Synchronizing Finite Automata

Lecture VII. Aperiodic Automata

Mikhail Volkov

Ural Federal University / Hunter College

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

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

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ whose action resets $\mathscr{A}$, that is, leaves the automaton in one particular state no matter which state in Q it started at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$.

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

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

1. Recap

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

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

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ whose action resets $\mathscr{A}$, that is, leaves the automaton in one particular state no matter which state in Q it started at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$.

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

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

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

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

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ whose action resets $\mathscr{A}$, that is, leaves the automaton in one particular state no matter which state in Q it started at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$.

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

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

1. Recap

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

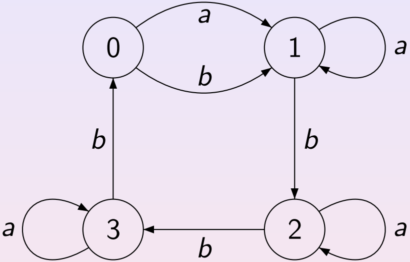
- Q the state set
- Σ the input alphabet
- $\delta : Q \times \Sigma \to Q$ the transition function

$\mathscr{A}$ is called synchronizing if there exists a word $w \in \Sigma^*$ whose action resets $\mathscr{A}$, that is, leaves the automaton in one particular state no matter which state in Q it started at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$.

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

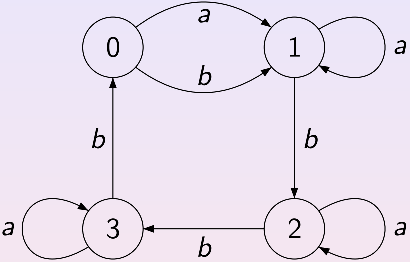
Any w with this property is a reset word for $\mathscr{A}$.

2. Example



A reset word is *abbbabbba*. In fact, we have verified that this is the shortest reset word for this automaton; that is, its reset threshold is 9.

2. Example



A reset word is *abbbabbba*. In fact, we have verified that this is the shortest reset word for this automaton; that is, its reset threshold is 9.

3. Černý Conjecture

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

3. Černý Conjecture

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

3. Černý Conjecture

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

3. Černý Conjecture

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

$$\text{(Černý, 1964)}\quad (n-1)^2 \leq C(n)$$

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

3. Černý Conjecture

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

$$\text{(Černý, 1964)}\quad (n-1)^2 \leq C(n) \leq \frac{n^3-n}{6}\quad \text{(Pin–Frankl, 1983)}.$$

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

The Černý conjecture is the claim that every synchronizing automaton with n states possesses a reset word of length $(n-1)^2$. The validity of the conjecture is main open problem of the area.

Define the *Černý function* $C(n)$ as the maximum reset threshold of all synchronizing automata with n states. In terms of this function, our current knowledge can be summarized in one line:

$$\text{(Černý, 1964)}\quad (n-1)^2 \leq C(n) \leq \frac{n^3-n}{6} \quad \text{(Pin–Frankl, 1983)}.$$

The Černý conjecture thus claims that in fact $C(n) = (n-1)^2$.

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

In this lecture, we encounter a restriction of a different nature: aperiodicity.

4. Restriction to Subclasses

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

In this lecture, we encounter a restriction of a different nature: aperiodicity.

4. Restriction to Subclasses

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

- Automata with a circular letter (Dubuc);

In this lecture, we encounter a restriction of a different nature: aperiodicity.

4. Restriction to Subclasses

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

- Automata with a circular letter (Dubuc);
- One-cluster automata (Béal, Berlinkov, Perrin, Steinberg);

In this lecture, we encounter a restriction of a different nature: aperiodicity.

4. Restriction to Subclasses

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

- Automata with a circular letter (Dubuc);
- One-cluster automata (Béal, Berlinkov, Perrin, Steinberg);
- Eulerian automata (Kari), see Lecture V;

In this lecture, we encounter a restriction of a different nature: aperiodicity.

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

- Automata with a circular letter (Dubuc);
- One-cluster automata (Béal, Berlinkov, Perrin, Steinberg);
- Eulerian automata (Kari), see Lecture V;
- Automata with a sink (Rystsov), see Lecture VI.

In this lecture, we encounter a restriction of a different nature: aperiodicity.

4. Restriction to Subclasses

Since the Černý problem has proved to be hard in general, a natural approach is to consider its restrictions to various special classes of automata.

There are several classes in which synchronizing automata have been investigated with (at least partial) success:

- Automata with a circular letter (Dubuc);
- One-cluster automata (Béal, Berlinkov, Perrin, Steinberg);
- Eulerian automata (Kari), see Lecture V;
- Automata with a sink (Rystsov), see Lecture VI.

In this lecture, we encounter a restriction of a different nature: aperiodicity.

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w) \colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w) \colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.

5. Aperiodic Automata

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w) \colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w)\colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.

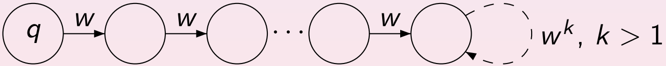
5. Aperiodic Automata

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w)\colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.



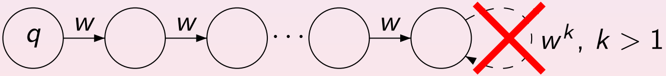
5. Aperiodic Automata

The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w)\colon Q \to Q$ induced by words $w \in \Sigma^*$.

A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.

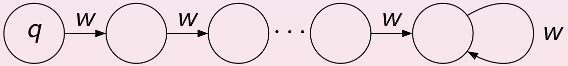


The transition monoid of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ consists of all transformations $\delta(\sqcup, w) \colon Q \to Q$ induced by words $w \in \Sigma^*$.

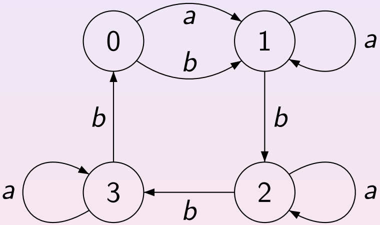
A monoid is said to be aperiodic if all its subgroups are singletons.

A DFA is called aperiodic (or counter-free) if its transition monoid is aperiodic.

An equivalent 'elementary' formulation: $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic iff for every $q \in Q$ and every $w \in \Sigma^*$ there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$.



The Černý automaton $\mathscr{C}_4$ is not aperiodic since the letter b acts as a cyclic permutation of the states and thus generates a 4-element subgroup in the transition monoid of $\mathscr{C}_4$.

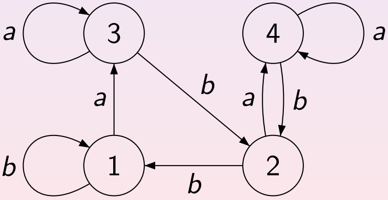


The following automaton is aperiodic:

6. Examples

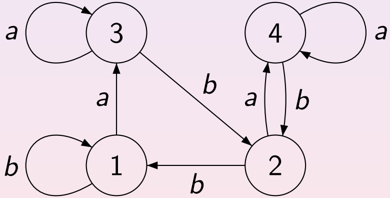
The Černý automaton $\mathscr{C}_4$ is not aperiodic since the letter b acts as a cyclic permutation of the states and thus generates a 4-element subgroup in the transition monoid of $\mathscr{C}_4$.

The following automaton is aperiodic:



The Černý automaton $\mathscr{C}_4$ is not aperiodic since the letter b acts as a cyclic permutation of the states and thus generates a 4-element subgroup in the transition monoid of $\mathscr{C}_4$.

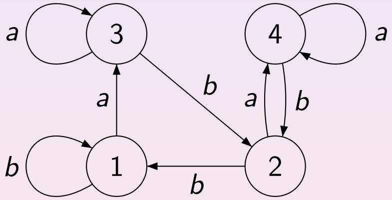
The following automaton is aperiodic:



	1	2	3	4
a	3	4	3	4
b	1	1	2	2

The Černý automaton $\mathscr{C}_4$ is not aperiodic since the letter b acts as a cyclic permutation of the states and thus generates a 4-element subgroup in the transition monoid of $\mathscr{C}_4$.

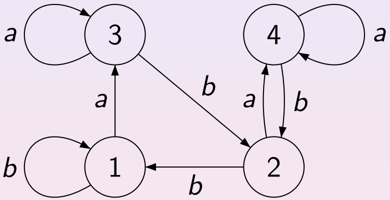
The following automaton is aperiodic:



	1	2	3	4
a	3	4	3	4
b	1	1	2	2
ab	2	2	2	2
ba	3	3	4	4
b^2	1	1	1	1

$a^2 = a$

The Černý automaton $\mathscr{C}_4$ is not aperiodic since the letter b acts as a cyclic permutation of the states and thus generates a 4-element subgroup in the transition monoid of $\mathscr{C}_4$.
The following automaton is aperiodic:



	1	2	3	4
a	3	4	3	4
b	1	1	2	2
ab	2	2	2	2
ba	3	3	4	4
b^2	1	1	1	1
aba	4	4	4	4
b^2a	3	3	3	3

$a^2 = a,\ ab^2 = b^2,\ bab = ab,\ b^3 = b^2$

7. Complexity

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to SHORT-RESET-WORD shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to SHORT-RESET-WORD shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to SHORT-RESET-WORD shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

7. Complexity

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to SHORT-RESET-WORD shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

7. Complexity

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to Short-Reset-Word shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

7. Complexity

In general, there is no way to verify whether or not a given DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is aperiodic avoiding the calculation of its transition monoid and the cardinality of the monoid can reach $|Q|^{|Q|}$. The problem is known to be PSPACE-complete (Sang Cho and Dung T. Huynh, "Finite-automaton aperiodicity is PSPACE-complete", Theor. Comput. Sci. 88 (1991) 99–116).

Also, the synchronization issues remain difficult when restricted to the class of aperiodic automata. Indeed, inspecting the reduction from SAT to $\textsc{Short-Reset-Word}$ shown in Lecture III, one can see that the construction results in an aperiodic automaton, and therefore, the question of whether or not a given aperiodic automaton admits a reset word whose length does not exceed a given positive integer is NP-complete.

8. Monotonicity

In some cases, however, aperiodicity is granted.

A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

Monotonic automata are aperiodic (known and easy).

8. Monotonicity

In some cases, however, aperiodicity is granted.

A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\_, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

Monotonic automata are aperiodic (known and easy).

8. Monotonicity

In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

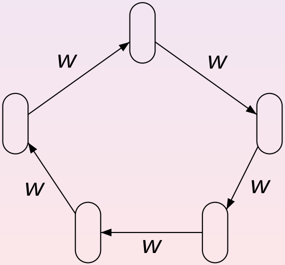
Monotonic automata are aperiodic (known and easy).

8. Monotonicity

In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

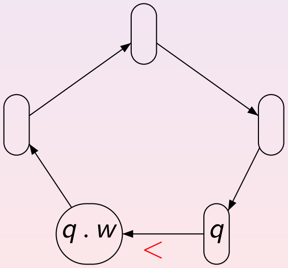
Monotonic automata are aperiodic (known and easy).



In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

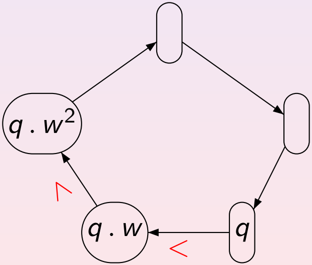
Monotonic automata are aperiodic (known and easy).



In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

Monotonic automata are aperiodic (known and easy).

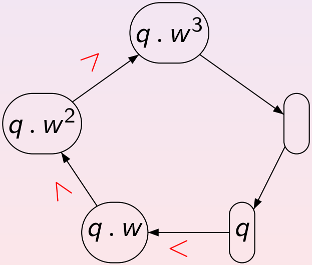


8. Monotonicity

In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\_, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

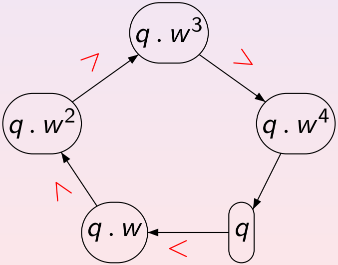
Monotonic automata are aperiodic (known and easy).



In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

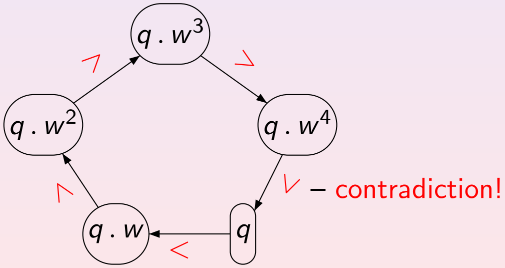
Monotonic automata are aperiodic (known and easy).



In some cases, however, aperiodicity is granted.
A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is monotonic if Q admits a linear order $\leq$ such that, for each $a \in \Sigma$, the transformation $\delta(\sqcup, a)$ preserves $\leq$:

$$p \leq q \Rightarrow \delta(p, a) \leq \delta(q, a).$$

Monotonic automata are aperiodic (known and easy).



9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \dots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is stable if $(p, q) \in \rho$ implies $\big(\delta(p, a), \delta(q, a)\big) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\bigl(\delta(p, a), \delta(q, a)\bigr) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

9. Generalized Monotonicity

A binary relation ρ on the state set of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is stable if $(p, q) \in \rho$ implies $\bigl(\delta(p, a), \delta(q, a)\bigr) \in \rho$ for all $p, q \in Q$ and $a \in \Sigma$.

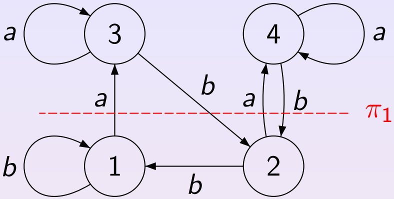
We call a DFA $\mathscr{A}$ generalized monotonic of level ℓ if it admits a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$, satisfying the following conditions:

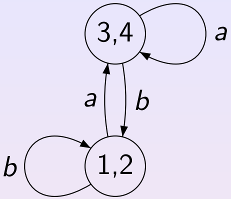
- ρ_0 is the equality;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a linear order on each π_i/π_{i-1}-class;
- π_ℓ is the universal relation.

Monotonic automata are precisely generalized monotonic automata of level 1.

The aperiodic automaton in our example is a generalized monotonic automaton of level 2.

10. Example Revisited

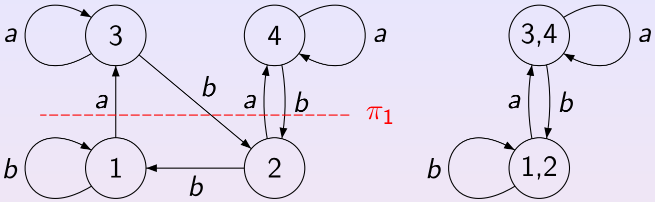




Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1,2\} <_2 \{3,4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots,$ and every generalized monotonic automaton is aperiodic.

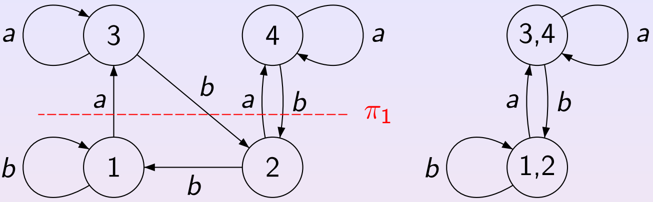
10. Example Revisited



Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1,2\} <_2 \{3,4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots$, and every generalized monotonic automaton is aperiodic.

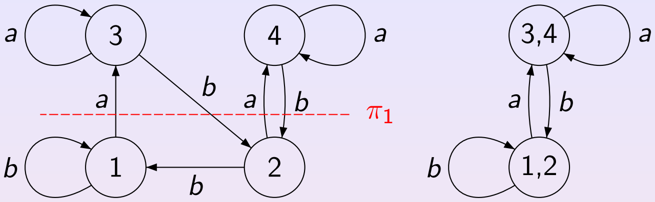
10. Example Revisited



Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1, 2\} <_2 \{3, 4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

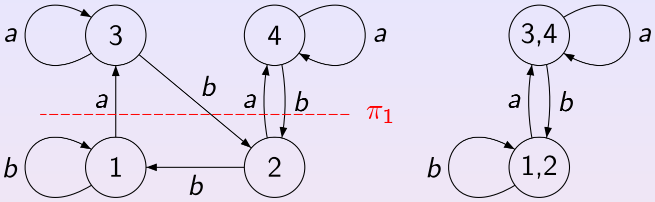
In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots$, and every generalized monotonic automaton is aperiodic.

10. Example Revisited



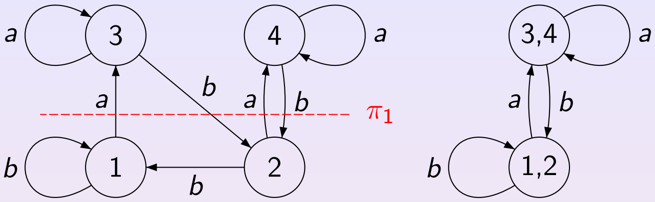
Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1,2\} <_2 \{3,4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots,$ and every generalized monotonic automaton is aperiodic.



Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1, 2\} <_2 \{3, 4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots,$ and every generalized monotonic automaton is aperiodic.



Endowing Q with the order $\leq_1$ such that $1 <_1 2$ and $3 <_1 4$, we get a linear order on each π_1-class. If we order Q/π_1 by letting $\{1, 2\} <_2 \{3, 4\}$, the quotient automaton becomes monotonic. It can be shown that the automaton is not monotonic. Moreover, it cannot be emulated by any monotonic automaton.

In fact, the hierarchy of generalized monotonic automata is strict: there are automata of each level $\ell = 1, 2, \ldots,$ and every generalized monotonic automaton is aperiodic.

The importance of aperiodic automata was understood at the beginning of 1960s. As usual, it has 3 sources and 3 components:

It is remarkable that each of these directions has led to a major open problem, and the 3 problems play nowadays a central role in the theory of finite automata.

11. Why Aperiodic Automata?

The importance of aperiodic automata was understood at the beginning of 1960s. As usual, it has 3 sources and 3 components:

- Star-free regular expressions (Schützenberger);

It is remarkable that each of these directions has led to a major open problem, and the 3 problems play nowadays a central role in the theory of finite automata.

11. Why Aperiodic Automata?

The importance of aperiodic automata was understood at the beginning of 1960s. As usual, it has 3 sources and 3 components:

- Star-free regular expressions (Schützenberger);
- Krohn–Rhodes decompositions of finite automata;

It is remarkable that each of these directions has led to a major open problem, and the 3 problems play nowadays a central role in the theory of finite automata.

The importance of aperiodic automata was understood at the beginning of 1960s. As usual, it has 3 sources and 3 components:

- Star-free regular expressions (Schützenberger);
- Krohn–Rhodes decompositions of finite automata;
- Logical characterizations of regular languages (McNaughton).

It is remarkable that each of these directions has led to a major open problem, and the 3 problems play nowadays a central role in the theory of finite automata.

11. Why Aperiodic Automata?

The importance of aperiodic automata was understood at the beginning of 1960s. As usual, it has 3 sources and 3 components:

- Star-free regular expressions (Schützenberger);
- Krohn–Rhodes decompositions of finite automata;
- Logical characterizations of regular languages (McNaughton).

It is remarkable that each of these directions has led to a major open problem, and the 3 problems play nowadays a central role in the theory of finite automata.

By Kleene's theorem every regular language can be described by a regular expression, say, $((a+ba)^*ab)^*(b+aa)^*$. Here words denote corresponding singleton languages, $+$ stands for union, concatenation means product and * is the Kleene star (iteration).

The Kleene star is clearly the most 'infinite' operation. One cannot eliminate it because neither union nor product can produce infinite languages from finite ones. However, one can use also complement (the class of regular languages is closed under complement by Kleene's theorem). An extended regular expression is built from words by using union, product, Kleene star, and complement, say, $((a+ba)^C ab)^*(b+(aa)^C)^*$.

The complement of a finite language is infinite. Can one get rid of the Kleene star in this setting?

By Kleene's theorem every regular language can be described by a regular expression, say, $((a+ba)^*ab)^*(b+aa)^*$. Here words denote corresponding singleton languages, $+$ stands for union, concatenation means product and * is the Kleene star (iteration).

The Kleene star is clearly the most 'infinite' operation. One cannot eliminate it because neither union nor product can produce infinite languages from finite ones. However, one can use also complement (the class of regular languages is closed under complement by Kleene's theorem). An extended regular expression is built from words by using union, product, Kleene star, and complement, say, $((a+ba)^C ab)^*(b+(aa)^C)^*$.

The complement of a finite language is infinite. Can one get rid of the Kleene star in this setting?

By Kleene's theorem every regular language can be described by a regular expression, say, $((a+ba)^*ab)^*(b+aa)^*$. Here words denote corresponding singleton languages, $+$ stands for union, concatenation means product and * is the Kleene star (iteration).

The Kleene star is clearly the most 'infinite' operation. One cannot eliminate it because neither union nor product can produce infinite languages from finite ones. However, one can use also complement (the class of regular languages is closed under complement by Kleene's theorem). An extended regular expression is built from words by using union, product, Kleene star, and complement, say, $((a+ba)^C ab)^*(b+(aa)^C)^*$.

The complement of a finite language is infinite. Can one get rid of the Kleene star in this setting?

12. Star-free Regular Expressions

By Kleene's theorem every regular language can be described by a regular expression, say, $((a+ba)^*ab)^*(b+aa)^*$. Here words denote corresponding singleton languages, $+$ stands for union, concatenation means product and * is the Kleene star (iteration).

The Kleene star is clearly the most 'infinite' operation. One cannot eliminate it because neither union nor product can produce infinite languages from finite ones. However, one can use also complement (the class of regular languages is closed under complement by Kleene's theorem). An extended regular expression is built from words by using union, product, Kleene star, and complement, say, $((a+ba)^C ab)^*(b+(aa)^C)^*$.

The complement of a finite language is infinite. Can one get rid of the Kleene star in this setting?

12. Star-free Regular Expressions

By Kleene's theorem every regular language can be described by a regular expression, say, $((a+ba)^*ab)^*(b+aa)^*$. Here words denote corresponding singleton languages, $+$ stands for union, concatenation means product and * is the Kleene star (iteration).

The Kleene star is clearly the most 'infinite' operation. One cannot eliminate it because neither union nor product can produce infinite languages from finite ones. However, one can use also complement (the class of regular languages is closed under complement by Kleene's theorem). An extended regular expression is built from words by using union, product, Kleene star, and complement, say, $((a+ba)^C ab)^*(b+(aa)^C)^*$.

The complement of a finite language is infinite. Can one get rid of the Kleene star in this setting?

13. Star-free Regular Expressions

In some cases we can:

$$(ab)^*$$

Here $E_1 \setminus E_2 = E_1 \cap E_2^C$ can be expressed as $(E_1^C + E_2)^C$ by De Morgan's law.
To understand the formula, observe that $a + a^C = \Sigma^*$.

However, for the language $(a^2)^*$ that looks alike $(ab)^*$ we would not be able to construct a star-free extended regular expression.

How can one distinguish between regular languages that need star and 'star-free' languages?

In some cases we can:

$$(ab)^* = \varepsilon + a(a+a^C)b \setminus \Big((a+a^C)aa(a+a^C) + (a+a^C)bb(a+a^C)\Big).$$

Here $E_1 \setminus E_2 = E_1 \cap E_2^C$ can be expressed as $(E_1^C + E_2)^C$ by De Morgan's law.

To understand the formula, observe that $a + a^C = \Sigma^*$.

However, for the language $(a^2)^*$ that looks alike $(ab)^*$ we would not be able to construct a star-free extended regular expression.

How can one distinguish between regular languages that need star and 'star-free' languages?

13. Star-free Regular Expressions

In some cases we can:

$$(ab)^* = \varepsilon + a(a+a^C)b \setminus \Big((a+a^C)aa(a+a^C)+(a+a^C)bb(a+a^C)\Big).$$

Here $E_1 \setminus E_2 = E_1 \cap E_2^C$ can be expressed as $(E_1^C + E_2)^C$ by De Morgan's law.
To understand the formula, observe that $a + a^C = \Sigma^*$.

However, for the language $(a^2)^*$ that looks alike $(ab)^*$ we would not be able to construct a star-free extended regular expression.

How can one distinguish between regular languages that need star and 'star-free' languages?

In some cases we can:

$$(ab)^* = \varepsilon + a(a+a^C)b \setminus \Big((a+a^C)aa(a+a^C) + (a+a^C)bb(a+a^C)\Big).$$

Here $E_1 \setminus E_2 = E_1 \cap E_2^C$ can be expressed as $(E_1^C + E_2)^C$ by De Morgan's law.
To understand the formula, observe that $a + a^C = \Sigma^*$.

However, for the language $(a^2)^*$ that looks alike $(ab)^*$ we would not be able to construct a star-free extended regular expression.

How can one distinguish between regular languages that need star and 'star-free' languages?

In some cases we can:

$$(ab)^* = \varepsilon + a(a+a^C)b \setminus \Big((a+a^C)aa(a+a^C) + (a+a^C)bb(a+a^C)\Big).$$

Here $E_1 \setminus E_2 = E_1 \cap E_2^C$ can be expressed as $(E_1^C + E_2)^C$ by De Morgan's law.
To understand the formula, observe that $a + a^C = \Sigma^*$.

However, for the language $(a^2)^*$ that looks alike $(ab)^*$ we would not be able to construct a star-free extended regular expression.

How can one distinguish between regular languages that need star and 'star-free' languages?

14. Schützenberger's Theorem

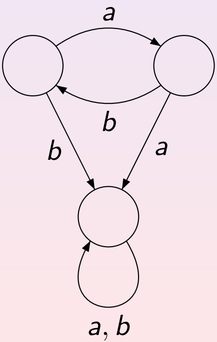
Schützenberger's Theorem, 1964

A regular language L admits a star-free extended regular expression iff the minimal automaton of L is aperiodic.

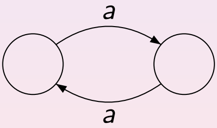
Schützenberger's Theorem, 1964

A regular language L admits a star-free extended regular expression iff the minimal automaton of L is aperiodic.

For instance, for $(ab)^*$ and $(a^2)^*$ the minimal automata are



and



By the (extended) star height of a regular language L we mean the minimum number of nested stars over all (extended) regular expressions representing L.

It is known that there exist regular languages of any given star height and that, given a language, its star height can be decided. However analogous problems are open for extended star height.

15. Extended Star Height Problem

By the (extended) star height of a regular language L we mean the minimum number of nested stars over all (extended) regular expressions representing L.
It is known that there exist regular languages of any given star height and that, given a language, its star height can be decided.
However analogous problems are open for extended star height.

15. Extended Star Height Problem

By the (extended) star height of a regular language L we mean the minimum number of nested stars over all (extended) regular expressions representing L.

It is known that there exist regular languages of any given star height and that, given a language, its star height can be decided. However analogous problems are open for extended star height.

15. Extended Star Height Problem

By the (extended) star height of a regular language L we mean the minimum number of nested stars over all (extended) regular expressions representing L.

It is known that there exist regular languages of any given star height and that, given a language, its star height can be decided. However analogous problems are open for extended star height.

Extended Star Height Problem

Is there a regular language of extended star height > 1?

Is the class of languages of extended star height 1 decidable?

16. Krohn–Rhodes Theorem

A DFA is said to be a group automaton if every letter acts as a permutation of the state set. Group automata are just Cayley graphs of groups and are antipodes of aperiodic automata.

Thus, $\mathscr{A}$ decomposes into counter (=group) and non-counter (=aperiodic) components. Group components can be further decomposed into cascade compositions of Cayley graphs of simple groups while aperiodic components are cascade compositions of flip-flops and their 1-letter subautomata.

A DFA is said to be a group automaton if every letter acts as a permutation of the state set. Group automata are just Cayley graphs of groups and are antipodes of aperiodic automata.

Thus, $\mathscr{A}$ decomposes into counter (=group) and non-counter (=aperiodic) components. Group components can be further decomposed into cascade compositions of Cayley graphs of simple groups while aperiodic components are cascade compositions of flip-flops and their 1-letter subautomata.

16. Krohn–Rhodes Theorem

A DFA is said to be a group automaton if every letter acts as a permutation of the state set. Group automata are just Cayley graphs of groups and are antipodes of aperiodic automata.

Krohn–Rhodes Theorem, 1962

Every finite automaton $\mathscr{A}$ can be emulated by a cascade composition of an alternating sequence of aperiodic and group automata derived from $\mathscr{A}$.

Thus, $\mathscr{A}$ decomposes into counter (=group) and non-counter (=aperiodic) components. Group components can be further decomposed into cascade compositions of Cayley graphs of simple groups while aperiodic components are cascade compositions of flip-flops and their 1-letter subautomata.

16. Krohn–Rhodes Theorem

A DFA is said to be a group automaton if every letter acts as a permutation of the state set. Group automata are just Cayley graphs of groups and are antipodes of aperiodic automata.

Krohn–Rhodes Theorem, 1962

Every finite automaton $\mathscr{A}$ can be emulated by a cascade composition of an alternating sequence of aperiodic and group automata derived from $\mathscr{A}$.

Thus, $\mathscr{A}$ decomposes into counter (=group) and non-counter (=aperiodic) components. Group components can be further decomposed into cascade compositions of Cayley graphs of simple groups while aperiodic components are cascade compositions of flip-flops and their 1-letter subautomata.

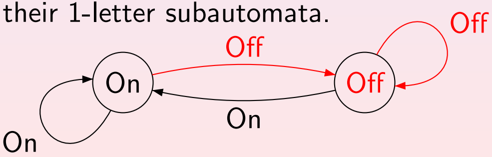
16. Krohn–Rhodes Theorem

A DFA is said to be a group automaton if every letter acts as a permutation of the state set. Group automata are just Cayley graphs of groups and are antipodes of aperiodic automata.

Krohn–Rhodes Theorem, 1962

Every finite automaton $\mathscr{A}$ can be emulated by a cascade composition of an alternating sequence of aperiodic and group automata derived from $\mathscr{A}$.

Thus, $\mathscr{A}$ decomposes into counter (=group) and non-counter (=aperiodic) components. Group components can be further decomposed into cascade compositions of Cayley graphs of simple groups while aperiodic components are cascade compositions of flip-flops and their 1-letter subautomata.



17. Group Complexity

The minimum number of group components in the Krohn–Rhodes decomposition of $\mathscr{A}$ is called the group complexity of $\mathscr{A}$. This parameter gives rise to an infinite hierarchy.

17. Group Complexity

The minimum number of group components in the Krohn–Rhodes decomposition of $\mathscr{A}$ is called the group complexity of $\mathscr{A}$. This parameter gives rise to an infinite hierarchy.

17. Group Complexity

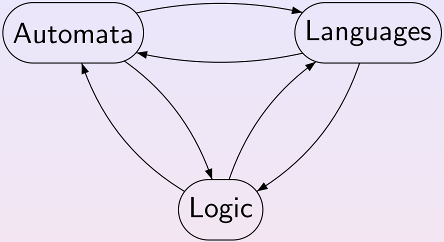
The minimum number of group components in the Krohn–Rhodes decomposition of $\mathscr{A}$ is called the group complexity of $\mathscr{A}$. This parameter gives rise to an infinite hierarchy.

Group Complexity Problem

Given a finite automaton $\mathscr{A}$, can one decide the group complexity of $\mathscr{A}$?
In particular, can we decide if the group complexity of $\mathscr{A}$ is 1?

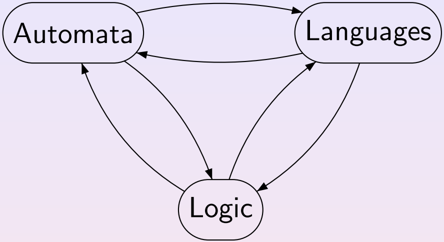
18. Logic for Words

There is a magic triangle



Logic for words has first order variables (positions) that take values in $\{1, 2, \dots\}$, second order variables (sets of positions) whose values are subsets of $\{1, 2, \dots\}$, the usual connectives and quantifiers, the predicate symbol $<$ with the usual meaning (and maybe some additional numerical predicates), and a special predicate Q_a for each letter a with the meaning: $Q_a x$ is true iff the position x holds the letter a.

There is a magic triangle



Logic for words has first order variables (positions) that take values in $\{1, 2, \dots\}$, second order variables (sets of positions) whose values are subsets of $\{1, 2, \dots\}$, the usual connectives and quantifiers, the predicate symbol $<$ with the usual meaning (and maybe some additional numerical predicates), and a special predicate Q_a for each letter a with the meaning: $Q_a x$ is true iff the position x holds the letter a.

19. Examples

Any closed formula of this logic defines a language.

$$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$$

$$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$$

$$\Psi_b : \Psi \mathbin{\&} \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$$

$$\Phi_a \mathbin{\&} \Psi_b \mathbin{\&} \forall x \forall y \left((y{=}x{+}1) \to \left((Q_a x \to Q_b y) \mathbin{\&} (Q_b x \to Q_a y)\right)\right)$$

$$\Psi \mathbin{\&} \forall x\,(Q_a x) \mathbin{\&} \exists H \Big(\forall x \forall y \left((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\right) \mathbin{\&}$$
$$\forall x\left((\neg(\exists y(y < x)) \to (x \in H)) \mathbin{\&} (\neg(\exists y(x < y)) \to \neg(x \in H))\right)\Big)$$

Any closed formula of this logic defines a language.

$$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$$

$$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$$

$$\Psi_b : \Psi \mathbin{\&} \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$$

$$\Phi_a \mathbin{\&} \Psi_b \mathbin{\&} \forall x \forall y \left((y{=}x{+}1) \to \left((Q_a x \to Q_b y) \mathbin{\&} (Q_b x \to Q_a y)\right)\right)$$

$$\Psi \mathbin{\&} \forall x\,(Q_a x) \mathbin{\&} \exists H \Big(\forall x \forall y \left((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\right) \mathbin{\&}$$

$$\forall x \left((\neg(\exists y(y < x)) \to (x \in H)) \mathbin{\&} (\neg(\exists y(x < y)) \to \neg(x \in H))\right)\Big)$$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$ all words starting with a

$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$

$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$

$$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$$

$$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \,\&$$
$$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H)))\Big)$$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y (y < x)) \to Q_a x \right)$ all words starting with a

$\Psi : \exists x \left(\neg(\exists y (x < y)) \right)$

$\Psi_b : \Psi \;\&\; \forall x \left(\neg(\exists y (x < y)) \to Q_b x \right)$

$$\Phi_a \;\&\; \Psi_b \;\&\; \forall x \forall y \left((y{=}x{+}1) \to \left((Q_a x \to Q_b y) \;\&\; (Q_b x \to Q_a y) \right) \right)$$

$$\Psi \;\&\; \forall x \,(Q_a x) \;\&\; \exists H \Big(\forall x \forall y \big((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H)) \big) \;\&$$
$$\forall x \big((\neg(\exists y (y < x)) \to (x \in H)) \;\&\; (\neg(\exists y (x < y)) \to \neg(x \in H)) \big) \Big)$$

19. Examples

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$ — all words starting with a

$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$ — all finite words

$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$

$$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$$

$$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y \left((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\right) \,\&$$
$$\forall x \left((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H))\right)\Big)$$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$ all words starting with a

$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$ all finite words

$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$

$$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$$

$$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y \left((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\right) \,\&$$
$$\forall x \left((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H))\right)\Big)$$

19. Examples

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x \right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y)) \right)$	all finite words
$\Psi_b : \Psi \mathbin{\&} \forall x \left(\neg(\exists y(x < y)) \to Q_b x \right)$	all finite words ending with b

$$\Phi_a \mathbin{\&} \Psi_b \mathbin{\&} \forall x \forall y\, ((y{=}x{+}1) \to ((Q_a x \to Q_b y) \mathbin{\&} (Q_b x \to Q_a y)))$$

$$\Psi \mathbin{\&} \forall x\, (Q_a x) \mathbin{\&} \exists H \Big(\forall x \forall y\, ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \mathbin{\&}$$
$$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \mathbin{\&} (\neg(\exists y(x < y)) \to \neg(x \in H))) \Big)$$

19. Examples

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y\, ((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y)))$$

$$\Psi \,\&\, \forall x\, (Q_a x) \,\&\, \exists H \Big(\forall x \forall y ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \,\&$$
$$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H)))\Big)$$

19. Examples

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$
Here $y{=}x{+}1$ abbreviates $(x < y) \,\&\, \neg\left(\exists z \; ((x < z) \& (z < y))\right)$.

$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y \big((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\big) \,\&$
$\forall x \big((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H))\big)\Big)$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \mathbin{\&} \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$\Phi_a \mathbin{\&} \Psi_b \mathbin{\&} \forall x \forall y\, ((y{=}x{+}1) \to ((Q_a x \to Q_b y) \mathbin{\&} (Q_b x \to Q_a y)))$
This (first order) formula defines the language $(ab)^*$.

$\Psi \mathbin{\&} \forall x\, (Q_a x) \mathbin{\&} \exists H \Big(\forall x \forall y ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \mathbin{\&}$
$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \mathbin{\&} (\neg(\exists y(x < y)) \to \neg(x \in H))) \Big)$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \,((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y)))$
This (first order) formula defines the language $(ab)^*$.

$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \,\&$
$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H)))\Big)$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \mathbin{\&} \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$\Phi_a \mathbin{\&} \Psi_b \mathbin{\&} \forall x \forall y\, ((y{=}x{+}1) \to ((Q_a x \to Q_b y) \mathbin{\&} (Q_b x \to Q_a y)))$

This (first order) formula defines the language $(ab)^*$.

$\Psi \mathbin{\&} \forall x\,(Q_a x) \mathbin{\&} \exists H \Big(\forall x \forall y ((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \mathbin{\&}$

$\forall x ((\neg(\exists y(y < x)) \to (x \in H)) \mathbin{\&} (\neg(\exists y(x < y)) \to \neg(x \in H))) \Big)$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$ all words starting with a

$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$ all finite words

$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$ all finite words ending with b

$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$
This (first order) formula defines the language $(ab)^*$.

$$\Psi \,\&\, \forall x\, (Q_a x) \,\&\, \exists H \Big(\forall x \forall y \big((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\big) \,\&$$
$$\forall x \big((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H))\big)\Big)$$

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y\, ((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y)))$$

This (first order) formula defines the language $(ab)^*$.

$$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H\Big(\forall x \forall y((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))) \,\&$$
$$\forall x((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H)))\Big)$$

This (monadic second order) formula defines the language $(a^2)^*$.

Any closed formula of this logic defines a language.

$\Phi_a : \forall x \left(\neg(\exists y(y < x)) \to Q_a x\right)$	all words starting with a
$\Psi : \exists x \left(\neg(\exists y(x < y))\right)$	all finite words
$\Psi_b : \Psi \,\&\, \forall x \left(\neg(\exists y(x < y)) \to Q_b x\right)$	all finite words ending with b

$\Phi_a \,\&\, \Psi_b \,\&\, \forall x \forall y \left((y{=}x{+}1) \to ((Q_a x \to Q_b y) \,\&\, (Q_b x \to Q_a y))\right)$
This (first order) formula defines the language $(ab)^*$.

$\Psi \,\&\, \forall x\,(Q_a x) \,\&\, \exists H \Big(\forall x \forall y \big((y{=}x{+}1) \to ((x \in H) \leftrightarrow \neg(y \in H))\big) \,\&$
$\forall x \big((\neg(\exists y(y < x)) \to (x \in H)) \,\&\, (\neg(\exists y(x < y)) \to \neg(x \in H))\big)\Big)$
This (monadic second order) formula defines the language $(a^2)^*$.

Monadic second order formulas define precisely regular languages (Büchi, 1960), but we would not be able to construct a first order formula defining $(a^2)^*$.

Can one distinguish between 'second' and 'first' order languages?

20. McNaughton's Theorem

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

A natural complexity measure for first order formulas is the number of alternations of logical quantifiers in the prenex form.

20. McNaughton's Theorem

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

A natural complexity measure for first order formulas is the number of alternations of logical quantifiers in the prenex form. The minimum number of quantifier alternations over all first order formulas representing a given star-free language L is called the dot-depth of L.

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

A natural complexity measure for first order formulas is the number of alternations of logical quantifiers in the prenex form. The minimum number of quantifier alternations over all first order formulas representing a given star-free language L is called the dot-depth of L. This parameter gives rise to an infinite hierarchy.

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

A natural complexity measure for first order formulas is the number of alternations of logical quantifiers in the prenex form. The minimum number of quantifier alternations over all first order formulas representing a given star-free language L is called the dot-depth of L. This parameter gives rise to an infinite hierarchy.

Dot-Depth Problem

Given a star-free language L, can one decide the dot-depth of L? In particular, can we decide if the dot-depth of L is 3?

20. McNaughton's Theorem

Can one distinguish between 'second' and 'first' order languages?

McNaughton's Theorem, 1966

A regular language L admits a description by a first order formula iff the minimal automaton of L is aperiodic.

A natural complexity measure for first order formulas is the number of alternations of logical quantifiers in the prenex form. The minimum number of quantifier alternations over all first order formulas representing a given star-free language L is called the dot-depth of L. This parameter gives rise to an infinite hierarchy.

Dot-Depth Problem

Given a star-free language L, can one decide the dot-depth of L? In particular, can we decide if the dot-depth of L is 3?

Dot-depth 1 and dot-depth 2 are known to be decidable (Knast, 1980, for 1 and Place-Zeitoun, 2014, for 2)

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.

Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing.*

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

21. Back to Synchronization

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.

Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ *is synchronizing.*

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.
Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing*.

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

21. Back to Synchronization

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.

Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing*.

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

21. Back to Synchronization

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.
Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing*.

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

21. Back to Synchronization

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.
Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing*.

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

21. Back to Synchronization

Here we aim to study aperiodic automata from the viewpoint of synchronization (in particular, to prove the Černý conjecture for aperiodic automata).

As discussed in Lecture VI, we may restrict to strongly connected automata.
Here we encounter a small surprise: *every strongly connected aperiodic automaton* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *is synchronizing*.

Indeed, take any $q, q' \in Q$. Since $\mathscr{A}$ is strongly connected, there exists $w \in \Sigma^*$ such that $q \,.\, w = q'$. On the other hand, $\mathscr{A}$ is aperiodic whence there exists a positive integer m such that $q \,.\, w^m = q \,.\, w^{m+1}$. Applying w^m to the equality $q \,.\, w = q'$, we get $q \,.\, w^{m+1} = q' \,.\, w^m$ whence $q \,.\, w^m = q' \,.\, w^m$. Thus, every pair of states can be synchronized, and by Černý's criterion, this ensures that $\mathscr{A}$ is synchronizing.

Partial Order

Avraham Trahtman (The Černý conjecture for aperiodic automata, Discrete Math. Theor. Comp. Sci. 9(2) (2007) 3–10) has proved that every synchronizing aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$ (so less than $(n-1)^2$).

Recall that $\frac{n(n-1)}{2}$ is precisely Rystsov's bound for n-state synchronizing automata having a sink. Thus, it remains to prove that every strongly connected aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

The key observation by Trahtman is that every strongly connected aperiodic automaton admits a non-trivial stable partial order.

Partial Order

Avraham Trahtman (The Černý conjecture for aperiodic automata, Discrete Math. Theor. Comp. Sci. 9(2) (2007) 3–10) has proved that every synchronizing aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$ (so less than $(n-1)^2$).

Recall that $\frac{n(n-1)}{2}$ is precisely Rystsov's bound for n-state synchronizing automata having a sink. Thus, it remains to prove that every strongly connected aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

The key observation by Trahtman is that every strongly connected aperiodic automaton admits a non-trivial stable partial order.

Partial Order

Avraham Trahtman (The Černý conjecture for aperiodic automata, Discrete Math. Theor. Comp. Sci. 9(2) (2007) 3–10) has proved that every synchronizing aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$ (so less than $(n-1)^2$).

Recall that $\frac{n(n-1)}{2}$ is precisely Rystsov's bound for n-state synchronizing automata having a sink. Thus, it remains to prove that every strongly connected aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

The key observation by Trahtman is that every strongly connected aperiodic automaton admits a non-trivial stable partial order.

Partial Order

Avraham Trahtman (The Černý conjecture for aperiodic automata, Discrete Math. Theor. Comp. Sci. 9(2) (2007) 3–10) has proved that every synchronizing aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$ (so less than $(n-1)^2$).

Recall that $\frac{n(n-1)}{2}$ is precisely Rystsov's bound for n-state synchronizing automata having a sink. Thus, it remains to prove that every strongly connected aperiodic automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

The key observation by Trahtman is that every strongly connected aperiodic automaton admits a non-trivial stable partial order.

23. Constructing Partial Order

Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, its square $\mathscr{A}^{[2]} = \langle Q \times Q, \Sigma, \delta^{[2]} \rangle$ is defined by $\delta^{[2]}\big((q,p),a\big) = \big(\delta(q,a), \delta(p,a)\big)$.
Warning: it is not quite the same as the automaton on all at most 2-element subsets that we considered in Lecture II.

If $\mathscr{A}$ is synchronizing and strongly connected, then $\mathscr{A}^{[2]}$ has a least strongly connected component $D = \{(q,q) \mid q \in Q\}$. Let K be a strongly connected component immediately following D in the natural order of strongly connected components.

23. Constructing Partial Order

Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, its square $\mathscr{A}^{[2]} = \langle Q \times Q, \Sigma, \delta^{[2]} \rangle$ is defined by $\delta^{[2]}\big((q,p),a\big) = \big(\delta(q,a), \delta(p,a)\big)$.

Warning: it is not quite the same as the automaton on all at most 2-element subsets that we considered in Lecture II.

If $\mathscr{A}$ is synchronizing and strongly connected, then $\mathscr{A}^{[2]}$ has a least strongly connected component $D = \{(q,q) \mid q \in Q\}$. Let K be a strongly connected component immediately following D in the natural order of strongly connected components.

Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, its square $\mathscr{A}^{[2]} = \langle Q \times Q, \Sigma, \delta^{[2]} \rangle$ is defined by $\delta^{[2]}\big((q,p),a\big) = \big(\delta(q,a), \delta(p,a)\big)$.
Warning: it is not quite the same as the automaton on all at most 2-element subsets that we considered in Lecture II.

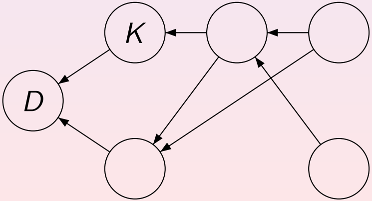
If $\mathscr{A}$ is synchronizing and strongly connected, then $\mathscr{A}^{[2]}$ has a least strongly connected component $D = \{(q,q) \mid q \in Q\}$. Let K be a strongly connected component immediately following D in the natural order of strongly connected components.

23. Constructing Partial Order

Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, its square $\mathscr{A}^{[2]} = \langle Q \times Q, \Sigma, \delta^{[2]} \rangle$ is defined by $\delta^{[2]}((q,p),a) = \big(\delta(q,a), \delta(p,a)\big)$.

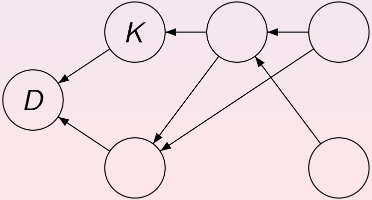
Warning: it is not quite the same as the automaton on all at most 2-element subsets that we considered in Lecture II.

If $\mathscr{A}$ is synchronizing and strongly connected, then $\mathscr{A}^{[2]}$ has a least strongly connected component $D = \{(q,q) \mid q \in Q\}$. Let K be a strongly connected component immediately following D in the natural order of strongly connected components.



Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$, its square $\mathscr{A}^{[2]} = \langle Q \times Q, \Sigma, \delta^{[2]} \rangle$ is defined by $\delta^{[2]}\big((q,p),a\big) = \big(\delta(q,a), \delta(p,a)\big)$.
Warning: it is not quite the same as the automaton on all at most 2-element subsets that we considered in Lecture II.

If $\mathscr{A}$ is synchronizing and strongly connected, then $\mathscr{A}^{[2]}$ has a least strongly connected component $D = \{(q,q) \mid q \in Q\}$. Let K be a strongly connected component immediately following D in the natural order of strongly connected components.



Then $K \cup D$ is a non-trivial stable reflexive relation on Q.

24. $\mathscr{A}$ is aperiodic

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

24. $\mathscr{A}$ is aperiodic

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \dots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \dots, k-1$. We choose the shortest such sequence $p_0, p_1, \dots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \, . \, w = (q, p)$, that is, $p \, . \, w = q$, $q \, . \, w = p$.

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$.
By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \dots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \dots, k-1$. We choose the shortest such sequence $p_0, p_1, \dots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

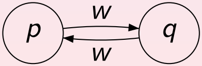
If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$.

24. $\mathscr{A}$ is aperiodic

Let $\succeq_K$ be the transitive closure of $K \cup D$. It is clear that $\succeq_K$ is non-trivial, stable, reflexive and transitive.
Now we show that $\succeq_K$ is antisymmetric whenever $\mathscr{A}$ is aperiodic.

Suppose that there are $p, q \in Q$ such that $p \neq q$ and $p \succeq_K q \succeq_K p$. Then there is a sequence of $p_0, p_1, \ldots, p_k \in Q$ such that $k > 1$, $p_0 = p = p_k$, $q = p_j$ for some j, $0 < j < k$, and $(p_i, p_{i+1}) \in K$ for all $i = 0, 1, \ldots, k-1$. We choose the shortest such sequence $p_0, p_1, \ldots, p_k$ (over all possible 'obstacles' (p, q) to antisymmetry).

If $k = 2$, then we have $p_0 = p = p_2$, $p_1 = q$ and $(p, q), (q, p) \in K$. By the definition of K, there exists $w \in \Sigma^*$ such that $(p, q) \,.\, w = (q, p)$, that is, $p \,.\, w = q$, $q \,.\, w = p$. This clearly contradicts the assumption that $\mathscr{A}$ is aperiodic:



25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) . w = (p_1, p_2)$, that is, $p_0 . w = p_1$, $p_0 . w^2 = p_1 . w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 . w^{m+1} = p_0 . w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 . w^2 \neq p_0 . w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i . w^{m-1}, p_{i+1} . w^{m-1}) \in K$ or $p_i . w^{m-1} = p_{i+1} . w^{m-1}$. The choice of m ensures $p_0 . w^{m-1} \neq p_1 . w^{m-1} = p_0 . w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 . w^{m-1} = p_0 . w^m = p_0 . w^{m+1} = p_2 . w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) . w = (p_1, p_2)$, that is, $p_0 . w = p_1$, $p_0 . w^2 = p_1 . w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 . w^{m+1} = p_0 . w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 . w^2 \neq p_0 . w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i . w^{m-1}, p_{i+1} . w^{m-1}) \in K$ or $p_i . w^{m-1} = p_{i+1} . w^{m-1}$. The choice of m ensures $p_0 . w^{m-1} \neq p_1 . w^{m-1} = p_0 . w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 . w^{m-1} = p_0 . w^m = p_0 . w^{m+1} = p_2 . w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) \,.\, w = (p_1, p_2)$, that is, $p_0 \,.\, w = p_1$, $p_0 \,.\, w^2 = p_1 \,.\, w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 \,.\, w^{m+1} = p_0 \,.\, w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 \,.\, w^2 \neq p_0 \,.\, w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i \,.\, w^{m-1}, p_{i+1} \,.\, w^{m-1}) \in K$ or $p_i \,.\, w^{m-1} = p_{i+1} \,.\, w^{m-1}$. The choice of m ensures $p_0 \,.\, w^{m-1} \neq p_1 \,.\, w^{m-1} = p_0 \,.\, w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 \,.\, w^{m-1} = p_0 \,.\, w^m = p_0 \,.\, w^{m+1} = p_2 \,.\, w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) \,.\, w = (p_1, p_2)$, that is, $p_0 \,.\, w = p_1$, $p_0 \,.\, w^2 = p_1 \,.\, w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 \,.\, w^{m+1} = p_0 \,.\, w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 \,.\, w^2 \neq p_0 \,.\, w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i \,.\, w^{m-1}, p_{i+1} \,.\, w^{m-1}) \in K$ or $p_i \,.\, w^{m-1} = p_{i+1} \,.\, w^{m-1}$. The choice of m ensures $p_0 \,.\, w^{m-1} \neq p_1 \,.\, w^{m-1} = p_0 \,.\, w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 \,.\, w^{m-1} = p_0 \,.\, w^m = p_0 \,.\, w^{m+1} = p_2 \,.\, w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) \mathbin{.} w = (p_1, p_2)$, that is, $p_0 \mathbin{.} w = p_1$, $p_0 \mathbin{.} w^2 = p_1 \mathbin{.} w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 \mathbin{.} w^{m+1} = p_0 \mathbin{.} w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 \mathbin{.} w^2 \neq p_0 \mathbin{.} w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i \mathbin{.} w^{m-1}, p_{i+1} \mathbin{.} w^{m-1}) \in K$ or $p_i \mathbin{.} w^{m-1} = p_{i+1} \mathbin{.} w^{m-1}$. The choice of m ensures $p_0 \mathbin{.} w^{m-1} \neq p_1 \mathbin{.} w^{m-1} = p_0 \mathbin{.} w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 \mathbin{.} w^{m-1} = p_0 \mathbin{.} w^m = p_0 \mathbin{.} w^{m+1} = p_2 \mathbin{.} w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) . w = (p_1, p_2)$, that is, $p_0 . w = p_1$, $p_0 . w^2 = p_1 . w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 . w^{m+1} = p_0 . w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 . w^2 \neq p_0 . w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \ldots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \ldots, k-1$ either $(p_i . w^{m-1}, p_{i+1} . w^{m-1}) \in K$ or $p_i . w^{m-1} = p_{i+1} . w^{m-1}$. The choice of m ensures $p_0 . w^{m-1} \neq p_1 . w^{m-1} = p_0 . w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 . w^{m-1} = p_0 . w^m = p_0 . w^{m+1} = p_2 . w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \ldots, p_k$.

25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) \,.\, w = (p_1, p_2)$, that is, $p_0 \,.\, w = p_1$, $p_0 \,.\, w^2 = p_1 \,.\, w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 \,.\, w^{m+1} = p_0 \,.\, w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 \,.\, w^2 \neq p_0 \,.\, w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \ldots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \ldots, k-1$ either $(p_i \,.\, w^{m-1}, p_{i+1} \,.\, w^{m-1}) \in K$ or $p_i \,.\, w^{m-1} = p_{i+1} \,.\, w^{m-1}$. The choice of m ensures $p_0 \,.\, w^{m-1} \neq p_1 \,.\, w^{m-1} = p_0 \,.\, w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 \,.\, w^{m-1} = p_0 \,.\, w^m = p_0 \,.\, w^{m+1} = p_2 \,.\, w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \ldots, p_k$.

25. Antisymmetry

Suppose that $k > 2$. Then p_0, p_1, p_2 are all distinct. By the definition of K, there exists $w \in \Sigma^*$ such that $(p_0, p_1) \,.\, w = (p_1, p_2)$, that is, $p_0 \,.\, w = p_1$, $p_0 \,.\, w^2 = p_1 \,.\, w = p_2$. Since $\mathscr{A}$ is aperiodic, there exists m such that $p_0 \,.\, w^{m+1} = p_0 \,.\, w^m$; we choose the least m with this property. Observe that $m > 1$ since $p_0 \,.\, w^2 \neq p_0 \,.\, w$.

Now we apply w^{m-1} to each state in the sequence $p_0, p_1, \dots, p_k$. Since $K \cup D$ is stable, we get that for all $i = 0, 1, \dots, k-1$ either $(p_i \,.\, w^{m-1}, p_{i+1} \,.\, w^{m-1}) \in K$ or $p_i \,.\, w^{m-1} = p_{i+1} \,.\, w^{m-1}$. The choice of m ensures $p_0 \,.\, w^{m-1} \neq p_1 \,.\, w^{m-1} = p_0 \,.\, w^m$ whence the new sequence still contains an obstacle to antisymmetry. On the other hand, $p_1 \,.\, w^{m-1} = p_0 \,.\, w^m = p_0 \,.\, w^{m+1} = p_2 \,.\, w^{m-1}$, and therefore, if we retain in the new sequence only the first state from each group of adjacent equal states, the sequence becomes shorter. This contradicts the minimality of $p_0, p_1, \dots, p_k$.

26. Using the Order

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?

We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.

The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p\,.\,u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p\,.\,u_1 \succeq_K r\,.\,u_1$ whence $q = r\,.\,u_1$ as q is minimal. If still $|T\,.\,u_1| > 1$, we can take a maximum state $p' \in T\,.\,u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p'\,.\,u_2 = q$, and so on.

26. Using the Order

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?
We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.
The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p \,.\, u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p \,.\, u_1 \succeq_K r \,.\, u_1$ whence $q = r \,.\, u_1$ as q is minimal. If still $|T \,.\, u_1| > 1$, we can take a maximum state $p' \in T \,.\, u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p' \,.\, u_2 = q$, and so on.

26. Using the Order

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?
We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.
The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p \,.\, u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p \,.\, u_1 \succeq_K r \,.\, u_1$ whence $q = r \,.\, u_1$ as q is minimal. If still $|T \,.\, u_1| > 1$, we can take a maximum state $p' \in T \,.\, u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p' \,.\, u_2 = q$, and so on.

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?
We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.
The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p \, . \, u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p \, . \, u_1 \succeq_K r \, . \, u_1$ whence $q = r \, . \, u_1$ as q is minimal. If still $|T \, . \, u_1| > 1$, we can take a maximum state $p' \in T \, . \, u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p' \, . \, u_2 = q$, and so on.

26. Using the Order

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?
We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.
The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p\,.\,u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p\,.\,u_1 \succeq_K r\,.\,u_1$ whence $q = r\,.\,u_1$ as q is minimal. If still $|T\,.\,u_1| > 1$, we can take a maximum state $p' \in T\,.\,u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p'\,.\,u_2 = q$, and so on.

Thus, $\succeq_K$ is a non-trivial partial order. How does it help?
We denote by π_K the symmetric closure of $\succeq_K$. Clearly, π_K is a congruence of the automaton $\mathscr{A}$.
The quotient automaton $\mathscr{A}/\pi_K$ has $< n$ states and so it has a short reset word v by the induction assumption (we induct on the number of states). In $\mathscr{A}$, the word v sends the whole state set into a single π_K-class, say, T. In order to quickly synchronize T choose a minimum state $q \in T$ and a maximum state $p \in T$ w.r.t. $\succeq_K$ ($p \neq q$ whenever $|T| > 1$). Since $\mathscr{A}$ is strongly connected, there is a word u_1 of length $< n$ such that $p\,.\,u_1 = q$. Then for each $r \in T$ such that $p \succeq_K r$ we have $q = p\,.\,u_1 \succeq_K r\,.\,u_1$ whence $q = r\,.\,u_1$ as q is minimal. If still $|T\,.\,u_1| > 1$, we can take a maximum state $p' \in T\,.\,u_1$ and repeat the process by taking a word u_2 of length $< n$ such that $p'\,.\,u_2 = q$, and so on.

How long can be a reset word constructed this way?

From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \dots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

27. Estimations

How long can be a reset word constructed this way?

From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n - 1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n - m + 1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \ldots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

27. Estimations

How long can be a reset word constructed this way?
From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \ldots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

27. Estimations

How long can be a reset word constructed this way?
From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \le \frac{n(n-1)}{2}$$

for all $m = 2, \dots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

27. Estimations

How long can be a reset word constructed this way?
From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \dots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

27. Estimations

How long can be a reset word constructed this way?

From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \dots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

How long can be a reset word constructed this way?

From the minimum-maximum symmetry it follows that the number of steps is at most $\frac{|T|}{2}$. In the case when $T = Q$ (actually, this is the worst case) we get at most $\frac{n}{2}$ steps and a word of length at most $n-1$ is added at each step. The resulting reset word is of length at most $\frac{n(n-1)}{2}$. If $|T| = m < n$, then the quotient automaton $\mathscr{A}/\pi_K$ has at most $n-m+1$ states and we first need a word v of length at most $\frac{(n-m+1)(n-m)}{2}$ to send Q to T and then a word of length at most $\frac{m(n-1)}{2}$ to synchronize T. It remains to calculate that

$$\frac{(n-m+1)(n-m)}{2} + \frac{m(n-1)}{2} \leq \frac{n(n-1)}{2}$$

for all $m = 2, \ldots, n-1$. If $m = 1$, then v itself is a reset word for $\mathscr{A}$.

28. Weakly Monotonic Automata

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

28. Weakly Monotonic Automata

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

28. Weakly Monotonic Automata

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \dots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

28. Weakly Monotonic Automata

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \dots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

28. Weakly Monotonic Automata

In my paper (Synchronizing automata preserving a chain of partial orders, Theor. Comput. Sci. 410 (2009) 3513–3519) Trahtman's theorem has been extended to a larger class automata.

A DFA $\mathscr{A}$ is weakly monotonic of level ℓ if it has a strictly increasing chain of stable binary relations $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_\ell$ satisfying the following conditions:

- ρ_0 is the equality relation;
- for each $i = 1, \ldots, \ell$, the congruence π_{i-1} generated by ρ_{i-1} is contained in ρ_i and the relation ρ_i/π_{i-1} is a partial order on Q/π_{i-1};
- π_ℓ is the universal relation.

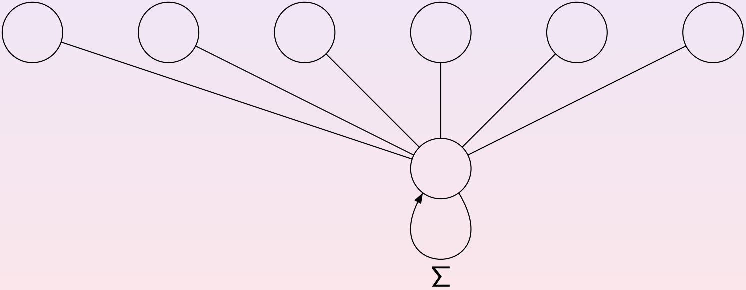
This differs from the notion of a generalized monotonic automaton by just dropping the restriction that the order ρ_i/π_{i-1} is linear on each π_i/π_{i-1}-class.

29. Examples

- every aperiodic automaton is weakly monotonic;
- every automaton with 0 is weakly monotonic (of level 1).

- every aperiodic automaton is weakly monotonic;
- every automaton with 0 is weakly monotonic (of level 1).

- every aperiodic automaton is weakly monotonic;
- every automaton with 0 is weakly monotonic (of level 1).



30. WM + SC Automata

- Every weakly monotonic automaton with a strongly connected underlying digraph is synchronizing. (A non-trivial generalization of the corresponding result for aperiodic automata.)

- Every weakly monotonic automaton with a strongly connected underlying digraph and n states has a reset word of length $\leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor$. (This upper bound is new even for the aperiodic case – recall that Trahtman's bound was 3 times higher, namely, $\frac{n(n-1)}{2}$.)

- Every weakly monotonic synchronizing automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

- Every weakly monotonic automaton with a strongly connected underlying digraph is synchronizing. (A non-trivial generalization of the corresponding result for aperiodic automata.)
- Every weakly monotonic automaton with a strongly connected underlying digraph and n states has a reset word of length $\leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor$. (This upper bound is new even for the aperiodic case – recall that Trahtman's bound was 3 times higher, namely, $\frac{n(n-1)}{2}$.)
- Every weakly monotonic synchronizing automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

- Every weakly monotonic automaton with a strongly connected underlying digraph is synchronizing. (A non-trivial generalization of the corresponding result for aperiodic automata.)

- Every weakly monotonic automaton with a strongly connected underlying digraph and n states has a reset word of length $\leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor$. (This upper bound is new even for the aperiodic case – recall that Trahtman's bound was 3 times higher, namely, $\frac{n(n-1)}{2}$.)

- Every weakly monotonic synchronizing automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

- Every weakly monotonic automaton with a strongly connected underlying digraph is synchronizing. (A non-trivial generalization of the corresponding result for aperiodic automata.)

- Every weakly monotonic automaton with a strongly connected underlying digraph and n states has a reset word of length $\leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor$. (This upper bound is new even for the aperiodic case – recall that Trahtman's bound was 3 times higher, namely, $\frac{n(n-1)}{2}$.)

- Every weakly monotonic synchronizing automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

- Every weakly monotonic automaton with a strongly connected underlying digraph is synchronizing. (A non-trivial generalization of the corresponding result for aperiodic automata.)

- Every weakly monotonic automaton with a strongly connected underlying digraph and n states has a reset word of length $\leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor$. (This upper bound is new even for the aperiodic case – recall that Trahtman's bound was 3 times higher, namely, $\frac{n(n-1)}{2}$.)

- Every weakly monotonic synchronizing automaton with n states has a reset word of length $\frac{n(n-1)}{2}$.

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

31. Lower Bounds

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

$$n - 1 \leq C_{SCA}(n)$$

31. Lower Bounds

A bad news is that there are no matching lower bounds for the upper bounds just discussed. No aperiodic (even no weakly monotonic) strongly connected automaton is known for which the minimum length of reset words would be greater than or equal to the number of states.

Let $C_{SCA}(n)$ denote the restriction of the Černý function to the class of all strongly connected aperiodic automata, that is, $C_{SCA}(n)$ is the maximum length of shortest reset words for strongly connected aperiodic automata with n states. Then our current knowledge can be summarized as follows:

$$n-1 \leq C_{SCA}(n) \leq \left\lfloor \frac{n(n+1)}{6} \right\rfloor \quad \text{(Volkov, 2009).}$$

Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

$$\text{(Ananichev, 2010)} \quad n + \left\lfloor \frac{n}{2} \right\rfloor - 2 \leq C_A(n)$$

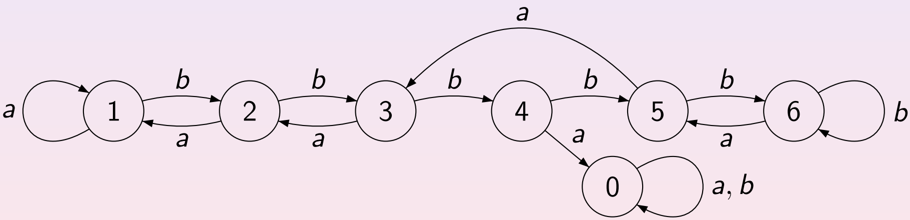
32. Ananichev's Series

Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

(Ananichev, 2010) $n+\left\lfloor\frac{n}{2}\right\rfloor-2 \le C_A(n) \le \frac{n(n-1)}{2}$ (Trahtman, 2007).

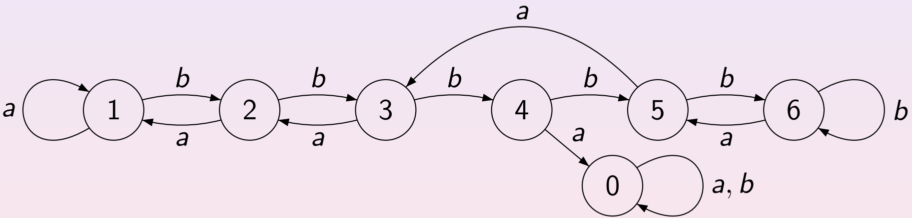
Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

(Ananichev, 2010) $n+\left\lfloor\frac{n}{2}\right\rfloor-2 \leq C_A(n) \leq \frac{n(n-1)}{2}$ (Trahtman, 2007).



Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

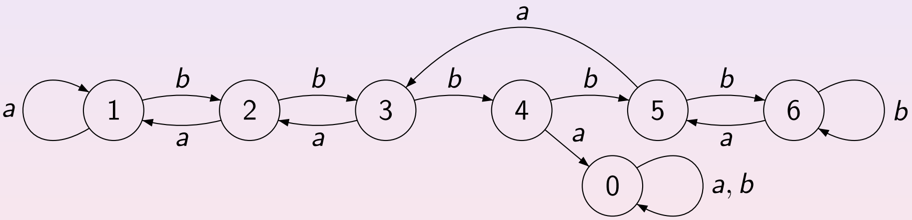
(Ananichev, 2010) $n + \left\lfloor \frac{n}{2} \right\rfloor - 2 \leq C_A(n) \leq \frac{n(n-1)}{2}$ (Trahtman, 2007).



This is the first automaton in Ananichev's series that yields the best up to now lower bound for $C_A(n)$.

Similarly, if $C_A(n)$ denotes the restriction of the Černý function to the class of all aperiodic automata, we have:

$$\text{(Ananichev, 2010)}\quad n+\left\lfloor\frac{n}{2}\right\rfloor-2 \le C_A(n) \le \frac{n(n-1)}{2}\quad \text{(Trahtman, 2007).}$$



This is the first automaton in Ananichev's series that yields the best up to now lower bound for $C_A(n)$. It has 7 states and its shortest reset word is a^4b^3a of length $7+\left\lfloor\frac{7}{2}\right\rfloor-2=8$.