Synchronizing Finite Automata

Lecture IV: The Černý Conjecture

Mikhail Volkov

Ural Federal University / Hunter College

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

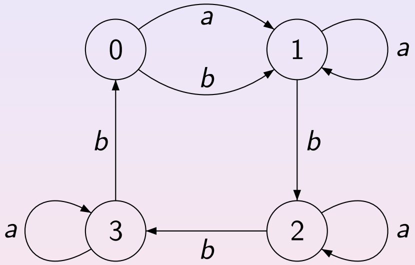
- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ 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 $Q \,.\, v = \{\delta(q, v) \mid q \in Q\}$.

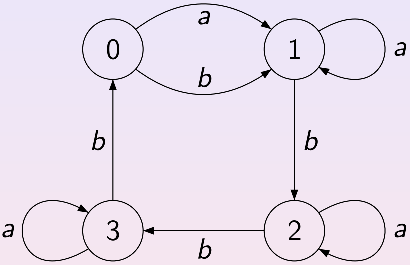
Any w with this property is a reset word for $\mathscr{A}$.

2. Example



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

2. Example



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 its **reset threshold**, i.e., the minimum length of its reset words?

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 its reset threshold, i.e., the minimum length of its reset words? 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 $\mathscr{C}_4$.

Suppose a synchronizing automaton has n states. What is its reset threshold, i.e., the minimum length of its reset words? 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 $\mathscr{C}_4$.

Suppose a synchronizing automaton has n states. What is its reset threshold, i.e., the minimum length of its reset words? 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

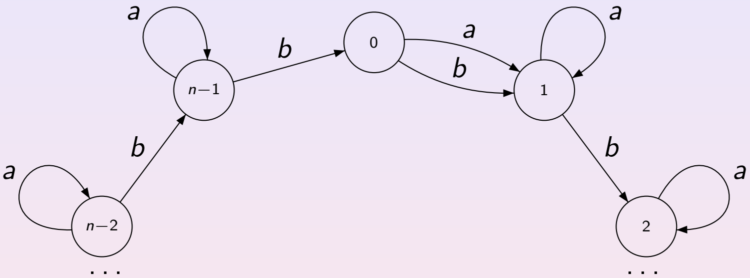
Suppose a synchronizing automaton has n states. What is its reset threshold, i.e., the minimum length of its reset words? We know an upper bound: there always exists a reset word of length $\frac{n^3-n}{6}$. What about a lower bound?
In his 1964 paper Jan Černý constructed a series $\mathscr{C}_n$, $n = 2, 3, \dots$, 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 $\mathscr{C}_4$.

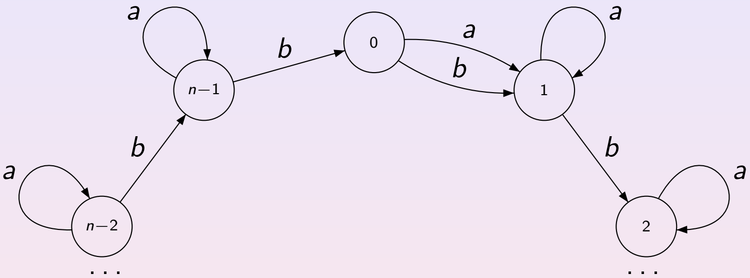
4. The Černý Series

Here is a generic automaton from the Černý series:



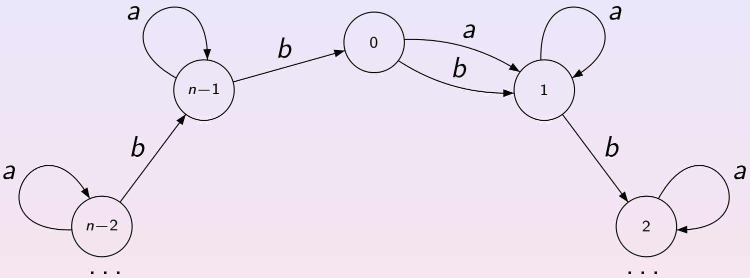
Černý has proved that the shortest reset word for $\mathscr{C}_n$ is $(ab^{n-1})^{n-2}a$ of length $(n-1)^2$. As other results from Černý's paper of 1964, this nice series of automata has been rediscovered many times.

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.

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.

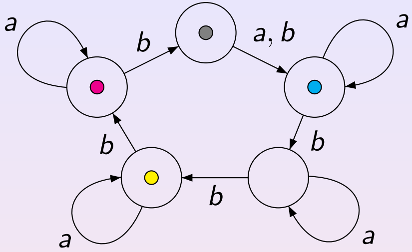
- The digraph of $\mathscr{C}_n$ — the game-board.
- The initial position — each state holds a coin, all coins are pairwise distinct.
- Each letter $c \in \{a, b\}$ defines a move — coins slide along the arrows labelled c and, whenever two coins meet at the state 1, the coin arriving from 0 is removed.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

5. Game

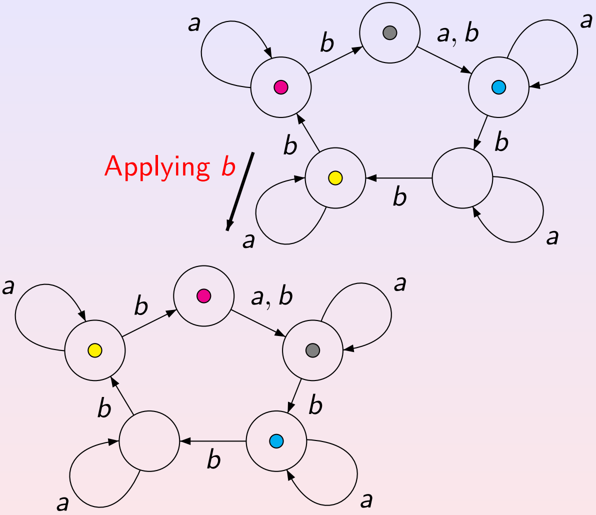
We present a proof of this result using a solitaire-like game.

- 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.

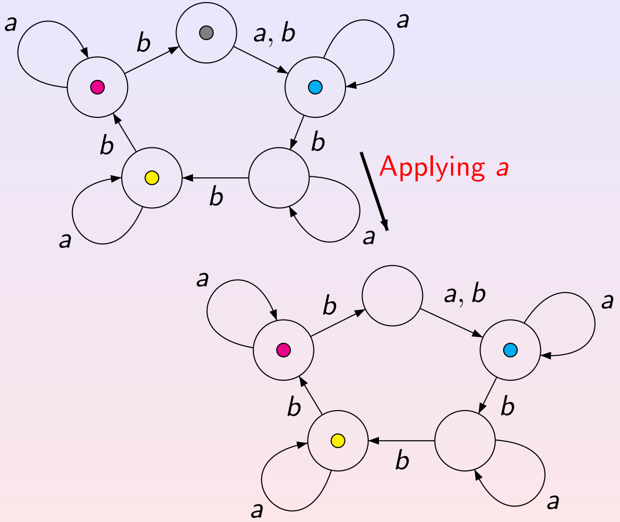
6. Example



6. Example



6. Example



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, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

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

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. 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 distance from $s_i(C)$ to the state 0 and $d_i(C)$ is the distance from $s_i(C)$ to the state holding the golden coin (recall that the latter is present in all positions.) Distances are measured on the 'main circle' of our automaton in the direction of arrows.

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. 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 distance from $s_i(C)$ to the state 0 and $d_i(C)$ is the distance from $s_i(C)$ to the state holding the golden coin (recall that the latter is present in all positions.) Distances are measured on the 'main circle' of our automaton in the direction of arrows.

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. 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 distance from $s_i(C)$ to the state 0 and $d_i(C)$ is the distance from $s_i(C)$ to the state holding the golden coin (recall that the latter is present in all positions.) Distances are measured on the 'main circle' of our automaton in the direction of arrows.

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

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. 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 distance from $s_i(C)$ to the state 0 and $d_i(C)$ is the distance from $s_i(C)$ to the state holding the golden coin (recall that the latter is present in all positions.) Distances are measured on the 'main circle' of our automaton in the direction of arrows.

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.

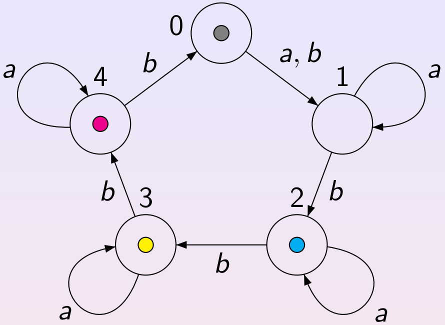
If a coin C is present in a position P_i, let $s_i(C)$ be the state covered with C in this position. Define the weight of C in the position P_i as

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

where $m_i(C)$ is the distance from $s_i(C)$ to the state 0 and $d_i(C)$ is the distance from $s_i(C)$ to the state holding the golden coin (recall that the latter is present in all positions.) Distances are measured on the 'main circle' of our automaton in the direction of arrows.

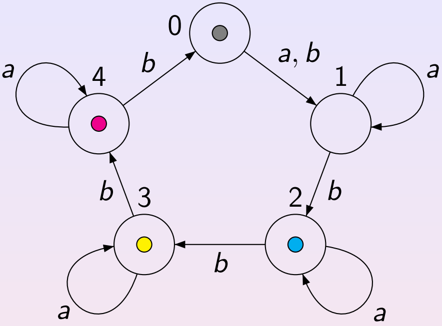
The weight of P_i is the maximum weight of the coins present in this position.

9. Example



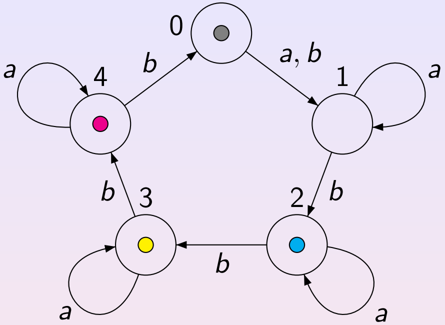
Assume the yellow coin is the golden one. Then its weight is 2.
The weight of the blue coin is $5 \cdot 1 + 3 = 8$.
The weight of the gray coin is $5 \cdot 3 + 0 = 15$.
The weight of the red coin is $5 \cdot 4 + 1 = 21$,
and this is the weight of the position.

9. Example



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

9. Example



Assume the yellow coin is the golden one. Then its weight is 2.

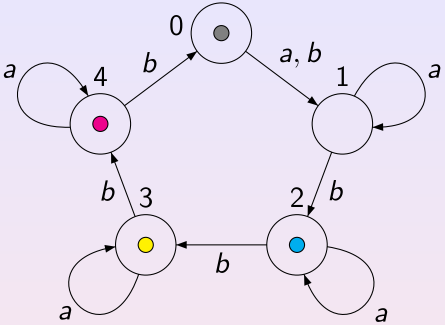
The weight of the blue coin is $5 \cdot 1 + 3 = 8$.

The weight of the gray coin is $5 \cdot 3 + 0 = 15$.

The weight of the red coin is $5 \cdot 4 + 1 = 21$,

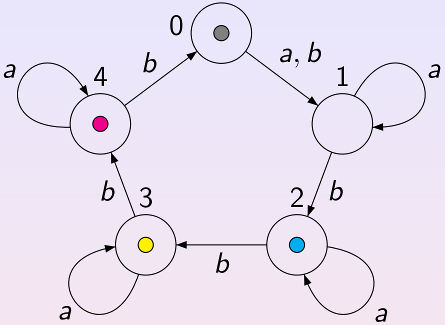
and this is the weight of the position.

9. Example



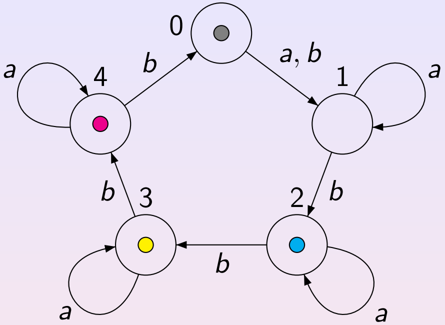
Assume the yellow coin is the golden one. Then its weight is 2.
The weight of the blue coin is $5 \cdot 1 + 3 = 8$.
The weight of the gray coin is $5 \cdot 3 + 0 = 15$.
The weight of the red coin is $5 \cdot 4 + 1 = 21$,
and this is the weight of the position.

9. Example



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

9. Example



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

10. Properties of the Weight Function. I

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 holding 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 $\mathrm{wg}(P_0) \geq n(n-1)$.

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 holding 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. I

We have to check that our weight function satisfies the conditions

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

(ii) $1 \geq \text{wg}(P_{i-1}) - \text{wg}(P_i)$ 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 holding the golden coin. Then $d_0(C) = n - 1$ whence

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

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

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 holding 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 $\mathrm{wg}(P_0) \geq n(n-1)$.

11. Properties of the Weight Function. II

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.

In particular, except for the final position, the golden coin can never be the coin of maximum weight: for any coin $C \neq G$, we have $d_i(C) \geq 1$ whence
$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq n > n - 1 \geq \mathrm{wg}(G, P_i)$.

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. II

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.
In particular, except for the final position, the golden coin can never be the coin of maximum weight: for any coin $C \neq G$, we have $d_i(C) \geq 1$ whence
$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq n > n - 1 \geq \mathrm{wg}(G, P_i)$.

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. II

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.
In particular, except for the final position, the golden coin can never be the coin of maximum weight: for any coin $C \neq G$, we have $d_i(C) \geq 1$ whence
$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq n > n - 1 \geq \mathrm{wg}(G, P_i)$.

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. II

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.
In particular, except for the final position, the golden coin can never be the coin of maximum weight: for any coin $C \neq G$, we have $d_i(C) \geq 1$ whence
$\mathrm{wg}(C, P_i) = n \cdot d_i(C) + m_i(C) \geq n > n - 1 \geq \mathrm{wg}(G, P_i)$.

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. III

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. III

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. III

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. III

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. The Černý Function

Define the Černý function $C(n)$ as the maximum reset threshold of all 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 just one line:

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

13. The Černý Function

Define the Černý function $C(n)$ as the maximum reset threshold of all 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 just one line:

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

13. The Černý Function

Define the Černý function $C(n)$ as the maximum reset threshold of all 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 just one line:

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

Define the Černý function $C(n)$ as the maximum reset threshold of all 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 just one line:

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

14. 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 reset threshold $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \dots$, with a few sporadic examples for $n \leq 6$.

14. 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 reset threshold $(n-1)^2$ 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 reset threshold $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots$, with a few sporadic examples for $n \leq 6$.

14. 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 reset threshold $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \dots,$ with a few sporadic examples for $n \leq 6$.

14. 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 reset threshold $(n-1)^2$ is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots,$ with a few sporadic examples for $n \leq 6$.

14. Why it is hard?

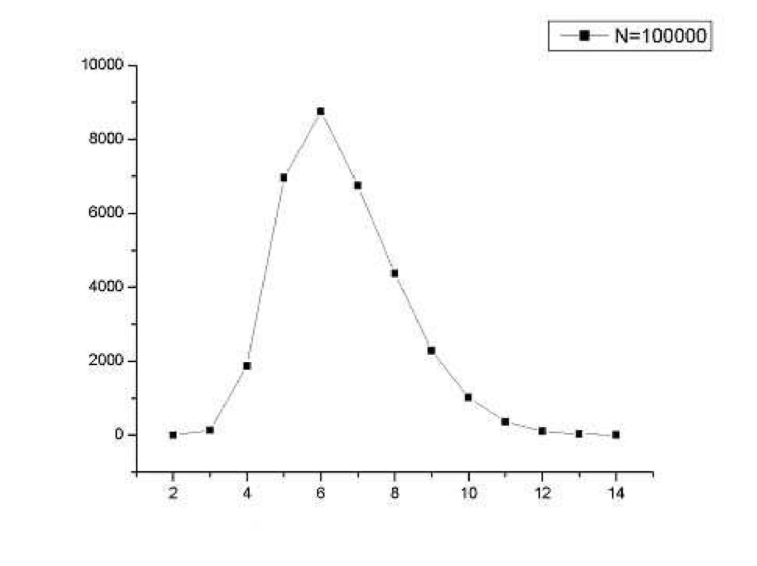
Why is the problem so surprisingly difficult?

We saw two reasons:

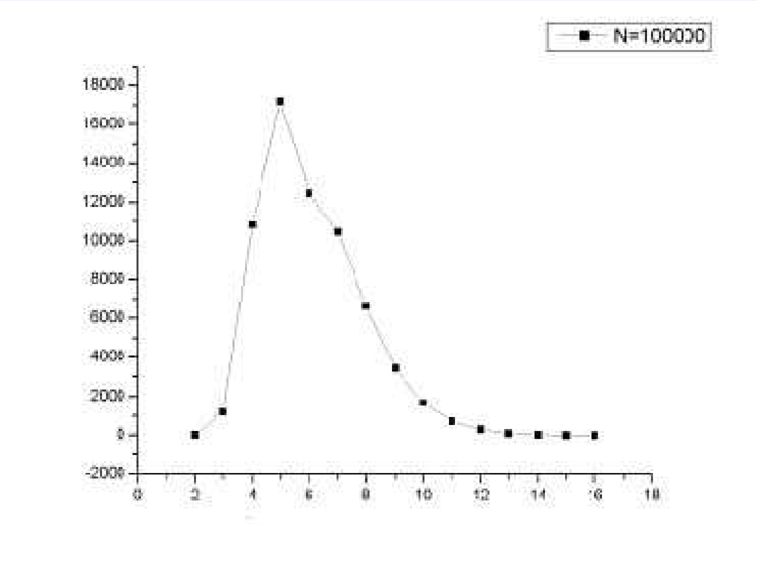
- non-locality: prefixes of optimal solutions need not be optimal (that is why the greedy algorithm fails);
- combinatorics of finite sets is encoded in the problem.

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

15. 20-State Experiment



16. 30-State Experiment



17. Random Automata

Recent massive experiments (see Andrzej Kisielewicz, Jakub Kowalski, and Marek Szykuła, Computing the shortest reset words of synchronizing automata, J. Comb. Optim., 29 (2015) 88–124) involved random DFAs with up to 350 states and up to 10 letters.

Almost all random DFAs are synchronizing and the mean value of reset thresholds for random n-state automata with 2 input letters turns out to be close to $2.5\sqrt{n-5}$.

Known theoretical results about random automata are still much weaker, but it has been proved (Mikhail Berlinkov and Marek Szykuła, Algebraic synchronization criterion and computing reset words, MFCS 2015, LNCS 9234 (2015) 103–115) that reset threshold of a random n-state automaton with 2 input letters is at most $n^{3/2+o(1)}$.

This implies that the Černý conjecture holds true almost surely.

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

17. Random Automata

Recent massive experiments (see Andrzej Kisielewicz, Jakub Kowalski, and Marek Szykuła, Computing the shortest reset words of synchronizing automata, J. Comb. Optim., 29 (2015) 88–124) involved random DFAs with up to 350 states and up to 10 letters. Almost all random DFAs are synchronizing and the mean value of reset thresholds for random n-state automata with 2 input letters turns out to be close to $2.5\sqrt{n-5}$.

Known theoretical results about random automata are still much weaker, but it has been proved (Mikhail Berlinkov and Marek Szykuła, Algebraic synchronization criterion and computing reset words, MFCS 2015, LNCS 9234 (2015) 103–115) that reset threshold of a random n-state automaton with 2 input letters is at most $n^{3/2+o(1)}$.

This implies that the Černý conjecture holds true almost surely.

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

17. Random Automata

Recent massive experiments (see Andrzej Kisielewicz, Jakub Kowalski, and Marek Szykuła, Computing the shortest reset words of synchronizing automata, J. Comb. Optim., 29 (2015) 88–124) involved random DFAs with up to 350 states and up to 10 letters. Almost all random DFAs are synchronizing and the mean value of reset thresholds for random n-state automata with 2 input letters turns out to be close to $2.5\sqrt{n-5}$.

Known theoretical results about random automata are still much weaker, but it has been proved (Mikhail Berlinkov and Marek Szykuła, Algebraic synchronization criterion and computing reset words, MFCS 2015, LNCS 9234 (2015) 103–115) that reset threshold of a random n-state automaton with 2 input letters is at most $n^{3/2+o(1)}$.

This implies that the Černý conjecture holds true almost surely.

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

17. Random Automata

Recent massive experiments (see Andrzej Kisielewicz, Jakub Kowalski, and Marek Szykuła, Computing the shortest reset words of synchronizing automata, J. Comb. Optim., 29 (2015) 88–124) involved random DFAs with up to 350 states and up to 10 letters.

Almost all random DFAs are synchronizing and the mean value of reset thresholds for random n-state automata with 2 input letters turns out to be close to $2.5\sqrt{n-5}$.

Known theoretical results about random automata are still much weaker, but it has been proved (Mikhail Berlinkov and Marek Szykuła, Algebraic synchronization criterion and computing reset words, MFCS 2015, LNCS 9234 (2015) 103–115) that reset threshold of a random n-state automaton with 2 input letters is at most $n^{3/2+o(1)}$.

This implies that the Černý conjecture holds true almost surely.

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

17. Random Automata

Recent massive experiments (see Andrzej Kisielewicz, Jakub Kowalski, and Marek Szykuła, Computing the shortest reset words of synchronizing automata, J. Comb. Optim., 29 (2015) 88–124) involved random DFAs with up to 350 states and up to 10 letters. Almost all random DFAs are synchronizing and the mean value of reset thresholds for random n-state automata with 2 input letters turns out to be close to $2.5\sqrt{n-5}$.

Known theoretical results about random automata are still much weaker, but it has been proved (Mikhail Berlinkov and Marek Szykuła, Algebraic synchronization criterion and computing reset words, MFCS 2015, LNCS 9234 (2015) 103–115) that reset threshold of a random n-state automaton with 2 input letters is at most $n^{3/2+o(1)}$.

This implies that the Černý conjecture holds true almost surely.

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

18. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is proper if none of the DFAs 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 $n = 2$:

18. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is proper if none of the DFAs 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 $n = 2$:

18. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is proper if none of the DFAs 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 $n = 2$:

18. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is proper if none of the DFAs 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.

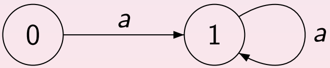
For the sake of completeness, we start with $n = 2$:

18. Sporadic Examples: $n = 2$

A synchronizing automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is proper if none of the DFAs 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 $n = 2$:

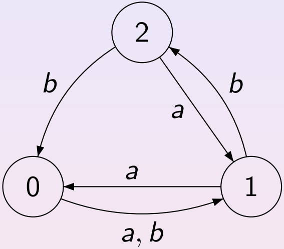


19. Sporadic Examples: $n = 3$

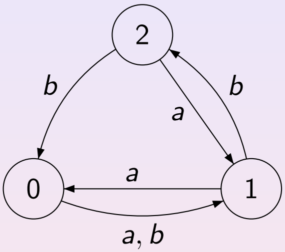
For $n = 3$ we have three sporadic automata:

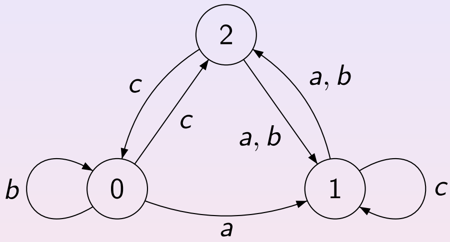
19. Sporadic Examples: $n = 3$

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



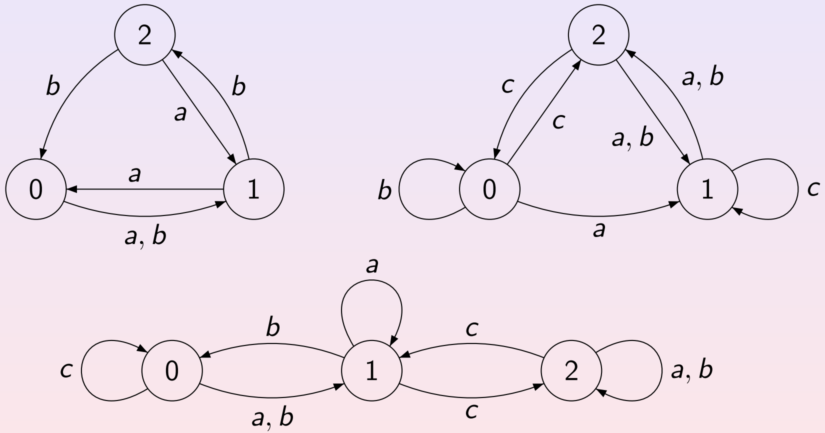
For $n = 3$ we have three sporadic automata:





19. Sporadic Examples: $n = 3$

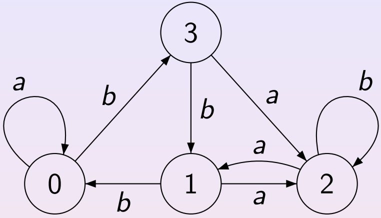
For $n = 3$ we have three sporadic automata:



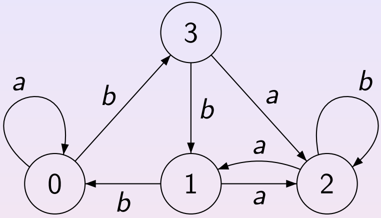
20. Sporadic Examples: $n = 4$

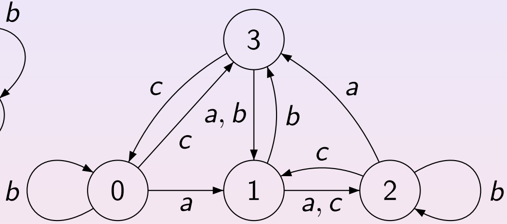
Also for $n = 4$ three sporadic automata are known:

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

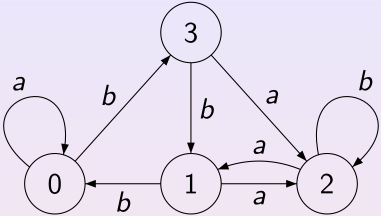


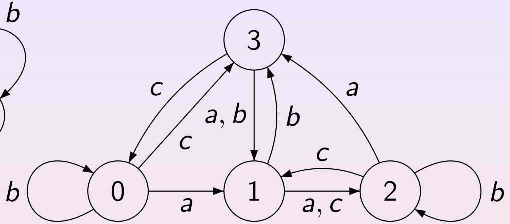
Also for $n = 4$ three sporadic automata are known:

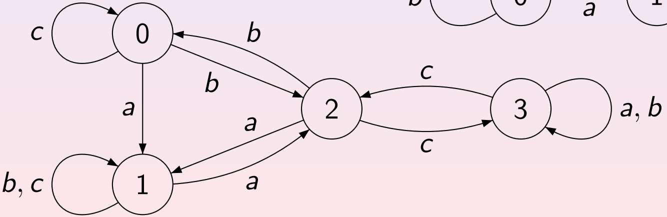




Also for $n = 4$ three sporadic automata are known:





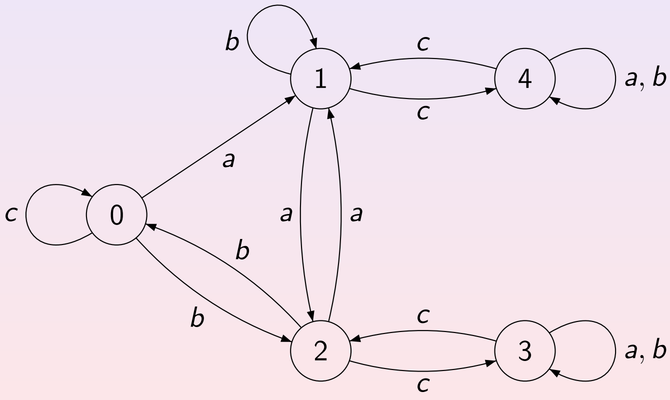


21. Roman's Automaton

A proper 5-state automaton reaching the Černý bound has been discovered by Adam Roman (A note on Černý conjecture for automata over 3-letter alphabet, J. Automata, Languages and Combinatorics, 13 (2008) 141–143).

21. Roman's Automaton

A proper 5-state automaton reaching the Černý bound has been discovered by Adam Roman (A note on Černý conjecture for automata over 3-letter alphabet, J. Automata, Languages and Combinatorics, 13 (2008) 141–143).

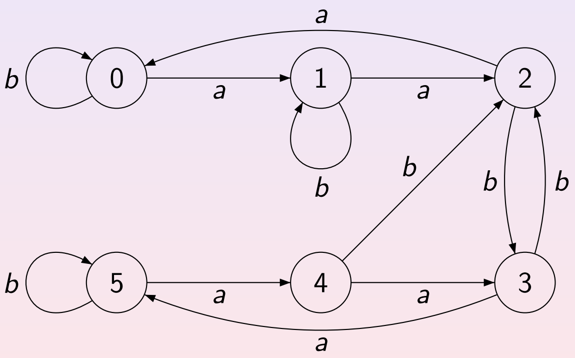


22. Kari's Automaton

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

22. Kari's Automaton

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



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 $|Q \,.\, w| = 2$.

Recent exhaustive search experiments (Andrzej Kisielewicz and Marek Szykuła, Synchronizing automata with large reset lengths, arXiv:1404.3311) indicated that likely $\mathscr{K}_6$ is the only 'proper' counter example to Pin's conjecture.

23. 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 $|Q \,.\, w| = 2$.

Recent exhaustive search experiments (Andrzej Kisielewicz and Marek Szykuła, Synchronizing automata with large reset lengths, arXiv:1404.3311) indicated that likely $\mathscr{K}_6$ is the only 'proper' counter example to Pin's conjecture.

23. 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 $|Q \,.\, w| = 2$.

Recent exhaustive search experiments (Andrzej Kisielewicz and Marek Szykuła, Synchronizing automata with large reset lengths, arXiv:1404.3311) indicated that likely $\mathscr{K}_6$ is the only 'proper' counter example to Pin's conjecture.

23. 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 $|Q \,.\, w| = 2$.

Recent exhaustive search experiments (Andrzej Kisielewicz and Marek Szykuła, Synchronizing automata with large reset lengths, arXiv:1404.3311) indicated that likely $\mathscr{K}_6$ is the only 'proper' counter example to Pin's conjecture.

23. 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 $|Q\,.\,w| = 2$.

Recent exhaustive search experiments (Andrzej Kisielewicz and Marek Szykuła, Synchronizing automata with large reset lengths, arXiv:1404.3311) indicated that likely $\mathscr{K}_6$ is the only 'proper' counter example to Pin's 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 $k = 1$.

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 $k = 1$.

24. Rank Conjecture. I

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 $k = 1$.

24. Rank Conjecture. I

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 $k = 1$.

24. Rank Conjecture. I

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.

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.

25. Rank Conjecture. II

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.

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.