Recognizing Synchronizing Automata with Finitely Many Minimal Synchronizing Words is PSPACE-Complete

Emanuele Rodaro
Join work with
Elena Pribavkina

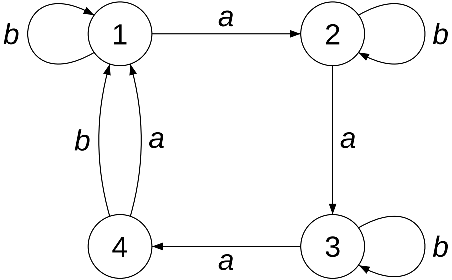
Dep. of Mathematics, University of Porto
CMUP

June 28, 2011

Synchronizing Automata

- A deterministic finite automaton (DFA) is a triple $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.
- The transition function $\delta : Q \times \Sigma \to Q$ naturally extends to the free monoid Σ^*, this extension is still denoted by δ; also for $S \subseteq Q$ and $w \in \Sigma^*$ we will write $\delta(S, w) = \{\delta(q, w) \mid q \in S\} = S \cdot w$.
- A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is called *synchronizing* if there is a word w whose action *resets* $\mathscr{A}$, that is, leaves the automaton in one particular state no matter which state in Q it started at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$. Equivalently, $|Q \cdot w| = |\delta(Q, w)| = 1$.
- Any such w is called *synchronizing* or *reset* word for $\mathscr{A}$.

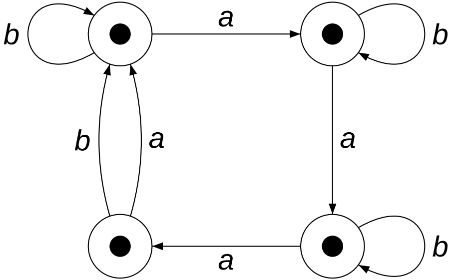
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

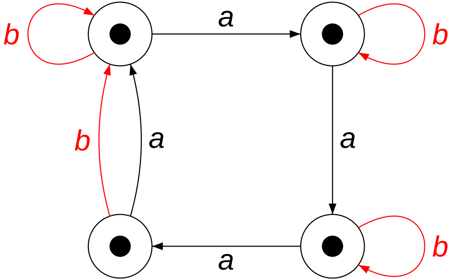
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

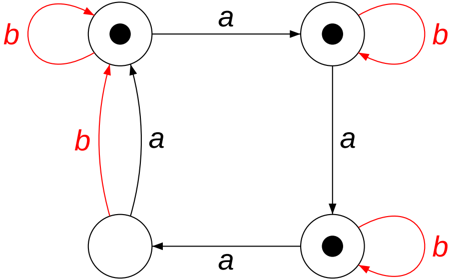
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

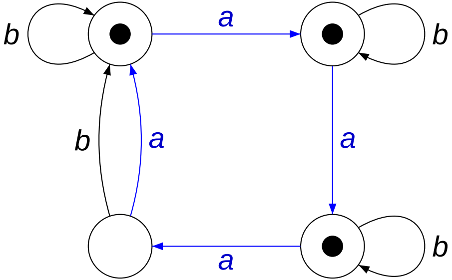
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

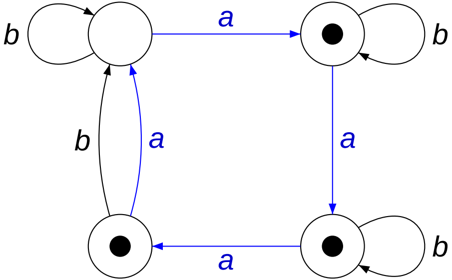
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

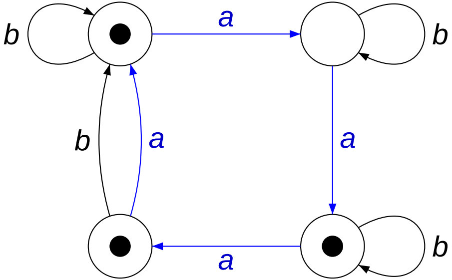
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

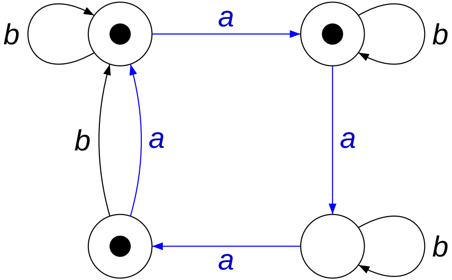
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

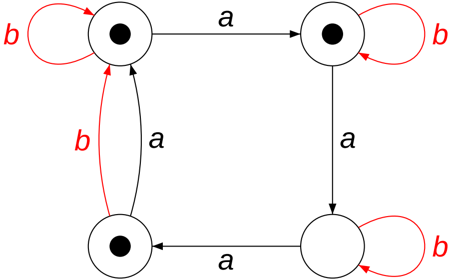
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

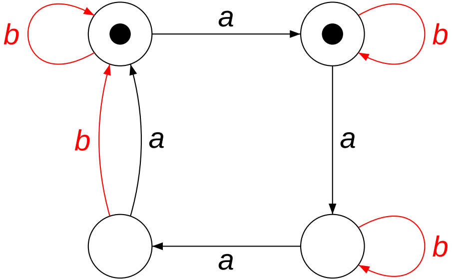
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

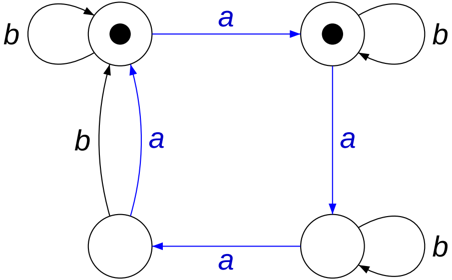
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

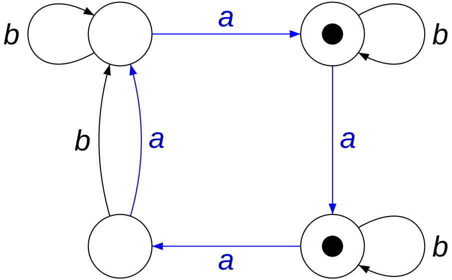
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

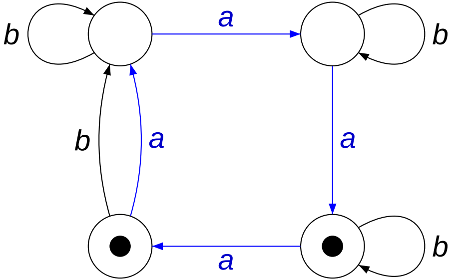
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

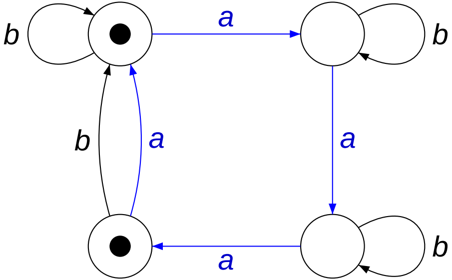
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

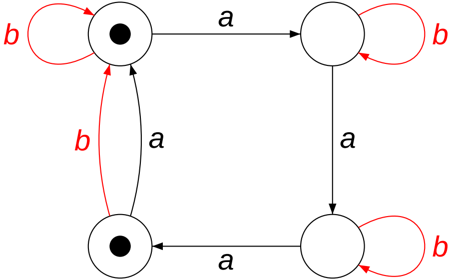
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

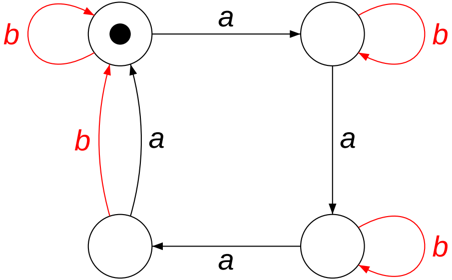
Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

Synchronizing Automata – An Example



A reset word is *baaabaaab*.

In fact it is the shortest reset word for this automaton.

Černý's Conjecture

Suppose a synchronizing automaton has n states. What is the length of the shortest synchronizing word?

In 1964 Jan Černý found an infinite series of n-state reset automata whose shortest reset word has length $(n-1)^2$.

Conjecture

Any synchronizing automaton with n states has a reset word of length at most $(n-1)^2$.

Černý's Conjecture

Suppose a synchronizing automaton has n states. What is the length of the shortest synchronizing word?

In 1964 Jan Černý found an infinite series of n-state reset automata whose shortest reset word has length $(n-1)^2$.

Conjecture

Any synchronizing automaton with n states has a reset word of length at most $(n-1)^2$.

Černý's Conjecture

Suppose a synchronizing automaton has n states. What is the length of the shortest synchronizing word?

In 1964 Jan Černý found an infinite series of n-state reset automata whose shortest reset word has length $(n-1)^2$.

Conjecture

Any synchronizing automaton with n states has a reset word of length at most $(n-1)^2$.

Some results

- Consider the pairs automaton $\langle Q \times Q, \Sigma, \delta \times \delta \rangle$. Then $\mathscr{A}$ is synchronizing iff for any $(p, q) \in Q \times Q$ there is a word $w' \in \Sigma^*$ with $(p, q) \cdot w' = (p', p')$. We have the trivial upper bound $\frac{n^2(n-1)}{2}$. Moreover the problem of checking if $\mathscr{A}$ is synchronizing is in **P**.
- Greedy Algorithm. Non trivial analysis gives $\frac{n^3-n}{6}$ [Pin]. An even better analysis gives $\frac{7n(n^2+12n-4)}{48}$ [Trahtman]. The problem of checking if there is a synchronizing word of length at most m is **NP**-complete [Eppstein].
- The conjecture is true for many particular classes: automata with zero, monotonic automata, aperiodic automata, automata whose underlying digraph is Eulerian, strongly transitive automata etc.

A New Class of Synchronizing Automata

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing DFA.
$Syn(\mathscr{A})$ denotes the language of all words synchronizing $\mathscr{A}$.

A synchronizing word v is said to be *minimal* if none of its proper prefixes nor suffixes is synchronizing. Equivalently none of its proper factors is synchronizing.

The language $Syn(\mathscr{A})$ of all synchronizing words is a two-sided ideal generated by the language $Syn^{min}(\mathscr{A})$ of all minimal synchronizing words:

$$Syn(\mathscr{A}) = \Sigma^* Syn^{min}(\mathscr{A}) \Sigma^*.$$

We consider the class **FG** of synchronizing automata whose language of minimal synchronizing words is finite. Such automata are referred to as *finitely generated synchronizing automata*.

A New Class of Synchronizing Automata

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing DFA.
$Syn(\mathscr{A})$ denotes the language of all words synchronizing $\mathscr{A}$.

A synchronizing word v is said to be *minimal* if none of its proper prefixes nor suffixes is synchronizing. Equivalently none of its proper factors is synchronizing.

The language $Syn(\mathscr{A})$ of all synchronizing words is a two-sided ideal generated by the language $Syn^{min}(\mathscr{A})$ of all minimal synchronizing words:

$$Syn(\mathscr{A}) = \Sigma^* Syn^{min}(\mathscr{A}) \Sigma^*.$$

We consider the class **FG** of synchronizing automata whose language of minimal synchronizing words is finite. Such automata are referred to as *finitely generated synchronizing automata*.

A New Class of Synchronizing Automata

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing DFA.
$Syn(\mathscr{A})$ denotes the language of all words synchronizing $\mathscr{A}$.

A synchronizing word v is said to be *minimal* if none of its proper prefixes nor suffixes is synchronizing. Equivalently none of its proper factors is synchronizing.

The language $Syn(\mathscr{A})$ of all synchronizing words is a two-sided ideal generated by the language $Syn^{min}(\mathscr{A})$ of all minimal synchronizing words:

$$Syn(\mathscr{A}) = \Sigma^* Syn^{min}(\mathscr{A}) \Sigma^*.$$

We consider the class **FG** of synchronizing automata whose language of minimal synchronizing words is finite. Such automata are referred to as *finitely generated synchronizing automata*.

A New Class of Synchronizing Automata

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing DFA.
$Syn(\mathscr{A})$ denotes the language of all words synchronizing $\mathscr{A}$.

A synchronizing word v is said to be *minimal* if none of its proper prefixes nor suffixes is synchronizing. Equivalently none of its proper factors is synchronizing.

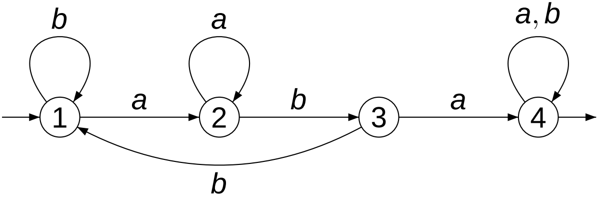
The language $Syn(\mathscr{A})$ of all synchronizing words is a two-sided ideal generated by the language $Syn^{min}(\mathscr{A})$ of all minimal synchronizing words:

$$Syn(\mathscr{A}) = \Sigma^* Syn^{min}(\mathscr{A}) \Sigma^*.$$

We consider the class **FG** of synchronizing automata whose language of minimal synchronizing words is finite. Such automata are referred to as *finitely generated synchronizing automata*.

Finitely Generated Synchronizing Automata

The minimal automaton $\mathscr{A}_{aba}$ recognizing the language $\Sigma^* aba \Sigma^*$:



$Syn(\mathscr{A}_{aba}) = \Sigma^* aba \Sigma^* \Rightarrow Syn^{min}(\mathscr{A}_{aba}) = \{aba\} \Rightarrow \mathscr{A}_{aba} \in$ **FG**.

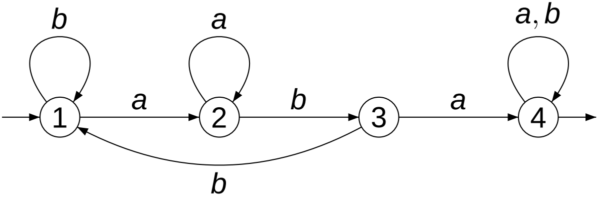
For any word $w \in \Sigma^*$ the automaton $\mathscr{A}_w \in$ **FG**.

$\mathscr{A}_w$ has $n = |w| + 1$ states $\Rightarrow$ its shortest reset word has length $n - 1$.

In general, the minimal automaton recognizing the language $\Sigma^* M \Sigma^*$ for a finite language M is in **FG**.

Finitely Generated Synchronizing Automata

The minimal automaton $\mathscr{A}_{aba}$ recognizing the language $\Sigma^* aba \Sigma^*$:



$Syn(\mathscr{A}_{aba}) = \Sigma^* aba \Sigma^* \Rightarrow Syn^{min}(\mathscr{A}_{aba}) = \{aba\} \Rightarrow \mathscr{A}_{aba} \in$ **FG.**

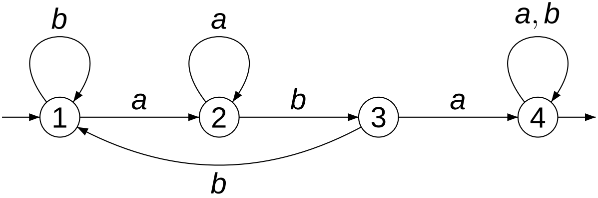
For any word $w \in \Sigma^*$ the automaton $\mathscr{A}_w \in$ **FG**.

$\mathscr{A}_w$ has $n = |w| + 1$ states $\Rightarrow$ its shortest reset word has length $n - 1$.

In general, the minimal automaton recognizing the language $\Sigma^* M \Sigma^*$ for a finite language M is in **FG**.

Finitely Generated Synchronizing Automata

The minimal automaton $\mathscr{A}_{aba}$ recognizing the language $\Sigma^* aba \Sigma^*$:



$Syn(\mathscr{A}_{aba}) = \Sigma^* aba \Sigma^* \Rightarrow Syn^{min}(\mathscr{A}_{aba}) = \{aba\} \Rightarrow \mathscr{A}_{aba} \in$ **FG**.

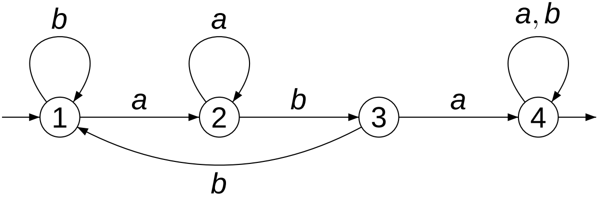
For any word $w \in \Sigma^*$ the automaton $\mathscr{A}_w \in$ **FG**.

$\mathscr{A}_w$ has $n = |w| + 1$ states $\Rightarrow$ its shortest reset word has length $n - 1$.

In general, the minimal automaton recognizing the language $\Sigma^* M \Sigma^*$ for a finite language M is in **FG**.

Finitely Generated Synchronizing Automata

The minimal automaton $\mathscr{A}_{aba}$ recognizing the language $\Sigma^* aba \Sigma^*$:



$Syn(\mathscr{A}_{aba}) = \Sigma^* aba \Sigma^* \Rightarrow Syn^{min}(\mathscr{A}_{aba}) = \{aba\} \Rightarrow \mathscr{A}_{aba} \in$ **FG.**

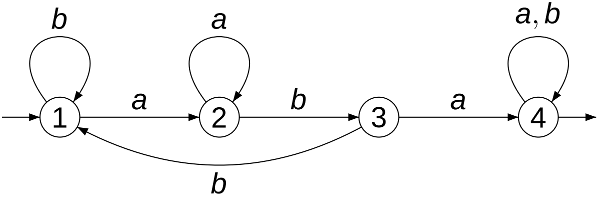
For any word $w \in \Sigma^*$ the automaton $\mathscr{A}_w \in$ **FG.**

$\mathscr{A}_w$ has $n = |w| + 1$ states $\Rightarrow$ its shortest reset word has length $n - 1$.

In general, the minimal automaton recognizing the language $\Sigma^* M \Sigma^*$ for a finite language M is in **FG.**

Finitely Generated Synchronizing Automata

The minimal automaton $\mathscr{A}_{aba}$ recognizing the language $\Sigma^* aba \Sigma^*$:



$Syn(\mathscr{A}_{aba}) = \Sigma^* aba \Sigma^* \Rightarrow Syn^{min}(\mathscr{A}_{aba}) = \{aba\} \Rightarrow \mathscr{A}_{aba} \in$ **FG**.

For any word $w \in \Sigma^*$ the automaton $\mathscr{A}_w \in$ **FG**.

$\mathscr{A}_w$ has $n = |w| + 1$ states $\Rightarrow$ its shortest reset word has length $n - 1$.

In general, the minimal automaton recognizing the language $\Sigma^* M \Sigma^*$ for a finite language M is in **FG**.

Auxiliary Definitions

- $T \subseteq Q$ is *reachable* if there is $v \in \Sigma^*$ with $T = Q \cdot v$.
- $Fix(T)$ is the set of all words *stabilizing* T:

$$Fix(T) = \{w \in \Sigma^* \mid T \cdot w = T\}.$$

- by $Syn(T)$ we denote the set of all words bringing T to a singleton:

$$Syn(T) = \{w \in \Sigma^* \mid |T \cdot w| = 1\}.$$

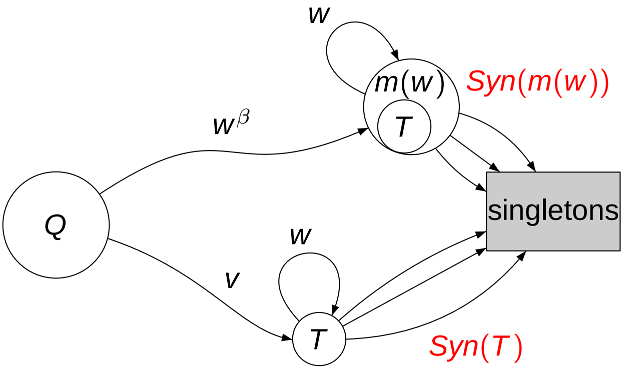
- Let $w \in \Sigma^*$, by $m(w) \subseteq Q$ we denote the *maximal fixed set with respect to* w: $m(w) \cdot w = m(w)$.

Characterization of **FG**

Theorem [Pribavkina, R. 2009]

A synchronizing automaton $\mathscr{A}$ is in ***FG*** *iff for any reachable non-singleton subset $T \subsetneq Q$, for each $w \in Fix(T)$ it holds*

$$Syn(T) = Syn(m(w)).$$



Černý's Conjecture for the Class **FG**

Theorem [Pribavkina, R. 2009]

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a finitely generated synchronizing automaton with n states. There is a synchronizing word of length at most $3n - 5$.

Remark

Take any letter $a \in \Sigma$, then either a^k or $a^k \tau a^k$ is synchronizing for some $k \leq n - |m(a)|$ and $|\tau| \leq n - 1$).

Open Problem

Is the bound $3n - 5$ for the class **FG** precise?

Černý's Conjecture for the Class **FG**

Theorem [Pribavkina, R. 2009]

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a finitely generated synchronizing automaton with n states. There is a synchronizing word of length at most $3n - 5$.

Remark

Take any letter $a \in \Sigma$, then either a^k or $a^k \tau a^k$ is synchronizing for some $k \leq n - |m(a)|$ and $|\tau| \leq n - 1$).

Open Problem

Is the bound $3n - 5$ for the class **FG** precise?

Černý's Conjecture for the Class **FG**

Theorem [Pribavkina, R. 2009]

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a finitely generated synchronizing automaton with n states. There is a synchronizing word of length at most $3n - 5$.

Remark

Take any letter $a \in \Sigma$, then either a^k or $a^k \tau a^k$ is synchronizing for some $k \leq n - |m(a)|$ and $|\tau| \leq n - 1$).

Open Problem

Is the bound $3n - 5$ for the class **FG** precise?

The FINITENESS Problem

Given a synchronizing automaton, how can we decide whether it is finitely generated? (It is not a property of a digraph as in case of many other classes of reset automata.)

FINITENESS

Input: A synchronizing DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.
Question: Is $\mathscr{A}$ finitely generated?

FINITENESS Problem is decidable

- $Syn^{min}(\mathscr{A}) = Syn(\mathscr{A}) \setminus (\Sigma Syn(\mathscr{A}) \cup Syn(\mathscr{A})\Sigma)$.
- The language $Syn(\mathscr{A})$ is regular (it is recognized by the power automaton $\mathcal{P}(\mathscr{A})$ with Q as an initial state and singletons as terminal ones).
- If $\mathscr{A}$ has n states, then $\mathcal{P}(\mathscr{A})$ has at most $2^n - 1$ states.
- $Syn^{min}(\mathcal{A})$ is recognized by an automaton with $O(2^{3n})$ states, thus checking the finiteness takes $O(2^{6n})$.

Another Algorithm

Our characterization gives rise to the following algorithm FINCHECK($\mathscr{A}$):

1 `From a synchronizing DFA` $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ `build the power automaton` $\mathcal{P}(\mathscr{A}) = \langle \mathcal{Q}, \Sigma, \delta \rangle$.

2 `For each state` T `of` $\mathcal{Q}$ `do:`

  2.1 `For each` H `of` $\mathcal{Q}$ `with` $T \subseteq H$ `do:`

    2.2 `If` $Fix(H) \cap Fix(T) \neq \varnothing$, `then`

      2.3 `If` $Syn(T) \neq Syn(H)$, `then exit and return NO`

3 `Otherwise exit and return YES`

Cost of this algorithm $O\left(2^{2n}3^n\right)$. With Savitch's trick:

Proposition

FINITENESS is in ***PSPACE****.*

Complexity of FINITENESS

Theorem [Pribavkina, R. 2009]

FINITENESS is co-NP-hard.

Theorem

*FINITENESS is **PSPACE**-complete.*

Proof Strategy: Well known **PSPACE**-complete problem:

NON-EMPTINESS DFA

Input: Given n DFA's $M_i = \langle Q_i, \Sigma, \delta_i, q_i, F_i \rangle$ for $i = 1, \ldots, n$.
Question: $\bigcap_i L[M_i] \neq \emptyset$?

Complexity of FINITENESS

Theorem [Pribavkina, R. 2009]

FINITENESS is co-NP-hard.

Theorem

FINITENESS is ***PSPACE****-complete.*

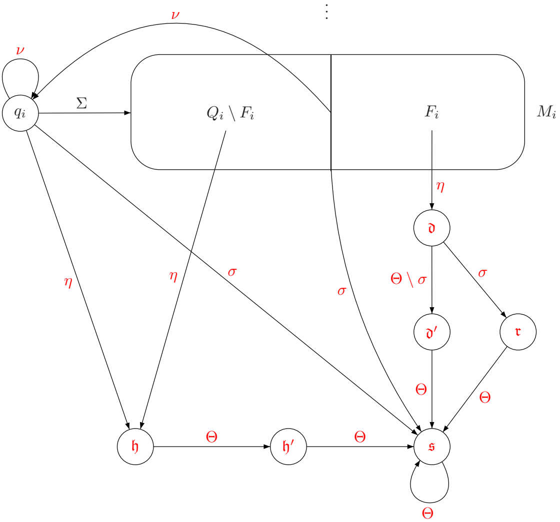
Proof Strategy: Well known **PSPACE**-complete problem:

NON-EMPTINESS DFA

Input: Given n DFA's $M_i = \langle Q_i, \Sigma, \delta_i, q_i, F_i \rangle$ for $i = 1, \ldots, n$.
Question: $\bigcap_i L[M_i] \neq \emptyset$?

Proof strategy (first step)

From $M_i = \langle Q_i, \Sigma, \delta_i, q_i, F_i \rangle$, $i = 1, \ldots, n$ we create another automaton $\mathcal{M} = \langle Q, \Theta, \phi \rangle$ with $Q = \cup_{i=1}^{n} Q_i \cup \{\mathfrak{d}, \mathfrak{d}', \mathfrak{h}, \mathfrak{h}', \mathfrak{r}, \mathfrak{s}\}$ and $\Theta = \Sigma \cup \{\sigma, \nu, \eta\}$



Proof strategy (first step)

Lemma (1)

There is a word $w \in \Theta^+$ *with* $|w| \geq 2$ *such that* $\phi(Q, w) = \{\mathfrak{s}, \mathfrak{r}\}$ *if and only if* $\bigcap_{i=0}^{n} L[M_i] \neq \varnothing$.

Lemma (2)

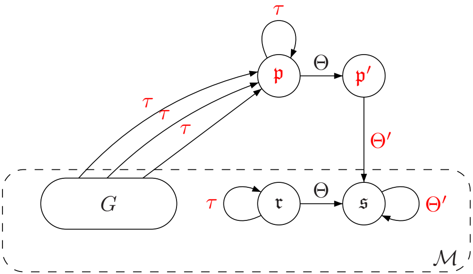
The automaton $\mathcal{M} = \langle Q, \Theta, \phi \rangle$ *previously defined belongs to the class* ***FG****.*

Proof strategy (second step)

From $\mathcal{M} = \langle Q, \Theta, \phi \rangle$ we build a new automaton $\mathcal{M}' = \langle Q', \Theta', \phi' \rangle$ with $Q' = Q \cup \{\mathfrak{p}, \mathfrak{p}'\}$, $\Theta' = \Theta \cup \{\tau\}$. Using Lemma (2) we get:

Lemma (3)

$Syn^{min}(\mathcal{M}')$ is infinite if and only if there exists $w \in \Theta^+$ such that $\phi(Q, w) = \{\mathfrak{r}, \mathfrak{s}\}$.



Experimental Results

- For the binary alphabet we have the following table:

n	Reset Automata	Finitely Generated Automata
1	1	1
2	12	12
3	549	405
4	51520	26032

Open Problem

Given a n-state synchronizing automaton, what is the probability that it is finitely generated?

Experimental Results

- For the binary alphabet we have the following table:

n	Reset Automata	Finitely Generated Automata
1	1	1
2	12	12
3	549	405
4	51520	26032

Open Problem

Given a n-state synchronizing automaton, what is the probability that it is finitely generated?

Other open problems for the class **FG**

- The characterization is given in terms of the power automaton. Is there a characterization in terms of the transition monoid of $\mathscr{A}$ or using linear algebra methods?
- If $\mathscr{A} \in$ **FG** give a bound for the length of the longest word in $Syn^{min}(\mathscr{A})$ and a bound to $|Syn^{min}(\mathscr{A})|$. We have proved

Theorem

*Let $\mathscr{A} \in$ **FG** with n states, let N be the number of non-singleton states of $\mathcal{P}(\mathscr{A})$ consisting of only reachable subsets. Then the length of any minimal synchronizing word is at most*

$$N^2 - N + 1 \leq (2^n - n - 1)^2 - 2^n - n$$

Is there a polynomial bound?

Thank you

for your attention!