

The Size of Power Automata

K. Sutner

Computer Science Department

Carnegie Mellon University

Pittsburgh, PA 15213

sutner@cs.cmu.edu, www.cs.cmu.edu/~sutner

Abstract

We describe a class of simple transitive semiautomata that exhibit full exponential blow-up during deterministic simulation. For arbitrary semiautomata we show that it is PSPACE-complete to decide whether the size of the accessible part of their power automata exceeds a given bound.

1 Motivation

Consider the following semiautomaton $\mathcal{A} = \langle [n], \Sigma, \delta \rangle$ where $[n] = \{1, \dots, n\}$, $\Sigma = \{a, b, c\}$ and the transition function is given by

- δ_a a cyclic shift on $[n]$,
- δ_b the transposition that interchanges 1 and 2,
- δ_c sends 1 and 2 to 2, identity elsewhere.

It is well-known that \mathcal{A} has a transition semigroup of maximal size n^n , see [13]. In other words, every function $f : [n] \rightarrow [n]$ is already of the form δ_w for some word w . Note that δ_a, δ_b can be replaced by any other pair of generators for the symmetric group on n points, and δ_c can be replaced by any function whose range has cardinality $n - 1$. It was shown by Salomaa that, for a three-letter alphabet Σ , those are the only choices that produce a maximal transition semigroup, see [15, 16]. If we think of the transition function as operating on sets of states, it follows that for all $A, B \subseteq [n]$ such that $|A| \geq |B| \geq 1$, there is a word w such that $\delta_w(A) = B$.

Now suppose we reverse all transitions in \mathcal{A} . In $\text{rev}(\mathcal{A})$, for any $B \subseteq [n]$, there is a word w such that $\delta_w(\{1\}) = B$. Indeed, if we augment the semiautomaton $\text{rev}(\mathcal{A})$

by selecting 1 as initial and final state, the corresponding power automaton has size 2^n , see [19, 3], and this power automaton turns out to be reduced. The same is true if we select all states to be initial and final. However, the semiautomaton $\text{rev}(\mathcal{A})$ is transitive (i.e., the underlying digraph is strongly connected), whereas its power automaton has 3 strongly connected components: $[n]$, \emptyset and the remaining subsets of $[n]$. The question arises whether there are transitive nondeterministic semiautomata on n states whose corresponding power automata have size 2^n , and where deletion of the sink \emptyset produces a transitive machine. Equivalently, can we find a semiautomaton $\langle Q, \Sigma, \delta \rangle$ such that for all $\emptyset \neq A, B \subseteq Q$ there is a word w such that $\delta_w(A) = B$? As is customary, we assume that the set of initial states as well as final states of a semiautomaton is the whole state set.

Let us fix some notation. For a nondeterministic automaton \mathcal{A} (with or without initial and final states), we write $\text{pow}(\mathcal{A})$ for the accessible part of the full power automaton of \mathcal{A} , $\pi(\mathcal{A})$ for the size of $\text{pow}(\mathcal{A})$, and $\mu(\mathcal{A})$ for the size of the minimal automaton of \mathcal{A} . Hence, we have the obvious bounds

$$1 \leq \mu(\mathcal{A}) \leq \pi(\mathcal{A}) \leq 2^n$$

Also, $\mu(\mathcal{A})$ can be computed in polynomial time from $\text{pow}(\mathcal{A})$. But can $\pi(\mathcal{A})$ be computed efficiently, without having to construct $\text{pow}(\mathcal{A})$ first? When are $\mu(\mathcal{A})$ and $\pi(\mathcal{A})$ equal to or close to the upper bound? In particular, what happens for semiautomata, and for transitive semiautomata? When is the sink-free version of $\text{pow}(\mathcal{A})$ again transitive?

These questions are originally motivated by the study of discrete dynamical systems, see [12, 1] for details and references on the topic. Briefly, let Σ be an alphabet, and denote by Σ^∞ the collection of all biinfinite words over Σ , usually referred to as *configurations* in this context. We can associate every configuration X with its *cover* $\text{cov}(X) \subseteq \Sigma^*$, the set of all finite factors of X . For a set \mathcal{X} of configurations define its cover $\text{cov}(\mathcal{X})$ to be the union of all the covers $\text{cov}(X)$ where $X \in \mathcal{X}$. A *shift space* is a subset \mathcal{X} of Σ^∞ that is topologically closed and invariant under the shift map $\sigma: \sigma(X)_i = X_{i+1}$. By compactness, a shift space can be reconstructed from its cover: a configuration X is in \mathcal{X} iff it is the limit of a sequence of words in the cover. Of particular interest are *sofic systems*, subshifts \mathcal{X} where $\text{cov}(\mathcal{X})$ is a regular language. A notable subclass of all sofic systems are the *subshifts of finite type*: shifts whose cover is a finite complement language: there is a finite set F of words over Σ^* such that $\text{cov}(\mathcal{X}) = \Sigma^* - \Sigma^* F \Sigma^*$, see Weiss [20].

The proper morphisms for shift spaces are given by continuous maps that commute with the shift. By the Curtis-Lyndon-Hedlund theorem [9], these maps are precisely the global maps of one-dimensional *cellular automata*. For our purposes, a (one-dimensional) cellular automaton is simply a local map $\rho: \Sigma^w \rightarrow \Sigma$. The local map naturally extends to a global map $\Sigma^\infty \rightarrow \Sigma^\infty$ that we also denote by ρ : $\rho(X)_i = \rho(X_{i-w+1} \dots X_i)$. Weiss [20] showed that every sofic system is the homomorphic image of a subshift of finite type.

By repeatedly applying the global map ρ to the full shift Σ^∞ , we obtain a descending sequence of sofic shifts. Let $L_t = \text{cov}(\rho^t(\Sigma^\infty))$ denote the corresponding regular cover languages. Clearly, all these languages are factorial, extensible and transitive (i.e., $uv \in L \implies u, v \in L$, $u \in L \implies \exists a, b \in \Sigma (aub \in L)$, and $u, v \in L \implies \exists x (uxv \in L)$). For languages of this type there is an alternative notion of minimal automaton, first introduced by Fischer [5] and discovered independently by Beauquier [2] in the form of the 0-minimal ideal in the syntactic semigroup of L . A *Fischer automaton* is a deterministic transitive semiautomaton. For each factorial, extensible and transitive language there is a unique Fischer automaton with the minimal number of states. Actually, the minimal Fischer automaton naturally embeds into the ordinary minimal DFA, see [17].

Each cellular automaton $\rho : \Sigma^{w+1} \rightarrow \Sigma$ is associated with a natural semiautomaton $B(\rho)$ whose underlying digraph is a de Bruijn graph $B(\Sigma, w) = \langle \Sigma^w, E \rangle$ where $E = \{ (ax, xb) \mid a, b \in \Sigma, x \in \Sigma^{w-1} \}$. If one labels edge (ax, xb) by $\rho(axb)$ one obtains a semiautomaton that accepts $\text{cov}(\rho(\Sigma^\infty))$. We will write $\mu(\rho)$ [$\mu_F(\rho)$, $\pi(\rho)$] for the size of the minimal DFA [minimal Fischer automaton, power automaton of $B(\rho)$, respectively] for $\text{cov}(\rho(\Sigma^\infty))$. In the mid eighties, Wolfram performed extensive calculations in an effort to understand the behavior of the sequence $(\mu(\rho^t))_t$, see [22]. In [21] he posed the question whether these sequences are generally increasing, and what can be said about the complexity of the limit set $L_\infty = \bigcap L_t$. It appears that for automata in Wolfram's classes *III* and *IV*, the sequence grows exponentially. For the lower classes the μ sequence appears to be bounded (and therefore eventually constant), or grow polynomially. The limit languages may be undecidable, though they are trivially co-recursively enumerable [10]. Surprisingly, there are examples of computationally universal cellular automata whose limit languages are regular [7], so one should not expect a simple answer to Wolfram's questions.

Recent work in computational mechanics also focuses on finite state machines as a tool to describe interesting behavior of cellular automata, see [8] for further references. One of the main objectives here is to find so-called *domains*, extensible, factorial, transitive, regular languages $L \subseteq \Sigma^*$ that are invariant or periodic under ρ (with cyclic boundary conditions). There are several examples where the typical dynamics of a cellular automaton can be described concisely on these domains, but not on the space of all configurations. In the reference, the authors use heuristic methods to construct candidate machines for domains. In general, one would expect a careful analysis of the Fischer automaton of the cellular automaton to provide a more systematic approach to the construction of domains.

Unfortunately, gathering computational data for the parameters $\mu(\rho)$ and $\mu_F(\rho)$ is rather difficult. For the sake of simplicity, let us only consider binary cellular automata, i.e., cellular automata over a two-symbol alphabet. The only obvious bounds are

$$\mu_F(\rho) \leq \mu(\rho) \leq \pi(\rho) \leq 2^{2^{w-1}}$$

where w is the width of the local map of ρ . Hence, even for elementary cellular automata ($w = 3$), the computations for the iterates easily get out of hand since

ρ^t has width $t(w - 1) + 1$.

We will first show some examples that demonstrate that full or nearly full exponential blow-up occurs quite often. In section 3, we prove that it is in general PSPACE-hard to determine even the size of the power automaton of a given semi-automaton. Our argument does not directly apply to the automata one encounters in the context of cellular automata, but it is not clear how the hardness obstruction could be avoided. Lastly, in section 4 we comment on some more related results in the study of cellular automata.

2 1-Permutation Automata and Blow-Up

2.1 Near Permutation Automata

The example given in the introduction shows that a transitive semiautomaton on a three-letter alphabet and n states may have a minimal automaton of size 2^n . In the following, we will limit our discussion to automata on two-letter alphabets. A *permutation automaton* is an automaton where each symbol $s = a, b$ induces a permutation δ_s of the state set. In other words, both the automaton and its reverse are deterministic. It is clear that the underlying graph of a permutation automaton has to be $(2, 2)$ -regular: every node has indegree 2 as well as outdegree 2. On the other hand, any $(2, 2)$ -regular graph admits labelings that produce a permutation automaton.

The following simple construction produces semiautomata with full blow-up. Start with a permutation automaton, and switch the label of one transition. These machines will be called *1-permutation automata* to indicate that the labeling has Hamming distance 1 to a permutation labeling. If the switched label is in particular a self-loop, we refer to the semiautomaton as a *loop-1-permutation automaton*. We have the following result, see [17] for a proof.

Theorem 2.1 *Let \mathcal{A} be a loop-1-permutation automaton of size n . Then the corresponding power automaton has 2^n states. Moreover, the power automaton is already reduced.*

The result applies in particular to cellular automata, since $B(\rho)$ is based on a de Bruijn graph. In fact, permutation automata are a standard way to construct cellular automata that have global maps that are open (in the sense of the usual product topology) and therefore surjective, but fail to be injective, see [9, 18]. The one-bit change moves the cover from being trivial, to having maximum possible complexity as a regular language.

It is shown in [17] that the minimal Fischer automaton of any factorial, extensible and transitive language L can be described as the uniquely determined strongly connected component of the minimal automaton of L that has transitions only to the sink. Minimal Fischer automata are *synchronizing*: there is a word w

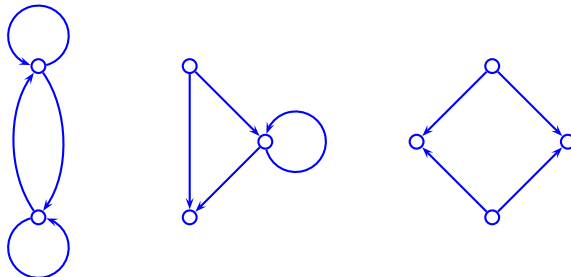
such that $\delta_w(Q) = \{p\}$ for any state p . Hence, in the case where L is given by a 1-permutation automaton \mathcal{A} for which $\text{pow}(\mathcal{A})$ is reduced, the construction of the minimal Fischer automaton can be construed as yet another power automaton problem: the minimal Fischer automaton is (isomorphic to) the kernel automaton $\text{pow}(S, \{p\})$ (the power automaton obtained by selecting $\{p\}$ as set of initial states), except for the sink. As we will see, this observation often provides a better way to compute the minimal Fischer automaton.

2.2 The Zig-zag Decomposition

To construct a permutation automaton start with an arbitrary transitive $(2, 2)$ -regular graph G . A *zig-zag* in G is an alternating cycle of the form

$$x_1 \rightarrow x_2 \leftarrow x_3 \rightarrow \dots \leftarrow x_{s-1} \rightarrow x_s \leftarrow x_1$$

where all edges are distinct, but the vertices x_1, \dots, x_n need not be distinct. A k -*zig-zag* is a zig-zag containing $2k$ edges, and therefore between k and $2k$ vertices, depending on the number of self-loops in the zig-zag. The three possible 2-zig-zags are shown below.



Since every edge belongs to exactly one zig-zag we can form a zig-zag decomposition of the graph. We denote by $\zeta(G)$ the number of zig-zags in G . For example, for binary de Bruijn graphs, all the zig-zags are isomorphic to one of the three graphs in the figure (the first is a complete de Bruijn graph). Hence, $\zeta(B(\mathbf{2}, w)) = 2^{w-1}$.

Our interest in zig-zags comes from the fact that in a permutation automaton, the label of a single edge on a zig-zag determines the labels of all the other edges. Hence, for any $(2, 2)$ -regular graph G there are $2^{\zeta(G)}$ permutation automata based on G . Note, though, that many of these automata may be isomorphic.

To produce a 1-permutation automaton \mathcal{A} we select one of the $2^{\zeta(G)}$ permutation automata on G , choose an edge in one zig-zag, and flip the label of that edge. This will usually produce a nondeterministic machine, but there is one exception: when the selected edge belongs to a 1-zig-zag $x \rightarrow y \leftarrow x$. In this case, we effectively remove the edge whose label was flipped, so that, say, δ_a is still a permutation, but δ_b is now a partial function. It follows that $|\delta_w(A)| \leq |A|$ for any

word w and $A \subseteq Q$. The power automaton of \mathcal{A} can not be transitive in this case, even after removal of the sink.

We need a method to determine $\pi(\mathcal{A})$, and in particular to show that $\pi(\mathcal{A}) = 2^n$ or, as the case may be, that $\pi(\mathcal{A}) < 2^n$. To avoid confusion, we refer to the states of the semiautomaton as *points*, and denote the set of all points by Q . In the following, we write the transition function as a right action $P' = P \cdot x$ of Σ^* on the semimodule $\text{pow}(Q)$. *State* then always refers to a state in the power automaton, i.e., a set $P \subseteq Q$ of points. P' is *reachable* from $P \subseteq Q$ if there is some word $x \in \Sigma^*$ such that $P' = P \cdot x$. State P is reachable if P is reachable from Q , in which case a word x such that $Q \cdot x = P$ is a *witness* for P . See [16] for an overview of results concerning the lengths of such witnesses in the special case where P has cardinality 1, and the automaton is deterministic.

To show the full blow-up occurs, the following method suggests itself. We can consider *representable operators*, maps $f : \mathcal{P}(Q) \rightarrow \mathcal{P}(Q)$, such that for each subset P of Q , there is a word x such that $Pf = P \cdot x$. Every word w gives rise to a representable operator $[w]$ where $P[w] = P \cdot w$. Representable operators are obviously closed under composition and thus form a monoid that acts naturally on $\mathcal{P}(Q)$ and the state set of $\text{pow}(\mathcal{A})$ is the orbit of Q under this monoid. In some cases one can show that the monoid of representable operators contains certain shift and delete operators that are sufficient to generate the whole power set of Q . If only a part of the full power set can be generated, one may succeed in giving simple membership conditions, which provide a count of the reachable states.

Before we establish full blow-up for a number of 1-permutation automata in the next section, we briefly comment on 2-permutation automata, automata whose labeling has Hamming distance 2 from the nearest permutation automaton.

Lemma 2.1 *Let \mathcal{A} be a 2-permutation automaton, based on some $(2, 2)$ -regular graph of size n . Then the power automaton of \mathcal{A} has size strictly less than 2^n .*

Proof.

It suffices to exhibit some non-reachable state $P \neq Q$.

Case 1. An a transition was switched to a b transition $p' \xrightarrow{b} p$, and a b transition was switched to an a transition $q' \xrightarrow{a} q$. Note that necessarily $p \neq q$. But then no state $P \neq Q$ such that $p, q \in P$ can be reachable since $p \notin Q \cdot a$ and $q \notin Q \cdot b$.

Case 2. The two switched transitions are $p' \xrightarrow{a} p$ and $q' \xrightarrow{a} q$ (without loss of generality).

Case 2.1. Both transitions lie in the same zig-zag.

If this zig-zag is a 2-zig-zag $b_1 \xrightarrow{a} e_1 \xleftarrow{a} b_2 \xrightarrow{a} e_2 \xleftarrow{a} b_1$ then no reachable state P can have $e_1 \in P \wedge e_2 \notin P$. If the zig-zag is k -zig-zag for $k > 2$, we can always obtain a non-reachable state by specifying three points. For example, consider

$$b_1 \xrightarrow{a} e_1 \xleftarrow{a} b_2 \xrightarrow{a} e_2 \xleftarrow{a} b_3 \xrightarrow{a} e_3 \xleftarrow{b} b_1.$$

Then no reachable state P can have $e_1, e_3 \in P \wedge e_2 \notin P$. The other cases are entirely similar.

Case 2.2. The two transitions lie in two distinct zig-zags.

First assume that the two zig-zags are vertex-disjoint. If both transitions belong to a 1-zig-zag, then no state P of cardinality $n - 1$ can be reachable: $Q \cdot a = Q$ and $p, q \notin Q \cdot b$, and our claim follows from the cardinality observation above.

Otherwise, note that for $p \in P$ we must have $P = P' \cdot a$ for some state P' . If further $q \notin P$, then $q' \notin P'$. Now consider the second transition $q' \xrightarrow{a} r$ with source q' (which exists since we may assume $q' \xrightarrow{a} q$ not to be part of a 1-zig-zag). But then we cannot have $r \in P$, since this would imply $q' \in P'$.

If the zig-zags have joint vertices the arguments still hold, since the membership restrictions are all expressed in terms of exits, i.e., vertices whose in-edges all belong to the zig-zag. But an overlap between two zig-zags always consists of an exit of one zig-zag being a begin (a vertex whose out-edges belong to the zig-zag) of the other. Similar considerations take care of self-loops. \square

The proof shows that indeed $\pi(\mathcal{A}) < c \cdot 2^n$ for some constant $c < 1$ in all cases except the one commented on earlier: the switched labels belong to 1-zig-zags.

2.3 Circulants

In contrast to the last lemma, we will now exhibit a class of 1-permutation automata that do exhibit full blow-up. Obvious candidates are machines based on *circulant graphs* since the latter are vertex-transitive. Recall that the circulant graph $C(n; d_1, d_2)$ has vertex set $\{0, 1, \dots, n - 1\}$ and edges $(i, i + d_s \bmod n)$, $s = 1, 2$. We will insist that $\gcd(n, d_1, d_2) = 1$ so that the graphs are always strongly connected and we can obtain a transitive semiautomaton, say, over the alphabet $\{a, b\}$, by attaching a label to each edge. It is easy to see that the number of zig-zags is

$$\zeta(C(n; d_1, d_2)) = \gcd(n, d_1 - d_2)$$

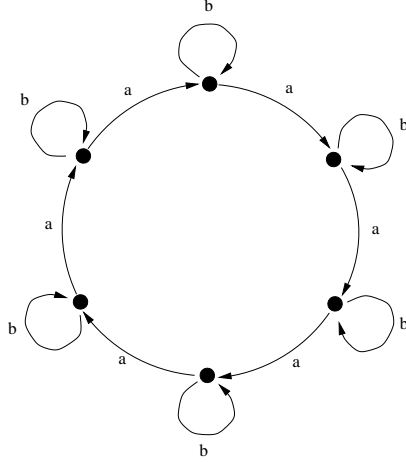
and they are all isomorphic as subgraphs of the circulant. Thus, there are $2^{\gcd(n, d_1 - d_2)}$ permutation automata based on a circulant graph, though many of them will be isomorphic.

In the following we will study a number of 1-permutation automata based on circulants $C(n; d_1, d_2)$ and compute their π , μ and μ_F values. We may safely assume that $0 \leq d_1 \leq d_2 < n$. Suppose e and n are coprime. Then $C(n; d_1, d_2)$ is isomorphic to $C(n; ed_1 \bmod n, ed_2 \bmod n)$, so we can cover a great many cases focusing on $d_1 = 0$ and $d_1 = 1$. E.g., for n prime all circulants are isomorphic to one of those two types.

2.3.1 Circulants $C(n; 0, 1)$

For circulants of the form $C(n; 0, 1)$ the whole graph consists of one n -zig-zag. Thus, there is essentially only one permutation automaton based on this type of graph: we may assume that all the cycle edges are labeled a , and all the self-loops

are labeled b . Since we can switch a label either at a self-loop or at a cycle edge there are two cases to consider.



First suppose that S is obtained from the permutation automaton on $C(n; 0, 1)$ by switching the label at a loop, say, the loop at node 0. We adopt the convention that the nodes are numbered $0, 1, \dots, n - 1$, and will assume that mods are taken whenever necessary. Thus, -1 is an alternative notation for node $n - 1$, and so forth. The following operators σ and κ_0 are representable:

$$\begin{aligned} P\sigma &= \{i + 1 \bmod n \mid i \in P\}, \\ P\kappa_0 &= P - \{0\}. \end{aligned}$$

Here, σ is a cyclic shift and κ_0 corresponds to deletion of point 0. To see that both are representable, note that $P\sigma = P \cdot a$ if $0 \notin P$ or $-1 \in P$. Otherwise, $P\sigma = P \cdot ab$. And $P\kappa_0 = P \cdot b$ for all P . By a sequence of shift and delete operations we can generate an arbitrary set $P \subseteq Q$ from Q . More precisely, consider an arbitrary state $P \subseteq Q$ and define the representable operator

$$f = \sigma \circ \tau_{-1} \circ \sigma \circ \tau_{-2} \circ \dots \circ \sigma \circ \tau_1 \circ \sigma \circ \tau_0.$$

Here τ_i is κ_0 if $i \notin P$, and the identity operator otherwise. It is easy to check that $Qf = P$.

Thus, $\pi(S) = 2^n$. Moreover, $\mu(S)$ must also be equal to 2^n . For suppose $p \notin P \subseteq Q$. Then there is a representable operator f similar to the one just used such that $pf \neq \emptyset$ but $Pf = \emptyset$: rotate and successively delete all the states in P . But then any two different states $P, P' \subseteq Q$ in $\text{pow}(S)$ must have distinct behavior and it follows that $\text{pow}(S)$ is the minimal automaton.

It remains to determine the minimal Fischer automaton. We claim that $\text{pow}(S)$ has exactly two strongly connected components: the sink \emptyset and the Fischer automaton. To see that other than \emptyset there is only one component, note that the

“sticky shift” operator σ_0 is representable:

$$P\sigma_0 = \begin{cases} P\sigma & \text{if } 0 \notin P, \\ P\sigma \cup \{0\} & \text{otherwise.} \end{cases}$$

It follows immediately that $P\sigma_0^{2n} = Q$ for any $P \neq \emptyset$.

Summarizing, we have established the following lemma.

Lemma 2.2 *Let S be a 1-permutation automaton based on the circulant graph $C(n; 0, 1)$, $n \geq 2$, where the switched edge is a self-loop. Then $\pi(S) = \mu(S) = 2^n$ and $\mu_F(S) = 2^n - 1$.*

What happens if we switch the label of an edge belonging to the big cycle of length n in $C(n; 0, 1)$? Without loss of generality, we may assume that the label of edge $(-1, 0)$ is switched to b . Then symbol a induces the operator σ_{κ_0} but symbol b induces a new operator

$$P_{-1}\alpha_0 = \begin{cases} P & \text{if } -1 \notin P, \\ P \cup \{0\} & \text{otherwise.} \end{cases}$$

Thus, $_{-1}\alpha_0$ adds point 0 to P provided that P contains point -1 . As a consequence, the modified shift operator

$$P\sigma' = \begin{cases} P\sigma & \text{if } -2 \in P, \\ P\sigma_{\kappa_0} & \text{otherwise,} \end{cases}$$

is representable. This suffices to apply the argument from the previous case and show that all states P can be reached from Q as long as $-1 \in P$ or $0 \notin P$. On the other hand, it is easy to see that the property “ $-1 \notin P$ & $0 \in P$ ” is preserved under all representable operators. Hence $\pi(S) = 2^n - 2^{n-2} = 3 \cdot 2^{n-2}$. As in the previous case, we can show that all states in the power automaton have distinct behavior, so that $\mu(S) = 3 \cdot 2^{n-2}$.

However, the minimal Fischer automaton is now smaller: its state set has cardinality n and has the form $\{-1\}, \{-1, 0\}, \{1\}, \{2\}, \dots$. This is one of the cases where the minimal Fischer automaton can be generated cheaply.

Lemma 2.3 *Let S be a 1-permutation automaton based on the circulant graph $C(n; 0, 1)$, $n \geq 2$, where the switched edge lies on the main cycle. Then $\pi(S) = \mu(S) = 3 \cdot 2^{n-2}$ and $\mu_F(S) = n$.*

2.3.2 Circulants $C(n; 1, 1)$

The circulant $C(n; 1, 1)$ can be construed as an ordinary cycle of length n where each edge has been replaced by a pair of parallel edges. All the zig-zags here are 1-zig-zags consisting of these pairs of parallel edges, and there are 2^n permutation labelings. However, the corresponding semiautomata are all isomorphic. Hence,

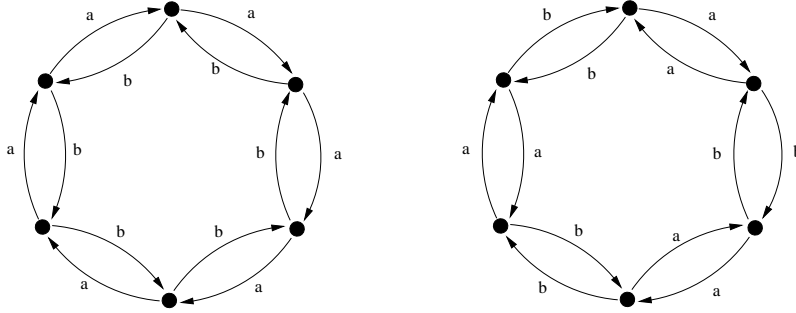
there is essentially only one 1-permutation automaton S arising from these labelings. For the sake of definiteness, suppose the label of one of the edges $(0, 1)$ has been switched to b . Then symbol b induces the cyclic shift operator σ and symbol a induces the delete operator κ_0 . As in the last section, we have the following characterization of the sizes of the power automaton, the minimal automaton and the minimal Fischer automaton.

Lemma 2.4 *Let S be a 1-permutation automaton based on the circulant graph $C(n; 1, 1)$, $n \geq 2$. Then $\pi(S) = \mu(S) = 2^n$ and $\mu_F(S) = n$.*

The state set in the minimal Fischer automaton here consists of all singletons.

2.3.3 Circulants $C(n; 1, d)$

We assume $1 < d < n$, so that the number of zig-zags can be any divisor of n . Consider the labeling obtained by labeling all the edges on the main cycle by a , and all the others by b . We will refer to this labeling as the *bi-cycle*. An example for $n = 6$, $d = n - 1$ is shown on the left in the figure below. There are two 6-zig-zags in the decomposition. The figure on the right shows the essentially only other permutation automaton on $C(6; 1, -1)$. We will refer to this second labeling as a *necklace*.



Lemma 2.5 *Let S be a 1-permutation automaton based on a bi-cycle where one of the secondary edges has been switched to label a . Then $\pi(S) = \mu(S) = 2^n$, and $\mu_F(S) = m(2^{n/m} - 1)$ where $m = \gcd(n, d - 1)$.*

Proof.

One can easily check that in this case

$$\begin{aligned} [a] &= {}_0\alpha_{d-1}\sigma &= \sigma {}_1\alpha_d, \\ [b] &= \kappa_0\sigma^d &= \sigma^d \kappa_d. \end{aligned}$$

The proof for the size of the power automaton is by induction on the cardinality r of P , the case $r = n$ being trivial. Suppose $P = (X, 1^k 0 Y)$, $k \geq 0$. Here we have written state P as a bit-vector; the first position after the comma indicates point

d. Then $P = P' \cdot ba^k$ for a suitable state P' of cardinality $r + 1$. On the other hand, consider $P = (1^k 0 X, \mathbf{1})$. Then $P = P' \cdot ba^{n-d+k}$ again for a suitable set P' of cardinality $r + 1$.

To see that the power automaton is already minimal, consider a state $P \subseteq Q$ and a point $q \notin P$. Let p be the maximal element in P . Then $|P \cdot ba^{n-p}| < |P|$, and $|q \cdot ba^{n-p}| \neq \emptyset$. By induction, it follows that there is a word w such that $P \cdot w = \emptyset \neq q \cdot w$.

To determine the size of the minimal Fischer automaton recall from above that it is isomorphic to the strongly connected component in $\text{pow}(S)$ that has edges leading only to the sink \emptyset . Thus, the component is generated by the singleton $\{0\}$. Consider the family of states whose points are have distance a multiple of $m = \text{gcd}(n, d - 1)$. More precisely, let

$$\begin{aligned} P_{r,d} &= \{r + i(d - 1) \bmod n \mid i \in \mathbb{Z}\} \\ Q_{r,d} &= \mathfrak{P}(P_{r,d}) \end{aligned}$$

where $0 \leq r < m$. It is clear that the cardinality of $Q_{r,d}$ is $2^{n/m}$, and that the union of all these families has size $m \cdot 2^{n/m} - m + 1$. Hence, it suffices to show that $\{0\}$ generates all sets in $Q_{r,d}$, for all shift factors r .

We will first verify this claim for $r = 0$. It is clear that $\{0\} \cdot a^{n^2} = P_{0,d}$. Furthermore, words $u = a^{d-1}$ and $v = ba^{d-2}$ produce operators $[u], [v] : Q_{0,d} \rightarrow Q_{0,d}$. We claim that indeed $P_{0,d} \cdot \{u, v\}^* = Q_{0,d}$. To avoid indexing problems, it is convenient to collapse $Q_{r,d}$ to $\mathfrak{P}(\{0, 1, \dots, n/m - 1\})$ by removing all irrelevant bits, and to think of $\{u, v\}$ as a new alphabet. On the collapsed sets, the effect of our two operators is as follows:

$$\begin{aligned} [u] &= {}_0\alpha_1\sigma &= \sigma_1\alpha_2, \\ [v] &= \kappa_0\sigma^2 &= \sigma^2\kappa_2. \end{aligned}$$

Thus, it suffices to show that the shift operator σ is representable. This is straightforward for states other than $P = (1, 0, X)$. Since $P \cdot u = (x_{-1}, 1, \underline{1}, x_2, x_3, \dots)$ introduces a spurious element, indicated in the bit-vector by an underline, we need to fully rotate the whole configuration to eliminate this extra element. The proper rules are to apply simple rotations induced by u except when the current configuration has the form $(0, y_1, \dots, y_s, 1)$ or $(\underline{1}, y_1, \dots, y_s, 1)$. In either case, we use v to obtain $(y_s, 1, 0, y_1, \dots, y_{s-1})$. The second case in particular produces the desired rotation of P .

Hence, we can generate all of $Q_{0,d}$. Since a induces a right shift on all states of the form $(0, X)$ it suffices to show that $Q_{1,d}$ can also be generated (assuming, of course, that $1 < m$). This is again easy for configurations whose collapse is not of the form $(1, 0, X)$: we can apply $[a]$ to the corresponding configuration in $Q_{0,d}$. For configurations of the critical form, we can apply $[b]$ to a left shift of the configuration. \square

Note that the argument actually shows that it suffices to have the operator τ defined to be σ for P such that $0 \notin P \vee (d-1) \in P$, and the identity otherwise, in conjunction with $[b]$ to generate all subsets of Q . Also, whenever n and $d-1$ are coprime, the minimal Fischer automaton has size $2^n - 1$.

For other labelings such as the necklace above it turns out that the size of the minimal Fischer automaton is still determined by $\gcd(n, d-1)$, see the comment at the end of the introduction. However, the size of the power automaton may be smaller: we may have $\pi(S) = \mu(S) = 2^n - 1$ or even $\pi(S) = \mu(S) = 15 \cdot 2^{n-4}$ (the latter holds for necklaces). Unfortunately, a general proof for this claim along the lines of the cases just discussed seems to be too tedious for consideration.

3 The Hardness Argument

We now turn to the *Power Automaton Size problem*, or PAS for short. More precisely, given a nondeterministic automaton \mathcal{A} , we wish to determine the size $\pi(\mathcal{A})$ of the accessible part of the power automaton associated with \mathcal{A} . To see that counting the number of states in $\text{pow}(\mathcal{A})$ can be done in nondeterministic linear space, it is best to consider a slightly more general problem: the succinct version of the Weak Connected Component Size problem. In the ordinary version of the problem, we are given a digraph G and a source node s , and we have to determine the size of the weakly connected component of s . In other words, we have to count the number of nodes of G reachable from s by some path. By the celebrated Immerman-Szelepsényi theorem, the Weak Connected Component Size problem is in NLOG. The corresponding reachability problem, where we are given an additional target node t and we have to decide if t is reachable from s , is NLOG-complete, see [6].

In the succinct version of the problem, the graph in question has $N = 2^n$ nodes and the adjacencies between the nodes are not given by a standard data structure, such as adjacency lists, but by a boolean circuit A . The circuit has $2n$ inputs and $A(x_1, \dots, x_n, y_1, \dots, y_n) = 1$ iff there is an edge from node $x_1 \dots x_n$ to node $y_1 \dots y_n$. The counting problem is now solvable in NSPACE(n), and the corresponding reachability problem is PSPACE-complete.

But PAS is clearly a special case of Weak Connected Component Size in its succinct form: the graph has as vertex set the full power set of Q , the source vertex is $I \subseteq Q$, the set of initial states of the nondeterministic machine, and the adjacencies are given by the condition

$$(P, P') \in E \iff \exists a \in \Sigma (P \cdot a = P').$$

The latter condition is easily expressed as a boolean circuit of size $O(kn)$ which codes the transition relation of the nondeterministic automaton.

To establish hardness, it is convenient to consider a decision version of PAS. For the next theorem we have chosen PAS_{\geq} where the input is a nondeterministic semiautomaton \mathcal{A} together with a positive bound B , and one has to deter-

mine if $\pi(\mathcal{A}) \geq B$. Note that since $\text{NSPACE}(n)$ is closed under complementation by Immerman-Szelepcsényi, it does not matter whether the query is phrased as “ $\pi(\mathcal{A}) \geq m$ ”, “ $\pi(\mathcal{A}) < m$ ”, or “ $\pi(\mathcal{A}) = m$ ”; the problem always remains in $\text{NSPACE}(n)$.

Theorem 3.1 PAS_{\geq} , the problem of determining whether the size of the power automaton of a given nondeterministic semiautomaton exceeds a given bound, is PSPACE-complete.

Proof. To establish PSPACE-hardness of the problem, we use Kozen’s result that it is PSPACE-hard to determine whether a given collection of DFAs determines an empty intersection language, see [11] or [6]. The construction in the first reference produces machines with unique final states.

So suppose we have a list $\mathcal{A}_1, \dots, \mathcal{A}_r$ of DFAs over some alphabet Σ . We assume that a, b, c are three new symbols not occurring in Σ , and set $\Gamma = \Sigma \cup \{a, b, c\}$. Let n_i be the size of machine \mathcal{A}_i . We may safely assume that $r \geq 4$ and that $n_i \leq n_{i+1}$. Let p_i be the least prime larger than 7, r , n_i and p_{i-1} , for $i = 1, \dots, r$.

Construct new machines \mathcal{A}'_i by attaching a cycle of length p_i to \mathcal{A}_i . The transitions on the cycle are all labeled by b . Furthermore, at each node of the cycle except for one, there is a self-loop labeled by the new symbol c . The one node without a self-loop will be called the base-point of the cycle. There are transitions labeled b from the unique final state of \mathcal{A}_i to all the points on the cycle C_i . Lastly, we attach a self-loop labeled a to the initial state of \mathcal{A} .

The semiautomaton \mathcal{A} also contains an auxiliary cycle C of length $m = 7$. Again, the cycle edges are labeled b . There are self-loops at all nodes of C labeled by all symbols in Σ as well as c . Furthermore, the loop at a selected base-point q_0 is in addition labeled a . There are no transitions from or to C from any other part of \mathcal{A} .

Hence, $\mathcal{A} = \bigoplus \mathcal{A}'_i \oplus C$ is the disjoint union of all the DFAs with their attached cycles plus the auxiliary cycle C . We have to count the number of states in $\text{pow}(\mathcal{A})$ that are reachable from $Q = \bigcup Q_i \cup \bigcup C_i \cup C$, the state set of \mathcal{A} . Again, we will refer to the states of \mathcal{A} as points, so that state from now on always refers to $\text{pow}(\mathcal{A})$, i.e., a set of points. We will write Q' for $\bigcup Q_i$ and C' for $\bigcup C_i$.

Let us say that an input string of the form $w = uaxbv$ is proper, where $u, x \in \Sigma^*$ and $v \in \Gamma^*$. Correspondingly, all states obtained from proper inputs are proper. First consider the case $u = v = \varepsilon$. After the first symbol a , all the Kozen automata are in their respective initial states and the base-point on C is also active. Let

$$A(x) = \{ i \in [r] \mid \mathcal{A}_i \text{ accepts } x \}.$$

The next input symbol b will then generate the state $\bigcup_{i \in A(x)} C_i$ plus one point on the auxiliary cycle C . As there is no self-loop at the base-points of the cycles C_i , the symbol c induces a delete operation on these cycles (delete all base-points). Symbol b , on the other hand, induces a cyclic shift. Since the lengths of these cycles as well as the auxiliary cycle C are relatively prime, it follows from the Chinese

remainder theorem that we can generate $m\alpha(A(x))$ states from $Q \cdot axb$. Here for any subset A of $[r]$: $\alpha(A) = \prod_{i \in A} 2^{p_i}$. Hence, from any proper input $uaxbv$ we can generate at least $m\alpha(A(x))$ states.

Improper inputs are somewhat more tedious to deal with. First, we have the reachable states Q and \emptyset . Then, there are at most $\prod_{i \in [r]} 2^{n_i}$ states due to inputs of the form Σ^* , and at most $\alpha([r])$ states due to inputs of the form $\Sigma^*(b+c)(\Gamma-a)^*$. Either of these states contain C as a subset. Then there are m states due to inputs of the form $\Sigma^*(b+c)(\Gamma-a)^*a\Gamma^*$, namely all the single points on C . Lastly, inputs of the form $\Sigma^*a\Gamma^*$ which fail to be proper can produce at most an additional $\prod_{i \in [r]} (p_i + 1)$ states, consisting of the base-point in C and at most one point in Q_i .

Since $n_i \leq p_i$ and $p_1 + 1 \leq 2^{p_1}$, the number of states reachable from Q is bounded from above by

$$K = 3\alpha([r]) + m(1 + \sum_x \alpha(A(x))),$$

where the summation is over suitably chosen factors $x \in \Sigma^*$ in proper inputs. On the other hand, the number of reachable states is bounded from below by

$$m \sum_x \alpha(A(x)).$$

Now consider the bound $B = m \cdot \alpha([r])$.

First suppose that the acceptance languages of the Kozen automata have non-empty intersection. Then there is some input $x \in \Sigma^*$ such that all DFAs are in their accepting state after scanning x . Hence, at least B states are reachable, as required.

On the other hand, suppose that the intersection language of the Kozen automata is empty. Then $A(x)$ has cardinality at most $r - 1$ and we have

$$\begin{aligned} K/B &\leq \frac{3\alpha([r]) + m(1 + \sum_x \alpha(A(x)))}{m \cdot \alpha([r])} \\ &= 3/m + \sum_x \alpha(A(x))/\alpha([r]) + 1/\alpha([r]) \\ &< 1/2 + \sum_i 2^{-p_i} + 2^{-\sum p_i} < 1. \end{aligned}$$

Thus, $K < B$, and we are done. \square

A minor modification of the last argument also shows that it is hard to determine the size of the *kernel automaton*. Recall that the (full) kernel automaton of a semiautomaton $\mathcal{A} = \langle Q, \Sigma, \tau \rangle$ is the subautomaton of the full power automaton of \mathcal{A} that is induced by the singleton states $\{p\}$ for $p \in Q$.

Corollary 3.1 *The problem of determining whether the size of the kernel automaton of a given nondeterministic semiautomaton exceeds a given bound, is PSPACE-complete.*

Proof. Modify the machine \mathcal{A} from the last theorem by attaching a new state q_0 and transitions labeled a from q_0 to the initial states of the Kozen automata as well as to the base point of the auxiliary cycle. The old self-loops labeled a are removed. The argument then proceeds almost verbatim as in the last proof. \square

It was pointed out by Ravikumar [14] that the uniqueness of the final states in the Kozen automata implies that the related State Reachability Problem for the power automaton of a nondeterministic machine is also PSPACE-complete. The input for State Reachability is a nondeterministic machine with initial states, say, $I \subseteq Q$, a target set $P \subseteq Q$, and one has to determine whether $P = I \cdot x$ for some input x . It is not hard to modify the last construction in order to show that State Reachability is hard even for semiautomata.

The semiautomaton \mathcal{A} in the proof of the the theorem is of course nowhere near transitive. We do not know if PAS remains hard for transitive semiautomata, but it appears likely that even for this restricted class of machines there is essentially no other way to determine $\pi(\mathcal{A})$ than to construct the accessible part of the power automaton. In fact, we do not know how to compute $\pi(\mathcal{A})$ efficiently even if \mathcal{A} is a 1-permutation automaton based on a circulant, or on a de Bruijn graph.

4 Conclusion

Let us return to cellular automata. In a binary de Bruijn automaton there are only 2-zig-zags of the form

$$0x \rightarrow x0 \leftarrow 1x \rightarrow x1 \leftarrow 0x,$$

though they can assume any of the three forms shown in the figure above. This fact can be exploited to strengthen theorem 2.1 a bit.

Theorem 4.1 *Let \mathcal{A} be a loop-1-permutation automaton of size n whose underlying graph is a de Bruijn graph. Then the minimal Fischer automaton of \mathcal{A} has size $2^n - 1$.*

This result is only the tip of an iceberg. For binary cellular automata of width $w \geq 2$, there are 2^{w-2} zig-zags and $2^w \cdot 2^{2^{w-2}}$ 1-permutation automata. The following table shows the frequencies of the differences $2^{16} - \pi(\mathcal{A})$ for all binary 1-permutation automata of width 5.

Δ	0	1	2	4	8	16	32	64
freq.	4096	512	896	480	240	208	328	352
Δ	124	128	170	256	512	1024	1052	2048
freq.	8	296	8	224	160	120	8	256

From the table, one can see that exactly half of all edge/permutation labeling combinations produce full blow-up. For example, in the 2-zig-zags containing a self-loop, the self-loop as well as the edge connecting the begin to the exit always produce full blow-up, regardless of the underlying permutation labeling (see the second entry in the figure of all 2-zig-zags above). Unfortunately, the techniques used in [17] only cover the self-loop, not the other edge. As one might expect from lemma 2.1, there is a more general connection between the size of the power automaton of $B(\rho)$ and the Hamming distance of the de Bruijn automaton to the nearest permutation labeling. The precise nature of this relationship is not known. It does follow from the lemma, though, that none of the iterates ρ^t can exhibit full blow-up; in fact, there has to be an exponential gap. Also, since the expected value of the Hamming distance of a random binary cellular automaton of width w is $5/16 \cdot 2^w$ one can expect larger cellular automata to produce relatively smaller Fischer automata on average.

Another automata construct that is of importance in symbolic dynamics is the kernel automaton. Surjectivity of a cellular automaton ρ is equivalent to $B(\rho)$ being unambiguous. As a consequence, for a surjective cellular automaton, the states of maximal size in the kernel automaton of $B(\rho)$ form a subautomaton, the so-called Welch automaton. The size of the states in this subautomaton is an important parameter, e.g., one can show that this so-called Welch index is a homomorphism from the monoid of all epimorphisms of Σ^∞ to the multiplicative monoid of the positive natural numbers, see [9]. As we have seen, determining the size of the kernel automaton of a semiautomaton is also PSPACE-hard.

It is a pleasure to acknowledge helpful discussions with S. Bloom and D. Landetta. Software that performs calculations with finite state machines is available at www.cs.cmu.edu/~sutner.

References

- [1] M.-P. Beal and D. Perrin. Symbolic dynamics and finite automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 2, chapter 10. Springer Verlag, 1997.
- [2] D. Beauquier. Minimal automaton for a factorial, transitive, rational language. *Theoretical Computer Science*, 67:65–73, 1989.
- [3] W. Brauer. On minimizing finite automata. *EATCS Bulletin*, 39:113–116, 1988.
- [4] M. Delorme and J. Mazoyer. *Cellular Automata: A Parallel Model*, volume 460 of *Mathematics and Its Applications*. Kluwer Academic Publishers, 1999.
- [5] R. Fischer. Sofic systems and graphs. *Monatshefte für Mathematik*, 80:179–186, 1975.

- [6] M. R. Garey and D. S. Johnson. *Computers and Intractability*. Freeman, 1979.
- [7] E. Goles, A. Maass, and S. Martinez. On the limit set of some universal cellular automata. *Theoretical Computer Science*, 110:53–78, 1993.
- [8] J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata. Technical Report 95-10-095, Santa Fe Institute, 1995.
- [9] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Math. Systems Theory*, 3:320–375, 1969.
- [10] L. Hurd. Formal language characterizations of cellular automata limit sets. *Complex Systems*, 1(1):69–80, 1987.
- [11] D. Kozen. Lower bounds for natural proof systems. In *Proc. 18-th Ann. Symp. on Foundations of Computer Science*, pages 254–266. IEEE Computer Society, 1977.
- [12] D. Lind and B. Marcus. *Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.
- [13] J. E. Pin. *Varieties of Formal Languages*. Foundations of Computer Science. Plenum Publishing Corporation, 1986.
- [14] B. Ravikumar. Private communication, 1994.
- [15] A. Salomaa. On the composition of functions of several variables ranging over a finite set. *Ann. Univ. Turkuensis*, 41, 1960.
- [16] A. Salomaa. Many-valued truth functions, Černý’s conjecture and road coloring. *Bulletin EATCS*, (68):134–150, June 1999.
- [17] K. Sutner. Linear cellular automata and Fischer automata. *Parallel Computing*, 23(11):1613–1634, 1997.
- [18] K. Sutner. *Linear Cellular Automata and De Bruijn Automata*, pages 303–320. Volume 460 of *Mathematics and Its Applications* [4], 1999.
- [19] R. A. Trakhtenbrot and Y. M. Barzdin. *Finite Automata: Behavior and Sythesis*. North-Holland, 1973.
- [20] B. Weiss. Subshifts of finite type and sofic systems. *Monatshefte für Mathematik*, 77:462–474, 1973.
- [21] S. Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, T9:170–183, 1985.
- [22] S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific, 1986.