Slowly synchronizing DFAs of 7 states and maximal slowly synchronizing DFAs

Michiel de Bondt

February 15, 2019

Abstract

We compute all synchronizing DFAs with 7 states and synchronization length $\geq 29$. Furthermore, we compute alphabet size ranges for maximal, minimal and semi-minimal synchronizing DFAs with up to 7 states.

1 Introduction

A Deterministic Finite Automaton (DFA) consists of a finite set of so-called states, and a finite alphabet of so-called transition symbols. The transition symbols are maps from the state set to itself. A DFA also has a begin state and a set of final states, but those are irrelevant for this paper.

Let $Q$ and $\Sigma$ be the state set and the alphabet of a DFA. Then the maps of the transition symbols combine to a transition function from $Q \times \Sigma$ to $\Sigma$. We denote this function by $\cdot \cdot \cdot$ is left-associative, and we will omit it mostly. We additionally define $\cdot \cdot : 2^Q \times \Sigma \rightarrow 2^Q$, namely by $S x = \bigcup_{s \in S} \{sx\}$.

With * being the Kleene star, $\Sigma^*$ is the sets of all words over $\Sigma$, i.e. all sequences of zero or more symbols of $\Sigma$. Each such word can be seen as either the empty word, or a symbol followed by another word. With respect to this structural definition, we define $qw$ inductively as follows for states $q \in Q$, subsets $S \subseteq Q$, and words $w \in \Sigma^*$:

$q \lambda = q 
q(xw) = (qx)w 
S \lambda = S 
S(xw) = (Sx)w$

Here, $\lambda$ is the empty word, $x$ is the first letter of the word $xw$ and $w$ is the rest of $xw$.

We say that a DFA with state set $Q$ and alphabet $\Sigma$ is synchronizing (in $l$ steps), if there exists a $w \in \Sigma^*$ (of length $l$), such that $Qw$ has size 1. If a DFA is synchronizing in $l$ steps, but not in fewer than $l$ steps, then we call $l$ the synchronization length. A conjecture by Černý in 1964 [3] is that for a DFA with $n$ states, the largest possible synchronization length is $(n - 1)^2$. Černý constructed a series of DFAs which reach this synchronization length, which is depicted on the right. The unique shortest synchronizing word of this DFA is $b(a^{n-1}b)^{n-2}$. 

1
In section 2, we will discuss our search for DFAs with 7 states and large synchronization lengths. This search extends results in [5] and [2]. To obtain a more efficient search algorithm, we improved the pruning of [2].

In section 3, we will define several types of synchronizing DFAs, and we discuss the search for DFAs with up to 7 states of these types. Some of these types were already discussed in [1], in which the search has already been done for DFAs with synchronization length \((n - 1)^2\).

### 2 Slowly synchronizing DFAs with 7 states

In [2], we computed all basic DFAs with 7 states with synchronization length at least 31. This yielded only 22 DFAs up to reordering states. Using better pruning, but pruning which only works for DFAs and not for PFAs in general, we extended this computation to synchronization length at least 29, yielding no less than 1850647 DFAs up to reordering states. The results are given below. Similar computations for smaller state sets can be found in [1].

| alph. size | sync. 36 | sync. 35 | sync. 34 | sync. 33 | sync. 32 | sync. 31 | sync. 30 | sync. 29 |
|------------|---------|---------|---------|---------|---------|---------|---------|---------|
| 1          | 1       |         |         |         |         |         |         |         |
| 2          |         | 3       | 3       | 13      | 39      |         |         |         |
| 3          |         | 3       | 8       | 44      | 373     |         |         |         |
| 4          |         |         | 4       | 90      | 1902    |         |         |         |
| 5          |         |         |         | 148     | 7416    |         |         |         |
| 6          |         |         |         | 194     | 23486   |         |         |         |
| 7          |         |         |         | 183     | 60544   |         |         |         |
| 8          |         |         |         | 113     | 126448  |         |         |         |
| 9          |         |         |         | 44      | 213970  |         |         |         |
| 10         |         |         |         | 10      | 294678  |         |         |         |
| 11         |         |         |         | 1       | 331780  |         |         |         |
| 12         |         |         |         |         | 306068  |         |         |         |
| 13         |         |         |         |         | 231142  |         |         |         |
| 14         |         |         |         |         | 142256  |         |         |         |
| 15         |         |         |         |         | 70713   |         |         |         |
| 16         |         |         |         |         | 27980   |         |         |         |
| 17         |         |         |         |         | 8620    |         |         |         |
| 18         |         |         |         |         | 2000    |         |         |         |
| 19         |         |         |         |         | 332     |         |         |         |
| 20         |         |         |         |         | 36      |         |         |         |
| 21         |         |         |         |         |         |         |         |         |

The computation took 8.5 CPU-years on a heterogeneous cluster, and the estimated single-thread time was about 5 years. The computation was performed by borrowing CPU-cycles from the science department of our university, especially the theoretical chemistry group.

There exists a basic DFA with 7 states, 39 (53) symbols, and synchronization length 28 (27). This shows that enumeration of basic DFA with 7 states and
synchronization length 28 (27) is not feasible. In the next section, we suggest computations which can be performed in practice instead.

As mentioned above, the algorithm differs from that in [2] in that the pruning has been improved. The pruning is done by finding an upper bound of the synchronization length of all synchronizing extensions $B$ of a DFA $A$. Here, $B$ is an extension of $A$ if $A$ and $B$ have the same state sets, and for every symbol $a$ of $A$, there exists a symbol $a'$ of $B$ which corresponds to $a$ as a (partial) mapping of states.

The pruning in [2] comes in three variants, with three upper bound $L$, $L'$, and $L''$. The first variant is the easiest.

1. Determine the size $|S|$ of a smallest reachable set $S$. Let $m$ be the minimal distance from $Q$ to a set of size $|S|$.

2. For each $k \leq |S|$, partition the collection of irreducible sets of size $k$ into strongly connected components. Let $m_k$ be the number of components plus the sum of their diameters.

3. For each reducible set $R$ of size $k \leq |S|$, find the length $l_R$ of its shortest reduction word. Let $l_k$ be the maximum of these lengths.

4. Now note that a synchronizing extension of $A$ will have a synchronizing word of length at most

$$L = \sum_{k=2}^{|S|} (m_k + l_k) + m.$$

The second variant improves the first variant as follows. Let $M$ be the maximum distance from $Q$ to a set of size $|S|$. Partition the irreducible sets of size $|S|$ which can be reached from $Q$ into strongly connected components, and let $c$ be the number of components plus the sum of their diameters. Then a synchronizing extension of $A$ will have a synchronizing word of length at most

$$L' = \sum_{k=2}^{|S|} (m_k + l_k) - c + 1 + M.$$

The third variant is the hardest variant. We take the upper bound $L''$ equal to $L''_0$, and we define inductively an upper bound $L''_R$ for the length of the the shortest synchronizing word for a reducible subset $R$, and an upper bound $L''_k$ for the maximum length of the shortest synchronizing word for any subset of size $k$. Define $S_R$, $m_R$, $M_R$ and $c_R$ as $S$, $m$, $M$ and $c$ respectively, but with $Q$ replaced by $R$.

$$L''_R = m_R$$

$$L''_R = \min\{L''_{|S_R|} - c_R + 1 + M_R, L''_{|R|-1} + l_R\}$$

$$L''_1 = 0$$

$$L''_k = m_k + \max\{L''_{k-1}, L''_R | R \text{ is reducible and } |R| = k\}$$

We improve the three upper bounds as follows.

- In $L$, we improve $m_k$ for each $k \leq |S|$;
• In $L'$, we improve $M$, $m_{|S|} - c$ and $m_k$ for each $k < |S|$;

• In $L''$, we improve $M_R$ and $m_{|S_R|} - c_R$ for each $R \subseteq Q$ and $m_k$ for each $k < |S|$.

Since $M = M_Q$, $S = S_Q$ and $c = c_Q$, it suffices to improve $m_k$ for each $k \leq |S|$, and $M_R$ and $m_{|S_R|} - c_R$ for each $R \subseteq Q$. We must preserve the following.

(a) Let $k \leq |S|$. For every synchronizing extension $B$ of $A$, the shortest path from any subset of size $k$ to a subset of size $\leq k$ which is either reducible in $A$ or of size $< k$, has length at most $m_k$.

(β) Let $R \subseteq Q$ be reducible in $A$. Notice that $|S_R|$ is the size of the smallest set which is reachable from $R$ in $A$. For every synchronizing extension $B$ of $A$, the shortest path from $R$ to a subset of size $\leq |S_R|$ which is reducible in $A$ or of size $< |S_R|$, has length at most $M_R + 1 + m_{|S_R|} - c_R$.

The first improvement is obtained by realizing that for subsets of size $k$ in (α) and of size $|S_R|$ in (β) which are not reducible in $A$, the only thing that matters is that they contain a pair, from which there exists a short path in $B$ to a subset of size $\leq 2$ which is either reducible in $A$ or of size $< 2$.

For $m_k$, the improvement is as follows. Let $\sigma$ be a strongly connected component of irreducible subsets of size $k$ of the power automaton of $A$. The purpose of $m_k$ is to estimate the number of subsets of $\sigma$ in a synchronization path of the power automaton of $B$, which is done by the diameter of $\sigma$, i.e.

$$\max \left\{ \max \left\{ d(S_1, S_2) \mid S_2 \in \sigma \right\} \mid S_1 \in \sigma \right\}$$

where $d(S_1, S_2)$ is the number of steps required to get from $S_1$ to $S_2$ in $A$. This can be improved to

$$\max \left\{ \max \left\{ \min \left\{ d(S_1, T) \mid T \supseteq P \right\} \mid P \text{ is a pair contained in some subset } S_2 \text{ of } \sigma \right\} \mid S_1 \in \sigma \right\}$$

The purpose of $c_R$ is to exclude some strongly connected components which are considered in $m_{|S_R|}$, which can be done in the same way as before.

For $M_R$, the improvement is as follows. Let $\tau$ be the collection of subset of size $|S_R|$ which are reachable from $R$ in $A$. Then we can improve

$$\max \left\{ d(R, T) \mid T \in \tau \right\}$$

to

$$\max \left\{ \min \left\{ d(R, T) \mid T \supseteq P \right\} \right\} \text{ \{ P is a pair contained in some subset of } \tau \right\}$$

For the second improvement, we use ideas of [4] and [5]. Let $k \geq 2$, and $S_1, S_2, \ldots, S_\ell$ be distinct $k$-subsets and $P_1, P_2, \ldots, P_\ell$ be distinct pairs of states. We say that

$$(S_1, P_1), (S_2, P_2), \ldots, (S_\ell, P_\ell)$$

is a Frankl-Pin sequence, if

(i) $P_i \subseteq S_i$ for all $i$;
Let $\sigma$ be a collection of $k$-subsets of states and let $\pi$ be a collection of pairs of states. Denote by $\text{fp}(\rho)$ (or $\text{fp}(\sigma, \pi)$) the length of the longest Frankl-Pin sequence $(S_1, P_1), (S_2, P_2), \ldots, (S_\ell, P_\ell)$, with $S_i \in \rho$ (and $P_i \in \pi$) for all $i$.

Theorem 2.1. Let $\rho_k$ be the collection of $k$-subsets of states which are reducible in $A$.

(i) Let $\tau$ be a collection of subsets of size $k$. Then there exists a $T \in \tau$, such that in $B$, it takes at most
$$1 - \text{fp}(\tau) + \left(\frac{n - k + 2}{2}\right) - \text{fp}(\rho_k, \rho_2)$$
steps to get from $T$ to a subset of size $\leq k$ which is either reducible in $A$ or of size $< k$.

(ii) Let $\tau$ be a collection of subsets of size $k$. Then there exists a $T \in \tau$, such that in $B$, it takes at most
$$\left(\frac{n - k + 2}{2}\right) - \text{fp}(\rho_k, \rho_2)$$
steps to get from $T$ to a subset of size $\leq k$ which is either reducible in $A$ or of size $< k$.

Proof. The proof of (i) is essentially that of [4, Theorem 2] and [5, Theorem 1], and the proof of (ii) is similar. □

Notice that Theorem 2.1 (i) is a special case of Theorem 2.1 (ii), namely the case where $|\tau| = 1$. On account of Theorem 2.1 (i), we can improve $m_k$ to
$$\min \left\{ m_k, \left(\frac{n - k + 2}{2}\right) - \text{fp}(\rho_k, \rho_2) \right\}$$
On account of Theorem 2.1 (ii), we can improve $m_k - c_R$ with $k = |S_r|$ to
$$\min \left\{ m_k - c_R, - \text{fp}(\tau) + \left(\frac{n - k + 2}{2}\right) - \text{fp}(\rho_k, \rho_2) \right\}$$
There is however one problem, namely computing $\text{fp}(\tau)$ and $\text{fp}(\rho_k, \rho_2)$. We do not compute $\text{fp}(\tau)$ and $\text{fp}(\rho_k, \rho_2)$, but take the lengths of Frankl-Pin sequences which are not necessarily maximal. This makes the improvements of $m_k$ and $m_{S_R} - c_R$ worse, but they remain valid.

We construct the Frankl-Pin sequences with length $\leq \text{fp}(\sigma, \pi)$ by a greedy approach. We take a pair $P$ of $\pi$ which is contained in the fewest subsets of $\sigma$. We make $\sigma'$ from $\sigma$ by removing all subsets which contain $P$. We compute a lower bound $f$ of $\text{fp}(\sigma', \pi \setminus \{P\})$ recursively. If $\sigma' \neq \sigma$, then the Frankl-Pin sequence with length $f$ can be extended at the front, and $1 + f$ is a lower bound of $\text{fp}(\sigma, \pi)$. If $\sigma' = \sigma$, then $f$ is a lower bound of $\text{fp}(\sigma, \pi)$.

A DFA is transitive or strongly connected if one can get from any state to any other state. A synchronizing DFA is minimal or irreducibly synchronizing if it becomes nonsynchronizing after removing any symbol. The authors of [4] and [5] count the synchronizing automata differently, namely they count only transitive minimal synchronizing DFAs up to reordering states. Below, we do this as well for 7 states.
Actually, all slowly synchronizing minimal DFAs with 7 states are counted above, because nontransitive synchronizing DFAs with 7 states have synchronization length at most 26.

**Theorem 2.2.** If the Cerny conjecture is true for less than \( n \geq 2 \) states, then the maximum length of the synchronizing word of a nontransitive synchronizing DFA with \( n \) states is

\[
\max\left\{ \frac{1}{2}n(n-1), (n-2)^2 + 1 \right\}
\]

which is \((n-2)^2 + 2\) if \( n = 3 \) or \( n = 4 \), and \((n-2)^2 + 1\) otherwise.

**Proof.** Let \( \mathcal{A} \) be a synchronizing DFA with \( n \) states, and suppose that \( \mathcal{A} \) has exactly \( m \) states which can be reached from every other state. Suppose that the Cerny conjecture holds for \( m \) states. Then these \( m \) states can be synchronized in at most \((m-1)^2\) steps. It takes

\[
1 + 2 + \cdots + n - m = \frac{1}{2}(n-m+1)(n-m)
\]

steps to reduce the set of all \( n \) states to those \( m \) states, so the synchronization length is at most

\[
f(m) = (m-1)^2 + \frac{1}{2}(n-m+1)(n-m)
\]

It is straightforward to show that \( f(m) \) can indeed be obtained as a synchronization length. Since \( f \) is a convex function, its maximum is obtained at \( m = 1 \) or \( m = n - 1 \).

In figure \( \text{I} \) we count transitive minimal synchronizing DFAs up to reordering states for less than 7 states.

A conjecture of Angéla Cardoso asserts that the maximum subset synchronization lengths of the Cerny automata are the best possible for synchronizing DFAs, see [5]. The maximum synchronization length of the Cerny automaton with \( n \) states are

\[
(n-1)^2\left(\left\lfloor \frac{n}{|S|} \right\rfloor - 1 \right) \left(2n - |S|\left\lfloor \frac{n}{|S|} \right\rfloor - 1 \right)
\]

for a subset \( S \). For nonsynchronizing DFAs, subset synchronization lengths can be exponential in the number of states.

We verified Cardoso’s conjecture for DFAs up to 7 states. In figure \( \text{II} \) the number of (transitive minimal) basic DFAs with \( n \) states in which it takes the
| alph. size | sync. | sync. | sync. | sync. | total |
|-----------|------|------|------|------|------|
| 1         | 4    | 3    | 2    | 1    | 0    |
| 2         | 2    | 3    | 3    | 2    | 8    |
| 3         | 2    | 3    | 3    | 2    | 10   |
| total     | 4    | 3    | 3    | 0    | 10   |

| alph. size | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | total |
|-----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1         | 1    | 4    | 11   | 23   | 43   | 46   | 139  | 224  | 380  | 622  | 986  | 1514 | 1547 | 893  | 99   |      | 0    |
| 2         | 1    | 8    | 31   | 89   | 448  | 841  | 1833 | 3892 | 7461 | 13471| 23144| 30931| 27044| 8344 |      |      | 6532 |
| 3         | 1    | 4    | 42   | 173  | 404  | 926  | 1944 | 3560 | 6619 | 10274| 12066| 3710 |      |      |      |      | 117538|
| 4         | 1    | 7    | 18   | 19   | 178  | 58   | 33   | 21   |      |      |      |      |      |      |      |      | 39723 |
| 5         | 2    | 2    | 11   | 26   | 42   | 1052 | 2925 | 1128 | 215  |      |      |      |      |      |      |      | 336   |
| total     | 2    | 13   | 46   | 156  | 671  | 1309 | 2917 | 6238 | 11459| 20745 | 34425 | 44511 | 32301 | 9237  | 99  |      | 164129|

| alph. size | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | sync. | total |
|-----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1         | 2    | 2    | 11   | 22   | 45   | 61   | 112  | 201  | 322  | 528  | 954  | 1761 | 2540 | 4077 |      |      | 6341 |
| 2         | 2    | 35   | 126  | 285  | 568  | 1355 | 4801 | 12092| 20636| 44871| 92738| 174948| 312377| 584993|    |      |      |
| 3         | 7    | 57   | 153  | 347  | 1319 | 5789 | 16414| 38463| 98340| 209987| 411502| 855834| 1658196|    |      |      |
| 4         | 1    | 4    | 10   | 41   | 285  | 1035 | 2895 | 11428| 41010| 96178| 179536| 827097| 1169501|    |      |      |
| 5         | 2    | 11   | 26   | 42   | 1552 | 2925 | 1128 | 215  |      |      |      |      |      |      |      |      |    | 3953  |
| total     | 2    | 0    | 4    | 54   | 209  | 493  | 1019 | 3082 | 11852| 31765 | 72107 | 188100| 401792| 768741| 2297812| 3452984|

Figure 1: The number of (slowly) synchronizing transitive minimal DFAs with 3 to 6 states, up to reordering states.
maximum number of steps to synchronize a subset of size $|S|$, up to reordering states, is given for $2 \leq |S| < n \leq 5$.

For 6 states, the only basic DFAs which require the maximum number of steps to synchronize subsets are the Cerny automaton with 6 states and the Kari automaton, the latter of which for $|S| \geq 4$ only. For 7 states, the only DFA which requires the maximum number of steps to synchronize subsets is the Cerny automaton. So it seems plausible that for $n \geq 7$ states, the Cerny automaton is the only automaton which reaches the Cardoso bound.

\section{Maximal and semi-minimal synchronizing DFAs}

In \cite{b}, we counted the number of basic synchronizing DFAs for up to 6 states and large synchronization lengths. We reduced the synchronization lengths until the number of basic synchronizing DFAs became too large.

To deal better with finding many synchronizing DFAs, we made two improvements to the search algorithm. In the search algorithm, the candidate symbols for extension are sorted in order of increasing number of synchronizing pairs. But this does not do anything if the DFA of the symbols that we have already chosen is synchronizing. The first improvement is to sort the symbols as well if the DFA of the symbols that we have already chosen is synchronizing. The candidate symbols for extension are sorted in order of increasing synchronization length.

The second improvement deals with the symmetry reduction of the synchronizing DFAs which are found by the algorithm. The algorithm itself performs symmetry reduction as an optimization, but this symmetry reduction is not perfect. But we need perfect symmetry reduction for for finding canonical representations to be stored and counting. This is done by applying all $n!$ symmetries on all symbols on candidate new synchronizing DFAs, where $n$ is the number of states. But applying symmetries on symbols takes some time. A lookup table for the symmetry applications would require $n! \cdot n^n$ entries for $n$.

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$|S|$ & $n = 3$ & $n = 4$ & $n = 4$ & $n = 5$ & $n = 5$ & $n = 5$
\hline
1 & 6 (5) & 3 (3) & 3 (3) & 1 (1) & 2 (2) & 1 (1)
2 & 23 (2) & 10 (4) & 11 (4) & 1 (1) & 2 (2) & 2 (2)
3 & 30 (0) & 9 (0) & 13 (0) & 1 (0) & 1 (0)
4 & 20 (0) & 5 (0) & 6 (0)
5 & 17 (0) & 1 (0) & 1 (0)
6 & 1 (0)
7 & 1 (0)
\hline
\text{total} & 87 (7) & 28 (7) & 34 (7) & 2 (2) & 5 (4) & 4 (3)
\hline
\end{tabular}
\end{center}

Figure 2: Number of DFAs with the largest subset synchronization lengths, up to reordering states.
states, which makes it too large. For that reason, we ordered the symmetries with the Johnson-Trotter algorithm for each \( n \), reducing the size of the lookup table to only \((n - 1) \cdot n^n\) entries for \( n \) states.

But these improvements do not solve the problem that there are too many synchronizing DFAs. To deal with that problem, we only searched for DFAs with additional properties for smaller synchronization bounds. A synchronizing basic DFA is

- **minimal**, if it becomes nonsynchronizing after removing any symbol.
- **semi-minimal**, if its synchronization length increases or it becomes nonsynchronizing after removing any symbol.
- **maximal**, if its synchronization length decreases after adding any new symbol.

Here, a symbol is new if it acts differently on the set of states.

For these types of DFAs, the number of DFAs appeared not to be very large even for smaller synchronization lengths. We counted the different types of synchronizing DFAs by testing found DFAs on having the type. With this, we kept track of symbols for the test for maximality, because testing all symbols takes very long. But we also optimized the search process. With the minimal DFAs, we did not search through synchronizing DFAs, because extensions of synchronizing DFAs are not minimal.

With the semi-minimal, maximal, and combined types, the collection of found DFAs is moved to another place in the code, namely to the new procedure described above, which sorts the symbols if the DFA is already synchronizing. The synchronizing DFA itself is collected as a candidate for a semi-minimal DFA. The candidate maximal DFA is made by saturating the synchronizing DFA with the sorted symbols, in such a way that the synchronization length is not affected. Next, the search process is continued, but extensions within the saturated DFA are skipped.

In the tables below, we do not give the number of DFAs, but we give ranges of possible alphabet sizes, for minimal, semi-minimal, maximal, maximal minimal, and maximal semi-minimal DFAs with a specific state set and synchronization length. The number of DFAs for each alphabet size in such a range can be found with the source code. Ranges are given for synchronizing DFAs which do not need to be transitive, but we verified that the corresponding ranges for transitive DFAs can be obtained by removing 1 (if present).

The ranges for general synchronizing basic DFAs were found as follows. Suppose that \( \mathcal{B} \) is a *maximum* DFA, i.e. a maximal DFA with the largest possible alphabet size. By removing symbols of \( \mathcal{B} \), we can obtain a semi-minimal DFA \( \mathcal{A} \) with the same synchronization length as \( \mathcal{B} \). Consequently, to conclude that the range for general synchronizing basic DFAs with \( \mathcal{A} \) and \( \mathcal{B} \) is continuous, it suffices to verify that the the range of semi-minimal synchronizing DFAs with \( \mathcal{A} \) is continuous.

But this does not work for the ranges of transitive general synchronizing basic DFAs, because \( \mathcal{A} \) may be not transitive. However, for the actual maximum DFAs \( \mathcal{B} \) which were printed by the search algorithm, it appeared that it was possible to make \( \mathcal{A} \) transitive by restoring one symbol of \( \mathcal{B} \). So the ranges for general basic DFAs can be deduced from the maximal and semi-minimal
Figure 3: Alphabet size ranges for synchronizing basic DFAs with 2, 3 and 4 states.

... ranges, and the corresponding transitive ranges can be obtained by removing 1 (if present), just as for the other ranges.

In figure 3, we give the results for up to 4 states. We were able to get through down to synchronization length 1.

For 5 states, we were able to get through only for minimal DFAs. For 6 states, we were not able to get through at all. The results are given in figure 4.

Notice that some additional ranges are given in the table for 5 states as well. The lines for synchronization lengths 1 and 2 were obtained by reasoning. This reasoning can be generalized to any number of states. The maximal minimal ranges were obtained by testing minimal DFAs for maximality, which was done by an algorithm to test the procedure of keeping track of the symbols for the test for maximality (not included in the source code).

Finally, we describe how we found the ranges for semi-minimal synchronizing
| sync. length | min | smin | max | max | max |
|--------------|-----|------|-----|-----|-----|
| 16           | 2-3 | 2-3  | 2-3 | 2-3 | 2-3 |
| 15           | 2-6 | 2-4  | 2-6 | 2-3 | 2-3 |
| 14           | 2-13| 2-4  | 2-8 | 13  | 2-3 |
| 13           | 2-15| 2-5  | 2-10, 12-13, 15 | 2-4 | 2-4 |
| 12           | 2-23| 2-5  | 2-17, 19-21, 23 | 2-3 | 2-3 |
| 11           | 2-29| 2-5  | 2-25, 27, 29 | 2-4 | 2-4 |
| 10           | 2-71| 2-5  | 2-27, 29, 31, 71 | 2-3 | 2-3 |
| 9            | 2-71| 2-5  | 2-41, 43-47, 49-51, 53, 55, 57, 59, 71 | 2-3 | 2-4 |
| 8            | 2-89| 2-5  | 3-57, 59-71, 73-75, 77, 83, 89 | ? | 3 |
| 7            | 2-215| 2-5  | 4-85, 87-89, 91-99, 101, 105, 167, 215 | ? | ? |
| 6            | ? | 2-5 | 2-6 | ? | ? |
| 5            | ? | 2-4 | 2-5 | ? | ? |
| 4            | ? | 1-4 | 1-4 | ? | ? |
| 3            | ? | 1-3 | 1-3 | ? | ? |
| 2            | 1-3119 | 1-2 | 1-2 | 3119 | ? | ? |
| 1            | 1-3124 | 1 | 1 | 3124 | ? | ? |
| all          | 1-3124 | 1-5 | 1-7 | 2-4 | ? | ? |

| sync. length | min | smin | max | max | max |
|--------------|-----|------|-----|-----|-----|
| 25           | 2   | 2    | 2   | 2   | 2   |
| 24           | ?   | ?    | ?   | ?   | ?   |
| 23           | 2-3 | 2-3  | 2-3 | 2-3 | 2-3 |
| 22           | 2-11 | 2-5 | 2-7, 10-11 | 2-3 | 2-4 |
| 21           | 2-15 | 2-5 | 2-15 | 2-4 | 2-4 |
| 20           | 2-21 | 2-5 | 2-17, 19, 21 | 2-4 | 2-4 |
| 19           | 2-47 | 2-6 | 2-17, 19, 25, 27, 47 | 2-3 | 2-5 |
| 18           | 2-53 | 2-6 | 2-25, 47, 53 | 2-4 | 2-5 |
| 17           | 2-59 | 2-6 | 2-29, 31-33, 35, 37, 39, 41, 43, 45, 59 | 2-4 | 2-5 |
| 16           | 2-95 | 2-6 | 2-41, 43, 45, 47, 49, 51, 53, 59, 61, 65, 77, 79, 83, 89, 95 | 2-4 | 2-6 |
| 15           | 2-101 | 2-6 | 2-71, 75, 77, 80-85, 101 | 2-4 | 2-5 |
| 14           | 2-143 | 2-6 | 2-93, 95-105, 107, 113, 115, 119, 123, 125, 127, 131, 137, 143 | 2-5 | 2-5 |
| 13           | ? | 2-6 | ? | ? | ? |
| 12           | ? | 2-6 | ? | ? | ? |
| 11           | ? | 2-6 | ? | ? | ? |
| 10           | ? | 2-6 | ? | ? | ? |

Figure 4: Alphabet size ranges for synchronizing basic DFAs with 5 and 6 states.
DFAs with 5 states. Notice first that these ranges contain the corresponding ranges for minimal DFAs, that non-minimal semi-minimal synchronizing DFAs have at least 2 symbols, and that the number of symbols of a semi-minimal synchronizing DFAs does not exceed its synchronization length. This yields the validity of the ranges for synchronization length \( \leq 4 \). Although the algorithm did not complete synchronization length 6, it did find semi-minimal synchronizing DFA with synchronization length 6 and up to 6 symbols. This yields the validity of the range for synchronization length 6. To complete the range for synchronization length 5, we need a construction with 5 symbols, which is given below, where self-transitions are omitted.

The shaded pair of states requires 5 steps to synchronize, and the other states synchronize as well. The construction can be generalized to \( n \geq 4 \) states, with \( n \) steps and \( n \) symbols (the construction is not semi-minimal for 3 states).

Synchronization length 13 is only included in the table for minimal synchronizing DFAs with 6 states. But we think the maximum DFA with 6 states and synchronization length 13 has 359 symbols. More generally, we think the maximum DFA with \( n \) states and synchronization length \( 3n - 5 \) has \( 3 \cdot (n - 1)! - 1 \) symbols.

This number of symbols is indeed obtainable. Take a state set \( Q \) of size \( n \), with distinct states \( q \) and \( q' \). We include (i) all \( (n - 1)! \) symbols which send \( Q \) to \( Q \) and \( q \) to \( q \), except the identity symbol, (ii) all \( (n - 1)! \) symbols which send \( Q \) to \( Q \) and \( q \) to \( q' \), and (iii) all \( (n - 1)! \) symbols which send \( Q \) to \( Q \setminus \{q\} \) and which send \( q \) and \( q' \) to the same state.

For 7 states, the idea was to start a search process to find all maximal and semi-minimal DFAs with synchronization length at least 27. A sample of 5 percent of this computation on a heterogeneous cluster indicated that this takes about 45 CPU years on that cluster (of which 2 years are already completed by the sample). But we did not get the time to do the whole computation. For that reason, I wrote a program to extract the maximal and semi-minimal DFAs with

| sync. length | min | smin | max | max | max |
|--------------|-----|------|-----|-----|-----|
| 36           | 2   | 2    | 2   | 2   | 2   |
| 35           |     |      |     |     |     |
| 34           |     |      |     |     |     |
| 33           |     |      |     |     |     |
| 32           |     |      | 2   | 2   | 2   |
| 31           | 2   | 2    | 2   | 3   |     |
| 30           | 2   | 2    | 2   | 2   | 2   |
| 29           | 2   | 2    | 2   | 2   | 2   |
synchronization length at least 29 from all basic DFAs with synchronization length at least 29. The selection of the maximal DFAs requires two passes. In the first pass, non-maximal DFAs are collected, by testing DFAs with one symbol removed to have the same synchronization length, for each DFA and each of its symbols.

Below are the alphabet size ranges for subset synchronization.

| $n$ | $|S|$ | sync. length | min | smin | max | max | max |
|-----|-----|-------------|-----|------|-----|-----|-----|
| 2   | 2   | 1           | 1-3 | 1    | 1   | 3   |
| 3   | 2   | 3           | 2-7 | 2-3  | 2-3 | 2-3 | 7   |
| 3   | 3   | 4           | 2-5 | 2-3  | 2-3 | 2-3 | 5   |
| 4   | 2   | 6           | 2-6 | 2-3  | 2-3 | 2-3 | 3, 6, 3, 3 |
| 4   | 3   | 8           | 2-6 | 2-3  | 2-3 | 2-3 | 2-3, 5-6, 2-3, 2-3 |
| 4   | 4   | 9           | 2-5 | 2-3  | 2-3 | 2-3 | 2-3, 5, 2-3, 2-3 |
| 5   | 2   | 10          | 2-3 | 2-3  | 2-3 | 2-3 | 2-3, 2-3, 2-3 |
| 5   | 3   | 13          | 2-4 | 2-3  | 2-3 | 2-3 | 2-3, 2-3, 2-3 |
| 5   | 4   | 15          | 2-4 | 2-3  | 2-3 | 2-3 | 2-3, 2-3, 2-3 |
| 5   | 5   | 16          | 2-3 | 2-3  | 2-3 | 2-3 | 2-3, 2-3, 2-3 |
| 6   | 2   | 15          | 2   | 2    | 2   | 2   | 2   |
| 6   | 3   | 20          | 2   | 2    | 2   | 2   | 2   |
| 6   | 4   | 22          | 2   | 2    | 2   | 2   | 2   |
| 6   | 5   | 24          | 2   | 2    | 2   | 2   | 2   |
| 6   | 6   | 25          | 2   | 2    | 2   | 2   | 2   |
| 7   | 2   | 21          | 2   | 2    | 2   | 2   | 2   |
| 7   | 3   | 28          | 2   | 2    | 2   | 2   | 2   |
| 7   | 4   | 31          | 2   | 2    | 2   | 2   | 2   |
| 7   | 5   | 33          | 2   | 2    | 2   | 2   | 2   |
| 7   | 6   | 35          | 2   | 2    | 2   | 2   | 2   |
| 7   | 7   | 36          | 2   | 2    | 2   | 2   | 2   |

We can observe the following in the results.

**Conjecture.** Let $n \geq 3$.

(i) The maximum number of symbols of a minimal DFA with $n$ states is $n$. This number of symbols is possible for minimal DFAs with $n$ states, if and only if the synchronization length is at least $n+1$ and at most $\frac{1}{2}n^2 + \frac{1}{2}n - 2$.

(ii) The maximum number of symbols of a semi-minimal DFA with $n$ states is $2n - 3$. If $n \geq 4$, then this number of symbols is possible for semi-minimal DFA with $n$ states, if and only if the synchronization length is at least $2n - 3$ and at most $\frac{1}{2}n^2 - \frac{1}{2}n - 1$.

Furthermore, transitive constructions are possible.

We show that transitive minimal DFAs with $n$ states and $n$ symbols as in (i) above indeed exist. Below on the left hand side, a construction is given for synchronization length $\frac{1}{2}n^2 + \frac{1}{2}n - 2$. Here, a single arrow represents a symbol which merges two states as indicated by the arrow, and preserves the other
states. Furthermore, a double arrow represents a symbol which interchanges two states and preserves the other states.

The white states can be moved to the left step by step, where each step yields a DFA of which the synchronization length is one less than that of its predecessor. This process ends with the DFA above on the right hand side, which has synchronization length $2n - 2$. By replacing the double arrows which attach the white states by single arrows towards the leftmost shaded state, one can decrease the synchronization length further and obtain all remaining synchronization lengths down to $n + 1$ inclusive.

But that construction is not transitive. For a transitive construction, we start with the semi-minimal DFA which we constructed before. This DFA is not minimal, because symbol $d$ is not needed for synchronization: removing symbol $d$ yields a DFA with a sink state which synchronizes in $2n - 3$ steps. Below on the left hand side, we attached a new state with a double arrow to the sink state, which we marked with an $\ast$. We will show that this new DFA is minimal with synchronization length $n + 1$.

One can attach more new states on state $\ast$ with double arrows, up to the DFA above on the right hand side. We will show that we obtain all synchronization lengths from $n + 2$ up to $2n - 3$ inclusive this way.

Just as before, the objective is to merge the shaded pair of states. But there is a second objective, namely to apply the interchange symbols. To make the first application of the interchange symbols effective, they have to be preceded by another interchange symbol or by symbol $d$, and we may assume the latter symbol to be the direct predecessor of the former. But a consecutive application of two interchange symbols will not occur in a shortest synchronizing word. So the second objective is that for each of the interchange symbols, there is an application which is immediately after symbol $d$.

Let $k$ be the number of states which is attached to state $\ast$ with an interchange symbol. To show that the length of the shortest synchronizing word is $n + k$, we need a third objective, which is that the last symbol is not an interchange symbol. This objective is justified because interchange symbols act as permutations on the state set, and therefore cannot be the last symbol of a shortest synchronizing word. Each of the time, an application of symbol $d$ does not contribute to the merge of the shaded pair of states, and neither do interchange symbols, except in the last step where the actual merge takes place by way of symbol $d$. This exception is compensated by the third objective. It
is also clear that a synchronizing word of length \( n + k \) exists, so we have all synchronization lengths from \( n + 1 \) up to \( 2n - 3 \) inclusive.

References

[1] M. de Bondt, H. Don and H. Zantema, Slowly synchronizing automata with fixed alphabet size, *Information and Computation*, to appear. Available at [https://arxiv.org/abs/1609.06853](https://arxiv.org/abs/1609.06853) (2017).

[2] M. de Bondt, H. Don and H. Zantema, Lower Bounds for Synchronizing Word Lengths in Partial Automata, *Int. J. Found. Comput. Sci.*, to appear. Available at [https://arxiv.org/abs/1801.10436](https://arxiv.org/abs/1801.10436) (2018).

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

[4] A. Kisielewicz and M. Szykuła, Generating Synchronizing Automata with Large Reset Lengths, [https://arxiv.org/abs/1404.3311](https://arxiv.org/abs/1404.3311) (2014).

[5] A. Kisielewicz, J. Kowalski and M. Szykuła, Experiments with synchronizing automata, *Implementation and Application of Automata*, eds. Y.-S. Han and K. Salomaa (Springer International Publishing, Cham, 2016), pp. 176–188.