and the fact that the letters have defect one. But what when letters of defect more than one are involved? What about letters that instead of being idempotent fulfill the property $Q.aa = Q.a$, i.e., the image $Q.a$ is permuted by a?

Acknowledgement. I thank the anonymous reviewers for careful reading, spotting typos and unclear formulations, and pointers to the literature. In particular, I thank one reviewer for spotting an error in the original formulation and proof of Theorem 8, which has been fixed, and another reviewer for giving a very easy argument related to checking complete reachability.

References

1. Araújo, J., Bentz, W., Cameron, P.J.: Groups synchronizing a transformation of non-uniform kernel. Theor. Comput. Sci. **498**, 1–9 (2013). https://doi.org/10.1016/j.tcs.2013.06.016
2. Berlinkov, M.V., Ferens, R., Szykula, M.: Preimage problems for deterministic finite automata. J. Comput. Syst. Sci. **115**, 214–234 (2021). https://doi.org/10.1016/j.jcss.2020.08.002
3. Bondar, E.A., Volkov, M.V.: Completely reachable automata. In: Câmpeanu, C., Manea, F., Shallit, J. (eds.) DCFS 2016. LNCS, vol. 9777, pp. 1–17. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41114-9_1
4. Bondar, E.A., Volkov, M.V.: A characterization of completely reachable automata. In: Hoshi, M., Seki, S. (eds.) DLT 2018. LNCS, vol. 11088, pp. 145–155. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98654-8_12
5. Bondar, E.A., Casas, D., Volkov, M.V.: Completely reachable automata: an interplay between automata, graphs, and trees. CoRR abs/2201.05075 (2022). https://arxiv.org/abs/2201.05075
6. Černý, J.: Poznámka k. homogénnym experimentom s konecnými automatmi. Mat. fyz. čas SAV **14**, 208–215 (1964)
7. Don, H.: The Černý conjecture and 1-contracting automata. Electron. J. Comb. **23**(3), P3.12 (2016)
8. Gonze, F., Jungers, R.M.: Hardly reachable subsets and completely reachable automata with 1-deficient words. J. Autom. Lang. Comb. **24**(2–4), 321–342 (2019). https://doi.org/10.25596/jalc-2019-321
9. Gusev, V.V., Maslennikova, M.I., Pribavkina, E.V.: Finitely generated ideal languages and synchronizing automata. In: Karhumäki, J., Lepistö, A., Zamboni, L. (eds.) WORDS 2013. LNCS, vol. 8079, pp. 143–153. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40579-2_16
10. Hoffmann, S.: Completely reachable automata, primitive groups and the state complexity of the set of synchronizing words. In: Leporati, A., Martín-Vide, C., Shapira, D., Zandron, C. (eds.) LATA 2021. LNCS, vol. 12638, pp. 305–317. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68195-1_24
11. Hoffmann, S.: State complexity of the set of synchronizing words for circular automata and automata over binary alphabets. In: Leporati, A., Martín-Vide, C., Shapira, D., Zandron, C. (eds.) LATA 2021. LNCS, vol. 12638, pp. 318–330. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68195-1_25
12. Hoffmann, S.: Sync-maximal permutation groups equal primitive permutation groups. In: Han, Y., Ko, S. (eds.) DCFS 2021. LNCS, vol. 13037, pp. 38–50. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93489-7_4
13. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley Publishing Company, Boston (1979)
14. Howie, J.M.: Fundamentals of Semigroup Theory. Oxford University Press, Oxford (1996)

15. Maslennikova, M.I.: Reset complexity of ideal languages over a binary alphabet. Int. J. Found. Comput. Sci. **30**(6–7), 1177–1196 (2019). https://doi.org/10.1142/S0129054119400343
16. Reis, R., Rodaro, E.: Ideal regular languages and strongly connected synchronizing automata. Theor. Comput. Sci. **653**, 97–107 (2016). https://doi.org/10.1016/j.tcs.2016.09.026
17. Rystsov, I.K.: Polynomial complete problems in automata theory. Inf. Process. Lett. **16**(3), 147–151 (1983). https://doi.org/10.1016/0020-0190(83)90067-4
18. Rystsov, I.K.: Estimation of the length of reset words for automata with simple idempotents. Cybern. Syst. Anal. **36**(3), 339–344 (2000). https://doi.org/10.1007/BF02732984
19. Sandberg, S.: 1 homing and synchronizing sequences. In: Broy, M., Jonsson, B., Katoen, J.-P., Leucker, M., Pretschner, A. (eds.) Model-Based Testing of Reactive Systems. LNCS, vol. 3472, pp. 5–33. Springer, Heidelberg (2005). https://doi.org/10.1007/11498490_2
20. Savitch, W.J.: Relationships between nondeterministic and deterministic tape complexities. J. Comput. Syst. Sci. **4**(2), 177–192 (1970). https://doi.org/10.1016/S0022-0000(70)80006-X
21. Shitov, Y.: An improvement to a recent upper bound for synchronizing words of finite automata. J. Autom. Lang. Combin. **24**(2–4), 367–373 (2019). https://doi.org/10.25596/jalc-2019-367
22. Volkov, M.V., Kari, J.: Černý's conjecture and the road colouring problem. In: Éric Pin, J. (ed.) Handbook of Automata Theory, vol. I, pp. 525–565. European Mathematical Society Publishing House (2021)