Synchronizing Automata

a problem everyone can understand but nobody can solve (so far)

Mikhail Volkov

Ural Federal University / Hunter College

February 17, 2016

Can Modern Math be Understood?

"*Most current mathematical research, since the 60s, is devoted to fancy situations: it brings solutions which nobody understands to questions nobody asked*" (quoted from Bernard Beauzamy, "Real Life Mathematics", Irish Math. Soc. Bull. 48 (2002), 43-46).

This provocative claim is certainly exaggerated but it does reflect a really serious problem: a communication barrier between math (and exact science in general) and human common sense.

The barrier results in a paradox: while the achievements of modern mathematics are widely used in many crucial aspects of everyday life, people tend to believe that today mathematicians do "abstract nonsense" of no use at all.

Can Modern Math be Understood?

"*Most current mathematical research, since the 60s, is devoted to fancy situations: it brings solutions which nobody understands to questions nobody asked*" (quoted from Bernard Beauzamy, "Real Life Mathematics", Irish Math. Soc. Bull. 48 (2002), 43-46).

This provocative claim is certainly exaggerated but it does reflect a really serious problem: a communication barrier between math (and exact science in general) and human common sense.

The barrier results in a paradox: while the achievements of modern mathematics are widely used in many crucial aspects of everyday life, people tend to believe that today mathematicians do "abstract nonsense" of no use at all.

Can Modern Math be Understood?

"*Most current mathematical research, since the 60s, is devoted to fancy situations: it brings solutions which nobody understands to questions nobody asked*" (quoted from Bernard Beauzamy, "Real Life Mathematics", Irish Math. Soc. Bull. 48 (2002), 43-46).

This provocative claim is certainly exaggerated but it does reflect a really serious problem: a communication barrier between math (and exact science in general) and human common sense.

The barrier results in a paradox: while the achievements of modern mathematics are widely used in many crucial aspects of everyday life, people tend to believe that today mathematicians do "abstract nonsense" of no use at all.

Can Modern Math be Understood?

"*Most current mathematical research, since the 60s, is devoted to fancy situations: it brings solutions which nobody understands to questions nobody asked*" (quoted from Bernard Beauzamy, "Real Life Mathematics", Irish Math. Soc. Bull. 48 (2002), 43-46).

This provocative claim is certainly exaggerated but it does reflect a really serious problem: a communication barrier between math (and exact science in general) and human common sense.

The barrier results in a paradox: while the achievements of modern mathematics are widely used in many crucial aspects of everyday life, people tend to believe that today mathematicians do "abstract nonsense" of no use at all.

Can Modern Math be Understood?

In many cases there is an inherent difficulty: mathematics is difficult. Normally non-mathematicians do accept the fact that a solution to a mathematical problem may be difficult to digest but the point is that it is usually hard to explain to them why the solution is worth the effort.

Can Modern Math be Understood?

In many cases there is an inherent difficulty: mathematics is difficult. Normally non-mathematicians do accept the fact that a solution to a mathematical problem may be difficult to digest but the point is that it is usually hard to explain to them why the solution is worth the effort.

Can Modern Math be Understood?

In many cases there is an inherent difficulty: mathematics is difficult. Normally non-mathematicians do accept the fact that a solution to a mathematical problem may be difficult to digest but the point is that it is usually hard to explain to them why the solution is worth the effort.

Can Modern Math be Understood?

In many cases there is an inherent difficulty: mathematics is difficult. Normally non-mathematicians do accept the fact that a solution to a mathematical problem may be difficult to digest but the point is that it is usually hard to explain to them why the solution is worth the effort.

Well, you have proved Fermat's Last Theorem, congratulations!

Can Modern Math be Understood?

In many cases there is an inherent difficulty: mathematics is difficult. Normally non-mathematicians do accept the fact that a solution to a mathematical problem may be difficult to digest but the point is that it is usually hard to explain to them why the solution is worth the effort.

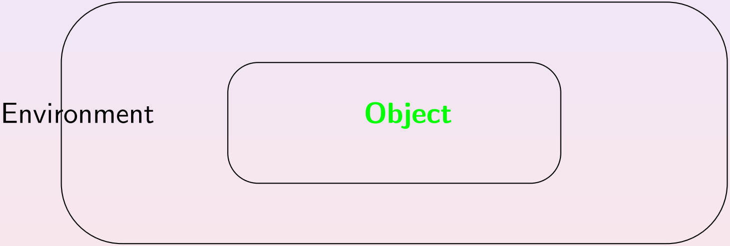
Well, you have proved Fermat's Last Theorem, congratulations!

Will my cows give more milk now?

Finite Automata

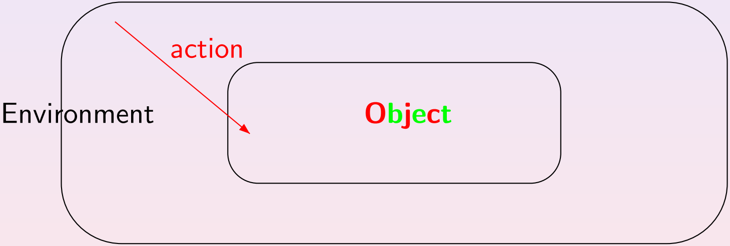
A finite automaton is a simple but extremely productive notion that captures the idea of an object interacting with an environment.

A finite automaton is a simple but extremely productive notion that captures the idea of an object interacting with an environment.

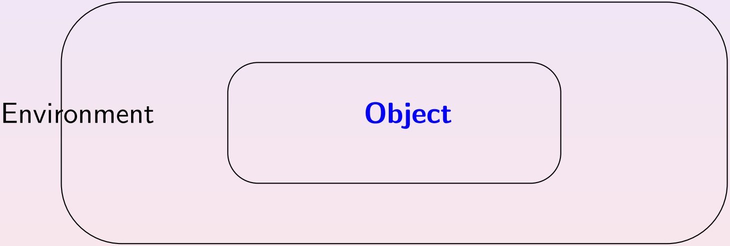


Finite Automata

A finite automaton is a simple but extremely productive notion that captures the idea of an object interacting with an environment.



A finite automaton is a simple but extremely productive notion that captures the idea of an object interacting with an environment.



This notion originates in the seminal work by Alan Turing ("On Computable Numbers, With an Application to the Entscheidungsproblem", Proc. London Math. Soc., Ser. 2, 42 (1936), 230–265).

"The behavior of the computer at any moment is determined by the symbols which he is observing, and his state of mind at that moment".

Another important source is the work by neurobiologists Warren McCulloch and Walter Pitts ("A Logical Calculus of the Ideas Immanent in Nervous Activity", Bull. Math. Biophys. 5 (1943), 115–133).

Finite Automata

This notion originates in the seminal work by Alan Turing ("On Computable Numbers, With an Application to the Entscheidungsproblem", Proc. London Math. Soc., Ser. 2, 42 (1936), 230–265).

"*The behavior of the computer at any moment is determined by the symbols which he is observing, and his state of mind at that moment*".

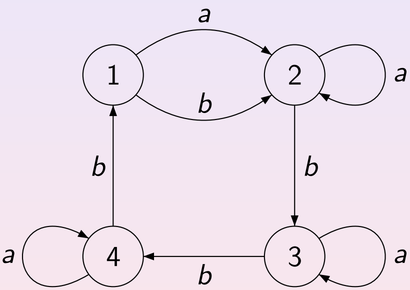
Another important source is the work by neurobiologists Warren McCulloch and Walter Pitts ("A Logical Calculus of the Ideas Immanent in Nervous Activity", Bull. Math. Biophys. 5 (1943), 115–133).

Visualization

Finite automata admit a convenient visual representation.

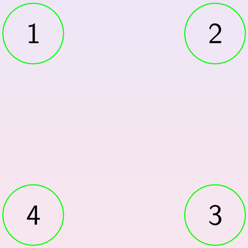
Visualization

Finite automata admit a convenient visual representation.



Visualization

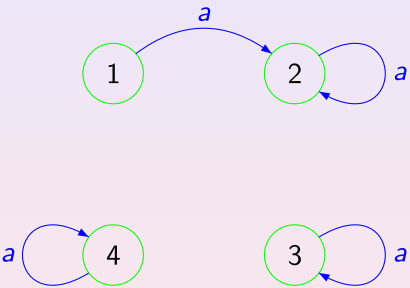
Finite automata admit a convenient visual representation.



Here one sees 4 **states** called 1,2,3,4,

Visualization

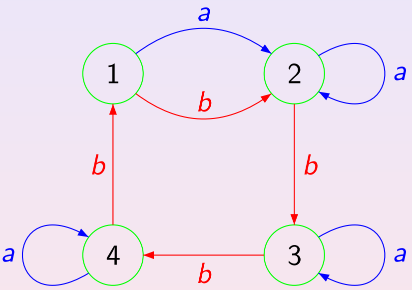
Finite automata admit a convenient visual representation.



Here one sees 4 **states** called 1,2,3,4, an action called *a*

Visualization

Finite automata admit a convenient visual representation.



Here one sees 4 **states** called 1,2,3,4, an action called a and another action called b.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q \,.\, w$ for $\delta(q, w)$ and $P \,.\, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q \,.\, w$ for $\delta(q, w)$ and $P \,.\, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.
Σ^* stands for the set of all words over Σ including the empty word.
The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.
To simplify notation we often write $q \,.\, w$ for $\delta(q, w)$ and $P \,.\, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q\,.\,w$ for $\delta(q, w)$ and $P\,.\,w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

- Q is the state set;
- Σ is the input alphabet;
- $\delta : Q \times \Sigma \to Q$ is the totally defined transition function.

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q \,.\, w$ for $\delta(q, w)$ and $P \,.\, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q \,.\, w$ for $\delta(q, w)$ and $P \,.\, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q\,.\,w$ for $\delta(q, w)$ and $P\,.\,w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.
Σ^* stands for the set of all words over Σ including the empty word.
The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.
To simplify notation we often write $q \, . \, w$ for $\delta(q, w)$
and $P \, . \, w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and Terminology

We consider complete deterministic finite automata:

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

Here

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

We need neither initial nor final states.

Σ^* stands for the set of all words over Σ including the empty word.

The function δ uniquely extends to a function $Q \times \Sigma^* \to Q$ still denoted by δ.

To simplify notation we often write $q\,.\,w$ for $\delta(q, w)$ and $P\,.\,w$ for $\{\delta(q, w) \mid q \in P\}$.

Definitions and terminology

An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ 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 $q, q' \in Q$.

We can also write this as $|Q \,.\, w| = 1$.

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

Other names:

- for automata: directable, cofinal, collapsible, etc;
- for words: directing, recurrent, synchronizing, etc.

Definitions and terminology

An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ 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 $q, q' \in Q$.

We can also write this as $|Q \,.\, w| = 1$.

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

Other names:

- for automata: directable, cofinal, collapsible, etc;
- for words: directing, recurrent, synchronizing, etc.

Definitions and terminology

An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ 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 $q, q' \in Q$.

We can also write this as $|Q \,.\, w| = 1$.

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

Other names:

- for automata: directable, cofinal, collapsible, etc;
- for words: directing, recurrent, synchronizing, etc.

Definitions and terminology

An automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ 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 $q, q' \in Q$.

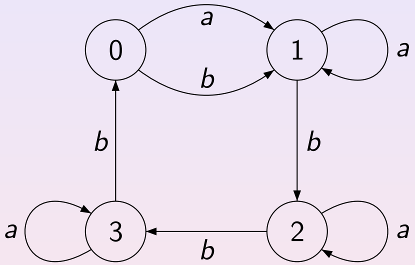
We can also write this as $|Q \,.\, w| = 1$.

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

Other names:

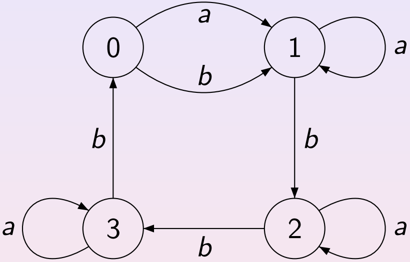
- for automata: directable, cofinal, collapsible, etc;
- for words: directing, recurrent, synchronizing, etc.

An Example



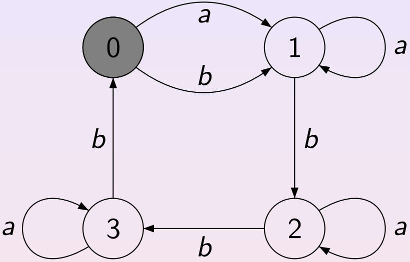
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



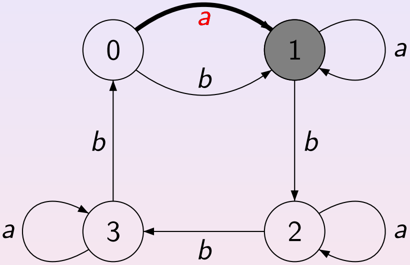
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



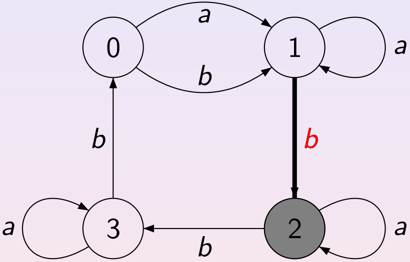
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



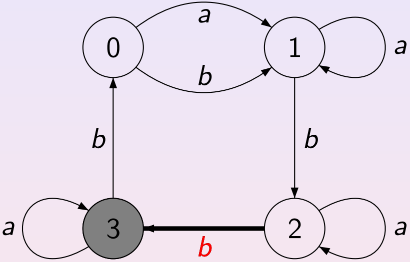
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



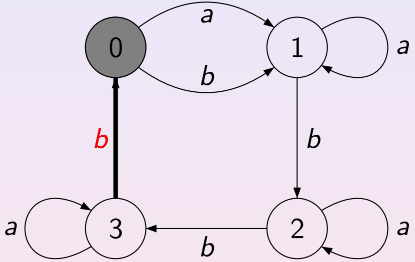
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



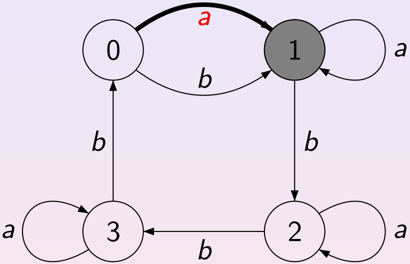
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



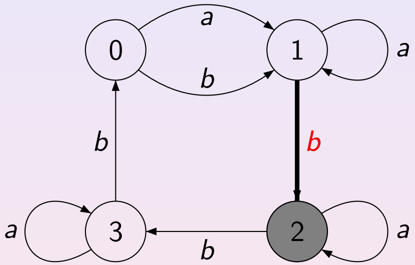
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



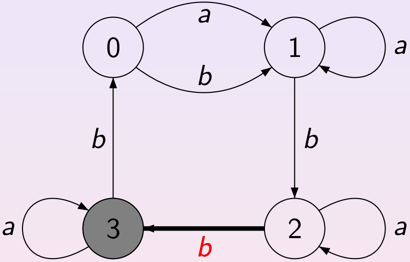
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



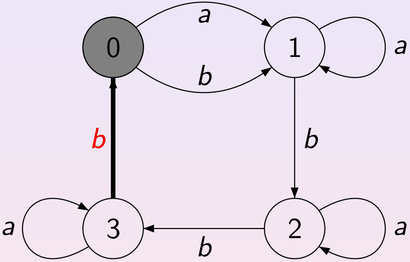
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



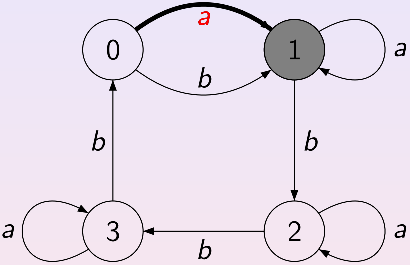
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



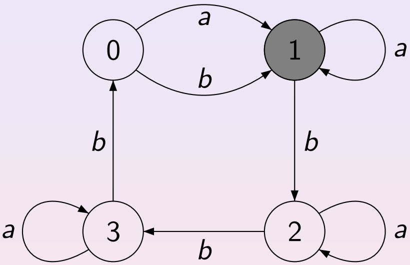
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



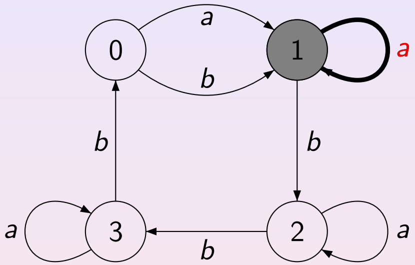
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



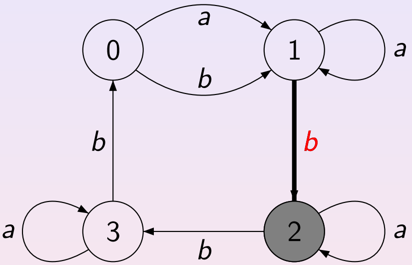
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



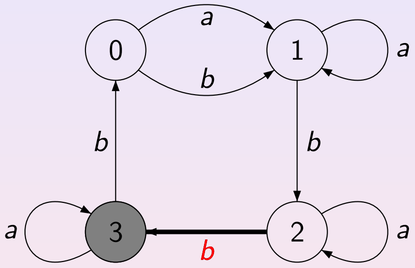
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



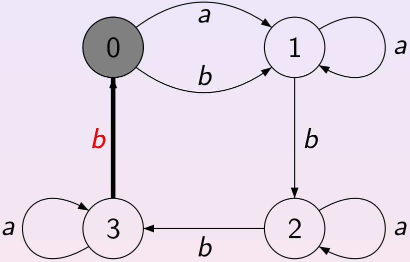
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



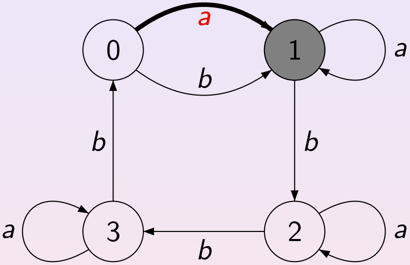
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



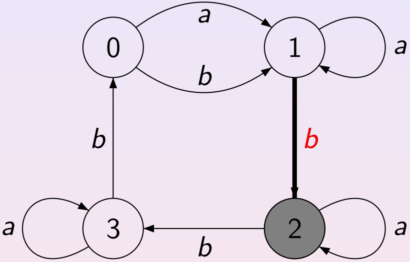
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



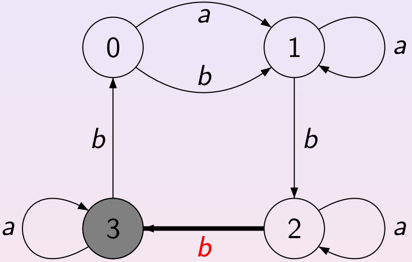
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



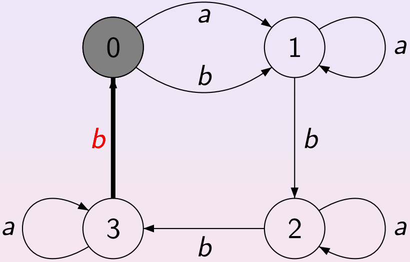
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



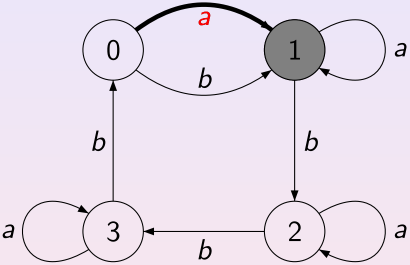
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



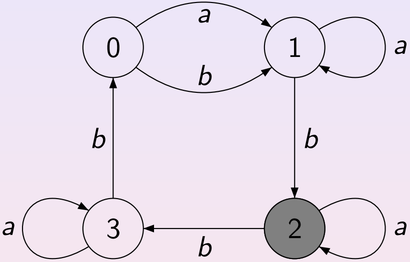
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



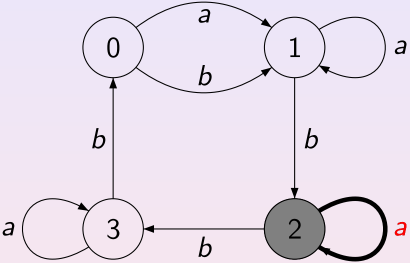
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



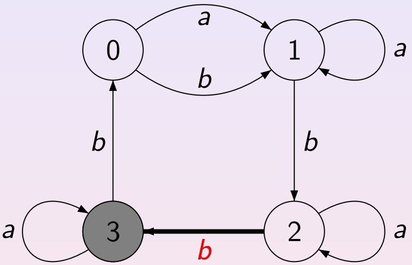
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



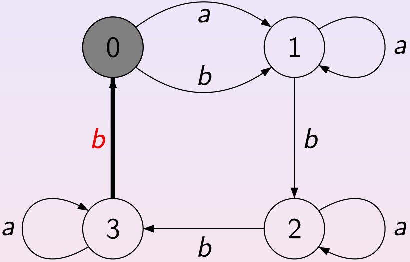
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



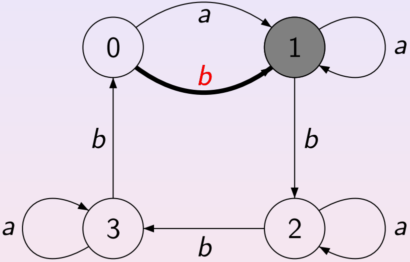
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



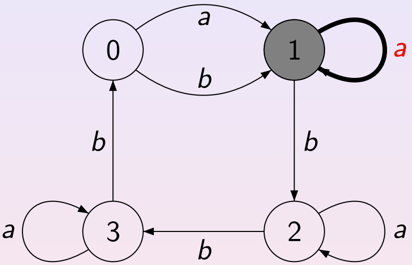
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



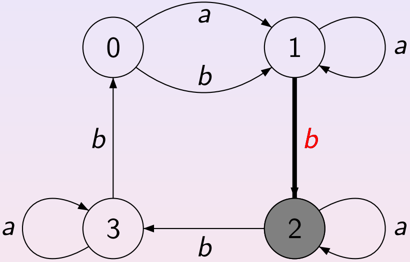
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



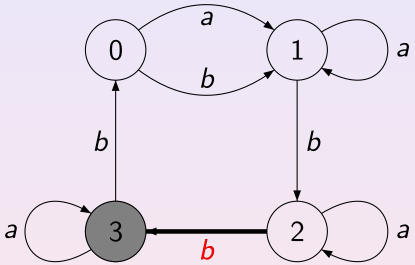
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



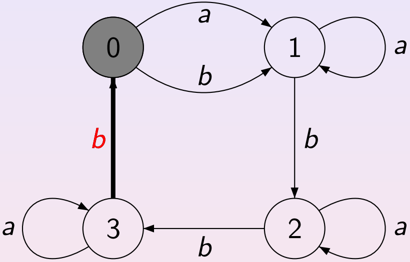
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



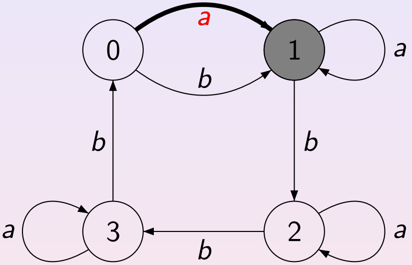
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



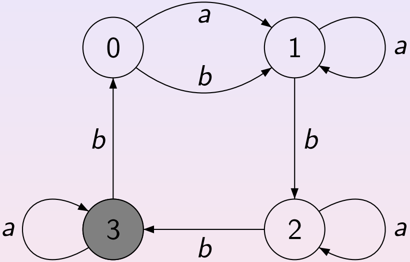
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



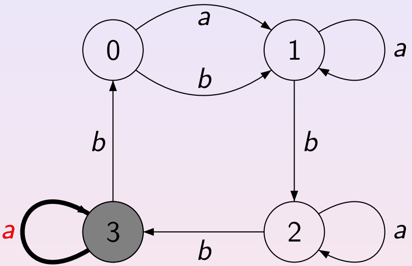
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



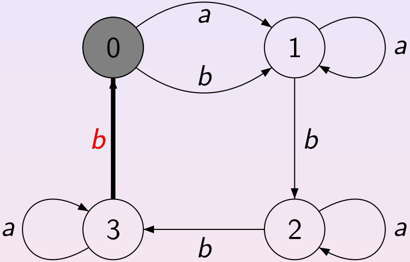
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



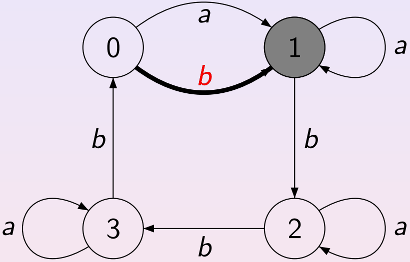
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



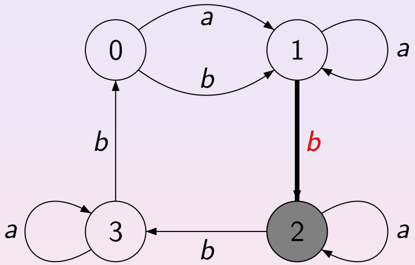
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



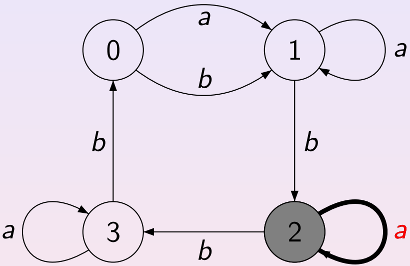
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



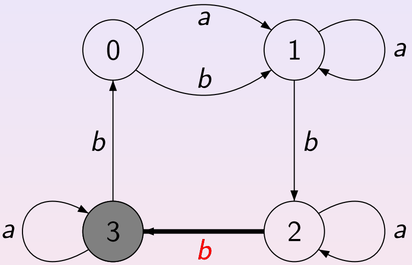
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



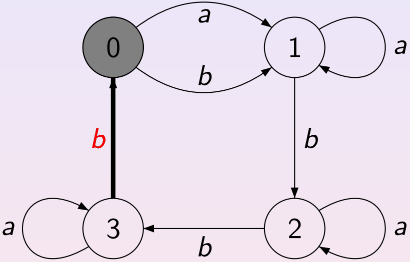
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



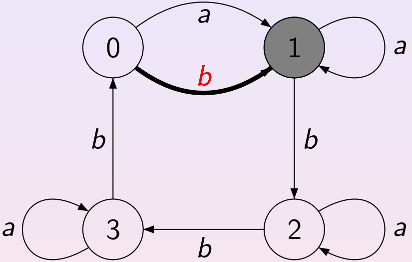
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



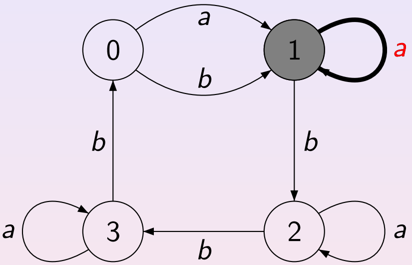
A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

An Example



A reset word is *abbbabbba*: applying it at any state brings the automaton to the state 1

Cerný's Paper

The notion was formalized in 1964 in a paper by Jan Černý (Poznámka k homogénnym eksperimentom s konečnými automatami, Matematicko-fyzikalny Časopis Slovensk. Akad. Vied, 14, no.3, 208–216 [in Slovak]) though implicitly it had been around since at least 1956.

The idea of synchronization is pretty natural and of obvious importance: we aim to restore control over a device whose current state is not known.

Think of a satellite which loops around the Moon and cannot be controlled from the Earth while "behind" the Moon (Černý's original motivation).

Cerný's Paper

The notion was formalized in 1964 in a paper by Jan Černý (Poznámka k homogénnym eksperimentom s konečnými automatami, Matematicko-fyzikalny Časopis Slovensk. Akad. Vied, 14, no.3, 208–216 [in Slovak]) though implicitly it had been around since at least 1956.

The idea of synchronization is pretty natural and of obvious importance: we aim to restore control over a device whose current state is not known.

Think of a satellite which loops around the Moon and cannot be controlled from the Earth while "behind" the Moon (Černý's original motivation).

Černý's Paper

The notion was formalized in 1964 in a paper by Jan Černý (Poznámka k homogénnym eksperimentom s konečnými automatami, Matematicko-fyzikalny Časopis Slovensk. Akad. Vied, 14, no.3, 208–216 [in Slovak]) though implicitly it had been around since at least 1956.

The idea of synchronization is pretty natural and of obvious importance: we aim to restore control over a device whose current state is not known.

Think of a satellite which loops around the Moon and cannot be controlled from the Earth while "behind" the Moon (Černý's original motivation).

It is not surprising that synchronizing automata were re-invented a number of times:

- The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.
- Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Other Sources

It is not surprising that synchronizing automata were re-invented a number of times:

- The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.
- Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

It is not surprising that synchronizing automata were re-invented a number of times:

- The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.
- Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Other Sources

It is not surprising that synchronizing automata were re-invented a number of times:

• The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.

• Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Other Sources

It is not surprising that synchronizing automata were re-invented a number of times:

• The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.

• Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Other Sources

It is not surprising that synchronizing automata were re-invented a number of times:

• The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.

• Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Other Sources

It is not surprising that synchronizing automata were re-invented a number of times:

• The notion was very natural by itself and fitted fairly well in what was considered as the mainstream of automata theory in the 1960s.

• Černý's paper published in Slovak language remained unknown in the English-speaking world for quite a long time.

Example: A. E. Laemmel, B. Rudner, Study of the application of coding theory, Report PIBEP-69-034, Polytechnic Inst. Brooklyn, Dept. Electrophysics, Farmingdale, N.Y., 94 pp.

Since the 60s synchronizing automata have been considered as a useful tool for testing of reactive systems (first circuits, later protocols) and have been also applied in coding theory.

In the 80s, the notion was reinvented by engineers working in a branch of robotics which deals with part handling problems in industrial automation.

Re-inventing by Dynamics Theorists

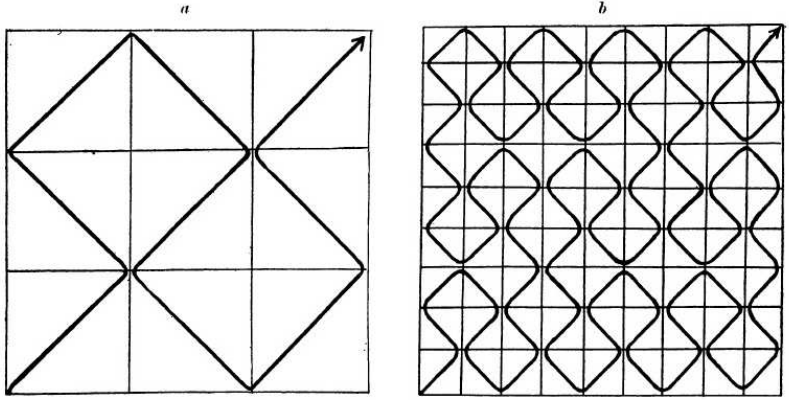
In the 90s, synchronizing automata were rediscovered by dynamic theorists who studied constant length substitutions. We use as a guiding example E. H. Moore's construction for the Peano curve ("On Certain Crinkly Curves", Trans. AMS, 1 (1900), 72–90).

Re-inventing by Dynamics Theorists

In the 90s, synchronizing automata were rediscovered by dynamic theorists who studied constant length substitutions. We use as a guiding example E. H. Moore's construction for the Peano curve ("On Certain Crinkly Curves", Trans. AMS, 1 (1900), 72–90).

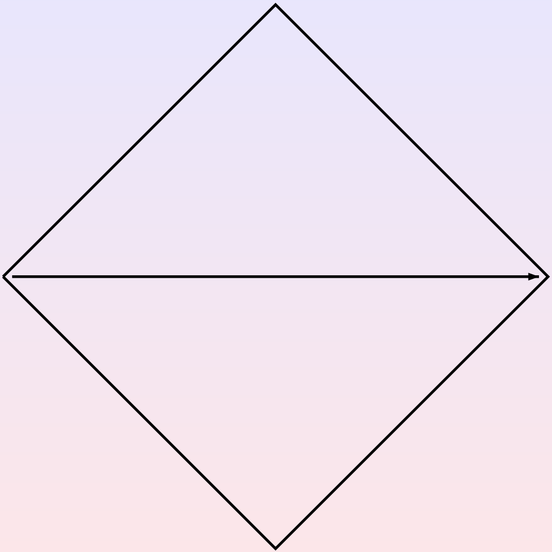
Re-inventing by Dynamics Theorists

In the 90s, synchronizing automata were rediscovered by dynamic theorists who studied constant length substitutions. We use as a guiding example E. H. Moore's construction for the Peano curve ("On Certain Crinkly Curves", Trans. AMS, 1 (1900), 72–90).

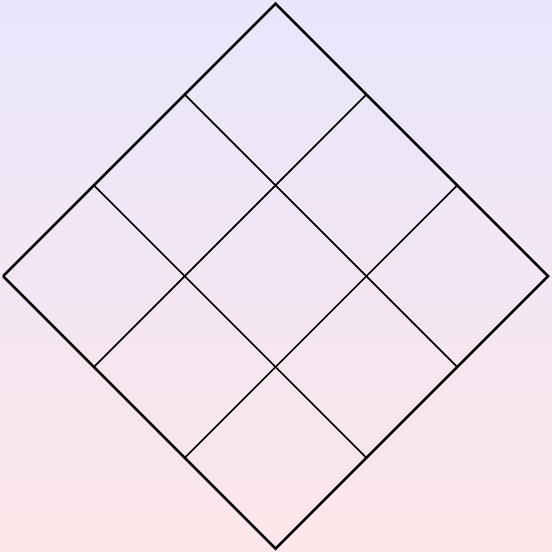


Line $C_1^{(3)}$ Line $C_2^{(3)}$

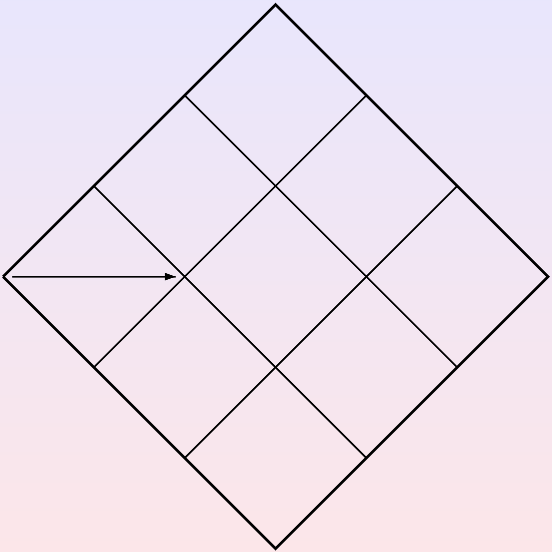
Moore's Curve as Substitution



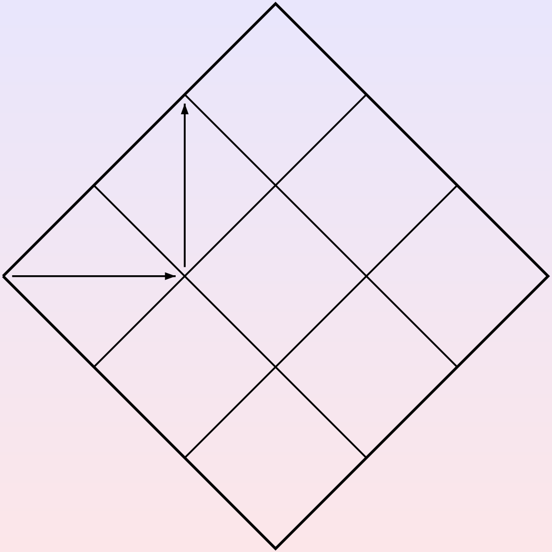
Moore's Curve as Substitution



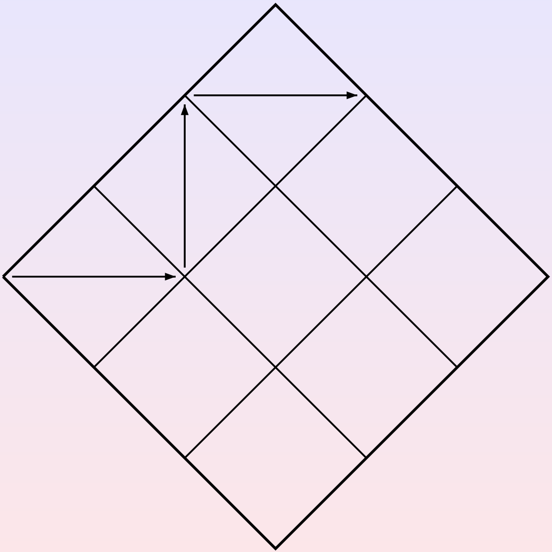
Moore's Curve as Substitution



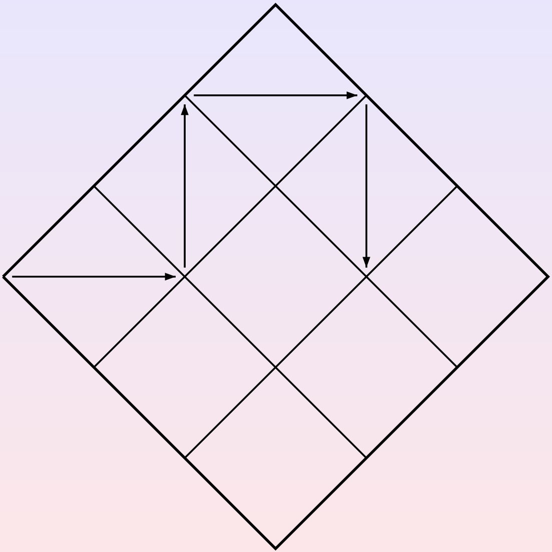
Moore's Curve as Substitution



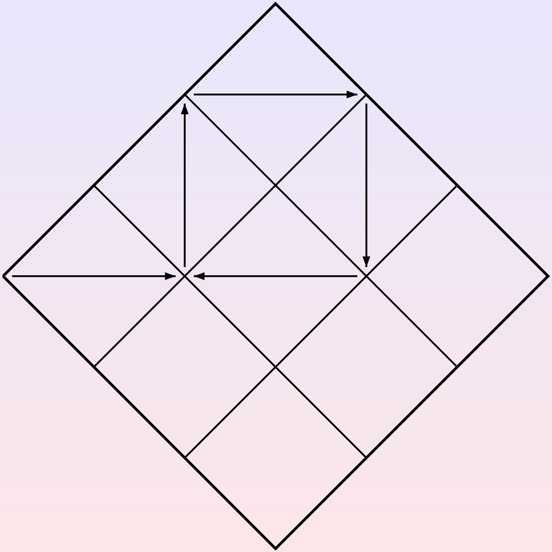
Moore's Curve as Substitution



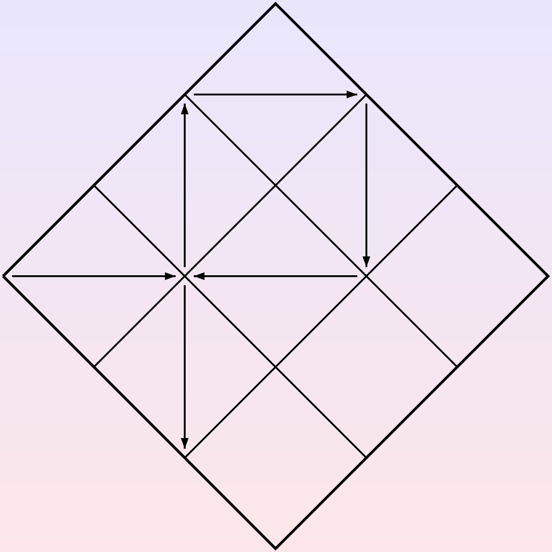
Moore's Curve as Substitution



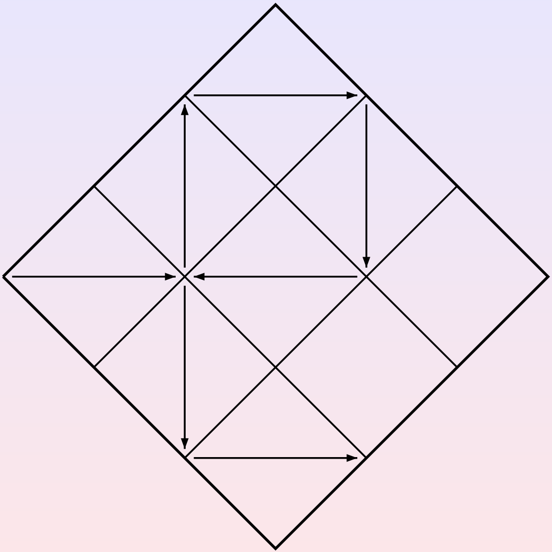
Moore's Curve as Substitution

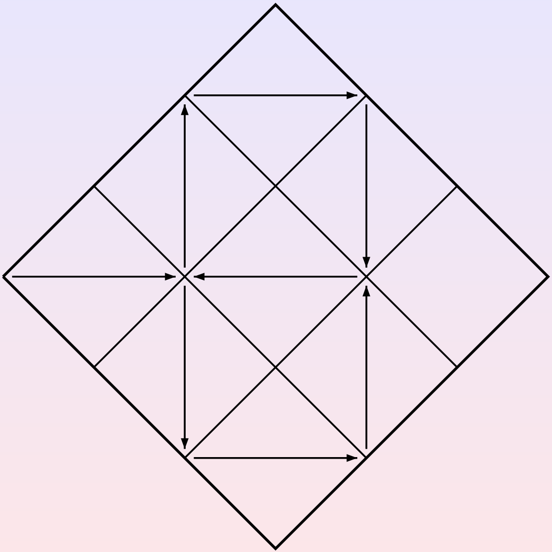


Moore's Curve as Substitution

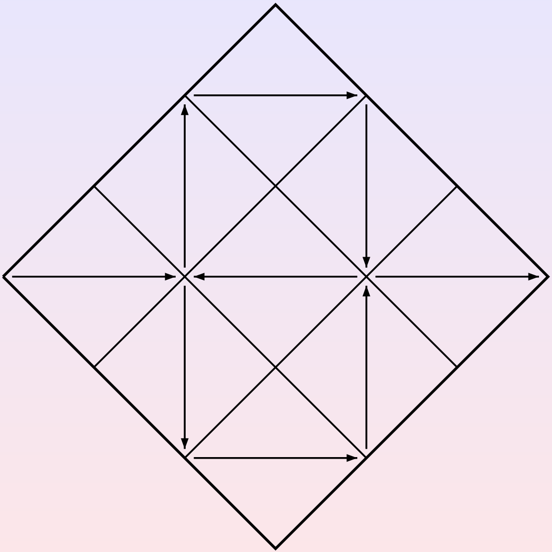


Moore's Curve as Substitution

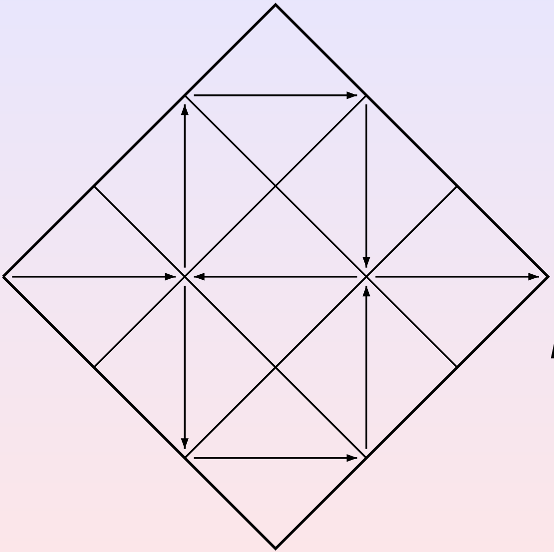




Moore's Curve as Substitution

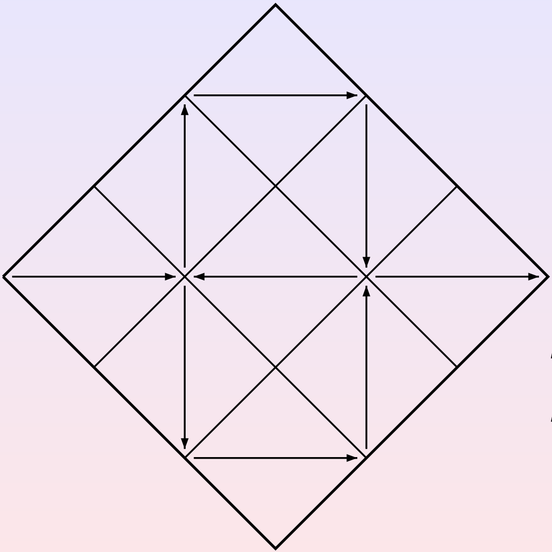


Moore's Curve as Substitution



$E \mapsto ENE\ SWS\ ENE$

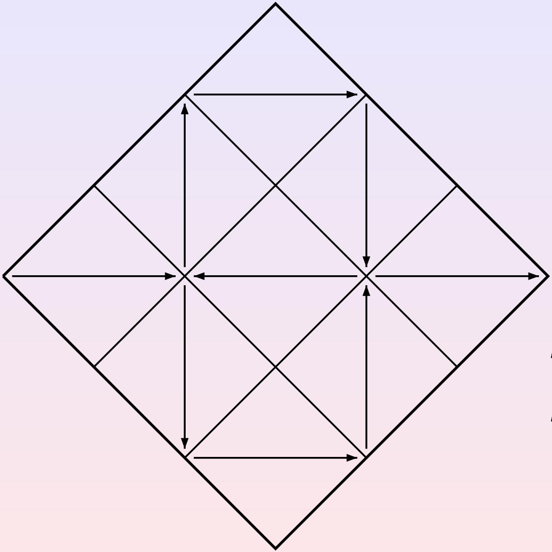
Moore's Curve as Substitution



$E \mapsto ENE\ SWS\ ENE$

$N \mapsto NWN\ ESE\ NWN$

Moore's Curve as Substitution

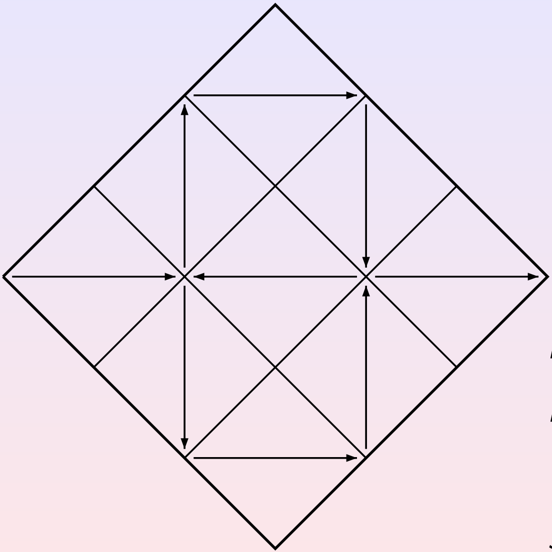


$E \mapsto ENE\ SWS\ ENE$

$N \mapsto NWN\ ESE\ NWN$

$W \mapsto WSW\ NEN\ WSW$

Moore's Curve as Substitution



$E \mapsto ENE\ SWS\ ENE$

$N \mapsto NWN\ ESE\ NWN$

$W \mapsto WSW\ NEN\ WSW$

$S \mapsto SES\ WNW\ SES$

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

$$\begin{array}{ccc} 0 & \mapsto & 11 \\ 1 & \mapsto & 12 \\ 2 & \mapsto & 20 \end{array}$$

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

$$\begin{array}{ccccc} 0 & \mapsto & 11 & \mapsto & 1212 \\ 1 & \mapsto & 12 & \mapsto & 1220 \\ 2 & \mapsto & 20 & \mapsto & 2011 \end{array}$$

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

$$\begin{array}{ccccccc} 0 & \mapsto & 11 & \mapsto & 1212 & \mapsto & 12201220 \\ 1 & \mapsto & 12 & \mapsto & 1220 & \mapsto & 12202011 \\ 2 & \mapsto & 20 & \mapsto & 2011 & \mapsto & 20111212 \end{array}$$

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

$$\begin{array}{ccccccccc} 0 & \mapsto & 11 & \mapsto & 1212 & \mapsto & 12201220 & \mapsto & 1220201112202011 \\ 1 & \mapsto & 12 & \mapsto & 1220 & \mapsto & 12202011 & \mapsto & 1220201120111212 \\ 2 & \mapsto & 20 & \mapsto & 2011 & \mapsto & 20111212 & \mapsto & 2011121212201220 \end{array}$$

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

0	$\mapsto$	11	$\mapsto$	1212	$\mapsto$	12201220	$\mapsto$	122020**1**112202011
1	$\mapsto$	12	$\mapsto$	1220	$\mapsto$	12202011	$\mapsto$	122020**1**120111212
2	$\mapsto$	20	$\mapsto$	2011	$\mapsto$	20111212	$\mapsto$	201112**1**212201220

Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$).
The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

A substitution on a finite alphabet X is a map $\sigma : X \to X^+$; the substitution is said to be of constant length if all words $\sigma(x)$, $x \in X$, have the same length. One says that σ satisfies the coincidence condition if there exist positive integers m and k such that all words $\sigma^k(x)$ have the same letter in the m-th position. For an example, consider the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$. Calculate the iterations of τ up to τ^4:

$$\begin{array}{ccccccccc} 0 & \mapsto & 11 & \mapsto & 1212 & \mapsto & 12201220 & \mapsto & 122020\mathbf{1}112202011 \\ 1 & \mapsto & 12 & \mapsto & 1220 & \mapsto & 12202011 & \mapsto & 122020\mathbf{1}120111212 \\ 2 & \mapsto & 20 & \mapsto & 2011 & \mapsto & 20111212 & \mapsto & 201112\mathbf{1}212201220 \end{array}$$

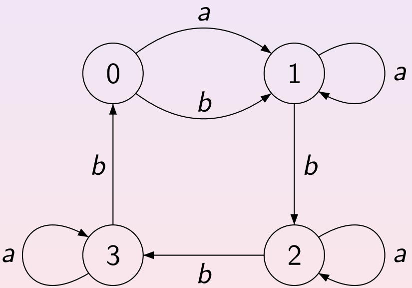
Thus, τ satisfies the coincidence condition (with $k = 4$, $m = 7$). The coincidence condition completely characterizes the constant length substitutions that give rise to dynamical systems measure-theoretically isomorphic to a translation on a compact Abelian group (Dekking, 1978).

Substitution Systems = DFAs

There is a straightforward bijection between DFAs and constant length substitutions. Each DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with $\Sigma = \{a_1, \ldots, a_\ell\}$ defines a length ℓ substitution on Q that maps every $q \in Q$ to the word $(q \, . \, a_1) \ldots (q \, . \, a_\ell) \in Q^+$.

Substitution Systems = DFAs

There is a straightforward bijection between DFAs and constant length substitutions. Each DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ with $\Sigma = \{a_1, \ldots, a_\ell\}$ defines a length ℓ substitution on Q that maps every $q \in Q$ to the word $(q \,.\, a_1) \ldots (q \,.\, a_\ell) \in Q^+$. For instance, the automaton



induces the substitution $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 23,\ 3 \mapsto 30.$

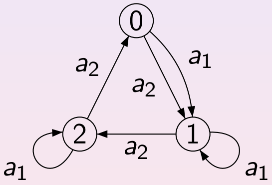
Substitution Systems = DFAs

Conversely, each substitution $\sigma : X \to X^+$ such that all words $\sigma(x)$, $x \in X$, have the same length ℓ gives rise to a DFA for which X is the state set and which has ℓ input letters $a_1, \dots, a_\ell$ acting on X as follows: $x \,.\, a_i$ is the symbol in the i-th position of the word $\sigma(x)$.

Under this bijection substitutions satisfying the coincidence condition correspond precisely to synchronizing automata, and moreover, given a substitution, the number of iterations at which the coincidence first occurs is equal to the minimum length of reset word for the corresponding automaton.

Substitution Systems = DFAs

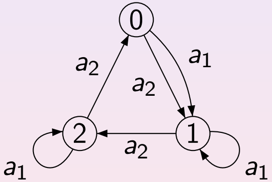
Conversely, each substitution $\sigma : X \to X^+$ such that all words $\sigma(x)$, $x \in X$, have the same length ℓ gives rise to a DFA for which X is the state set and which has ℓ input letters $a_1, \ldots, a_\ell$ acting on X as follows: $x \,.\, a_i$ is the symbol in the i-th position of the word $\sigma(x)$. For instance, the substitution τ on $X = \{0, 1, 2\}$ defined by $0 \mapsto 11$, $1 \mapsto 12$, $2 \mapsto 20$ induces the automaton:



Under this bijection substitutions satisfying the coincidence condition correspond precisely to synchronizing automata, and moreover, given a substitution, the number of iterations at which the coincidence first occurs is equal to the minimum length of reset word for the corresponding automaton.

Substitution Systems = DFAs

Conversely, each substitution $\sigma : X \to X^+$ such that all words $\sigma(x)$, $x \in X$, have the same length ℓ gives rise to a DFA for which X is the state set and which has ℓ input letters $a_1, \dots, a_\ell$ acting on X as follows: $x\,.\,a_i$ is the symbol in the i-th position of the word $\sigma(x)$. For instance, the substitution τ on $X = \{0,1,2\}$ defined by $0 \mapsto 11,\ 1 \mapsto 12,\ 2 \mapsto 20$ induces the automaton:



Under this bijection substitutions satisfying the coincidence condition correspond precisely to synchronizing automata, and moreover, given a substitution, the number of iterations at which the coincidence first occurs is equal to the minimum length of reset word for the corresponding automaton.

The Černý Series

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

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 Černý Series

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

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:

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

$$\delta(m, b) = m + 1 \pmod{n}.$$

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

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

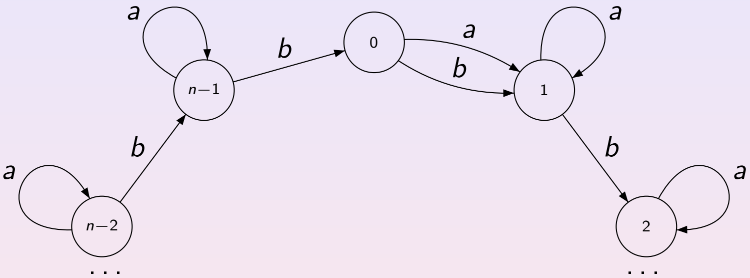
The states of $\mathscr{C}_n$ are the residues modulo n, and the input letters a and b act as follows:

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

$$\delta(m, b) = m + 1 \pmod{n}.$$

The Černý Series

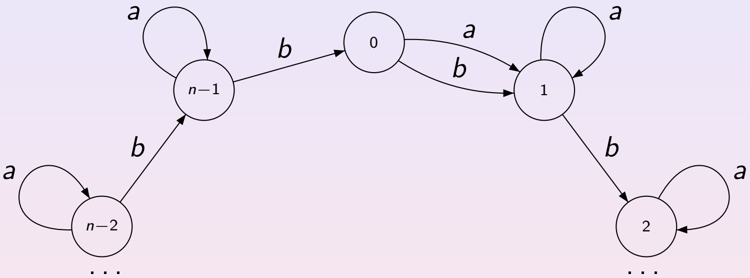
Here is a generic automaton from the Černý series:



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.

The Černý Series

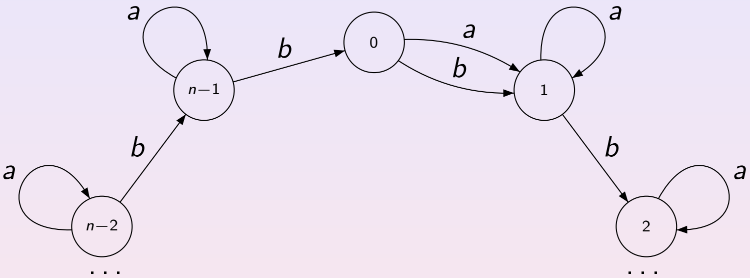
Here is a generic automaton from the Černý series:



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.

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.

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

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

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

The second inequality was proved in 1983 by J.-É. Pin.

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

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

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

The second inequality was proved in 1983 by J.-É. Pin.

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

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

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

The second inequality was proved in 1983 by J.-É. Pin.

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

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

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

The second inequality was proved in 1983 by J.-É. Pin.

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

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

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

The second inequality was proved in 1983 by J.-É. Pin.

Why so Difficult?

Why is the problem so surprisingly difficult?

One of the reasons: "slowly" synchronizing automata turn out to be extremely rare. Only one infinite series of n-state synchronizing automata with reset threshold $(n-1)^2$ is known (due to Černý, 1964), with a few (actually, 8) sporadic examples for $n \leq 6$.

Why so Difficult?

Why is the problem so surprisingly difficult?

One of the reasons: "slowly" synchronizing automata turn out to be extremely rare. Only one infinite series of n-state synchronizing automata with reset threshold $(n-1)^2$ is known (due to Černý, 1964), with a few (actually, 8) sporadic examples for $n \leq 6$.

Why so Difficult?

Why is the problem so surprisingly difficult?

One of the reasons: "slowly" synchronizing automata turn out to be extremely rare. Only one infinite series of n-state synchronizing automata with reset threshold $(n-1)^2$ is known (due to Černý, 1964), with a few (actually, 8) sporadic examples for $n \leq 6$.

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

Episode IV: A New Hope

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

Episode IV: A New Hope

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

8 states:

Reset threshold	49	48	47	46	45	44	43	42	41	40
# of automata	1	0	0	0	0	1	1	3	1	5

Episode IV: A New Hope

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

8 states:

Reset threshold	49	48	47	46	45	44	43	42	41	40
# of automata	1	0	0	0	0	1	1	3	1	5

Episode IV: A New Hope

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

8 states:

Reset threshold	49	48	47	46	45	44	43	42	41	40
# of automata	1	0	0	0	0	1	1	3	1	5

9 states:

Reset threshold	64	63	62	61	60	59	58	57	56	55
# of automata	1	0	0	0	0	0	1	2	3	0

Reset threshold	54	53	52	51
# of automata	0	0	4	4

In 2009/10, Vladimir Gusev, at that time a PhD student of mine, has performed a massive series of experiments searching exhaustively through automata with a modest number of states in order to find new examples of "slowly" synchronizing automata. The next tables present the distribution of non-isomorphic synchronizing automata with 8 and 9 states and 2 letters with respect to their reset thresholds.

8 states:

Reset threshold	49	48	47	46	45	44	43	42	41	40
# of automata	1	0	0	0	0	1	1	3	1	5

9 states:

Reset threshold	64	63	62	61	60	59	58	57	56	55
# of automata	1	0	0	0	0	0	1	2	3	0

Reset threshold	54	53	52	51
# of automata	0	0	4	4

Advantage of Being Old

Thus, the pattern is:

$(n-1)^2$ the first gap the "island" the second gap

The second gap first appears at 9 states and grows rather regularly with the number of states. The size of the island depends only on the parity of the number of states.

The very same pattern appears in the distribution of exponents of non-negative matrices.

Advantage of Being Old

Thus, the pattern is:

$(n-1)^2$ the first gap the "island" the second gap

The second gap first appears at 9 states and grows rather regularly with the number of states. The size of the island depends only on the parity of the number of states.

The very same pattern appears in the distribution of exponents of non-negative matrices.

Advantage of Being Old

Thus, the pattern is:

$(n-1)^2$ the first gap the "island" the second gap

The second gap first appears at 9 states and grows rather regularly with the number of states. The size of the island depends only on the parity of the number of states.

The very same pattern appears in the distribution of exponents of non-negative matrices.

Exponents of Non-negative Matrices

A non-negative matrix A is said to be primitive if some power A^k is positive. The minimum k with this property is called the exponent of A, denoted $\exp A$.

Helmut Wielandt proved in 1950 that for any primitive $n \times n$-matrix A, one has $\exp A \leq n^2 - 2n + 2 = (n-1)^2 + 1$, and this bound is tight. Possible exponents of $n \times n$-matrices were intensively studied in the 1960s, and it was discovered that two extreme values are each attained by a unique matrix, then there is a gap followed by an island followed by another gap. The sizes of the gaps and of the island perfectly match the sizes of the gaps and of the islands in possible reset thresholds of synchronizing automata with n states — basically one has the same picture shifted by 1. Clearly, this cannot be a mere coincidence.

Exponents of Non-negative Matrices

A non-negative matrix A is said to be primitive if some power A^k is positive. The minimum k with this property is called the exponent of A, denoted $\exp A$.

Helmut Wielandt proved in 1950 that for any primitive $n \times n$-matrix A, one has $\exp A \leq n^2 - 2n + 2 = (n-1)^2 + 1$, and this bound is tight. Possible exponents of $n \times n$-matrices were intensively studied in the 1960s, and it was discovered that two extreme values are each attained by a unique matrix, then there is a gap followed by an island followed by another gap. The sizes of the gaps and of the island perfectly match the sizes of the gaps and of the islands in possible reset thresholds of synchronizing automata with n states — basically one has the same picture shifted by 1. Clearly, this cannot be a mere coincidence.

Exponents of Non-negative Matrices

A non-negative matrix A is said to be primitive if some power A^k is positive. The minimum k with this property is called the exponent of A, denoted $\exp A$.

Helmut Wielandt proved in 1950 that for any primitive $n \times n$-matrix A, one has $\exp A \leq n^2 - 2n + 2 = (n-1)^2 + 1$, and this bound is tight. Possible exponents of $n \times n$-matrices were intensively studied in the 1960s, and it was discovered that two extreme values are each attained by a unique matrix, then there is a gap followed by an island followed by another gap. The sizes of the gaps and of the island perfectly match the sizes of the gaps and of the islands in possible reset thresholds of synchronizing automata with n states — basically one has the same picture shifted by 1. Clearly, this cannot be a mere coincidence.

Exponents of Non-negative Matrices

A non-negative matrix A is said to be primitive if some power A^k is positive. The minimum k with this property is called the exponent of A, denoted $\exp A$.

Helmut Wielandt proved in 1950 that for any primitive $n \times n$-matrix A, one has $\exp A \leq n^2 - 2n + 2 = (n-1)^2 + 1$, and this bound is tight. Possible exponents of $n \times n$-matrices were intensively studied in the 1960s, and it was discovered that two extreme values are each attained by a unique matrix, then there is a gap followed by an island followed by another gap. The sizes of the gaps and of the island perfectly match the sizes of the gaps and of the islands in possible reset thresholds of synchronizing automata with n states — basically one has the same picture shifted by 1. Clearly, this cannot be a mere coincidence.

Exponents of Non-negative Matrices

A non-negative matrix A is said to be primitive if some power A^k is positive. The minimum k with this property is called the exponent of A, denoted $\exp A$.

Helmut Wielandt proved in 1950 that for any primitive $n \times n$-matrix A, one has $\exp A \leq n^2 - 2n + 2 = (n-1)^2 + 1$, and this bound is tight. Possible exponents of $n \times n$-matrices were intensively studied in the 1960s, and it was discovered that two extreme values are each attained by a unique matrix, then there is a gap followed by an island followed by another gap. The sizes of the gaps and of the island perfectly match the sizes of the gaps and of the islands in possible reset thresholds of synchronizing automata with n states — basically one has the same picture shifted by 1. Clearly, this cannot be a mere coincidence.

Digraphs and Matrices

A directed graph (digraph) is a pair $D = \langle V, E \rangle$.

- V set of vertices
- $E \subseteq V \times V$ set of edges

This definition allows loops but excludes multiple edges.

The matrix of a digraph $D = \langle V, E \rangle$ is just the incidence matrix of the edge relation, that is, a $V \times V$-matrix whose entry in the row v and the column v' is 1 if $(v, v') \in E$ and 0 otherwise.

A directed graph (digraph) is a pair $D = \langle V, E \rangle$.

- V set of vertices
- $E \subseteq V \times V$ set of edges

This definition allows loops but excludes multiple edges.

The matrix of a digraph $D = \langle V, E \rangle$ is just the incidence matrix of the edge relation, that is, a $V \times V$-matrix whose entry in the row v and the column v' is 1 if $(v, v') \in E$ and 0 otherwise.

Digraphs and Matrices

A directed graph (digraph) is a pair $D = \langle V, E \rangle$.

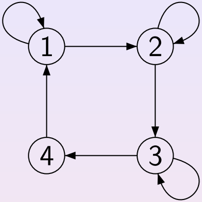
- V set of vertices
- $E \subseteq V \times V$ set of edges

This definition allows loops but excludes multiple edges.

The matrix of a digraph $D = \langle V, E \rangle$ is just the incidence matrix of the edge relation, that is, a $V \times V$-matrix whose entry in the row v and the column v' is 1 if $(v, v') \in E$ and 0 otherwise.

Digraphs and Matrices

For instance, the matrix of the digraph

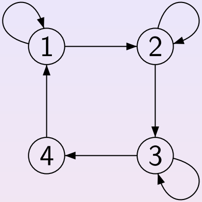


(with respect to the chosen numbering of its vertices) is $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$.

Conversely, given an $n \times n$-matrix $P = (p_{ij})$ with non-negative real entries, we assign to it a digraph $D(P)$ on the set $\{1, 2, \ldots, n\}$ as follows: (i, j) is an edge of $D(P)$ if and only if $p_{ij} > 0$.

This 'two-way' correspondence allows us to reformulate in terms of digraphs several important notions and results which originated in the classical Perron–Frobenius theory of non-negative matrices.

For instance, the matrix of the digraph



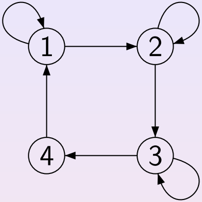
(with respect to the chosen numbering of its vertices) is $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$.

Conversely, given an $n \times n$-matrix $P = (p_{ij})$ with non-negative real entries, we assign to it a digraph $D(P)$ on the set $\{1, 2, \ldots, n\}$ as follows: (i, j) is an edge of $D(P)$ if and only if $p_{ij} > 0$.

This 'two-way' correspondence allows us to reformulate in terms of digraphs several important notions and results which originated in the classical Perron–Frobenius theory of non-negative matrices.

Digraphs and Matrices

For instance, the matrix of the digraph



(with respect to the chosen numbering of its vertices) is $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$.

Conversely, given an $n \times n$-matrix $P = (p_{ij})$ with non-negative real entries, we assign to it a digraph $D(P)$ on the set $\{1, 2, \ldots, n\}$ as follows: (i, j) is an edge of $D(P)$ if and only if $p_{ij} > 0$.

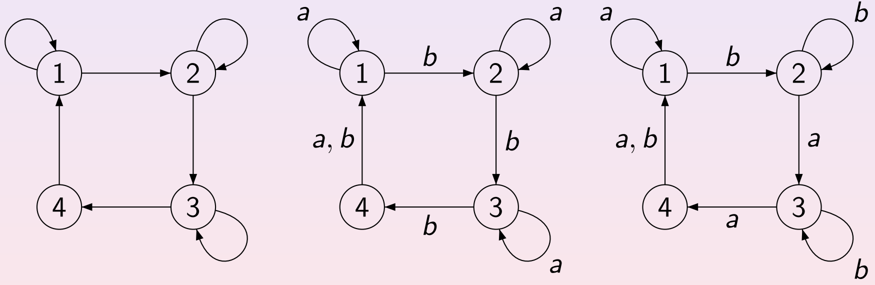
This 'two-way' correspondence allows us to reformulate in terms of digraphs several important notions and results which originated in the classical Perron–Frobenius theory of non-negative matrices.

Digraphs and Colorings

By a coloring of a digraph we mean assigning labels from an alphabet Σ to edges such that the digraph labeled this way becomes a DFA.

Digraphs and Colorings

By a coloring of a digraph we mean assigning labels from an alphabet Σ to edges such that the digraph labeled this way becomes a DFA.



A digraph D is primitive if D is strongly connected and the greatest common divisor of the lengths of all cycles in D is equal to 1.

Adler, Goodwyn, Weiss (Equivalence of topological Markov shifts, Israel J. Math. 27 (1977) 49–63):
Underlying digraphs of strongly connected synchronizing automata are primitive.

The Road Coloring Conjecture: Every primitive digraph admits a synchronizing coloring.

This was confirmed by Avraham Trahtman (The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Primitive Digraphs

A digraph D is primitive if D is strongly connected and the greatest common divisor of the lengths of all cycles in D is equal to 1.

Adler, Goodwyn, Weiss (Equivalence of topological Markov shifts, Israel J. Math. 27 (1977) 49–63):
Underlying digraphs of strongly connected synchronizing automata are primitive.

The Road Coloring Conjecture: Every primitive digraph admits a synchronizing coloring.

This was confirmed by Avraham Trahtman (The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

A digraph D is primitive if D is strongly connected and the greatest common divisor of the lengths of all cycles in D is equal to 1.

Adler, Goodwyn, Weiss (Equivalence of topological Markov shifts, Israel J. Math. 27 (1977) 49–63):
Underlying digraphs of strongly connected synchronizing automata are primitive.

The Road Coloring Conjecture: Every primitive digraph admits a synchronizing coloring.

This was confirmed by Avraham Trahtman (The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Primitive Digraphs

A digraph D is primitive if D is strongly connected and the greatest common divisor of the lengths of all cycles in D is equal to 1.

Adler, Goodwyn, Weiss (Equivalence of topological Markov shifts, Israel J. Math. 27 (1977) 49–63):
Underlying digraphs of strongly connected synchronizing automata are primitive.

The Road Coloring Conjecture: Every primitive digraph admits a synchronizing coloring.

This was confirmed by Avraham Trahtman (The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.
If $n > 4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6 < \gamma(D) < (n-1)^2$.
If $n > 3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4 < \gamma(D) < (n-1)^2$, or
$n^2-4n+6 < \gamma(D) < n^2-3n+2$.

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.
If $n > 4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6 < \gamma(D) < (n-1)^2$.
If $n > 3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4 < \gamma(D) < (n-1)^2$, or
$n^2-4n+6 < \gamma(D) < n^2-3n+2$.

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.
If $n > 4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6 < \gamma(D) < (n-1)^2$.
If $n > 3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4 < \gamma(D) < (n-1)^2$, or
$n^2-4n+6 < \gamma(D) < n^2-3n+2$.

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.
If $n>4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6<\gamma(D)<(n-1)^2$.
If $n>3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4<\gamma(D)<(n-1)^2$, or
$n^2-4n+6<\gamma(D)<n^2-3n+2$.

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.

If $n>4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6<\gamma(D)<(n-1)^2$.

If $n>3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4<\gamma(D)<(n-1)^2$, or $n^2-4n+6<\gamma(D)<n^2-3n+2$.

Exponents

A digraph D is primitive iff there exists t such that for each pair of vertices there exists a path between them of length exactly t. (This is equivalent to saying that the t-th power of the matrix of D is positive.) The least t with this property is called the exponent of the digraph D and is denoted by $\gamma(D)$.

1950, Wielandt: The exponent of every primitive digraph on n vertices is not greater than $(n-1)^2+1$ and this bound is tight.

1964, Dulmage–Mendelsohn: There is exactly one primitive digraph on n vertices with exponent $(n-1)^2+1$ and exactly one primitive digraph on n vertices with exponent $(n-1)^2$.
If $n>4$ is even, then there is no primitive digraph D on n vertices such that $n^2-4n+6<\gamma(D)<(n-1)^2$.
If $n>3$ is odd, then there is no primitive digraph D on n vertices such that $n^2-3n+4<\gamma(D)<(n-1)^2$, or $n^2-4n+6<\gamma(D)<n^2-3n+2$.

Exponents vs Reset Lengths

Exponents of primitive digraphs with 9 vertices vs reset thresholds of 2-letter strongly connected synchronizing automata with 9 states

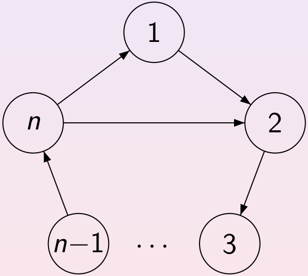
N	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51
# of primitive digraphs with exponent N	1	1	0	0	0	0	0	1	1	2	0	0	0	0	3
# of 2-letter synchronizing automata with reset threshold N	0	1	0	0	0	0	0	1	2	3	0	0	0	4	4

The Wielandt Automaton

The Wielandt automaton $\mathscr{W}_n$ is a (unique) coloring of the Wielandt digraph W_n with $\gamma(W_n) = (n-1)^2 + 1$. The Wielandt digraph has n vertices $1, 2, \ldots, n$, say, and the following $n+1$ edges: $(i, i+1)$ for $i = 1, \ldots, n-1$, $(n, 1)$, and $(n, 2)$.

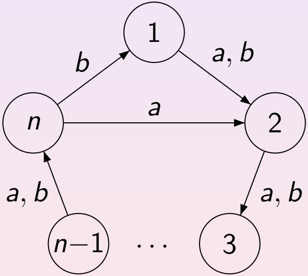
The Wielandt Automaton

The Wielandt automaton $\mathscr{W}_n$ is a (unique) coloring of the Wielandt digraph W_n with $\gamma(W_n) = (n-1)^2 + 1$. The Wielandt digraph has n vertices $1, 2, \ldots, n$, say, and the following $n+1$ edges: $(i, i+1)$ for $i = 1, \ldots, n-1$, $(n, 1)$, and $(n, 2)$.



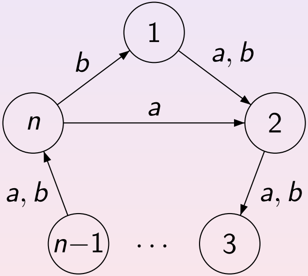
The Wielandt Automaton

The Wielandt automaton $\mathscr{W}_n$ is a (unique) coloring of the Wielandt digraph W_n with $\gamma(W_n) = (n-1)^2 + 1$. The Wielandt digraph has n vertices $1, 2, \ldots, n$, say, and the following $n+1$ edges: $(i, i+1)$ for $i = 1, \ldots, n-1$, $(n, 1)$, and $(n, 2)$.



The Wielandt Automaton

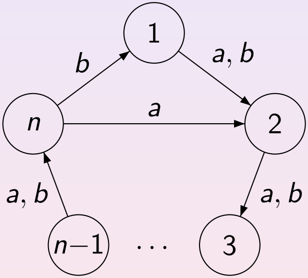
The Wielandt automaton $\mathscr{W}_n$ is a (unique) coloring of the Wielandt digraph W_n with $\gamma(W_n) = (n-1)^2 + 1$. The Wielandt digraph has n vertices $1, 2, \dots, n$, say, and the following $n + 1$ edges: $(i, i + 1)$ for $i = 1, \dots, n - 1$, $(n, 1)$, and $(n, 2)$.



It is easy to show that the reset threshold of $\mathscr{W}_n$ is $n^2 - 3n + 3$.

The Wielandt Automaton

The Wielandt automaton $\mathscr{W}_n$ is a (unique) coloring of the Wielandt digraph W_n with $\gamma(W_n) = (n-1)^2 + 1$. The Wielandt digraph has n vertices $1, 2, \ldots, n$, say, and the following $n+1$ edges: $(i, i+1)$ for $i = 1, \ldots, n-1$, $(n, 1)$, and $(n, 2)$.



It is easy to show that the reset threshold of $\mathscr{W}_n$ is $n^2 - 3n + 3$.

In a similar way, each digraph with large exponent generates slowly synchronizing automata.

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

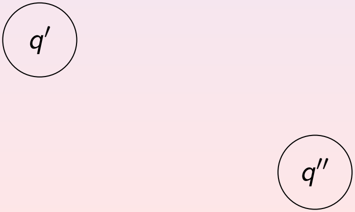
$$\gamma(D) \leq t + n - 1.$$

Colorings of Digraphs with Large Exponents

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

$$\gamma(D) \leq t + n - 1.$$



Colorings of Digraphs with Large Exponents

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

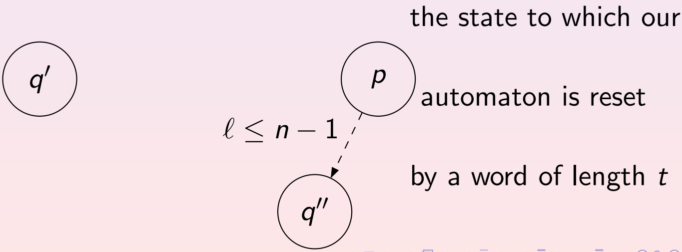
$$\gamma(D) \leq t + n - 1.$$

the state to which our automaton is reset by a word of length t

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

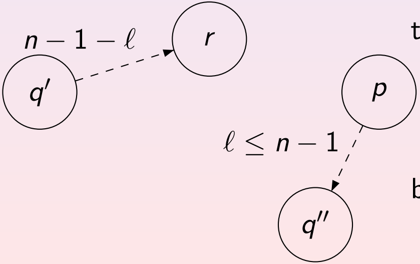
$$\gamma(D) \leq t + n - 1.$$



Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

$$\gamma(D) \leq t + n - 1.$$

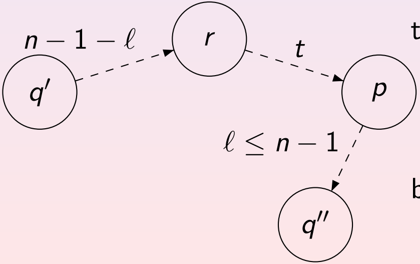


the state to which our automaton is reset by a word of length t

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

$$\gamma(D) \leq t + n - 1.$$



the state to which our automaton is reset by a word of length t

Colorings of Digraphs with Large Exponents

Observation

Let a strongly connected synchronizing automaton with n states and reset threshold t be a coloring of a digraph D. Then

$$\gamma(D) \leq t + n - 1.$$

For instance, the reset threshold t of the Wielandt automaton $\mathscr{W}_n$ must satisfy

$$t \geq \gamma(W_n) - n + 1 = (n-1)^2 + 1 - n + 1 = n^2 - 3n + 3,$$

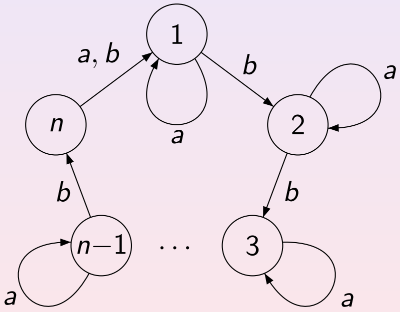
and it is easy to find a reset word of length $n^2 - 3n + 3$.

The Černý Automaton

There are slowly synchronizing automata that cannot be obtained as colorings of a digraph with large exponent. For instance, the Černý automaton $\mathscr{C}_n$ has reset threshold $(n-1)^2$ while its underlying digraph has exponent $n-1$.

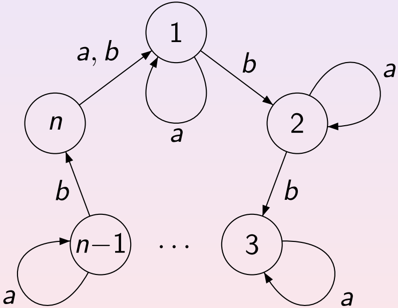
The Černý Automaton

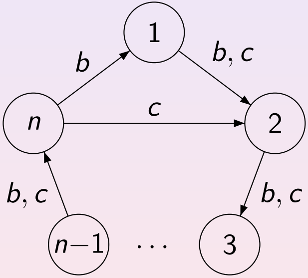
There are slowly synchronizing automata that cannot be obtained as colorings of a digraph with large exponent. For instance, the Černý automaton $\mathscr{C}_n$ has reset threshold $(n-1)^2$ while its underlying digraph has exponent $n-1$.



The Černý Automaton

There are slowly synchronizing automata that cannot be obtained as colorings of a digraph with large exponent. For instance, the Černý automaton $\mathscr{C}_n$ has reset threshold $(n-1)^2$ while its underlying digraph has exponent $n-1$.





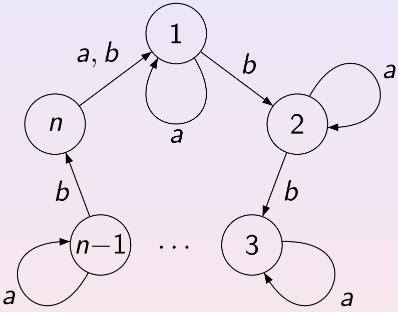
However, $\mathscr{C}_n$ becomes $\mathscr{W}_n$ under the action of b and $c = ab$.

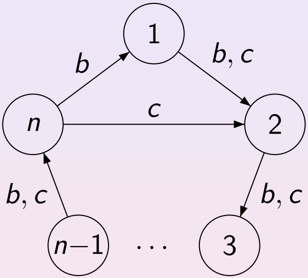
The Černý Automaton

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b. Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.

The Černý Automaton

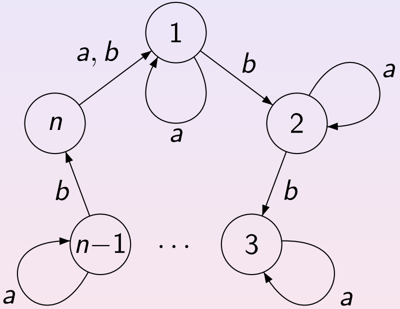
Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b.

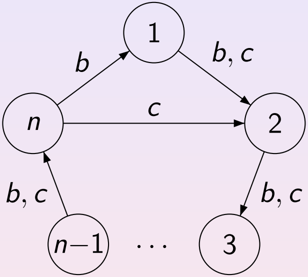




Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b.

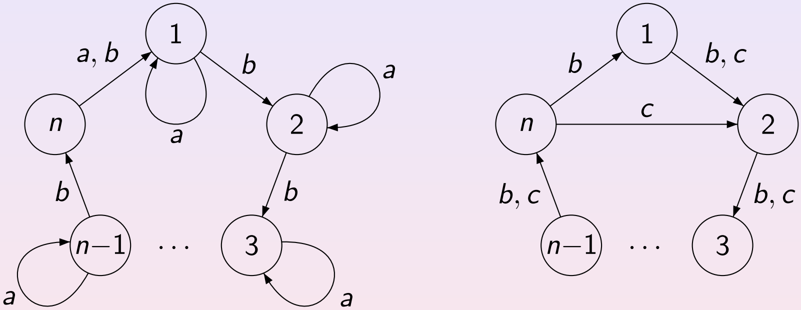




Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.

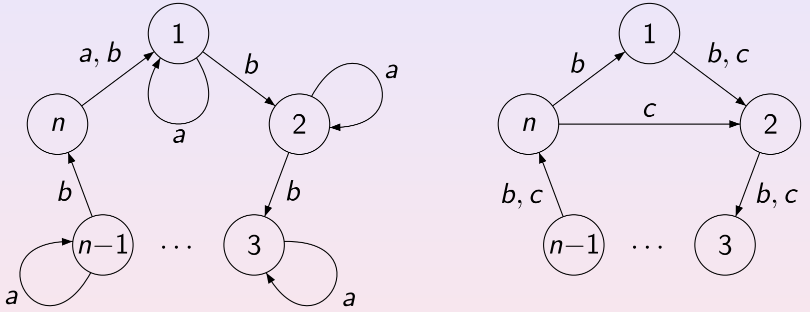
The Černý Automaton

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b.



Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b.



Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b. Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.
Further, v contains at least $n - 2$ occurrences of c. Since each occurrence of c in v corresponds to an occurrence of ab in w', we conclude that $|w'| \geq n^2 - 3n + 2 + n - 2 = n^2 - 2n$.

The Černý Automaton

Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b. Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.
Further, v contains at least $n-2$ occurrences of c. Since each occurrence of c in v corresponds to an occurrence of ab in w', we conclude that $|w'| \geq n^2 - 3n + 2 + n - 2 = n^2 - 2n$. Thus, $|w| = |w'a| \geq n^2 - 2n + 1 = (n-1)^2$.

The Černý Automaton

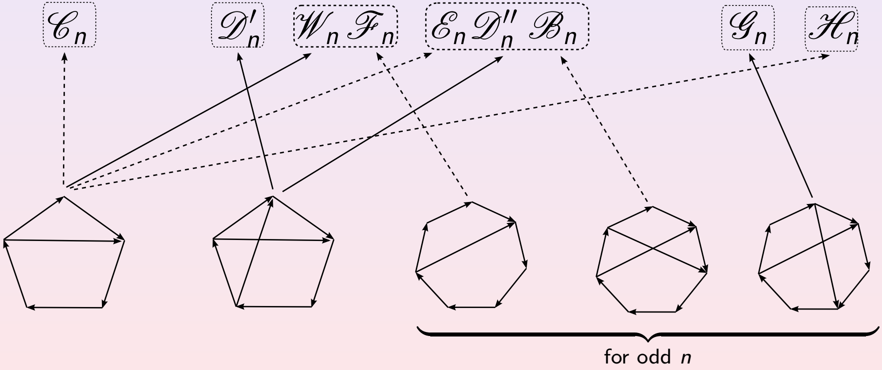
Let w be a shortest reset word for $\mathscr{C}_n$. It must end with a and every other occurrence of a in w is followed by an occurrence of b. Thus, $w = w'a$ where w' can be rewritten into a word v over the alphabet $\{b, c\}$. Since w' and v act in the same way, the word vc is a reset word for $\mathscr{W}_n$. Hence $|v| \geq n^2 - 3n + 2$.
Further, v contains at least $n - 2$ occurrences of c. Since each occurrence of c in v corresponds to an occurrence of ab in w', we conclude that $|w'| \geq n^2 - 3n + 2 + n - 2 = n^2 - 2n$. Thus, $|w| = |w'a| \geq n^2 - 2n + 1 = (n-1)^2$.

Thus, it is the Wielandt digraph that stays behind the Černý automaton!

In a similar manner it is easy to recover every known slowly synchronizing automaton from a suitable digraph with large exponent.

Digraphs vs Automata

In a similar manner it is easy to recover every known slowly synchronizing automaton from a suitable digraph with large exponent.



New Conjectures

(a) (The Černý conjecture) The reset threshold of every synchronizing n-automaton does not exceed $(n-1)^2$.

(b) If $n > 6$, then there exists exactly one n-automaton with reset threshold $(n-1)^2$, namely, $\mathscr{C}_n$.

(c) If $n > 6$, then there exists no n-automaton whose reset threshold is greater than $n^2 - 3n + 4$ but less than $(n-1)^2$.

(d) If $n > 7$ is odd, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly two n-automata with reset threshold $n^2 - 3n + 3$, and exactly three n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 8$ and $n^2 - 3n + 1$.

(e) If $n > 8$ is even, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly one n-automaton with reset threshold $n^2 - 3n + 3$, and exactly two n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 7$ and $n^2 - 3n + 1$.

New Conjectures

(a) (The Černý conjecture) The reset threshold of every synchronizing n-automaton does not exceed $(n-1)^2$.

(b) If $n > 6$, then there exists exactly one n-automaton with reset threshold $(n-1)^2$, namely, $\mathscr{C}_n$.

(c) If $n > 6$, then there exists no n-automaton whose reset threshold is greater than $n^2 - 3n + 4$ but less than $(n-1)^2$.

(d) If $n > 7$ is odd, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly two n-automata with reset threshold $n^2 - 3n + 3$, and exactly three n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 8$ and $n^2 - 3n + 1$.

(e) If $n > 8$ is even, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly one n-automaton with reset threshold $n^2 - 3n + 3$, and exactly two n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 7$ and $n^2 - 3n + 1$.

New Conjectures

(a) (The Černý conjecture) The reset threshold of every synchronizing n-automaton does not exceed $(n-1)^2$.
(b) If $n > 6$, then there exists exactly one n-automaton with reset threshold $(n-1)^2$, namely, $\mathscr{C}_n$.
(c) If $n > 6$, then there exists no n-automaton whose reset threshold is greater than $n^2 - 3n + 4$ but less than $(n-1)^2$.
(d) If $n > 7$ is odd, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly two n-automata with reset threshold $n^2 - 3n + 3$, and exactly three n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 8$ and $n^2 - 3n + 1$.
(e) If $n > 8$ is even, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly one n-automaton with reset threshold $n^2 - 3n + 3$, and exactly two n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 7$ and $n^2 - 3n + 1$.

New Conjectures

(a) (The Černý conjecture) The reset threshold of every synchronizing n-automaton does not exceed $(n-1)^2$.
(b) If $n > 6$, then there exists exactly one n-automaton with reset threshold $(n-1)^2$, namely, $\mathscr{C}_n$.
(c) If $n > 6$, then there exists no n-automaton whose reset threshold is greater than $n^2 - 3n + 4$ but less than $(n-1)^2$.
(d) If $n > 7$ is odd, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly two n-automata with reset threshold $n^2 - 3n + 3$, and exactly three n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 8$ and $n^2 - 3n + 1$.
(e) If $n > 8$ is even, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly one n-automaton with reset threshold $n^2 - 3n + 3$, and exactly two n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 7$ and $n^2 - 3n + 1$.

New Conjectures

(a) (The Černý conjecture) The reset threshold of every synchronizing n-automaton does not exceed $(n-1)^2$.
(b) If $n > 6$, then there exists exactly one n-automaton with reset threshold $(n-1)^2$, namely, $\mathscr{C}_n$.
(c) If $n > 6$, then there exists no n-automaton whose reset threshold is greater than $n^2 - 3n + 4$ but less than $(n-1)^2$.
(d) If $n > 7$ is odd, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly two n-automata with reset threshold $n^2 - 3n + 3$, and exactly three n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 8$ and $n^2 - 3n + 1$.
(e) If $n > 8$ is even, then there exists exactly one n-automaton with reset threshold $n^2 - 3n + 4$, exactly one n-automaton with reset threshold $n^2 - 3n + 3$, and exactly two n-automata with reset threshold $n^2 - 3n + 2$. There exists no n-automaton whose reset threshold is between $n^2 - 4n + 7$ and $n^2 - 3n + 1$.