Synchronizing Finite Automata
III. The Černý Conjecture

Mikhail Volkov

Ural State University, Ekaterinburg, Russia

CSClub, St Petersburg, November 14, 2010

1. Recap

Deterministic finite automata: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.

- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ 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 $q, q' \in Q$.

$|Q\,.\,w| = 1$. Here $Q\,.\,v = \{\delta(q, v) \mid q \in Q\}$.

Any w with this property is a reset word for $\mathscr{A}$.

1. Recap

Deterministic finite automata: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.

- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ 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 $q, q' \in Q$.

$|Q\,.\,w| = 1$. Here $Q\,.\,v = \{\delta(q, v) \mid q \in Q\}$.

Any w with this property is a reset word for $\mathscr{A}$.

1. Recap

Deterministic finite automata: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.

- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ 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 $q, q' \in Q$.
$|Q . w| = 1$. Here $Q . v = \{\delta(q, v) \mid q \in Q\}$.

Any w with this property is a reset word for $\mathscr{A}$.

1. Recap

Deterministic finite automata: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$.

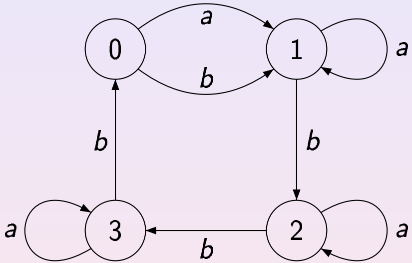
- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ 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 $q, q' \in Q$.

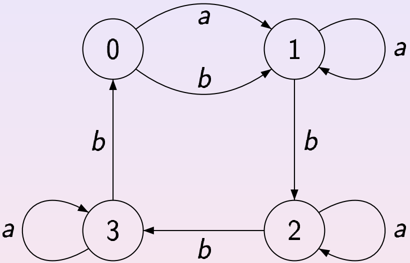
$|Q \,.\, w| = 1$. Here $Q \,.\, v = \{\delta(q, v) \mid q \in Q\}$.

Any w with this property is a reset word for $\mathscr{A}$.

2. Example



A reset word is *abbbabbba*. In fact, we have verified that this is the shortest reset word for this automaton.



A reset word is *abbbabbba*. In fact, we have verified that this is the shortest reset word for this automaton.

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

We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?

In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \ldots,$ of synchronizing automata over 2 letters.

The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

$$\delta(0, a) = 1,\ \delta(m, a) = m \text{ for } 0 < m < n,\ \delta(m, b) = m+1 \pmod{n}.$$

The automaton in the previous slide is $\mathscr{C}_4$.

3. The Černý Series

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

We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?

In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \dots$, of synchronizing automata over 2 letters.

The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

$$\delta(0, a) = 1,\ \delta(m, a) = m \text{ for } 0 < m < n, \delta(m, b) = m+1 \pmod{n}.$$

The automaton in the previous slide is $\mathscr{C}_4$.

3. The Černý Series

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

We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?

In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \dots,$ of synchronizing automata over 2 letters.

The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

$$\delta(0, a) = 1, \ \delta(m, a) = m \text{ for } 0 < m < n, \delta(m, b) = m+1 \pmod{n}.$$

The automaton in the previous slide is $\mathscr{C}_4$.

3. The Černý Series

Suppose a synchronizing automaton has n states. What is the length of its shortest reset word?
We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?
In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \dots$, of synchronizing automata over 2 letters.
The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

$$\delta(0, a) = 1,\ \delta(m, a) = m \text{ for } 0 < m < n, \delta(m, b) = m+1 \pmod{n}.$$

The automaton in the previous slide is $\mathscr{C}_4$.

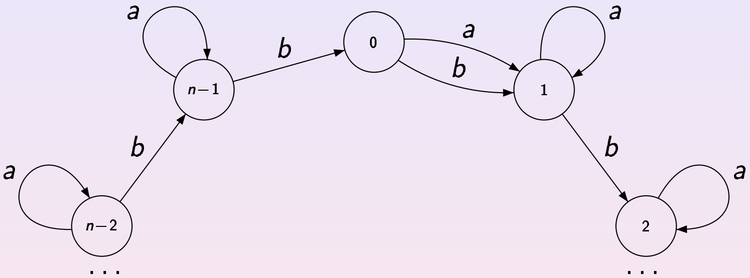
3. The Černý Series

Suppose a synchronizing automaton has n states. What is the length of its shortest reset word?
We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?
In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \ldots$, of synchronizing automata over 2 letters.
The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

$$\delta(0, a) = 1,\ \delta(m, a) = m \text{ for } 0 < m < n, \delta(m, b) = m+1 \pmod{n}.$$

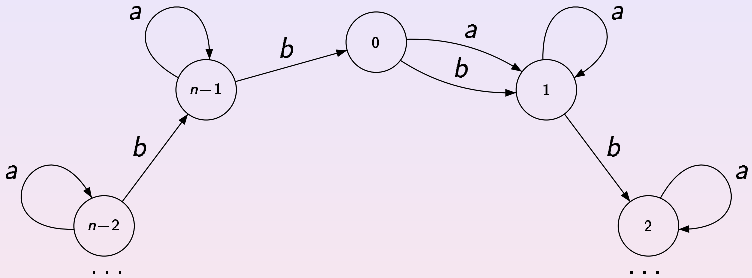
The automaton in the previous slide is $\mathscr{C}_4$.

Here is a generic automaton from the Černý series:



Černý has proved that the shortest reset word for $\mathscr{C}_n$ is $(ab^{n-1})^{n-2}a$ of length $(n-1)^2$. As other results from Černý's paper of 1964, this nice series of automata has been rediscovered many times.

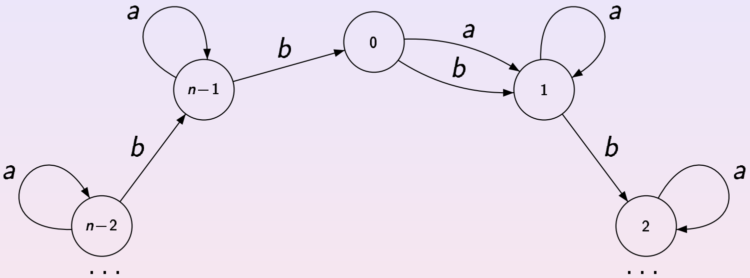
Here is a generic automaton from the Černý series:



Černý has proved that the shortest reset word for $\mathscr{C}_n$ is $(ab^{n-1})^{n-2}a$ of length $(n-1)^2$. As other results from Černý's paper of 1964, this nice series of automata has been rediscovered many times.

4. The Černý Series

Here is a generic automaton from the Černý series:



Černý has proved that the shortest reset word for $\mathscr{C}_n$ is $(ab^{n-1})^{n-2}a$ of length $(n-1)^2$. As other results from Černý's paper of 1964, this nice series of automata has been rediscovered many times.

5. Game

We present a proof of this result using a solitaire-like game.

- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

5. Game

We present a proof of this result using a solitaire-like game.

- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

5. Game

We present a proof of this result using a solitaire-like game.

- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

5. Game

We present a proof of this result using a solitaire-like game.

- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

5. Game

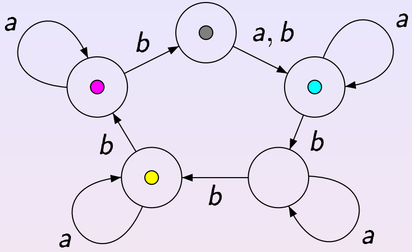
We present a proof of this result using a solitaire-like game.

- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

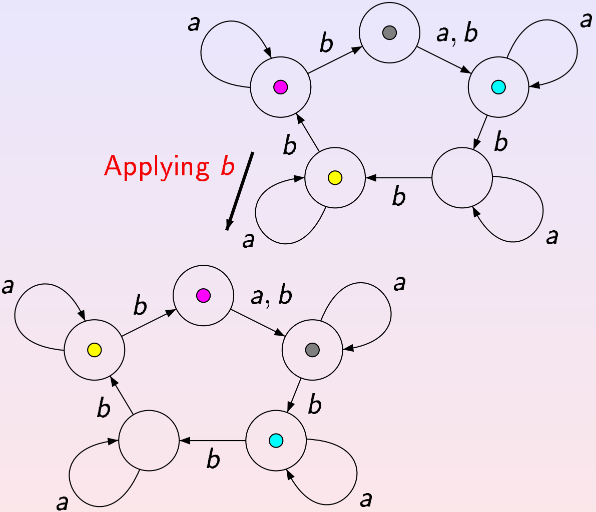
We present a proof of this result using a solitaire-like game.

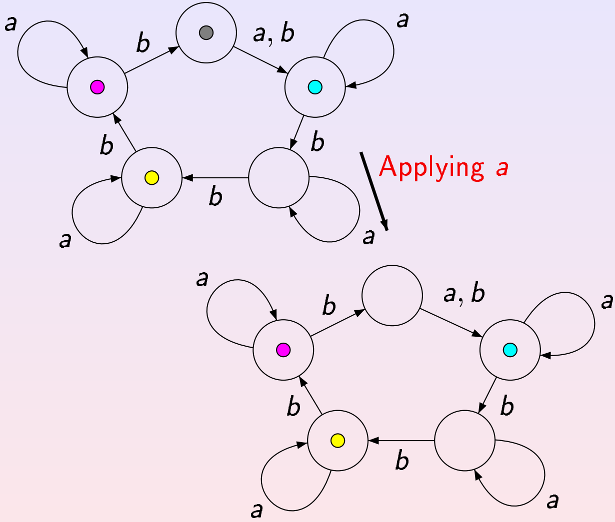
- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.
- The goal — to free all but one states.
- The only coin that remains at the end of the game is the golden coin G.

6. Example



6. Example





CSClub, St Petersburg, November 14, 2010

7. Key Idea

Let P_0 be an initial distribution of coins, w a reset word. Denote by P_i the position that arises when we apply the prefix of w of length i to the position P_0. We want to define the weight $\mathrm{wg}(P_i)$ of the position such that

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) for each $i = 1, \ldots, |w|$, the action of the i^{th} letter of w decreases the weight by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

Then $|w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} (\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)) =$

$$\mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq n(n-1) - (n-1) = (n-1)^2.$$

7. Key Idea

Let P_0 be an initial distribution of coins, w a reset word. Denote by P_i the position that arises when we apply the prefix of w of length i to the position P_0. We want to define the weight $\mathrm{wg}(P_i)$ of the position such that

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) for each $i = 1, \dots, |w|$, the action of the i^{th} letter of w decreases the weight by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

Then $$|w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} (\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)) =$$

$$\mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq n(n-1) - (n-1) = (n-1)^2.$$

7. Key Idea

Let P_0 be an initial distribution of coins, w a reset word. Denote by P_i the position that arises when we apply the prefix of w of length i to the position P_0. We want to define the weight $\mathrm{wg}(P_i)$ of the position such that

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) for each $i = 1, \dots, |w|$, the action of the i^{th} letter of w decreases the weight by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

$$\text{Then } |w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} (\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)) =$$

$$\mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq n(n-1) - (n-1) = (n-1)^2.$$

7. Key Idea

Let P_0 be an initial distribution of coins, w a reset word. Denote by P_i the position that arises when we apply the prefix of w of length i to the position P_0. We want to define the weight $\mathrm{wg}(P_i)$ of the position such that

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) for each $i = 1, \ldots, |w|$, the action of the i^{th} letter of w decreases the weight by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

Then $|w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} (\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)) =$

$$\mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq n(n-1) - (n-1) = (n-1)^2.$$

7. Key Idea

Let P_0 be an initial distribution of coins, w a reset word. Denote by P_i the position that arises when we apply the prefix of w of length i to the position P_0. We want to define the weight $\mathrm{wg}(P_i)$ of the position such that

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) for each $i = 1, \dots, |w|$, the action of the i^{th} letter of w decreases the weight by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

Then $|w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} (\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)) =$

$$\mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq n(n-1) - (n-1) = (n-1)^2.$$

The trick consists in letting the weight of each coin depend on its relative location w.r.t. the golden coin.

If a coin C is present in a position P_i, let $s_i(C)$ be the state covered with C in this position. We define the *weight of C in the position P_i* as

$$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C)$$

where $m_i(C)$ is the residue of $n - s_i(C)$ modulo n and $d_i(C)$ is the number of steps from $s_i(C)$ to $s_i(G)$ in the 'main circle' of our automaton. (Recall that G stands for the golden coin G which is present in all positions.)

The weight of P_i is the maximum weight of the coins present in this position.

8. Constructing the Weight Function

The trick consists in letting the weight of each coin depend on its relative location w.r.t. the golden coin.

If a coin C is present in a position P_i, let $s_i(C)$ be the state covered with C in this position. We define the *weight of* C *in the position* P_i as

$$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C)$$

where $m_i(C)$ is the residue of $n - s_i(C)$ modulo n and $d_i(C)$ is the number of steps from $s_i(C)$ to $s_i(G)$ in the 'main circle' of our automaton. (Recall that G stands for the golden coin G which is present in all positions.)

The weight of P_i is the maximum weight of the coins present in this position.

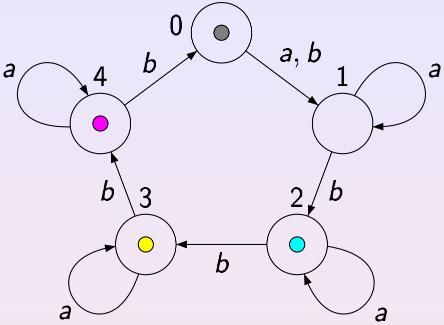
8. Constructing the Weight Function

The trick consists in letting the weight of each coin depend on its relative location w.r.t. the golden coin.
If a coin C is present in a position P_i, let $s_i(C)$ be the state covered with C in this position. We define the *weight of C in the position P_i* as

$$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C)$$

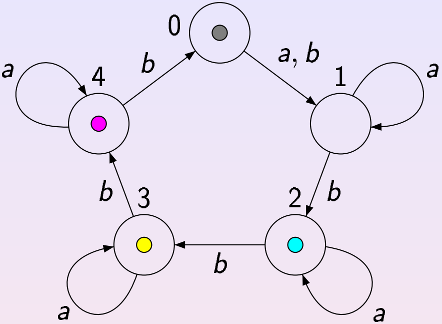
where $m_i(C)$ is the residue of $n - s_i(C)$ modulo n and $d_i(C)$ is the number of steps from $s_i(C)$ to $s_i(G)$ in the 'main circle' of our automaton. (Recall that G stands for the golden coin G which is present in all positions.)
The weight of P_i is the maximum weight of the coins present in this position.

9. Example



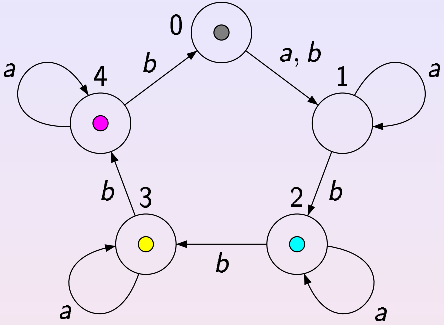
Assume that the yellow coin is the golden one. Then its weight is $5 - 3 = 2$. The weight of the cyan coin is $5 \cdot 1 + (5 - 2) = 8$. The weight of the gray coin is $5 \cdot 3 + 0 = 15$. The weight of the magenta coin is $5 \cdot 4 + (5 - 4) = 21$, and this is the weight of the position.

9. Example

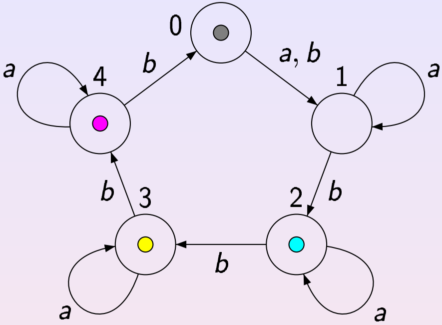


Assume that the yellow coin is the golden one. Then its weight is $5-3=2$. The weight of the cyan coin is $5\cdot 1+(5-2)=8$. The weight of the gray coin is $5\cdot 3+0=15$. The weight of the magenta coin is $5\cdot 4+(5-4)=21$, and this is the weight of the position.

9. Example



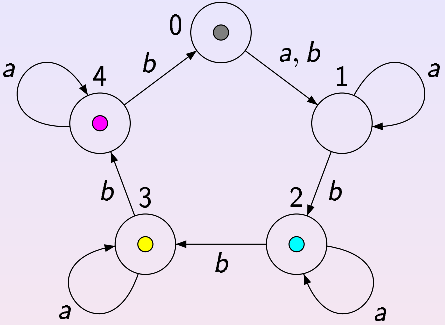
Assume that the yellow coin is the golden one. Then its weight is $5 - 3 = 2$. The weight of the cyan coin is $5 \cdot 1 + (5 - 2) = 8$. The weight of the gray coin is $5 \cdot 3 + 0 = 15$. The weight of the magenta coin is $5 \cdot 4 + (5 - 4) = 21$, and this is the weight of the position.



Assume that the yellow coin is the golden one. Then its weight is $5 - 3 = 2$. The weight of the cyan coin is $5 \cdot 1 + (5 - 2) = 8$. The weight of the gray coin is $5 \cdot 3 + 0 = 15$. The weight of the magenta coin is $5 \cdot 4 + (5 - 4) = 21$, and this is the weight of the position.

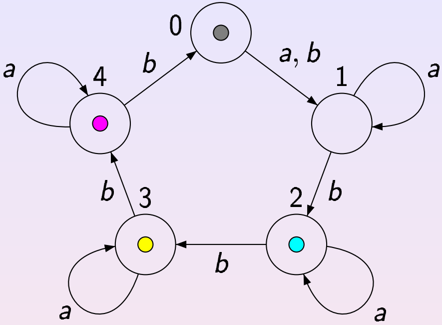
CSClub, St Petersburg, November 14, 2010

9. Example



Assume that the yellow coin is the golden one. Then its weight is $5-3=2$. The weight of the cyan coin is $5\cdot 1+(5-2)=8$. The weight of the gray coin is $5\cdot 3+0=15$. The weight of the magenta coin is $5\cdot 4+(5-4)=21$, and this is the weight of the position.

9. Example



Assume that the yellow coin is the golden one. Then its weight is $5 - 3 = 2$. The weight of the cyan coin is $5 \cdot 1 + (5 - 2) = 8$. The weight of the gray coin is $5 \cdot 3 + 0 = 15$. The weight of the magenta coin is $5 \cdot 4 + (5 - 4) = 21$, and this is the weight of the position.

10. Properties of the Weight Function

We have to check that our weight function satisfies the conditions

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$ for each $i = 1, \dots, |w|$.

In the initial position all states are covered with coins. Consider the coin C that covers the state $s_0(G) + 1 \pmod{n}$, that is the state in one step clockwise after the state covered with the golden coin. Then $d_0(C) = n - 1$ whence

$$\mathrm{wg}(C, P_0) = n \cdot (n-1) + m_0(C) \geq n(n-1).$$

Since the weight of a position is not less than the weight of any coin in this position, we have $\mathrm{wg}(P_0) \geq n(n-1)$.

10. Properties of the Weight Function

We have to check that our weight function satisfies the conditions

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$ for each $i = 1, \dots, |w|$.

In the initial position all states are covered with coins. Consider the coin C that covers the state $s_0(G) + 1 (\mathrm{mod}\ n)$, that is the state in one step clockwise after the state covered with the golden coin. Then $d_0(C) = n - 1$ whence

$$\mathrm{wg}(C, P_0) = n \cdot (n-1) + m_0(C) \geq n(n-1).$$

Since the weight of a position is not less than the weight of any coin in this position, we have $\mathrm{wg}(P_0) \geq n(n-1)$.

10. Properties of the Weight Function

We have to check that our weight function satisfies the conditions

(i) $\mathrm{wg}(P_0) \geq n(n-1)$ and $\mathrm{wg}(P_{|w|}) \leq n-1$;

(ii) $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$ for each $i = 1, \ldots, |w|$.

In the initial position all states are covered with coins. Consider the coin C that covers the state $s_0(G) + 1 \pmod{n}$, that is the state in one step clockwise after the state covered with the golden coin. Then $d_0(C) = n - 1$ whence

$$\mathrm{wg}(C, P_0) = n \cdot (n-1) + m_0(C) \geq n(n-1).$$

Since the weight of a position is not less than the weight of any coin in this position, we have $\mathrm{wg}(P_0) \geq n(n-1)$.

In the final position only the golden coin G remains, whence the weight of $P_{|w|}$ is the weight of G. Clearly, $\mathrm{wg}(G, P_i) = m_i(G) \leq n-1$ for any position P_i.

Let C be a coin of maximum weight in P_{i-1}. If the transition from P_{i-1} to P_i is caused by b, then $d_i(C) = d_{i-1}(C)$ (because the relative location of the coins does not change) and $m_i(C) = m_{i-1}(C) - 1$ if $m_{i-1}(C) > 0$, otherwise $m_i(C) = n-1$. We see that

$$\mathrm{wg}(P_i) \geq \mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq$$
$$n \cdot d_{i-1}(C) + m_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1.$$

11. Properties of the Weight Function

In the final position only the golden coin G remains, whence the weight of $P_{|w|}$ is the weight of G. Clearly, $\mathrm{wg}(G, P_i) = m_i(G) \leq n - 1$ for any position P_i.

Let C be a coin of maximum weight in P_{i-1}. If the transition from P_{i-1} to P_i is caused by b, then $d_i(C) = d_{i-1}(C)$ (because the relative location of the coins does not change) and $m_i(C) = m_{i-1}(C) - 1$ if $m_{i-1}(C) > 0$, otherwise $m_i(C) = n - 1$. We see that

$$\mathrm{wg}(P_i) \geq \mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq$$
$$n \cdot d_{i-1}(C) + m_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1.$$

11. Properties of the Weight Function

In the final position only the golden coin G remains, whence the weight of $P_{|w|}$ is the weight of G. Clearly, $\mathrm{wg}(G, P_i) = m_i(G) \leq n - 1$ for any position P_i.

Let C be a coin of maximum weight in P_{i-1}. If the transition from P_{i-1} to P_i is caused by b, then $d_i(C) = d_{i-1}(C)$ (because the relative location of the coins does not change) and $m_i(C) = m_{i-1}(C) - 1$ if $m_{i-1}(C) > 0$, otherwise $m_i(C) = n - 1$. We see that

$$\mathrm{wg}(P_i) \geq \mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq$$
$$n \cdot d_{i-1}(C) + m_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1.$$

12. Properties of the Weight Function

Suppose the transition from P_{i-1} to P_i is caused by a. If $s_{i-1}(C) \neq 0$, then $m_i(C) = m_{i-1}(C)$ and $d_i(C) = d_{i-1}(C)$ if $s_{i-1}(G) \neq 0$, otherwise $d_i(C) = d_{i-1}(C) + 1$. Thus, the transition from P_{i-1} to P_i cannot decrease the weight.

Assume that C covers 0 in P_{i-1}. Then in P_i the state 1 holds a coin C' (which may or may not coincide with C). In P_{i-1} the golden coin G does not cover 0 whence it does not move and $d_i(C') = d_{i-1}(C) - 1$. Therefore

$$\mathrm{wg}(P_i) \geq \mathrm{wg}(C', P_i) = n \cdot d_i(C') + n - 1 = n \cdot (d_{i-1}(C) - 1) + n - 1$$
$$= n \cdot d_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1.$$

12. Properties of the Weight Function

Suppose the transition from P_{i-1} to P_i is caused by a. If $s_{i-1}(C) \neq 0$, then $m_i(C) = m_{i-1}(C)$ and $d_i(C) = d_{i-1}(C)$ if $s_{i-1}(G) \neq 0$, otherwise $d_i(C) = d_{i-1}(C) + 1$. Thus, the transition from P_{i-1} to P_i cannot decrease the weight.

Assume that C covers 0 in P_{i-1}. Then in P_i the state 1 holds a coin C' (which may or may not coincide with C). In P_{i-1} the golden coin G does not cover 0 whence it does not move and $d_i(C') = d_{i-1}(C) - 1$. Therefore

$$\begin{aligned} \mathrm{wg}(P_i) \geq \mathrm{wg}(C', P_i) = n \cdot d_i(C') + n - 1 = n \cdot (d_{i-1}(C) - 1) + n - 1 \\ = n \cdot d_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1. \end{aligned}$$

12. Properties of the Weight Function

Suppose the transition from P_{i-1} to P_i is caused by a. If $s_{i-1}(C) \neq 0$, then $m_i(C) = m_{i-1}(C)$ and $d_i(C) = d_{i-1}(C)$ if $s_{i-1}(G) \neq 0$, otherwise $d_i(C) = d_{i-1}(C) + 1$. Thus, the transition from P_{i-1} to P_i cannot decrease the weight.

Assume that C covers 0 in P_{i-1}. Then in P_i the state 1 holds a coin C' (which may or may not coincide with C). In P_{i-1} the golden coin G does not cover 0 whence it does not move and $d_i(C') = d_{i-1}(C) - 1$. Therefore

$$\begin{aligned} \mathrm{wg}(P_i) \geq \mathrm{wg}(C', P_i) &= n \cdot d_i(C') + n - 1 = n \cdot (d_{i-1}(C) - 1) + n - 1 \\ &= n \cdot d_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1. \end{aligned}$$

12. Properties of the Weight Function

Suppose the transition from P_{i-1} to P_i is caused by a. If $s_{i-1}(C) \neq 0$, then $m_i(C) = m_{i-1}(C)$ and $d_i(C) = d_{i-1}(C)$ if $s_{i-1}(G) \neq 0$, otherwise $d_i(C) = d_{i-1}(C) + 1$. Thus, the transition from P_{i-1} to P_i cannot decrease the weight.

Assume that C covers 0 in P_{i-1}. Then in P_i the state 1 holds a coin C' (which may or may not coincide with C). In P_{i-1} the golden coin G does not cover 0 whence it does not move and $d_i(C') = d_{i-1}(C) - 1$. Therefore

$$\begin{aligned} \mathrm{wg}(P_i) \geq \mathrm{wg}(C', P_i) &= n \cdot d_i(C') + n - 1 = n \cdot (d_{i-1}(C) - 1) + n - 1 \\ &= n \cdot d_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1. \end{aligned}$$

13. More on Games

Assume that there are two players: Alice (Synchronizer) and Bob (Desynchronizer) whose moves alternate. Alice (who pays first) wants to synchronize the given automaton, Bob aims to make her task as hard as possible.

- Bob can win on a synchronizing automaton (for instance, he wins on $\mathscr{C}_n$).
- Given $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can decide who wins in $O(|Q|^2 \cdot |\Sigma|)$ time.
- If Alice wins, she can win in $O(|Q|^3|)$ moves.
- For every synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can construct an automaton $\mathscr{B}$ with $2|Q|$ states such that Alice wins on $\mathscr{B}$ but the minimum number of moves she needs to win is no less than the minimum length of reset words for $\mathscr{A}$.

13. More on Games

Assume that there are two players: Alice (Synchronizer) and Bob (Desynchronizer) whose moves alternate. Alice (who pays first) wants to synchronize the given automaton, Bob aims to make her task as hard as possible.

- Bob can win on a synchronizing automaton (for instance, he wins on $\mathscr{C}_n$).
- Given $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can decide who wins in $O(|Q|^2 \cdot |\Sigma|)$ time.
- If Alice wins, she can win in $O(|Q|^3|)$ moves.
- For every synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can construct an automaton $\mathscr{B}$ with $2|Q|$ states such that Alice wins on $\mathscr{B}$ but the minimum number of moves she needs to win is no less than the minimum length of reset words for $\mathscr{A}$.

13. More on Games

Assume that there are two players: Alice (Synchronizer) and Bob (Desynchronizer) whose moves alternate. Alice (who pays first) wants to synchronize the given automaton, Bob aims to make her task as hard as possible.

- Bob can win on a synchronizing automaton (for instance, he wins on $\mathscr{C}_n$).
- Given $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can decide who wins in $O(|Q|^2 \cdot |\Sigma|)$ time.
- If Alice wins, she can win in $O(|Q|^3|)$ moves.
- For every synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can construct an automaton $\mathscr{B}$ with $2|Q|$ states such that Alice wins on $\mathscr{B}$ but the minimum number of moves she needs to win is no less than the minimum length of reset words for $\mathscr{A}$.

13. More on Games

Assume that there are two players: Alice (Synchronizer) and Bob (Desynchronizer) whose moves alternate. Alice (who pays first) wants to synchronize the given automaton, Bob aims to make her task as hard as possible.

- Bob can win on a synchronizing automaton (for instance, he wins on $\mathscr{C}_n$).
- Given $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can decide who wins in $O(|Q|^2 \cdot |\Sigma|)$ time.
- If Alice wins, she can win in $O(|Q|^3|)$ moves.
- For every synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can construct an automaton $\mathscr{B}$ with $2|Q|$ states such that Alice wins on $\mathscr{B}$ but the minimum number of moves she needs to win is no less than the minimum length of reset words for $\mathscr{A}$.

13. More on Games

Assume that there are two players: Alice (Synchronizer) and Bob (Desynchronizer) whose moves alternate. Alice (who pays first) wants to synchronize the given automaton, Bob aims to make her task as hard as possible.

- Bob can win on a synchronizing automaton (for instance, he wins on $\mathscr{C}_n$).
- Given $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can decide who wins in $O(|Q|^2 \cdot |\Sigma|)$ time.
- If Alice wins, she can win in $O(|Q|^3|)$ moves.
- For every synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, one can construct an automaton $\mathscr{B}$ with $2|Q|$ states such that Alice wins on $\mathscr{B}$ but the minimum number of moves she needs to win is no less than the minimum length of reset words for $\mathscr{A}$.

14. The Černý function

Define the *Černý function* $C(n)$ as the maximum length of shortest reset words for synchronizing automata with n states. The above property of the series $\{\mathscr{C}_n\}$, $n = 2, 3, \ldots$, yields the inequality $C(n) \geq (n-1)^2$.

The Černý conjecture is the claim that in fact the equality $C(n) = (n-1)^2$ holds true. This simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. Everything we know about the conjecture in general can be summarized in one line:

$$(n-1)^2 \leq C(n) \leq \frac{n^3 - n}{6}.$$

14. The Černý function

Define the *Černý function* $C(n)$ as the maximum length of shortest reset words for synchronizing automata with n states. The above property of the series $\{\mathscr{C}_n\}$, $n = 2, 3, \dots$, yields the inequality $C(n) \geq (n-1)^2$.

The Černý conjecture is the claim that in fact the equality $C(n) = (n-1)^2$ holds true. This simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. Everything we know about the conjecture in general can be summarized in one line:

$$(n-1)^2 \leq C(n) \leq \frac{n^3 - n}{6}.$$

14. The Černý function

Define the *Černý function* $C(n)$ as the maximum length of shortest reset words for synchronizing automata with n states. The above property of the series $\{\mathscr{C}_n\}$, $n = 2, 3, \dots$, yields the inequality $C(n) \geq (n-1)^2$.

The Černý conjecture is the claim that in fact the equality $C(n) = (n-1)^2$ holds true. This simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. Everything we know about the conjecture in general can be summarized in one line:

$$(n-1)^2 \leq C(n) \leq \frac{n^3 - n}{6}.$$

14. The Černý function

Define the *Černý function* $C(n)$ as the maximum length of shortest reset words for synchronizing automata with n states. The above property of the series $\{\mathscr{C}_n\}$, $n = 2, 3, \dots$, yields the inequality $C(n) \geq (n-1)^2$.

The Černý conjecture is the claim that in fact the equality $C(n) = (n-1)^2$ holds true. This simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. Everything we know about the conjecture in general can be summarized in one line:

$$(n-1)^2 \leq C(n) \leq \frac{n^3 - n}{6}.$$

15. Why it is hard?

Why is the problem so surprisingly difficult?

We saw two reasons:

- non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);
- combinatorics of finite sets is encoded in the problem.

Yet another reason: "slowly" synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \dots$, with a few sporadic examples for $n \leq 6$.

15. Why it is hard?

Why is the problem so surprisingly difficult?

We saw two reasons:

- non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);
- combinatorics of finite sets is encoded in the problem.

Yet another reason: "slowly" synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \dots$, with a few sporadic examples for $n \leq 6$.

15. Why it is hard?

Why is the problem so surprisingly difficult?

We saw two reasons:

- non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);
- combinatorics of finite sets is encoded in the problem.

Yet another reason: “slowly” synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots$, with a few sporadic examples for $n \leq 6$.

15. Why it is hard?

Why is the problem so surprisingly difficult?

We saw two reasons:

• non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);

• combinatorics of finite sets is encoded in the problem.

Yet another reason: "slowly" synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \dots$, with a few sporadic examples for $n \leq 6$.

15. Why it is hard?

Why is the problem so surprisingly difficult?

We saw two reasons:

• non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);

• combinatorics of finite sets is encoded in the problem.

Yet another reason: "slowly" synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots$, with a few sporadic examples for $n \leq 6$.

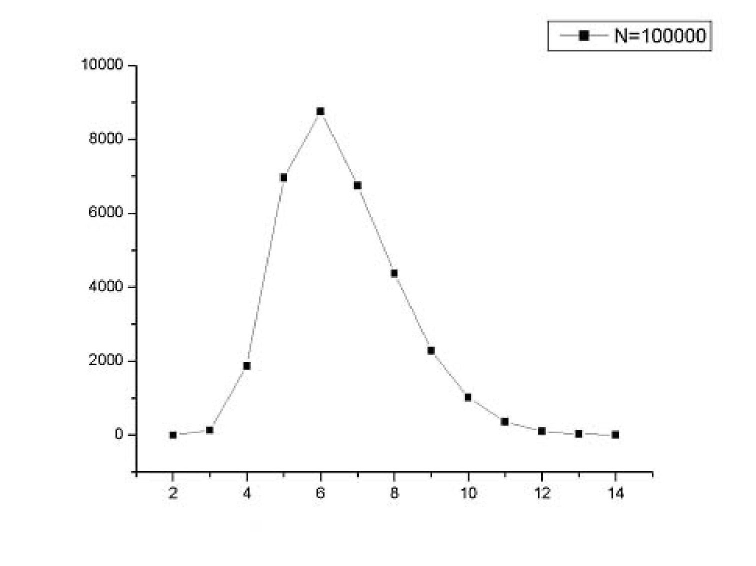
Why is the problem so surprisingly difficult?

We saw two reasons:

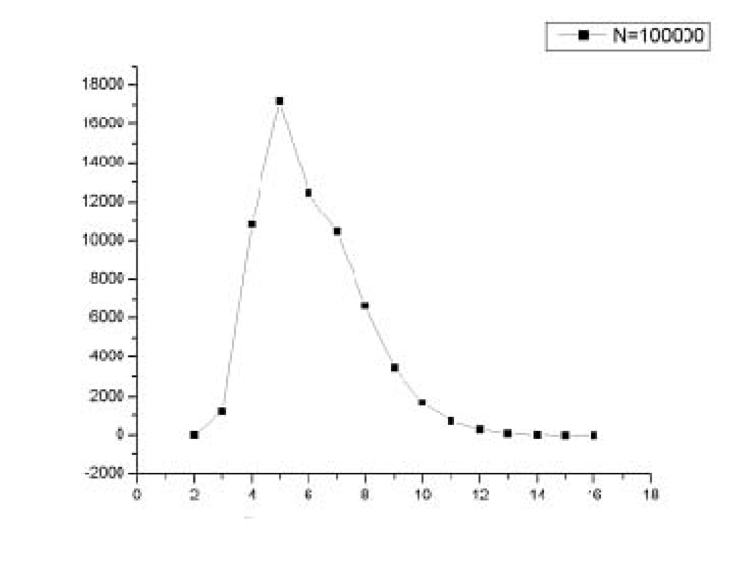
- non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);
- combinatorics of finite sets is encoded in the problem.

Yet another reason: "slowly" synchronizing automata turn out to be extremely rare. The only known infinite series of n-state synchronizing automata with shortest reset words of length $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots$, with a few sporadic examples for $n \leq 6$.

16. 20-State Experiment



17. 30-State Experiment



18. Random Automata

A (partial) explanation of these experimental observations: if Q is an n-set (with n large enough), then, on average, any product of $2n$ randomly chosen transformations of Q is a constant map (Peter Higgins, The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum, 37 (1988) 31–36). In automata-theoretic terms, this fact means that a randomly chosen DFA with n states and a sufficiently large input alphabet tends to be synchronizing and is reset by any word of length $\geq 2n$.

Thus, "slowly" synchronizing automata cannot be discovered via a random sampling.

18. Random Automata

A (partial) explanation of these experimental observations: if Q is an n-set (with n large enough), then, on average, any product of $2n$ randomly chosen transformations of Q is a constant map (Peter Higgins, The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum, 37 (1988) 31–36). In automata-theoretic terms, this fact means that a randomly chosen DFA with n states and a sufficiently large input alphabet tends to be synchronizing and is reset by any word of length $\geq 2n$.

Thus, "slowly" synchronizing automata cannot be discovered via a random sampling.

18. Random Automata

A (partial) explanation of these experimental observations: if Q is an n-set (with n large enough), then, on average, any product of $2n$ randomly chosen transformations of Q is a constant map (Peter Higgins, The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum, 37 (1988) 31–36). In automata-theoretic terms, this fact means that a randomly chosen DFA with n states and a sufficiently large input alphabet tends to be synchronizing and is reset by any word of length $\geq 2n$.

Thus, "slowly" synchronizing automata cannot be discovered via a random sampling.

A (partial) explanation of these experimental observations: if Q is an n-set (with n large enough), then, on average, any product of $2n$ randomly chosen transformations of Q is a constant map (Peter Higgins, The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum, 37 (1988) 31–36). In automata-theoretic terms, this fact means that a randomly chosen DFA with n states and a sufficiently large input alphabet tends to be synchronizing and is reset by any word of length $\geq 2n$.

Thus, "slowly" synchronizing automata cannot be discovered via a random sampling.

19. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *proper* if none of the automata obtained from $\mathscr{A}$ by erasing any letter in Σ are synchronizing. E.g., the Černý automata $\mathscr{C}_n$ with $n > 2$ are proper while $\mathscr{C}_2$ is not.

A synchronizing automaton with n states *reaches the Černý bound* if the minimum length of its reset words is $(n-1)^2$. We present here all known proper synchronizing automata beyond the Černý series $\mathscr{C}_n$, $n = 3, 4, \dots$, that reach the Černý bound.

For the sake of completeness, we start with $n = 2$:

19. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *proper* if none of the automata obtained from $\mathscr{A}$ by erasing any letter in Σ are synchronizing. E.g., the Černý automata $\mathscr{C}_n$ with $n > 2$ are proper while $\mathscr{C}_2$ is not.

A synchronizing automaton with n states *reaches the Černý bound* if the minimum length of its reset words is $(n-1)^2$. We present here all known proper synchronizing automata beyond the Černý series $\mathscr{C}_n$, $n = 3, 4, \ldots$, that reach the Černý bound.

For the sake of completeness, we start with $n = 2$:

19. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *proper* if none of the automata obtained from $\mathscr{A}$ by erasing any letter in Σ are synchronizing. E.g., the Černý automata $\mathscr{C}_n$ with $n > 2$ are proper while $\mathscr{C}_2$ is not.

A synchronizing automaton with n states *reaches the Černý bound* if the minimum length of its reset words is $(n-1)^2$. We present here all known proper synchronizing automata beyond the Černý series $\mathscr{C}_n$, $n = 3, 4, \dots,$ that reach the Černý bound.

For the sake of completeness, we start with $n = 2$:

19. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *proper* if none of the automata obtained from $\mathscr{A}$ by erasing any letter in Σ are synchronizing. E.g., the Černý automata $\mathscr{C}_n$ with $n > 2$ are proper while $\mathscr{C}_2$ is not.

A synchronizing automaton with n states *reaches the Černý bound* if the minimum length of its reset words is $(n-1)^2$. We present here all known proper synchronizing automata beyond the Černý series $\mathscr{C}_n$, $n = 3, 4, \dots$, that reach the Černý bound.

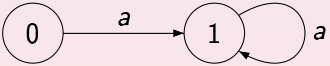
For the sake of completeness, we start with $n = 2$:

19. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *proper* if none of the automata obtained from $\mathscr{A}$ by erasing any letter in Σ are synchronizing. E.g., the Černý automata $\mathscr{C}_n$ with $n > 2$ are proper while $\mathscr{C}_2$ is not.

A synchronizing automaton with n states *reaches the Černý bound* if the minimum length of its reset words is $(n-1)^2$. We present here all known proper synchronizing automata beyond the Černý series $\mathscr{C}_n$, $n = 3, 4, \ldots$, that reach the Černý bound.

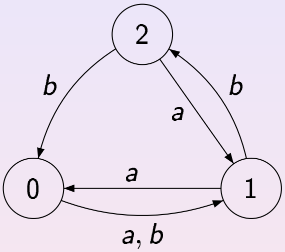
For the sake of completeness, we start with $n = 2$:



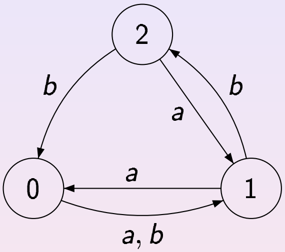
20. Sporadic Examples: $n = 3$

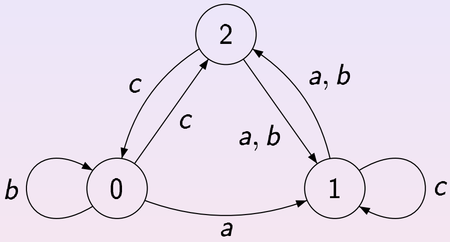
For $n = 3$ we have three sporadic automata:

For $n = 3$ we have three sporadic automata:

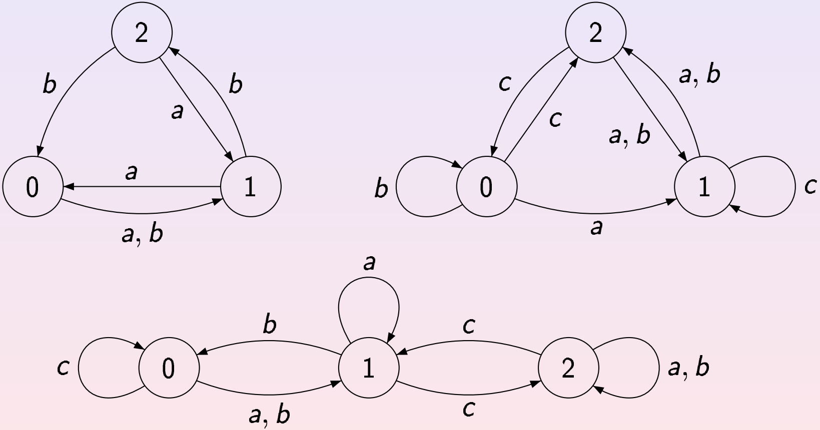


For $n = 3$ we have three sporadic automata:





For $n = 3$ we have three sporadic automata:

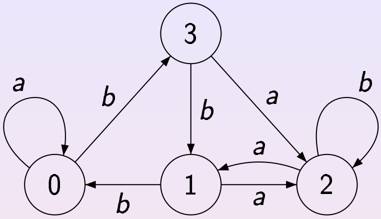


21. Sporadic Examples: $n = 4$

Also for $n = 4$ three sporadic automata are known:

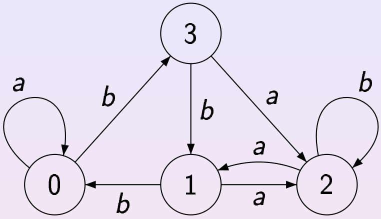
21. Sporadic Examples: $n = 4$

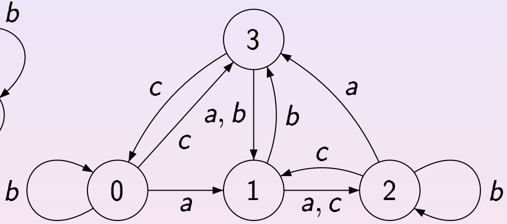
Also for $n = 4$ three sporadic automata are known:



21. Sporadic Examples: $n = 4$

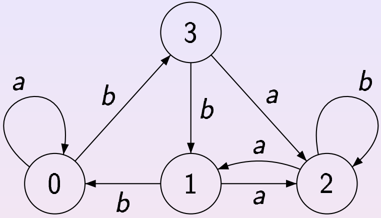
Also for $n = 4$ three sporadic automata are known:

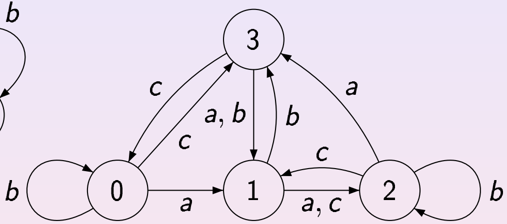


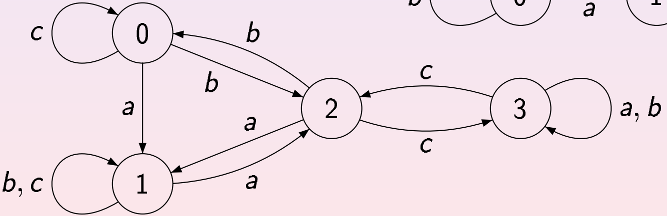


21. Sporadic Examples: $n = 4$

Also for $n = 4$ three sporadic automata are known:





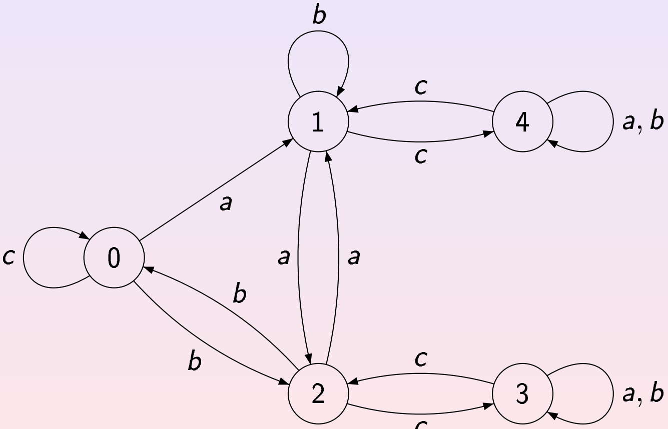


22. Roman's Automaton

A proper 5-state automaton reaching the Černý bound has been recently discovered by Adam Roman.

22. Roman's Automaton

A proper 5-state automaton reaching the Černý bound has been recently discovered by Adam Roman.

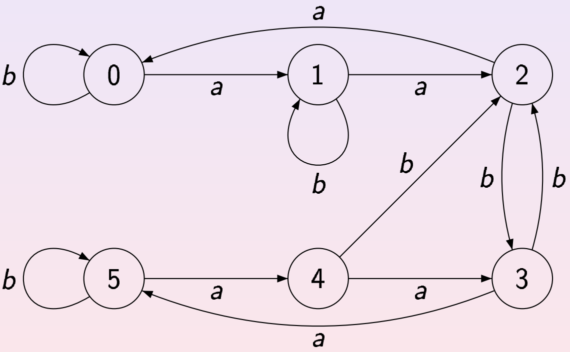


23. Kari's Automaton

The last in our list and the most remarkable example was published in 2001 by Jarkko Kari (A counter example to a conjecture concerning synchronizing words in finite automata, EATCS Bull., 73, 146).

23. Kari's Automaton

The last in our list and the most remarkable example was published in 2001 by Jarkko Kari (A counter example to a conjecture concerning synchronizing words in finite automata, EATCS Bull., 73, 146).



24. Pin's Conjecture

Kari's automaton $\mathscr{K}_6$ has refuted several conjectures.

The most well known of them was suggested by Jean-Éric Pin in 1978. Pin conjectured that if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states admits a word $w \in \Sigma^*$ such that $|Q\,.\,w| = k$, $1 \leq k \leq n$, then $\mathscr{A}$ possesses a word of length at most $(n-k)^2$ with the same property. (The Černý conjecture corresponds to the case $k = 1$.)

However, in $\mathscr{K}_6$ there is no word w of length $16 = (6-2)^2$ such that $|Q\,.\,w| = 2$.

Kari's automaton $\mathscr{K}_6$ has refuted several conjectures.

The most well known of them was suggested by Jean-Éric Pin in 1978. Pin conjectured that if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ with n states admits a word $w \in \Sigma^*$ such that $|Q \,.\, w| = k$, $1 \leq k \leq n$, then $\mathscr{A}$ possesses a word of length at most $(n-k)^2$ with the same property. (The Černý conjecture corresponds to the case $k = 1$.)

However, in $\mathscr{K}_6$ there is no word w of length $16 = (6-2)^2$ such that $|Q \,.\, w| = 2$.

Kari's automaton $\mathscr{K}_6$ has refuted several conjectures.

The most well known of them was suggested by Jean-Éric Pin in 1978. Pin conjectured that if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states admits a word $w \in \Sigma^*$ such that $|Q\,.\,w| = k$, $1 \leq k \leq n$, then $\mathscr{A}$ possesses a word of length at most $(n-k)^2$ with the same property. (The Černý conjecture corresponds to the case $k = 1$.)

However, in $\mathscr{K}_6$ there is no word w of length $16 = (6-2)^2$ such that $|Q\,.\,w| = 2$.

24. Pin's Conjecture

Kari's automaton $\mathscr{K}_6$ has refuted several conjectures.

The most well known of them was suggested by Jean-Éric Pin in 1978. Pin conjectured that if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states admits a word $w \in \Sigma^*$ such that $|Q\,.\,w| = k$, $1 \leq k \leq n$, then $\mathscr{A}$ possesses a word of length at most $(n-k)^2$ with the same property. (The Černý conjecture corresponds to the case $k = 1$.)

However, in $\mathscr{K}_6$ there is no word w of length $16 = (6-2)^2$ such that $|Q\,.\,w| = 2$.

25. Rank Conjecture

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q \,.\, w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$. Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q \,.\, w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q \,.\, w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$.

Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q \,.\, w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

25. Rank Conjecture

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q\,.\,w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$. Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q\,.\,w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

25. Rank Conjecture

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q\,.\,w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$. Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q\,.\,w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q\,.\,w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$. Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q\,.\,w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

25. Rank Conjecture

The *rank* of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is the minimum cardinality of the sets $Q \,.\, w$ where w runs over Σ^*. This is the minimum score that can be achieved in the solitaire game on the automaton $\mathscr{A}$. Synchronizing automata are precisely those of rank 1.

A corrected (and perhaps correct) version of Pin's conjecture is the following rank conjecture: if an automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with n states has rank k, then there exists a word $w \in \Sigma^*$ of length at most $(n-k)^2$ such that $|Q \,.\, w| = k$.

Again, the Černý conjecture corresponds to the case $k = 1$.

Kari's automaton does not refute the rank conjecture!

In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R \,.\, w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

Kari's automaton does not refute the rank conjecture!

In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R\,.\,w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

26. Extensibility Conjecture

Kari's automaton does not refute the rank conjecture!

In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R\,.\,w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

Kari's automaton does not refute the rank conjecture!
In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R \, . \, w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

26. Extensibility Conjecture

Kari's automaton does not refute the rank conjecture!

In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R\,.\,w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

26. Extensibility Conjecture

Kari's automaton does not refute the rank conjecture!

In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

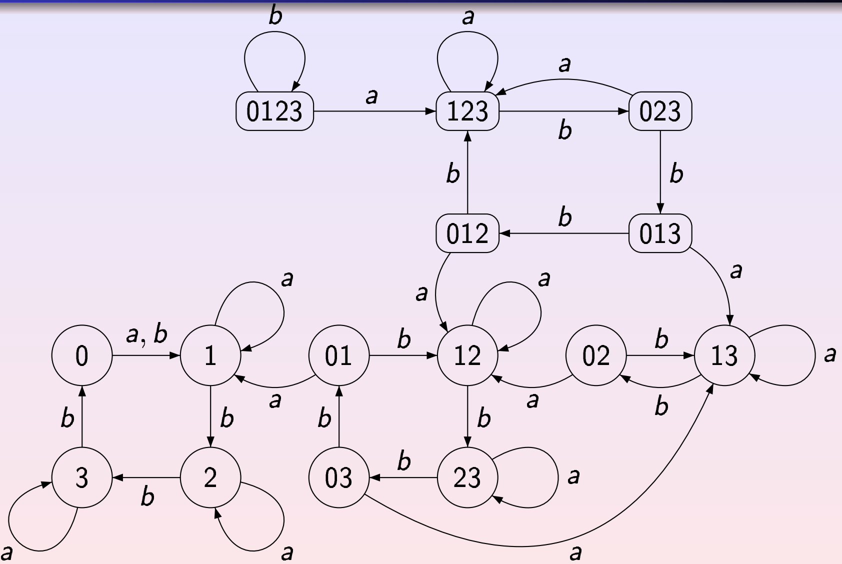
Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R\,.\,w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

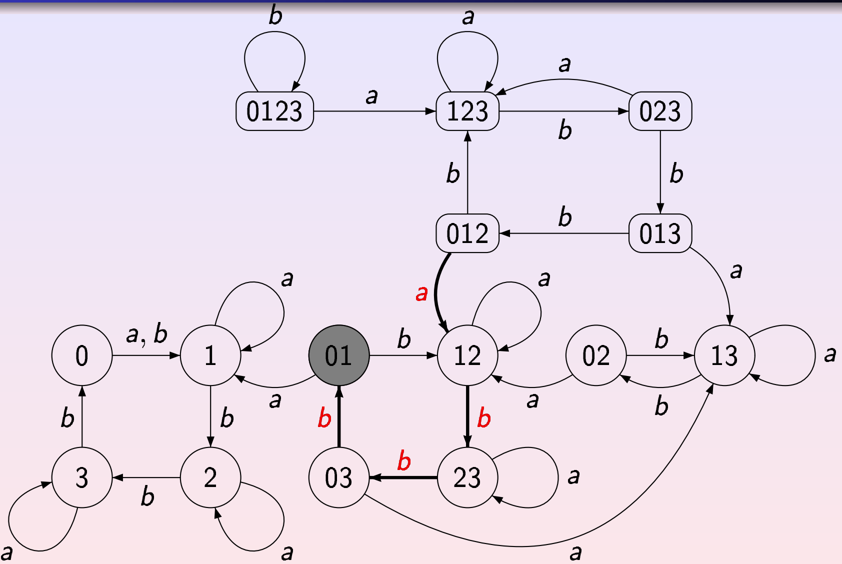
26. Extensibility Conjecture

Kari's automaton does not refute the rank conjecture!
In the solitaire game on $\mathscr{K}_6$, no sequence of 16 moves removes 4 coins. However, 4 is not the maximum number of tokens that can be removed! One can show that 5 states can be freed by a sequence of 25 moves — in full accordance with the rank conjecture.

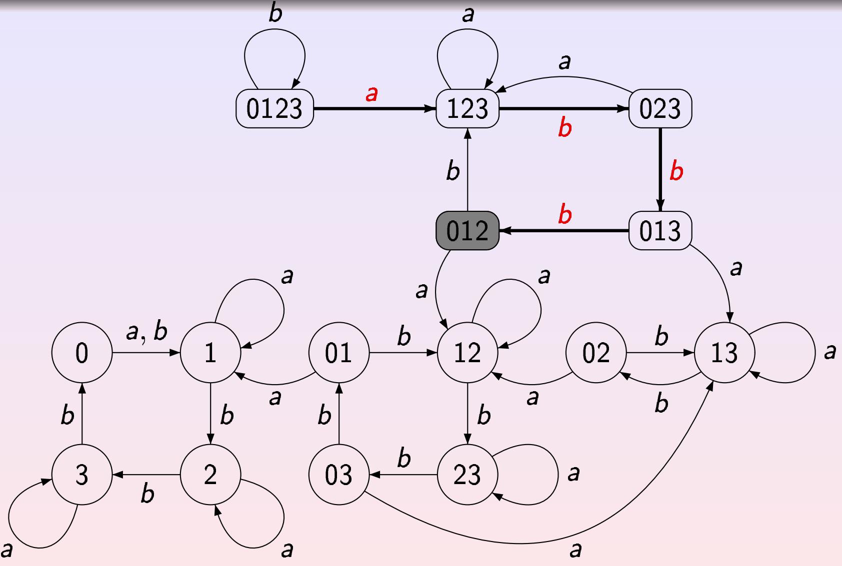
Yet another hope killed by Kari's example is the extensibility conjecture. For $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, a subset $P \subset Q$ is *extensible* if $P = R\,.\,w$ for some $w \in \Sigma^*$ of length at most $n = |Q|$ and some $R \subseteq Q$ with $|R| > |P|$. It was conjectured that in synchronizing automata every proper non-singleton subset is extensible.

27. Example




0123
123
023
012
013
0
1
01
12
02
13
3
2
03
23
a
b
a, b

27. Example



Observe that the extensibility conjecture implies the Černý conjecture.

Indeed, if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is synchronizing, then some letter $a \in \Sigma$ should sent two states $q, q' \in Q$ to the same state p. Let $P_0 = \{q, q'\}$ and, for $i > 0$, let P_i be such that $|P_i| > |P_{i-1}|$ and $P_{i-1} = P_i \,.\, w_i$ for some word w_i of length $\leq n$. Then in at most $n-2$ steps the sequence $P_0, P_1, P_2, \ldots$ reaches Q and

$$Q \,.\, w_{n-1} w_{n-2} \cdots w_1 a = \{p\},$$

that is, $w_{n-1} w_{n-2} \cdots w_1 a$ is a reset word. The length of this reset word is at most $n(n-2)+1 = (n-1)^2$.

28. Extensibility

Observe that the extensibility conjecture implies the Černý conjecture.

Indeed, if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is synchronizing, then some letter $a \in \Sigma$ should sent two states $q, q' \in Q$ to the same state p. Let $P_0 = \{q, q'\}$ and, for $i > 0$, let P_i be such that $|P_i| > |P_{i-1}|$ and $P_{i-1} = P_i \,.\, w_i$ for some word w_i of length $\leq n$. Then in at most $n - 2$ steps the sequence $P_0, P_1, P_2, \ldots$ reaches Q and

$$Q \,.\, w_{n-1} w_{n-2} \cdots w_1 a = \{p\},$$

that is, $w_{n-1} w_{n-2} \cdots w_1 a$ is a reset word. The length of this reset word is at most $n(n-2) + 1 = (n-1)^2$.

28. Extensibility

Observe that the extensibility conjecture implies the Černý conjecture.

Indeed, if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is synchronizing, then some letter $a \in \Sigma$ should sent two states $q, q' \in Q$ to the same state p. Let $P_0 = \{q, q'\}$ and, for $i > 0$, let P_i be such that $|P_i| > |P_{i-1}|$ and $P_{i-1} = P_i \,.\, w_i$ for some word w_i of length $\leq n$. Then in at most $n - 2$ steps the sequence $P_0, P_1, P_2, \ldots$ reaches Q and

$$Q \,.\, w_{n-1} w_{n-2} \cdots w_1 a = \{p\},$$

that is, $w_{n-1} w_{n-2} \cdots w_1 a$ is a reset word. The length of this reset word is at most $n(n-2) + 1 = (n-1)^2$.

28. Extensibility

Observe that the extensibility conjecture implies the Černý conjecture.

Indeed, if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is synchronizing, then some letter $a \in \Sigma$ should sent two states $q, q' \in Q$ to the same state p. Let $P_0 = \{q, q'\}$ and, for $i > 0$, let P_i be such that $|P_i| > |P_{i-1}|$ and $P_{i-1} = P_i \,.\, w_i$ for some word w_i of length $\leq n$. Then in at most $n - 2$ steps the sequence $P_0, P_1, P_2, \dots$ reaches Q and

$$Q \,.\, w_{n-1} w_{n-2} \cdots w_1 a = \{p\},$$

that is, $w_{n-1} w_{n-2} \cdots w_1 a$ is a reset word. The length of this reset word is at most $n(n-2) + 1 = (n-1)^2$.

28. Extensibility

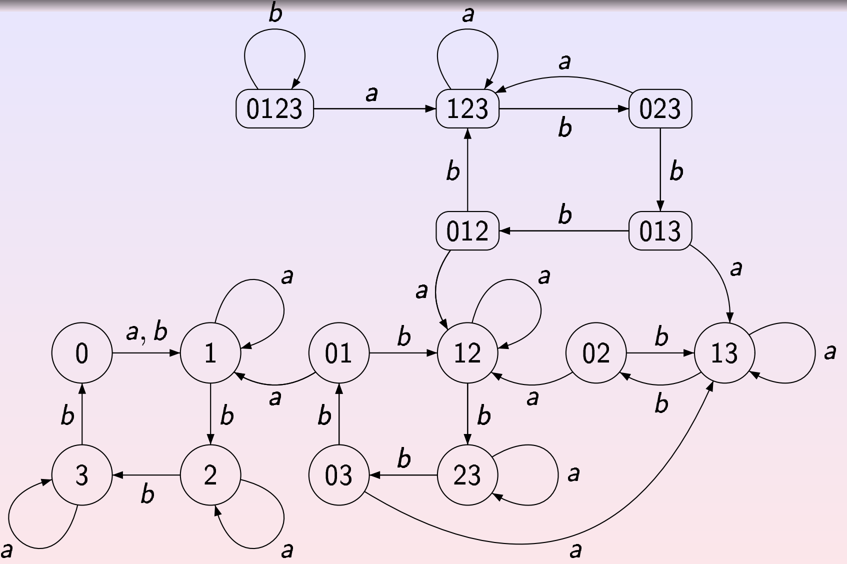
Observe that the extensibility conjecture implies the Černý conjecture.

Indeed, if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is synchronizing, then some letter $a \in \Sigma$ should sent two states $q, q' \in Q$ to the same state p. Let $P_0 = \{q, q'\}$ and, for $i > 0$, let P_i be such that $|P_i| > |P_{i-1}|$ and $P_{i-1} = P_i \,.\, w_i$ for some word w_i of length $\leq n$. Then in at most $n - 2$ steps the sequence $P_0, P_1, P_2, \dots$ reaches Q and

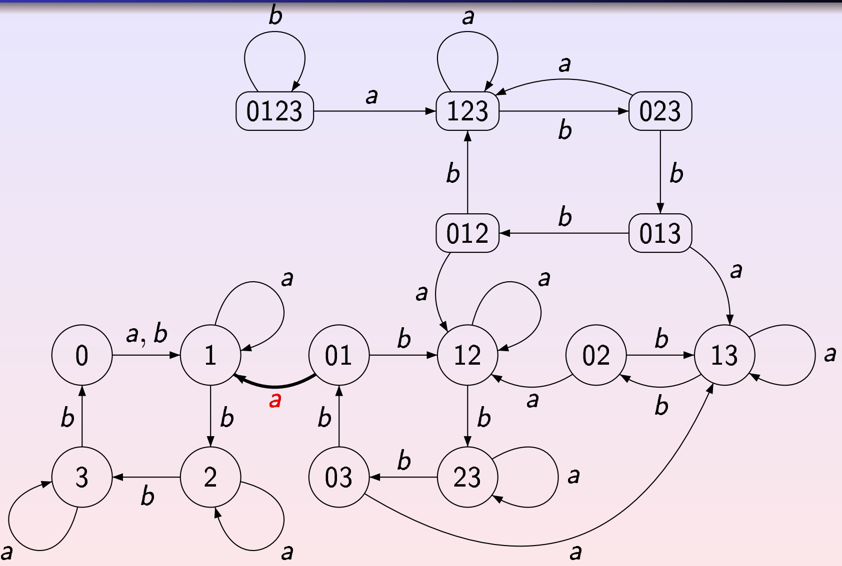
$$Q \,.\, w_{n-1} w_{n-2} \cdots w_1 a = \{p\},$$

that is, $w_{n-1} w_{n-2} \cdots w_1 a$ is a reset word. The length of this reset word is at most $n(n-2) + 1 = (n-1)^2$.

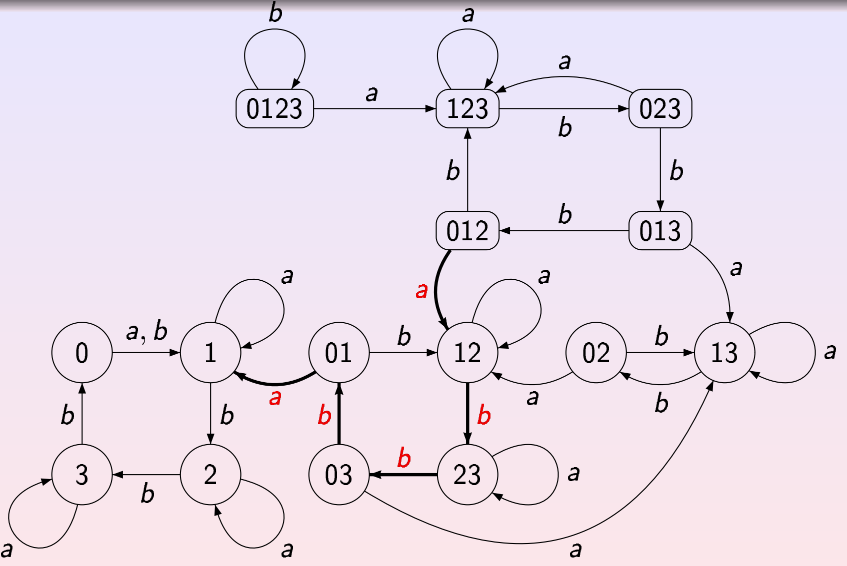
29. Example



29. Example

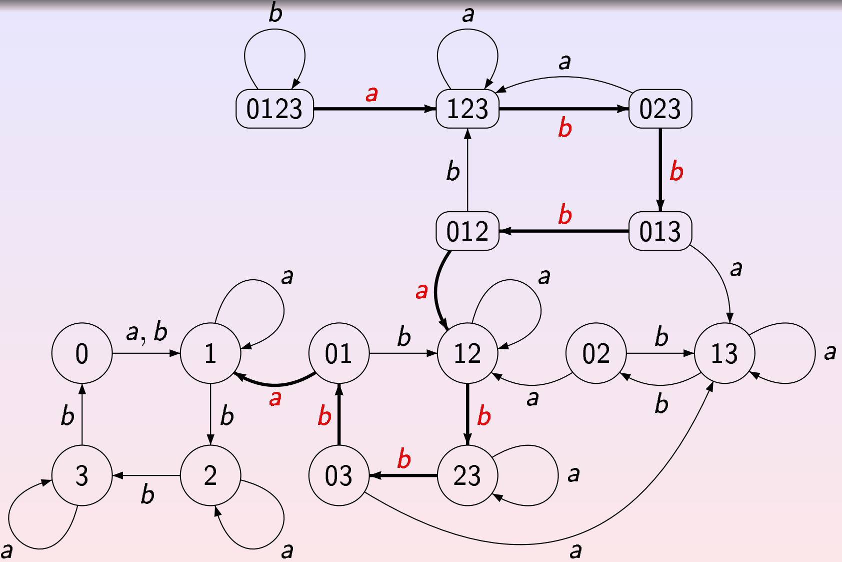


29. Example



CSClub, St Petersburg, November 14, 2010

29. Example



30. Extensibility

Several important results confirming the Černý conjecture for various partial cases have been proved by verifying the extensibility conjecture for the corresponding automata. This includes:

- Louis Dubuc's result for automata in which a letter acts on the state set Q as a cyclic permutation of order $|Q|$ (Sur le automates circulaires et la conjecture de Černý, RAIRO Inform. Theor. Appl., 32 (1998) 21–34 [in French]).
- Jarkko Kari's result for automata with Eulerian digraphs (Synchronizing finite automata on Eulerian digraphs, Theoret. Comput. Sci., 295 (2003) 223–232.)
- Benjamin Steinberg's result for automata in which a letter labels only one cycle (one-cluster automata) and this cycle is of prime length.

30. Extensibility

Several important results confirming the Černý conjecture for various partial cases have been proved by verifying the extensibility conjecture for the corresponding automata. This includes:

• Louis Dubuc's result for automata in which a letter acts on the state set Q as a cyclic permutation of order $|Q|$ (Sur le automates circulaires et la conjecture de Černý, RAIRO Inform. Theor. Appl., 32 (1998) 21–34 [in French]).

• Jarkko Kari's result for automata with Eulerian digraphs (Synchronizing finite automata on Eulerian digraphs, Theoret. Comput. Sci., 295 (2003) 223–232.)

• Benjamin Steinberg's result for automata in which a letter labels only one cycle (one-cluster automata) and this cycle is of prime length.

30. Extensibility

Several important results confirming the Černý conjecture for various partial cases have been proved by verifying the extensibility conjecture for the corresponding automata. This includes:

• Louis Dubuc's result for automata in which a letter acts on the state set Q as a cyclic permutation of order $|Q|$ (Sur le automates circulaires et la conjecture de Černý, RAIRO Inform. Theor. Appl., 32 (1998) 21–34 [in French]).
• Jarkko Kari's result for automata with Eulerian digraphs (Synchronizing finite automata on Eulerian digraphs, Theoret. Comput. Sci., 295 (2003) 223–232.)
• Benjamin Steinberg's result for automata in which a letter labels only one cycle (one-cluster automata) and this cycle is of prime length.

Several important results confirming the Černý conjecture for various partial cases have been proved by verifying the extensibility conjecture for the corresponding automata. This includes:

- Louis Dubuc's result for automata in which a letter acts on the state set Q as a cyclic permutation of order $|Q|$ (Sur le automates circulaires et la conjecture de Černý, RAIRO Inform. Theor. Appl., 32 (1998) 21–34 [in French]).
- Jarkko Kari's result for automata with Eulerian digraphs (Synchronizing finite automata on Eulerian digraphs, Theoret. Comput. Sci., 295 (2003) 223–232.)
- Benjamin Steinberg's result for automata in which a letter labels only one cycle (one-cluster automata) and this cycle is of prime length.

However, in $\mathscr{K}_6$ there exists a 2-subset that cannot be extended to a larger subset by any word of length 6 (and even by any word of length 7).

Thus, the extensibility conjecture fails, and the approach based on it cannot prove the Černý conjecture in general.

However, studying the extensibility phenomenon in synchronizing automata appears to be worthwhile: if there is a linear bound on the minimum length of words extending non-singleton proper subsets of a synchronizing automaton, then there is a quadratic bound on the minimum length of reset words for the automaton.

31. Extensibility vs Kari's Example

However, in $\mathscr{K}_6$ there exists a 2-subset that cannot be extended to a larger subset by any word of length 6 (and even by any word of length 7).

Thus, the extensibility conjecture fails, and the approach based on it cannot prove the Černý conjecture in general.

However, studying the extensibility phenomenon in synchronizing automata appears to be worthwhile: if there is a linear bound on the minimum length of words extending non-singleton proper subsets of a synchronizing automaton, then there is a quadratic bound on the minimum length of reset words for the automaton.

31. Extensibility vs Kari's Example

However, in $\mathscr{K}_6$ there exists a 2-subset that cannot be extended to a larger subset by any word of length 6 (and even by any word of length 7).

Thus, the extensibility conjecture fails, and the approach based on it cannot prove the Černý conjecture in general.

However, studying the extensibility phenomenon in synchronizing automata appears to be worthwhile: if there is a linear bound on the minimum length of words extending non-singleton proper subsets of a synchronizing automaton, then there is a quadratic bound on the minimum length of reset words for the automaton.

32. α-Extensibility

Let α be a positive real number. An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *α-extensible* if for any subset $P \subset Q$ there are $w \in \Sigma^*$ of length at most $\alpha|Q|$ and $R \subseteq Q$ with $|R| > |P|$ such that $P = R\,.\,w$.

An *α-extensible* automaton with n states has a reset word of length $\alpha n^2 + O(n)$.

Several important classes of synchronizing automata are known to be 2-extensible, for instance, one-cluster automata (Marie-Pierre Béal, Mikhail Berlinkov, Dominique Perrin, in print).

On the other hand, for any $\alpha < 2$ Berlinkov (DLT 2010) has constructed a synchronizing automaton that is not α-extensible.

32. α-Extensibility

Let α be a positive real number. An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *α-extensible* if for any subset $P \subset Q$ there are $w \in \Sigma^*$ of length at most $\alpha|Q|$ and $R \subseteq Q$ with $|R| > |P|$ such that $P = R\,.\,w$.

An *α-extensible* automaton with n states has a reset word of length $\alpha n^2 + O(n)$.

Several important classes of synchronizing automata are known to be 2-extensible, for instance, one-cluster automata (Marie-Pierre Béal, Mikhail Berlinkov, Dominique Perrin, in print).

On the other hand, for any $\alpha < 2$ Berlinkov (DLT 2010) has constructed a synchronizing automaton that is not α-extensible.

32. α-Extensibility

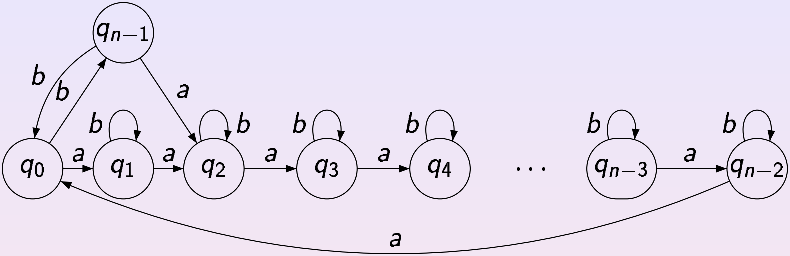
Let α be a positive real number. An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is *α-extensible* if for any subset $P \subset Q$ there are $w \in \Sigma^*$ of length at most $\alpha|Q|$ and $R \subseteq Q$ with $|R| > |P|$ such that $P = R\,.\,w$.

An *α-extensible* automaton with n states has a reset word of length $\alpha n^2 + O(n)$.

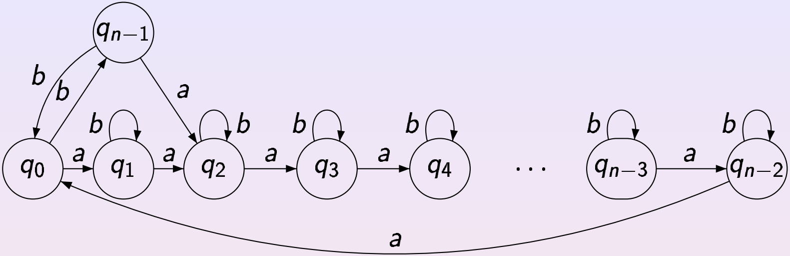
Several important classes of synchronizing automata are known to be 2-extensible, for instance, one-cluster automata (Marie-Pierre Béal, Mikhail Berlinkov, Dominique Perrin, in print).

On the other hand, for any $\alpha < 2$ Berlinkov (DLT 2010) has constructed a synchronizing automaton that is not α-extensible.

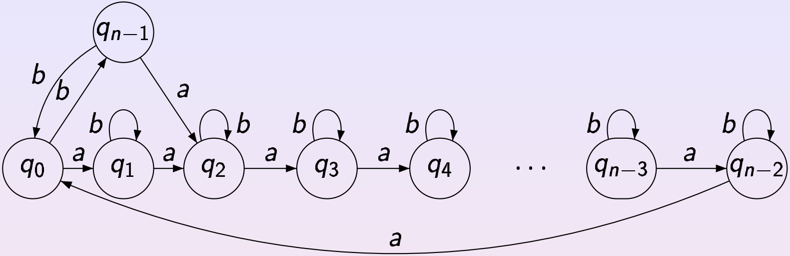
33. Berlinkov's Series



For $n > \frac{3}{2-\alpha}$, this automaton is not α-extensible.



For $n > \frac{3}{2-\alpha}$, this automaton is not α-extensible.



For $n > \frac{3}{2-\alpha}$, this automaton is not α-extensible.

Open problems: to investigate the worst-case/average-case behaviour of the greedy extension algorithm.