Synchronizing Finite Automata
Lecture VI. Automata with Zero

Mikhail Volkov

Ural Federal University

Spring of 2021

1. Recap

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

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

2. Algebraic Perspective

One can treat DFAs as unary algebras: each letter of the input alphabet defines a unary operation on the state set.

One can treat DFAs as unary algebras: each letter of the input alphabet defines a unary operation on the state set.

This allows us to apply to automata all standard algebraic notions, e.g., the notions of a subalgebra (subautomaton), a congruence, a quotient automaton.

One can treat DFAs as unary algebras: each letter of the input alphabet defines a unary operation on the state set.

This allows us to apply to automata all standard algebraic notions, e.g., the notions of a subalgebra (subautomaton), a congruence, a quotient automaton.

Subautomata: if $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is a DFA, and $S \subseteq Q$ is such that $\delta(s, a) \in S$ for all $s \in S$ and $a \in \Sigma$, consider the DFA $\mathscr{S} := \langle S, \Sigma, \tau\rangle$ where $\tau := \delta|_{S\times\Sigma}$.

One can treat DFAs as unary algebras: each letter of the input alphabet defines a unary operation on the state set.

This allows us to apply to automata all standard algebraic notions, e.g., the notions of a subalgebra (subautomaton), a congruence, a quotient automaton.

Subautomata: if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA, and $S \subseteq Q$ is such that $\delta(s, a) \in S$ for all $s \in S$ and $a \in \Sigma$, consider the DFA $\mathscr{S} := \langle S, \Sigma, \tau \rangle$ where $\tau := \delta|_{S \times \Sigma}$. The latter equality means that $\tau(s, a) := \delta(s, a)$ for all $s \in S$ and $a \in \Sigma$.

One can treat DFAs as unary algebras: each letter of the input alphabet defines a unary operation on the state set.

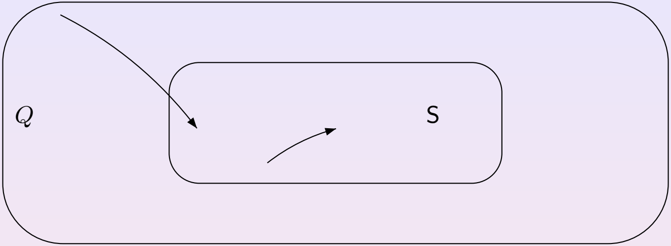
This allows us to apply to automata all standard algebraic notions, e.g., the notions of a subalgebra (subautomaton), a congruence, a quotient automaton.

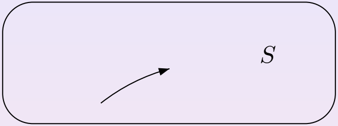
Subautomata: if $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA, and $S \subseteq Q$ is such that $\delta(s, a) \in S$ for all $s \in S$ and $a \in \Sigma$, consider the DFA $\mathscr{S} := \langle S, \Sigma, \tau \rangle$ where $\tau := \delta|_{S \times \Sigma}$. The latter equality means that $\tau(s, a) := \delta(s, a)$ for all $s \in S$ and $a \in \Sigma$.

Any such DFA is said to be a subautomaton of $\mathscr{A}$.

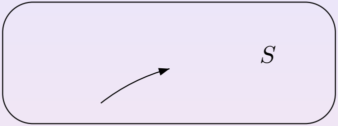
3. Subautomata

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





$$\mathscr{S} = \langle S, \Sigma, \tau \rangle$$



$$\mathscr{S} = \langle S, \Sigma, \tau \rangle$$

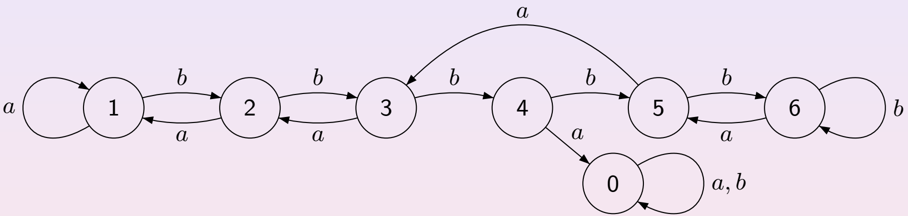
Exercise: show that a DFA has no proper subautomata iff it is strongly connected.

A singleton subautomaton is normally called a sink state or just a sink. At a sink state each letter must have a loop.

A singleton subautomaton is normally called a sink state or just a sink. At a sink state each letter must have a loop.
We study synchronizing automata and, clearly, a synchronizing automaton may have at most one sink.

A singleton subautomaton is normally called a sink state or just a sink. At a sink state each letter must have a loop.
We study synchronizing automata and, clearly, a synchronizing automaton may have at most one sink.
If a DFA has a unique sink state, this state is called a zero state or just a zero.

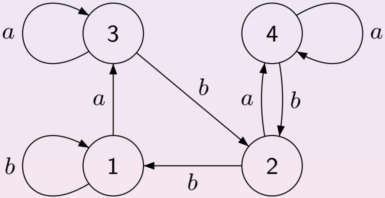
A singleton subautomaton is normally called a sink state or just a sink. At a sink state each letter must have a loop.
We study synchronizing automata and, clearly, a synchronizing automaton may have at most one sink.
If a DFA has a unique sink state, this state is called a zero state or just a zero.



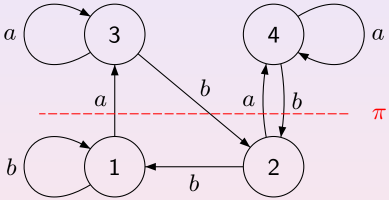
An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is called a congruence if $(p, q) \in \pi$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$.

An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is called a congruence if $(p, q) \in \pi$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$. For π being a congruence, $[q]_\pi$ is the π-class containing the state q.

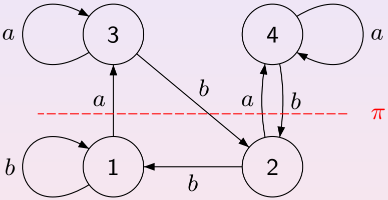
An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is called a congruence if $(p, q) \in \pi$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$. For π being a congruence, $[q]_\pi$ is the π-class containing the state q.



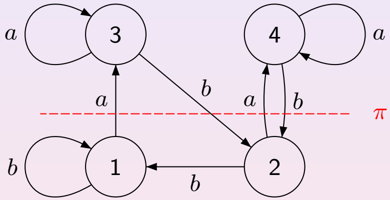
An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is called a congruence if $(p, q) \in \pi$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$. For π being a congruence, $[q]_\pi$ is the π-class containing the state q.

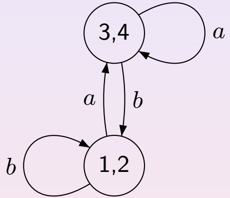


An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is called a congruence if $(p, q) \in \pi$ implies $\bigl(\delta(p, a), \delta(q, a)\bigr) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$. For π being a congruence, $[q]_\pi$ is the π-class containing the state q. The *quotient* $\mathscr{A}/\pi$ is the DFA $\langle Q/\pi, \Sigma, \delta_\pi \rangle$ where $Q/\pi := \{[q]_\pi \mid q \in Q\}$ and the function δ_π is defined by the rule $\delta_\pi([q]_\pi, a) := [\delta(q, a)]_\pi$.



An equivalence π on the state set Q of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is called a congruence if $(p, q) \in \pi$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \pi$ for all $p, q \in Q$ and all $a \in \Sigma$. For π being a congruence, $[q]_\pi$ is the π-class containing the state q. The *quotient* $\mathscr{A}/\pi$ is the DFA $\langle Q/\pi, \Sigma, \delta_\pi\rangle$ where $Q/\pi := \{[q]_\pi \mid q \in Q\}$ and the function δ_π is defined by the rule $\delta_\pi([q]_\pi, a) := [\delta(q, a)]_\pi$.





Suppose that $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA and $\mathscr{S} = \langle S, \Sigma, \tau \rangle$ is a subautomaton of $\mathscr{A}$.

Suppose that $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA and $\mathscr{S} = \langle S, \Sigma, \tau \rangle$ is a subautomaton of $\mathscr{A}$.

The partition of Q into classes one of which is S and all others are singletons is a congruence of $\mathscr{A}$.

Suppose that $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA and $\mathscr{S} = \langle S, \Sigma, \tau \rangle$ is a subautomaton of $\mathscr{A}$.

The partition of Q into classes one of which is S and all others are singletons is a congruence of $\mathscr{A}$.

It is called the Rees congruence corresponding to $\mathscr{S}$ and is denoted by $\rho_{\mathscr{S}}$.

Suppose that $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a DFA and $\mathscr{S} = \langle S, \Sigma, \tau \rangle$ is a subautomaton of $\mathscr{A}$.

The partition of Q into classes one of which is S and all others are singletons is a congruence of $\mathscr{A}$.

It is called the Rees congruence corresponding to $\mathscr{S}$ and is denoted by $\rho_{\mathscr{S}}$.

Clearly, in the quotient automaton $\mathscr{A}/\rho_{\mathscr{S}}$ the state S is a sink.

1. Any subautomaton of a synchronizing automaton is synchronizing, and every reset word for an automaton also serves as a reset word for any of its subautomata.

7. Useful Observations

1. Any subautomaton of a synchronizing automaton is synchronizing, and every reset word for an automaton also serves as a reset word for any of its subautomata.

2. Any quotient of a synchronizing automaton is synchronizing, and every reset word for an automaton also serves as a reset word for any of its quotients.

Let $\mathbf{C}$ be any class of automata closed under taking subautomata and quotients, and let $\mathbf{C}_n$ stand for the class of all automata with n states in $\mathbf{C}$.

8. A Reduction

Let $\mathbf{C}$ be any class of automata closed under taking subautomata and quotients, and let $\mathbf{C}_n$ stand for the class of all automata with n states in $\mathbf{C}$. Consider any function $f : \mathbb{Z}^+ \to \mathbb{N}$ such that

$$f(n) \geq f(n-m+1) + f(m) \text{ whenever } \ n \geq m \geq 1.$$

Let $\mathbf{C}$ be any class of automata closed under taking subautomata and quotients, and let $\mathbf{C}_n$ stand for the class of all automata with n states in $\mathbf{C}$. Consider any function $f : \mathbb{Z}^+ \to \mathbb{N}$ such that

$$f(n) \geq f(n-m+1) + f(m) \text{ whenever } \ n \geq m \geq 1.$$

Examples: $f(n) = n-1$, $f(n) = (n-1)^2$

Let $\mathbf{C}$ be any class of automata closed under taking subautomata and quotients, and let $\mathbf{C}_n$ stand for the class of all automata with n states in $\mathbf{C}$. Consider any function $f : \mathbb{Z}^+ \to \mathbb{N}$ such that

$$f(n) \geq f(n-m+1) + f(m) \text{ whenever } \ n \geq m \geq 1.$$

Examples: $f(n) = n-1$, $f(n) = (n-1)^2$

Theorem (Folklore)

If each synchronizing automaton in $\mathbf{C}_n$ which either is strongly connected or possesses a zero has a reset word of length $f(n)$, then the same holds true for all synchronizing automata in $\mathbf{C}_n$.

9. A Reduction: Proof

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.
Consider the set S of all states to which the automaton $\mathscr{A}$ can be reset and let $m = |S|$.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.
Consider the set S of all states to which the automaton $\mathscr{A}$ can be reset and let $m = |S|$.
If $q \in S$, then there exists a reset word $w \in \Sigma^*$ such that $Q.w = \{q\}$.
Then wa also is a reset word and $Q.wa = \{\delta(q, a)\}$ whence $\delta(q, a) \in S$.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.
Consider the set S of all states to which the automaton $\mathscr{A}$ can be reset and let $m = |S|$.
If $q \in S$, then there exists a reset word $w \in \Sigma^*$ such that $Q.w = \{q\}$.
Then wa also is a reset word and $Q.wa = \{\delta(q, a)\}$ whence $\delta(q, a) \in S$.
This means that, restricting the transition function δ to $S \times \Sigma$, we get a subautomaton $\mathscr{S}$ with the state set S.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.
Consider the set S of all states to which the automaton $\mathscr{A}$ can be reset and let $m = |S|$.
If $q \in S$, then there exists a reset word $w \in \Sigma^*$ such that $Q.w = \{q\}$.
Then wa also is a reset word and $Q.wa = \{\delta(q, a)\}$ whence $\delta(q, a) \in S$.
This means that, restricting the transition function δ to $S \times \Sigma$, we get a subautomaton $\mathscr{S}$ with the state set S.
Since $\mathscr{S}$ is synchronizing and strongly connected and since the class $\mathbf{C}$ is closed under taking subautomata, we have $\mathscr{S} \in \mathbf{C}$.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a synchronizing automaton in $\mathbf{C}_n$.
Consider the set S of all states to which the automaton $\mathscr{A}$ can be reset and let $m = |S|$.
If $q \in S$, then there exists a reset word $w \in \Sigma^*$ such that $Q.w = \{q\}$.
Then wa also is a reset word and $Q.wa = \{\delta(q, a)\}$ whence $\delta(q, a) \in S$.
This means that, restricting the transition function δ to $S \times \Sigma$, we get a subautomaton $\mathscr{S}$ with the state set S.
Since $\mathscr{S}$ is synchronizing and strongly connected and since the class $\mathbf{C}$ is closed under taking subautomata, we have $\mathscr{S} \in \mathbf{C}$.
Hence, $\mathscr{S}$ has a reset word v of length $f(m)$.

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.

10. A Reduction: End of the Proof

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.
Hence $\mathscr{A}/\rho_{\mathscr{S}}$ has a reset word u of length $f(n-m+1)$.

10. A Reduction: End of the Proof

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.
Hence $\mathscr{A}/\rho_{\mathscr{S}}$ has a reset word u of length $f(n-m+1)$.
Since $Q.u \subseteq S$ and $S.v$ is a singleton, we conclude that also $Q.uv \subseteq S.v$ is a singleton.

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.
Hence $\mathscr{A}/\rho_{\mathscr{S}}$ has a reset word u of length $f(n-m+1)$.
Since $Q.u \subseteq S$ and $S.v$ is a singleton, we conclude that also $Q.uv \subseteq S.v$ is a singleton.
Thus, uv is reset word for $\mathscr{A}$, and the length of this word does not exceed $f(n-m+1)+f(m) \leq f(n)$ according to the condition imposed on the function f.

Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.
Hence $\mathscr{A}/\rho_{\mathscr{S}}$ has a reset word u of length $f(n-m+1)$.
Since $Q.u \subseteq S$ and $S.v$ is a singleton, we conclude that also $Q.uv \subseteq S.v$ is a singleton.
Thus, uv is reset word for $\mathscr{A}$, and the length of this word does not exceed $f(n-m+1)+f(m) \leq f(n)$ according to the condition imposed on the function f.

Recall that the function $f(n)=(n-1)^2$ satisfies the condition $f(n) \geq f(n-m+1)+f(m)$ for $n \geq m \geq 1$.

10. A Reduction: End of the Proof

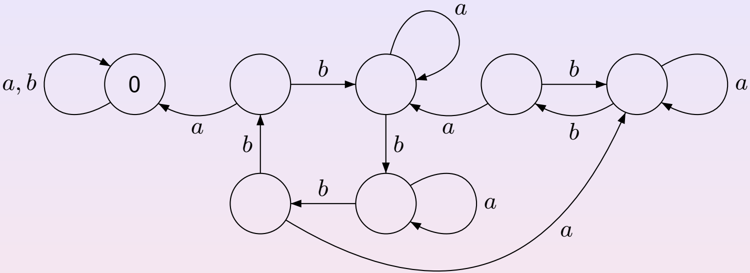
Now consider the Rees congruence $\rho_{\mathscr{S}}$ of the automaton $\mathscr{A}$.
The quotient $\mathscr{A}/\rho_{\mathscr{S}}$ is synchronizing, has S as a zero, and has $n-m+1$ states.
Since the class $\mathbf{C}$ is closed under taking quotients, we have $\mathscr{A}/\rho_{\mathscr{S}} \in \mathbf{C}$.
Hence $\mathscr{A}/\rho_{\mathscr{S}}$ has a reset word u of length $f(n-m+1)$.
Since $Q.u \subseteq S$ and $S.v$ is a singleton, we conclude that also $Q.uv \subseteq S.v$ is a singleton.
Thus, uv is reset word for $\mathscr{A}$, and the length of this word does not exceed $f(n-m+1) + f(m) \leq f(n)$ according to the condition imposed on the function f.

Recall that the function $f(n) = (n-1)^2$ satisfies the condition $f(n) \geq f(n-m+1) + f(m)$ for $n \geq m \geq 1$.
We see that it suffices to prove the Černý conjecture
1) for strongly connected automata and
2) for automata with zero.

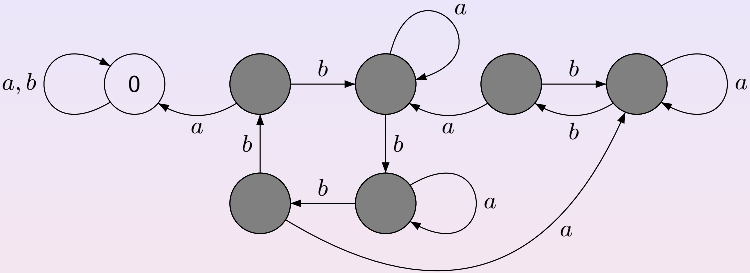
11. Automata with Zero

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.

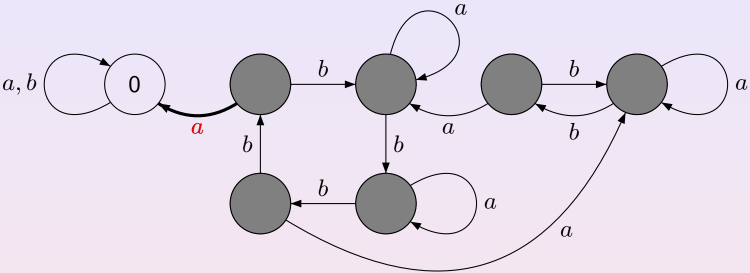


If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



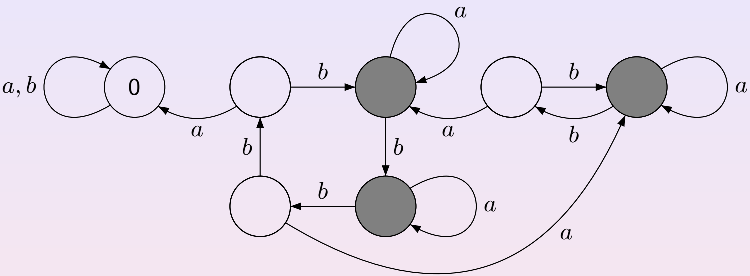
We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



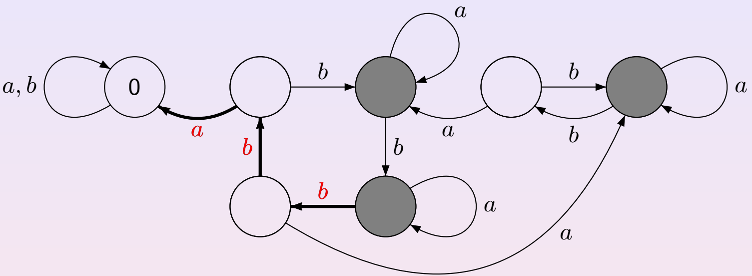
We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



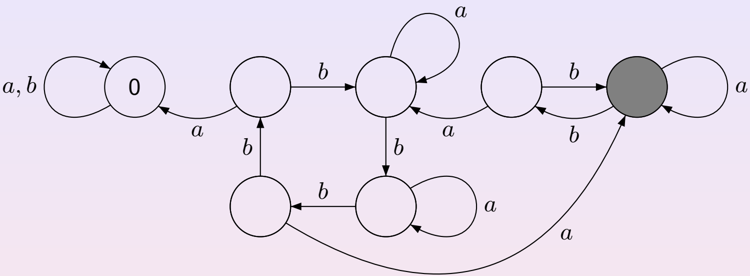
We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



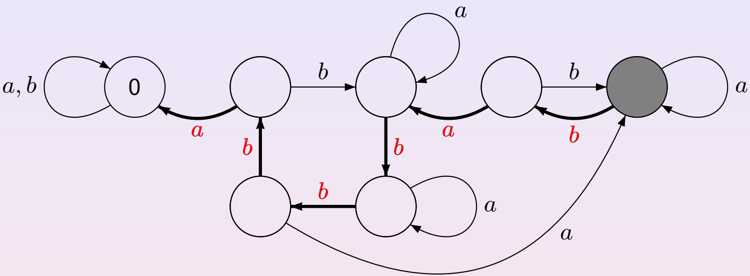
We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



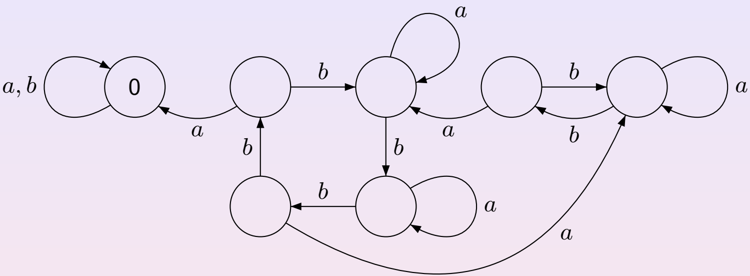
We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.

If a synchronizing automaton with n states has a zero, then it has a reset word of length $\leq \frac{n(n-1)}{2} \leq (n-1)^2$.



We cover all non-zero states with coins and move the closest coin to 0 until all coins disappear.
The algorithm makes at most $n-1$ steps and the length of the segment added in the step when t states still hold coins $(n-1 \geq t \geq 1)$ is at most $n-t$. The total length is $\leq 1+2+\cdots+(n-1) = \frac{n(n-1)}{2}$.

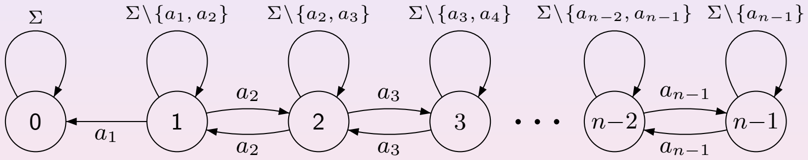
The upper bound $\frac{n(n-1)}{2}$ for the reset threshold of n-state synchronizing automata with 0 is tight.

The upper bound $\frac{n(n-1)}{2}$ for the reset threshold of n-state synchronizing automata with 0 is tight.

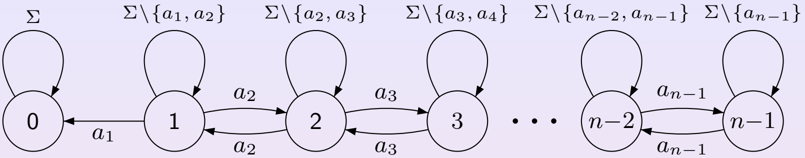
For each $n \geq 2$, Igor Rystsov (Reset words for commutative and solvable automata, Theoret. Comput. Sci. 172, 273–279 (1997)) constructed an n-state and $(n-1)$-letter synchronizing automaton $\mathscr{R}_n = \langle Q, \Sigma, \delta \rangle$ with zero and reset threshold equal to $\frac{n(n-1)}{2}$.

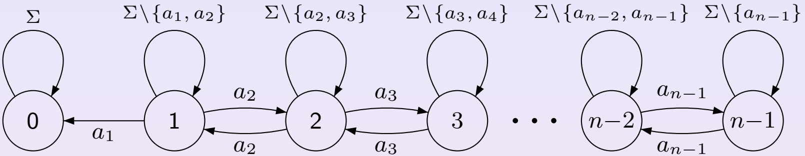
The upper bound $\frac{n(n-1)}{2}$ for the reset threshold of n-state synchronizing automata with 0 is tight.

For each $n \geq 2$, Igor Rystsov (Reset words for commutative and solvable automata, Theoret. Comput. Sci. 172, 273–279 (1997)) constructed an n-state and $(n-1)$-letter synchronizing automaton $\mathscr{R}_n = \langle Q, \Sigma, \delta \rangle$ with zero and reset threshold equal to $\frac{n(n-1)}{2}$.

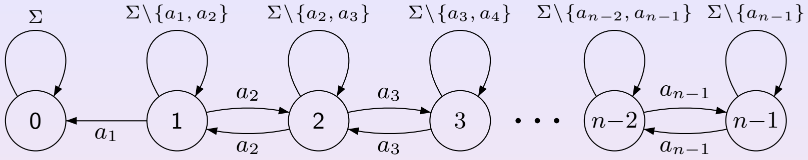


13. Rystsov's Series: Proof

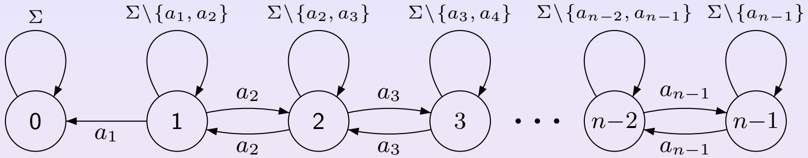




Let w be a reset word for $\mathscr{R}_n$.

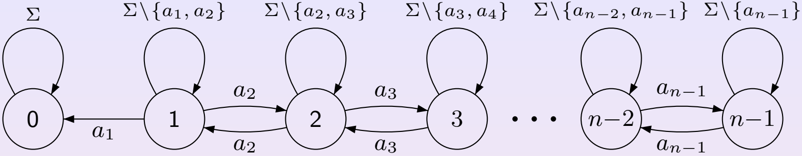


Let w be a reset word for $\mathscr{R}_n$. Clearly, $Q\,.\,w = \{0\}$.



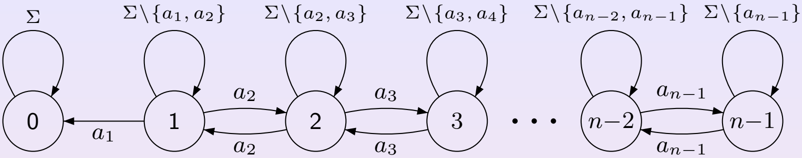
Let w be a reset word for $\mathscr{R}_n$. Clearly, $Q\,.\,w = \{0\}$.
For $S = \{s_1, \ldots, s_t\} \subseteq Q$, let $f(S) := \sum_{i=1}^{t} s_i$. Then $f(\{0\}) = 0$ and $f(Q) = \frac{n(n-1)}{2}$.

13. Rystsov's Series: Proof



Let w be a reset word for $\mathscr{R}_n$. Clearly, $Q\,.\,w = \{0\}$.
For $S = \{s_1, \dots, s_t\} \subseteq Q$, let $f(S) := \sum_{i=1}^{t} s_i$. Then $f(\{0\}) = 0$ and $f(Q) = \frac{n(n-1)}{2}$. For any S and any letter a_j, we have $f(S\,.\,a_j) \geq f(S) - 1$ since each letter only swaps two neighbor states or maps 1 and 0 to 0.

13. Rystsov's Series: Proof



Let w be a reset word for $\mathscr{R}_n$. Clearly, $Q \,.\, w = \{0\}$.
For $S = \{s_1, \dots, s_t\} \subseteq Q$, let $f(S) := \sum_{i=1}^{t} s_i$. Then $f(\{0\}) = 0$ and $f(Q) = \frac{n(n-1)}{2}$. For any S and any letter a_j, we have $f(S \,.\, a_j) \geq f(S) - 1$ since each letter only swaps two neighbor states or maps 1 and 0 to 0. Thus,

$$0 = f(\{0\}) = f(Q \,.\, w) \geq f(Q) - |w| = \frac{n(n-1)}{2} - |w|$$

whence $|w| \geq \frac{n(n-1)}{2}$.

In Rystsov's series the alphabet grows with the number of states.

In Rystsov's series the alphabet grows with the number of states.
This contrasts with the Černý series in which the alphabet is independent of the state number and leads to the following natural problem: to determine the reset threshold of n-state synchronizing automata with 0 over a fixed alphabet.

In Rystsov's series the alphabet grows with the number of states.
This contrasts with the Černý series in which the alphabet is independent of the state number and leads to the following natural problem: to determine the reset threshold of n-state synchronizing automata with 0 over a fixed alphabet.
This is an open problem, which is of independent interest and has connections with some questions of language theory.

14. Binary Case

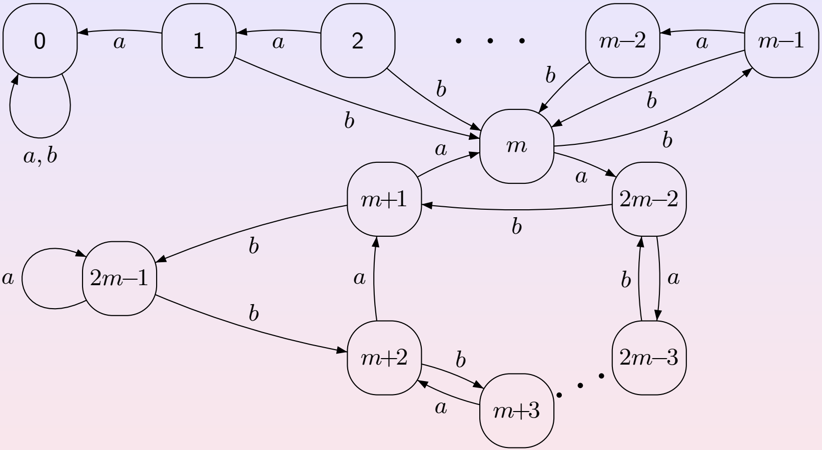
In Rystsov's series the alphabet grows with the number of states.
This contrasts with the Černý series in which the alphabet is independent of the state number and leads to the following natural problem: to determine the reset threshold of n-state synchronizing automata with 0 over a fixed alphabet.
This is an open problem, which is of independent interest and has connections with some questions of language theory.

In the binary case (2 input letters), for quite a long time, the lower bound for the reset threshold of n-state synchronizing automata with 0 found by Pavel Martyugin (A series of slowly synchronizing automata with zero state over a small alphabet, Inf. Comput. 19, 517–536 (2009)) remained the best.

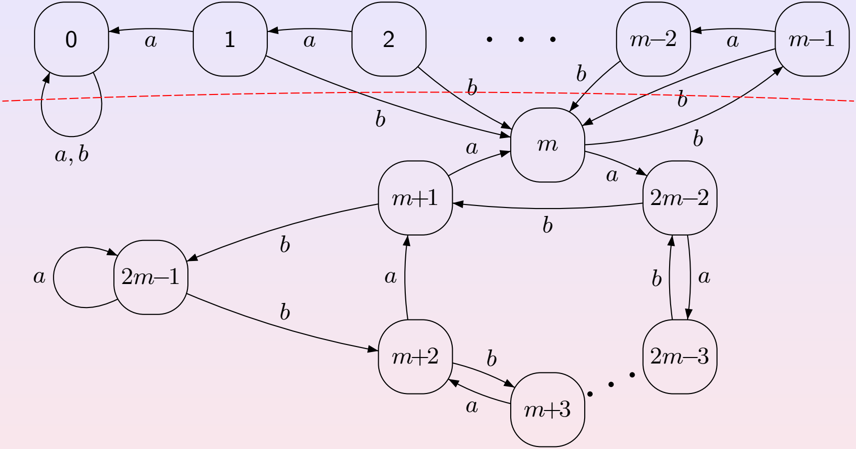
In Rystsov's series the alphabet grows with the number of states.
This contrasts with the Černý series in which the alphabet is independent of the state number and leads to the following natural problem: to determine the reset threshold of n-state synchronizing automata with 0 over a fixed alphabet.
This is an open problem, which is of independent interest and has connections with some questions of language theory.

In the binary case (2 input letters), for quite a long time, the lower bound for the reset threshold of n-state synchronizing automata with 0 found by Pavel Martyugin (A series of slowly synchronizing automata with zero state over a small alphabet, Inf. Comput. 19, 517–536 (2009)) remained the best. Namely, he has constructed, for every $n \geq 8$, a binary n-state synchronizing automaton with 0 and reset threshold equal to $\left\lceil \frac{n^2+6n-16}{4} \right\rceil$.

Here is the automaton $\mathscr{M}_n$ from Martyugin's series for $n = 2m$.

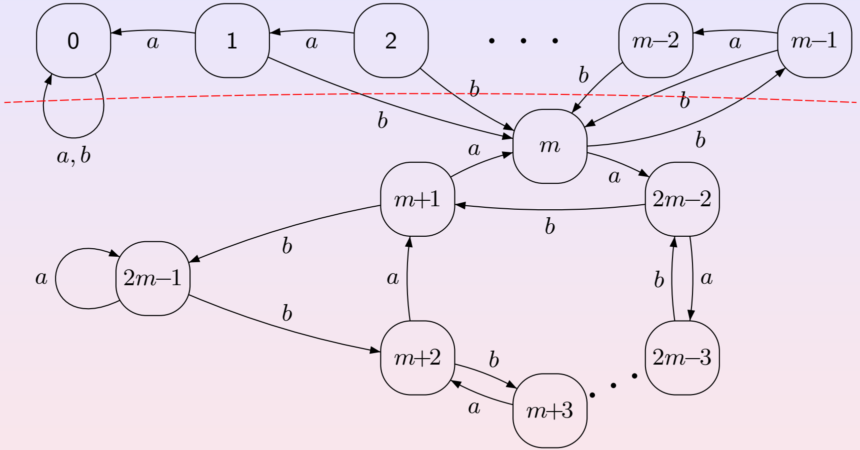


Here is the automaton $\mathscr{M}_n$ from Martyugin's series for $n = 2m$.



$\mathscr{M}_n$ consists of the "body" formed by $m, m+1, \ldots, 2m-1$

Here is the automaton $\mathscr{M}_n$ from Martyugin's series for $n = 2m$.



$\mathscr{M}_n$ consists of the "body" formed by $m, m+1, \ldots, 2m-1$ and the "tail" formed by $0, 1, \ldots, m-1$.

Vojtěch Vorel (Synchronization, Road Coloring, and Jumps in Finite Automata. Master Thesis, Charles University, Prague, 2015) has described a general construction for appending a tail to an almost permutation automaton with sink state.

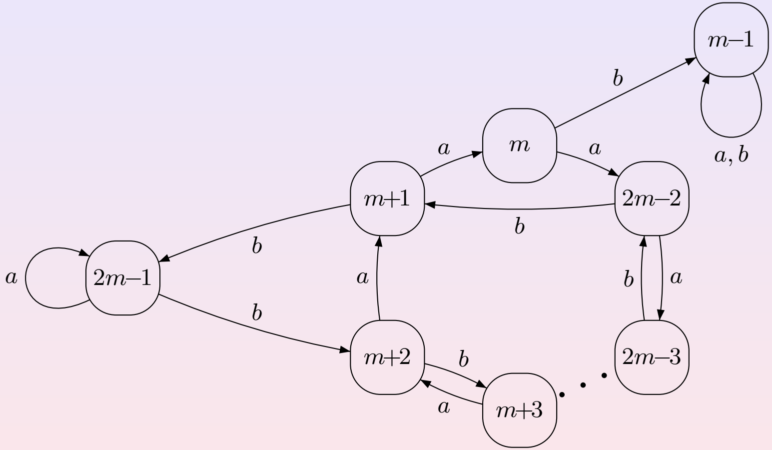
Vojtěch Vorel (Synchronization, Road Coloring, and Jumps in Finite Automata. Master Thesis, Charles University, Prague, 2015) has described a general construction for appending a tail to an almost permutation automaton with sink state. Using this idea, Ananichev and Vorel (A new lower bound for reset threshold of synchronizing automata with sink state. J. Automata, Languages and Combinatorics **24**(2-4), 153–164 (2019)) have constructed for each $n \geq 16$, $n \equiv 4 \pmod{12}$, a binary n-state synchronizing automaton with 0 that has reset threshold $\frac{1}{4}n^2 + 2n - 9$.

Vojtěch Vorel (Synchronization, Road Coloring, and Jumps in Finite Automata. Master Thesis, Charles University, Prague, 2015) has described a general construction for appending a tail to an almost permutation automaton with sink state. Using this idea, Ananichev and Vorel (A new lower bound for reset threshold of synchronizing automata with sink state. J. Automata, Languages and Combinatorics **24**(2-4), 153–164 (2019)) have constructed for each $n \geq 16$, $n \equiv 4 \pmod{12}$, a binary n-state synchronizing automaton with 0 that has reset threshold $\frac{1}{4}n^2 + 2n - 9$.

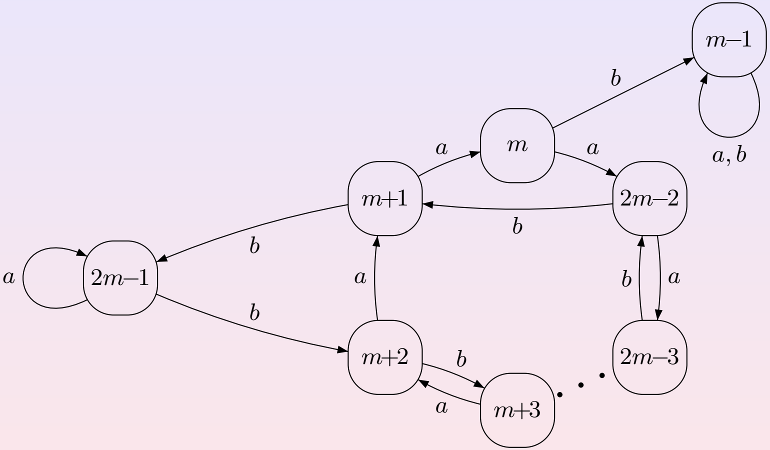
A synchronizing binary DFA $(Q, \{a, b\}, \delta)$ with a sink state $q_0 \in Q$ is an *almost permutation automaton* if it fulfils the following three conditions:

1. There is a unique state *pre-sink* $r \in Q \setminus \{q_0\}$ such that $\delta(r, b) = q_0$.
2. The letter b acts as a permutation on the set $Q \setminus \{r\}$.
3. The letter a acts as a permutation on Q.

For instance, the "body" of Martyugin's automaton $\mathscr{M}_{2m}$ is the following almost permutation automaton:



For instance, the "body" of Martyugin's automaton $\mathscr{M}_{2m}$ is the following almost permutation automaton:



Here the state $m-1$ is a sink and m is pre-sink state.

If $\mathscr{A} = (Q, \{a, b\}, \delta)$ is an almost permutation automaton, the least k such that a^k acts as the identity permutation is called the *order* of a. Clearly, the order of a is the least common multiple of the lengths of cycles with respect to a.

If $\mathscr{A} = (Q, \{a, b\}, \delta)$ is an almost permutation automaton, the least k such that a^k acts as the identity permutation is called the *order* of a. Clearly, the order of a is the least common multiple of the lengths of cycles with respect to a.

Vorel's Lemma

Let $\mathscr{A} = \langle Q, \{a, b\}, \delta\rangle$ be an n-state synchronizing almost permutation automaton and let k be a multiple of the order of a. Then one can add a "tail" to $\mathscr{A}$ so that the reset threshold of the resulting automaton is $\mathrm{rt}(\mathscr{A}) + nk$.

If $\mathscr{A} = (Q, \{a, b\}, \delta)$ is an almost permutation automaton, the least k such that a^k acts as the identity permutation is called the *order* of a. Clearly, the order of a is the least common multiple of the lengths of cycles with respect to a.

Vorel's Lemma

Let $\mathscr{A} = \langle Q, \{a, b\}, \delta \rangle$ be an n-state synchronizing almost permutation automaton and let k be a multiple of the order of a. Then one can add a "tail" to $\mathscr{A}$ so that the reset threshold of the resulting automaton is $\mathrm{rt}(\mathscr{A}) + nk$.

Thus, *in order to obtain a series of binary synchronizing automata with 0 and large reset threshold, it is sufficient to construct a series of almost permutation automata with large reset threshold.*

If $\mathscr{A} = (Q, \{a, b\}, \delta)$ is an almost permutation automaton, the least k such that a^k acts as the identity permutation is called the *order* of a. Clearly, the order of a is the least common multiple of the lengths of cycles with respect to a.

Vorel's Lemma

Let $\mathscr{A} = \langle Q, \{a, b\}, \delta\rangle$ be an n-state synchronizing almost permutation automaton and let k be a multiple of the order of a. Then one can add a "tail" to $\mathscr{A}$ so that the reset threshold of the resulting automaton is $\mathrm{rt}(\mathscr{A}) + nk$.

Thus, *in order to obtain a series of binary synchronizing automata with 0 and large reset threshold, it is sufficient to construct a series of almost permutation automata with large reset threshold.*

Suppose there is a series of n-state almost permutation automata $\mathscr{A}_n$ such that

$$\mathrm{rt}(\mathscr{A}_n) = An^2 + Bn + C,$$

where A, B and C are some constants. (Of course, $0 \leq A \leq 1/2$.)

If $\mathscr{A} = (Q, \{a, b\}, \delta)$ is an almost permutation automaton, the least k such that a^k acts as the identity permutation is called the *order* of a. Clearly, the order of a is the least common multiple of the lengths of cycles with respect to a.

Vorel's Lemma

Let $\mathscr{A} = \langle Q, \{a, b\}, \delta\rangle$ be an n-state synchronizing almost permutation automaton and let k be a multiple of the order of a. Then one can add a "tail" to $\mathscr{A}$ so that the reset threshold of the resulting automaton is $\mathrm{rt}(\mathscr{A}) + nk$.

Thus, *in order to obtain a series of binary synchronizing automata with 0 and large reset threshold, it is sufficient to construct a series of almost permutation automata with large reset threshold.*

Suppose there is a series of n-state almost permutation automata $\mathscr{A}_n$ such that

$$\mathrm{rt}(\mathscr{A}_n) = An^2 + Bn + C,$$

where A, B and C are some constants. (Of course, $0 \leq A \leq 1/2$.)
Then one can add tails of lengths $k = k(n)$ and obtain a series of binary automata $\mathscr{B}_N$ with 0 and $N = n + k$ states. If $k(n)$ is chosen to be the order of the letter a in $\mathscr{A}_n$, then Vorel's Lemma implies that

$$\mathrm{rt}(\mathscr{B}_N) = An^2 + Bn + C + nk.$$

Suppose that $k = Dn + E$, where D and E are constants. Then,

$$\mathrm{rt}(\mathscr{B}_N) = \frac{A+D}{(1+D)^2} \cdot N^2 + O(N).$$

If $D = 1 - 2A$, then the first coefficient is maximal and is equal to $\frac{1}{4(1-A)}$. This implies that if $A < 1/2$, then $\mathrm{rt}(\mathscr{B}_N)$ grows faster than $\mathrm{rt}(\mathscr{A}_n)$.

Suppose that $k = Dn + E$, where D and E are constants. Then,

$$\mathrm{rt}(\mathscr{B}_N) = \frac{A + D}{(1 + D)^2} \cdot N^2 + O(N).$$

If $D = 1 - 2A$, then the first coefficient is maximal and is equal to $\frac{1}{4(1-A)}$. This implies that if $A < 1/2$, then $\mathrm{rt}(\mathscr{B}_N)$ grows faster than $\mathrm{rt}(\mathscr{A}_n)$.

In Martyugin's example, the reset threshold of the "body" grows as $3n + C$, thus, $A = 0$ (and hence $D = 1$). Also $k = n - 2$. This gives

$$\mathrm{rt}(\mathscr{M}_N) = \frac{1}{4}N^2 + \frac{3}{2}N + O(1).$$

Suppose that $k = Dn + E$, where D and E are constants. Then,

$$\mathrm{rt}(\mathscr{B}_N) = \frac{A + D}{(1 + D)^2} \cdot N^2 + O(N).$$

If $D = 1 - 2A$, then the first coefficient is maximal and is equal to $\frac{1}{4(1-A)}$. This implies that if $A < 1/2$, then $\mathrm{rt}(\mathscr{B}_N)$ grows faster than $\mathrm{rt}(\mathscr{A}_n)$.

In Martyugin's example, the reset threshold of the "body" grows as $3n + C$, thus, $A = 0$ (and hence $D = 1$). Also $k = n - 2$. This gives

$$\mathrm{rt}(\mathscr{M}_N) = \frac{1}{4}N^2 + \frac{3}{2}N + O(1).$$

Ananichev and Vorel (loc. cit.) constructed n-state synchronizing almost permutation automata with reset threshold $4n - 13$. This gives

$$\mathrm{rt}(\mathscr{A}\mathscr{V}_N) = \frac{1}{4}N^2 + 2N + O(1).$$

Suppose that $k = Dn + E$, where D and E are constants. Then,

$$\mathrm{rt}(\mathscr{B}_N) = \frac{A + D}{(1 + D)^2} \cdot N^2 + O(N).$$

If $D = 1 - 2A$, then the first coefficient is maximal and is equal to $\frac{1}{4(1-A)}$. This implies that if $A < 1/2$, then $\mathrm{rt}(\mathscr{B}_N)$ grows faster than $\mathrm{rt}(\mathscr{A}_n)$.

In Martyugin's example, the reset threshold of the "body" grows as $3n + C$, thus, $A = 0$ (and hence $D = 1$). Also $k = n - 2$. This gives

$$\mathrm{rt}(\mathscr{M}_N) = \frac{1}{4}N^2 + \frac{3}{2}N + O(1).$$

Ananichev and Vorel (loc. cit.) constructed n-state synchronizing almost permutation automata with reset threshold $4n - 13$. This gives

$$\mathrm{rt}(\mathscr{A}\mathscr{V}_N) = \frac{1}{4}N^2 + 2N + O(1).$$

An intriguing question: is there a series of n-state almost permutation synchronizing automata with reset threshold $An^2 + O(n)$ where $A > 0$?

Suppose that $k = Dn + E$, where D and E are constants. Then,

$$\mathrm{rt}(\mathscr{B}_N) = \frac{A+D}{(1+D)^2} \cdot N^2 + O(N).$$

If $D = 1 - 2A$, then the first coefficient is maximal and is equal to $\frac{1}{4(1-A)}$. This implies that if $A < 1/2$, then $\mathrm{rt}(\mathscr{B}_N)$ grows faster than $\mathrm{rt}(\mathscr{A}_n)$.

In Martyugin's example, the reset threshold of the "body" grows as $3n + C$, thus, $A = 0$ (and hence $D = 1$). Also $k = n - 2$. This gives

$$\mathrm{rt}(\mathscr{M}_N) = \frac{1}{4}N^2 + \frac{3}{2}N + O(1).$$

Ananichev and Vorel (loc. cit.) constructed n-state synchronizing almost permutation automata with reset threshold $4n - 13$. This gives

$$\mathrm{rt}(\mathscr{A}\mathscr{V}_N) = \frac{1}{4}N^2 + 2N + O(1).$$

An intriguing question: is there a series of n-state almost permutation synchronizing automata with reset threshold $An^2 + O(n)$ where $A > 0$? If so, then $\mathrm{rt}(\mathscr{B}_N)$ would grow faster than $\frac{1}{4}N^2$.