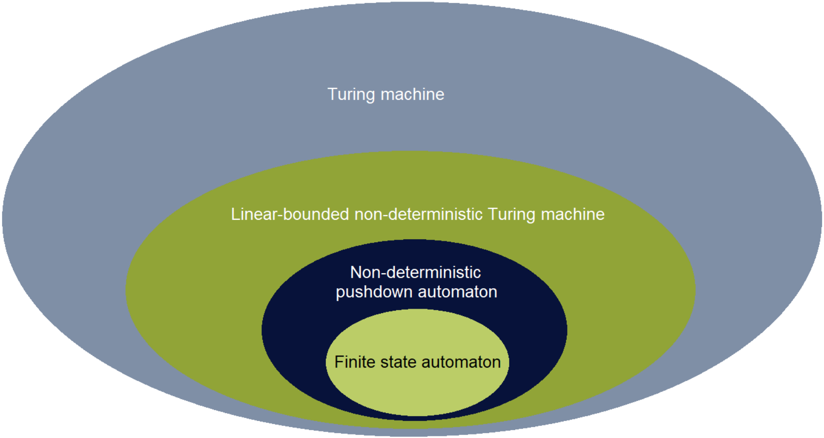
An introduction to automata theory and state complexity of the multiples of the Thue-Morse set

Adeline Massuir

Young Mathematicians Symposium of the Greater Region

September 25th 2018

Chomsky hierarchy



Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Definition

A *word* over an alphabet is a finite ordered sequence of letters.

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Definition

A *word* over an alphabet is a finite ordered sequence of letters.

Σ : $a, b, c, ab, abba, abaabcaca, \ldots$

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Definition

A *word* over an alphabet is a finite ordered sequence of letters.

Σ : $a, b, c, ab, abba, abaabcaca, \ldots$
Γ : $0, 1, 01, 01100001110, \ldots$

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Definition

A *word* over an alphabet is a finite ordered sequence of letters.

Σ : $a, b, c, ab, abba, abaabcaca, \ldots$
Γ : $0, 1, 01, 01100001110, \ldots$
DNA

Alphabet – Letter – Word

Definition

An *alphabet* is a finite set. Every element of an alphabet is a *letter*.

$\Sigma = \{a, b, c\}$
$\Gamma = \{0, 1\}$
DNA : $\{A, C, G, T\}$ (adenine – cytosine – guanine – thymine)
War : $\{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Jack}, \text{Queen}, \text{King}, \text{Ace}\}$

Definition

A *word* over an alphabet is a finite ordered sequence of letters.

Σ : $a, b, c, ab, abba, abaabcaca, \ldots$
Γ : $0, 1, 01, 01100001110, \ldots$
DNA
War

About words I

Let Σ be an alphabet and w a word over Σ.

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

$\Sigma = \{0, 1\}$

$$|01101|$$

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

$\Sigma = \{0, 1\}$

$$|01101| = 5$$

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

$\Sigma = \{0, 1\}$

$$|01101| = 5, |111|$$

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

$\Sigma = \{0, 1\}$

$$|01101| = 5, |111| = 3, \ldots$$

About words I

Let Σ be an alphabet and w a word over Σ.

Notation

The set of all words over Σ is denoted by Σ^*.

Definition

The *length* of w is the number of letters in w. It is denoted by $|w|$.

$\Sigma = \{0, 1\}$

$$|01101| = 5, |111| = 3, \ldots$$

Definition

The *empty word* ε is the only word of length 0.

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

$\Sigma = \{a, b, c\}$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{ i \in \{1, \dots, n\} : w_i = \sigma \} .$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_a =$$

Let Σ be an alphabet, $w = w_1 \ldots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{ i \in \{1, \ldots, n\} : w_i = \sigma \} .$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_a =$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_a = 5$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_b =$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{ i \in \{1, \dots, n\} : w_i = \sigma \} .$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_b =$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_b = 3$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{i \in \{1, \dots, n\} : w_i = \sigma\}.$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_c =$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{ i \in \{1, \dots, n\} : w_i = \sigma \}.$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_c =$$

About words II

Let Σ be an alphabet, $w = w_1 \dots w_n \in \Sigma^*$ and $\sigma \in \Sigma$.

Definition

We set

$$|w|_\sigma = \sharp \{ i \in \{1, \dots, n\} : w_i = \sigma \} .$$

$\Sigma = \{a, b, c\}$

$$|baabcbacaa|_c = 2$$

About words III

Let Σ be an alphabet, $u = u_1 \dots u_n \in \Sigma^*$ and $v = v_1 \dots v_m \in \Sigma^*$.

Definition

The *concatenation* of the words u and v is the word $w \in \Sigma^*$ defined as follows :

$$w = w_1 \dots w_{n+m} \qquad \text{where} \qquad \begin{cases} w_i = u_i & \text{if } 0 \leq i \leq n \\ w_{n+i} = v_i & \text{if } 0 \leq i \leq m. \end{cases}$$

We often denote the concatenation of u and v by $u.v$ or simply uv.

About words III

Let Σ be an alphabet, $u = u_1 \dots u_n \in \Sigma^*$ and $v = v_1 \dots v_m \in \Sigma^*$.

Definition

The *concatenation* of the words u and v is the word $w \in \Sigma^*$ defined as follows :

$$w = w_1 \dots w_{n+m} \qquad \text{where} \qquad \begin{cases} w_i = u_i & \text{if } 0 \leq i \leq n \\ w_{n+i} = v_i & \text{if } 0 \leq i \leq m. \end{cases}$$

We often denote the concatenation of u and v by $u.v$ or simply uv.

$\Sigma = \{0, 1\}$, $u = 01$ and $v = 1001110$.

About words III

Let Σ be an alphabet, $u = u_1 \dots u_n \in \Sigma^*$ and $v = v_1 \dots v_m \in \Sigma^*$.

Definition

The *concatenation* of the words u and v is the word $w \in \Sigma^*$ defined as follows :

$$w = w_1 \dots w_{n+m} \qquad \text{where} \qquad \begin{cases} w_i = u_i & \text{if } 0 \leq i \leq n \\ w_{n+i} = v_i & \text{if } 0 \leq i \leq m. \end{cases}$$

We often denote the concatenation of u and v by $u.v$ or simply uv.

$\Sigma = \{0, 1\}$, $u = 01$ and $v = 1001110$.

$$uv = 011001110$$
$$vu = 100111001$$

About words IV

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

For all $k \in \mathbb{N}_0$, we set

$$w^k = \underbrace{w \dots w}_{k \text{ times}}.$$

About words IV

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

For all $k \in \mathbb{N}_0$, we set

$$w^k = \underbrace{w \dots w}_{k \text{ times}}.$$

We also set $w^0 = \varepsilon$.

About words IV

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

For all $k \in \mathbb{N}_0$, we set

$$w^k = \underbrace{w \dots w}_{k \text{ times}}.$$

We also set $w^0 = \varepsilon$.

$\Sigma = \{\clubsuit, \heartsuit\}$, $w = \heartsuit\clubsuit$

About words IV

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

For all $k \in \mathbb{N}_0$, we set

$$w^k = \underbrace{w \dots w}_{k \text{ times}}.$$

We also set $w^0 = \varepsilon$.

$\Sigma = \{\clubsuit, \heartsuit\}$, $w = \heartsuit\clubsuit$

$$w^4$$

About words IV

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

For all $k \in \mathbb{N}_0$, we set

$$w^k = \underbrace{w \dots w}_{k \text{ times}}.$$

We also set $w^0 = \varepsilon$.

$\Sigma = \{\clubsuit, \heartsuit\}$, $w = \heartsuit\clubsuit$

$$w^4 = \heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit = (\heartsuit\clubsuit)^4$$

About words V

Let Σ be an alphabet and $w \in \Sigma^*$.

About words V

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

A word $u \in \Sigma^*$ is a *prefix* of the word w if there exists a word $y \in \Sigma^*$ such that
$$w = uy.$$

About words V

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

A word $u \in \Sigma^*$ is a *prefix* of the word w if there exists a word $y \in \Sigma^*$ such that

$$w = uy.$$

Definition

A word $u \in \Sigma^*$ is a *suffix* of the word w if there exists a word $x \in \Sigma^*$ such that

$$w = xu.$$

About words V

Let Σ be an alphabet and $w \in \Sigma^*$.

Definition

A word $u \in \Sigma^*$ is a *prefix* of the word w if there exists a word $y \in \Sigma^*$ such that

$$w = uy.$$

Definition

A word $u \in \Sigma^*$ is a *suffix* of the word w if there exists a word $x \in \Sigma^*$ such that

$$w = xu.$$

Definition

A word $u \in \Sigma^*$ is a *factor* of the word w if there exists two words $x, y \in \Sigma^*$ such that

$$w = xuy.$$

About words V

$\Sigma = \{a, b, c\}, \; w = baccabab.$

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes :

About words V

$\Sigma = \{a, b, c\},\ w = baccabab.$

Prefixes : ε

About words V

$\Sigma = \{a, b, c\},\ w = baccabab.$

Prefixes : ε, b

About words V

$\Sigma = \{a, b, c\},\ w = baccabab.$

Prefixes : ε, b, ba

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

Suffixes :

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

<u>Prefixes</u> : ε, b, ba, bac, $bacc$, $bacca$, $baccab$, $baccaba$, $baccabab$

<u>Suffixes</u> : ε

About words V

$\Sigma = \{a, b, c\},\ w = baccabab.$

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

Suffixes : ε, b

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes : ε, b, ba, bac, $bacc$, $bacca$, $baccab$, $baccaba$, $baccabab$

Suffixes : ε, b, ab

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes : ε, b, ba, bac, $bacc$, $bacca$, $baccab$, $baccaba$, $baccabab$

Suffixes : ε, b, ab, bab, $abab$, $cabab$, $ccabab$, $accabab$, $baccabab$

About words V

$\Sigma = \{a, b, c\}$, $w = baccabab$.

Prefixes : ε, b, ba, bac, $bacc$, $bacca$, $baccab$, $baccaba$, $baccabab$

Suffixes : ε, b, ab, bab, $abab$, $cabab$, $ccabab$, $accabab$, $baccabab$

Factors :

Length	Factors
0	ε

About words V

$\Sigma = \{a, b, c\}, \ w = baccabab.$

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

Suffixes : $\varepsilon, b, ab, bab, abab, cabab, ccabab, accabab, baccabab$

Factors :

Length	Factors
0	ε
1	a, b, c

About words V

$\Sigma = \{a, b, c\}, \; w = baccabab.$

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

Suffixes : $\varepsilon, b, ab, bab, abab, cabab, ccabab, accabab, baccabab$

Factors :

Length	Factors
0	ε
1	a, b, c
2	ab, ac, ba, ca, cc

About words V

$\Sigma = \{a, b, c\}, \ w = baccabab.$

Prefixes : $\varepsilon, b, ba, bac, bacc, bacca, baccab, baccaba, baccabab$

Suffixes : $\varepsilon, b, ab, bab, abab, cabab, ccabab, accabab, baccabab$

Factors :

Length	Factors
0	ε
1	a, b, c
2	ab, ac, ba, ca, cc
3	$aba, acc, bac, bab, cab, cca$
4	$abab, acca, bacc, caba, ccab$
5	$accab, bacca, cabab, ccaba$
6	$accaba, baccab, ccabab$
7	$accabab, baccaba$
8	$baccabab$

Combinatorics on words

Combinatorics on words

Proposition

Let $\Sigma = \{0,1\}$ and $w \in \Sigma^*$.

$$|w| \geq 4 \Rightarrow \exists u \in \Sigma^* : u^2 \in \text{Fac}(w).$$

Proposition

Let $\Sigma = \{0, 1\}$ and $w \in \Sigma^*$.

$$|w| \geq 4 \Rightarrow \exists u \in \Sigma^* : u^2 \in \mathrm{Fac}(w).$$

Definition

An *infinite word* over an alphabet Σ is a map

$$w : \mathbb{N} \to \Sigma.$$

We denote by Σ^ω the set of infinite words over Σ.

Thue-Morse sequence

0

Thue-Morse sequence

0

1

Thue-Morse sequence

01

1

Thue-Morse sequence

01

10

Thue-Morse sequence

0110

10

Thue-Morse sequence

0110

1001

Thue-Morse sequence

01101001

1001

Thue-Morse sequence

01101001

10010110

Thue-Morse sequence

0110100110010110

10010110

Thue-Morse sequence

0110100110010110 . . .

$$0110100110010110\ldots$$

Definition

An *overlap* is a finite word of the form

$$auaua,$$

where Σ is an alphabet, $u \in \Sigma^*$ and $a \in \Sigma$.

Thue-Morse sequence

$$0110100110010110\ldots$$

Definition

An *overlap* is a finite word of the form

$$auaua,$$

where Σ is an alphabet, $u \in \Sigma^*$ and $a \in \Sigma$.

Proposition

The Thue-Morse sequence is an infinite word without overlap.

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:$\{aa, b, ca\}$

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:$\{aa, b, ca\}$, $\{w \in \Sigma^* : |w|_a = |w|_b\}$

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:$\{aa, b, ca\}$, $\{w \in \Sigma^* : |w|_a = |w|_b\}$
$\Gamma = \{0, 1\}$: $\{u \in \Gamma^* : |u|_0 \in 2\,\mathbb{N}\}$

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:$\{aa, b, ca\}$, $\{w \in \Sigma^* : |w|_a = |w|_b\}$
$\Gamma = \{0, 1\}$: $\{u \in \Gamma^* : |u|_0 \in 2\,\mathbb{N}\}$
DNA

Language

Let Σ be an alphabet.

Definition

A *language* over Σ is subset of Σ^*.
The empty language is denoted by $\emptyset$.

Remark

$\emptyset \neq \{\varepsilon\}$!

$\Sigma = \{a, b, c\}$:$\{aa, b, ca\}$, $\{w \in \Sigma^* : |w|_a = |w|_b\}$
$\Gamma = \{0, 1\}$: $\{u \in \Gamma^* : |u|_0 \in 2\,\mathbb{N}\}$
DNA
War

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\}$, $L = \{a, aa\}, M = \{b\}$:

$$LM \;=$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \ldots w_n : \forall i \in \{1, \ldots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$LM = \{ab, aab\}$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{aligned} LM &= \{ab, aab\} \\ L^3 &= \end{aligned}$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{rcl} LM & = & \{ab, aab\} \\ L^3 & = & \{a.a.a, a.a.aa, a.aa.aa, aa.aa.aa\} \end{array}$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{aligned} LM &= \{ab, aab\} \\ L^3 &= \{a.a.a, a.a.aa, a.aa.aa, aa.aa.aa\} \\ M^4 &= \end{aligned}$$

Concatenation of languages

Definition

The *concatenation* of two languages L, M is the language

$$LM = \{uv : u \in L, v \in M\}.$$

In particular, if $n > 0$, we set

$$L^n = \{w_1 \dots w_n : \forall i \in \{1, \dots, n\}, w_i \in L\}$$

and we set $L^0 = \{\varepsilon\}$.

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{lcl} LM & = & \{ab, aab\} \\ L^3 & = & \{a.a.a, a.a.aa, a.aa.aa, aa.aa.aa\} \\ M^4 & = & \{bbbb\} \end{array}$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$L^* \quad = $$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$L^* \quad = \quad \{a^n : n \in \mathbb{N}\}$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{lcl} L^* & = & \{a^n : n \in \mathbb{N}\} \\ (LM)^* & = & \end{array}$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{lcl} L^* & = & \{a^n : n \in \mathbb{N}\} \\ (LM)^* & = & \{\varepsilon, ab, aab, abab, abaab, aabab, aabaab, \ldots\} \end{array}$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{lcl} L^* & = & \{a^n : n \in \mathbb{N}\} \\ (LM)^* & = & \{\varepsilon, ab, aab, abab, abaab, aabab, aabaab, \ldots\} \\ \emptyset^* & = & \end{array}$$

Kleene star

Definition

The *Kleene star* of a language L is the language

$$L^* = \bigcup_{i \geq 0} L^i.$$

$\Sigma = \{a, b\},\ L = \{a, aa\}, M = \{b\}$:

$$\begin{array}{lcl} L^* & = & \{a^n : n \in \mathbb{N}\} \\ (LM)^* & = & \{\varepsilon, ab, aab, abab, abaab, aabab, aabaab, \ldots\} \\ \emptyset^* & = & \{\varepsilon\} \end{array}$$

One more example

One more example

2049

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2		

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5		

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	$(3, 1, 1, 4, 4)$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	$(3, 1, 1, 4, 4)$

In general, in base $b \in \mathbb{N}_{\geq 2}$, the alphabet is

$$A_b := \{0, \ldots, b-1\}.$$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	$(3, 1, 1, 4, 4)$

In general, in base $b \in \mathbb{N}_{\geq 2}$, the alphabet is

$$A_b := \{0, \ldots, b-1\}.$$

Examples of languages : $\mathrm{rep}_b(\mathbb{N})$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	$(3, 1, 1, 4, 4)$

In general, in base $b \in \mathbb{N}_{\geq 2}$, the alphabet is

$$A_b := \{0, \ldots, b-1\}.$$

Examples of languages : $\text{rep}_b(\mathbb{N}), 0^* \text{rep}_b(\mathbb{N})$

One more example

2049

Base	Decomposition	Representation
10	$2 \times 10^3 + 4 \times 10^1 + 9 \times 10^0$	$(2, 0, 4, 9)$
2	$1 \times 2^{11} + 1 \times 2^0$	$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$
5	$3 \times 5^4 + 5^3 + 5^2 + 4 \times 5^1 + 4 \times 5^0$	$(3, 1, 1, 4, 4)$

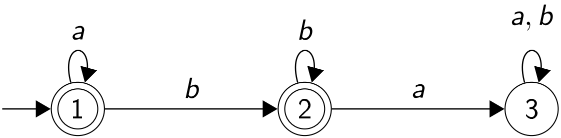
In general, in base $b \in \mathbb{N}_{\geq 2}$, the alphabet is

$$A_b := \{0, \ldots, b-1\}.$$

Examples of languages : $\text{rep}_b(\mathbb{N}), 0^* \text{rep}_b(\mathbb{N}), \text{rep}_b(2\,\mathbb{N}), \ldots$

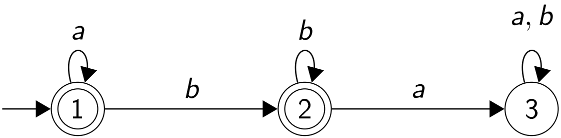
Automaton

Deterministic finite automaton (DFA) : $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$



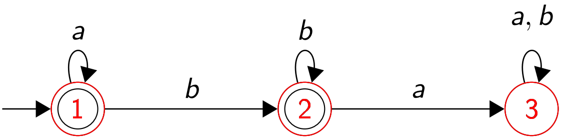
Automaton

Deterministic finite automaton (DFA) : $\mathscr{A} = ({\color{red}Q}, q_0, F, \Sigma, \delta)$



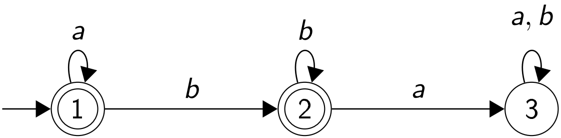
Automaton

Deterministic finite automaton (DFA) : $\mathscr{A} = (\{1, 2, 3\}, q_0, F, \Sigma, \delta)$

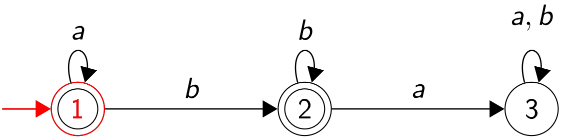


Automaton

Deterministic finite automaton (DFA) : $\mathscr{A} = (\{1, 2, 3\}, q_0, F, \Sigma, \delta)$

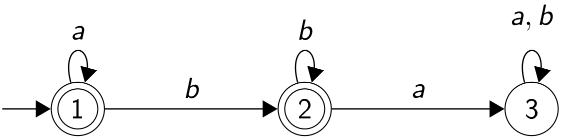


Deterministic finite automaton (DFA) : $\mathscr{A} = (\{1,2,3\}, 1, F, \Sigma, \delta)$



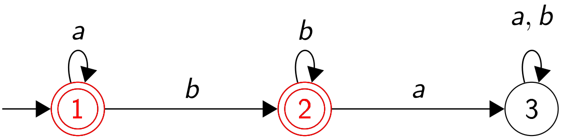
Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, F, \Sigma, \delta)$



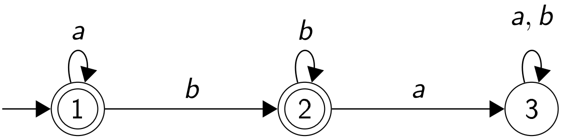
Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \Sigma, \delta)$



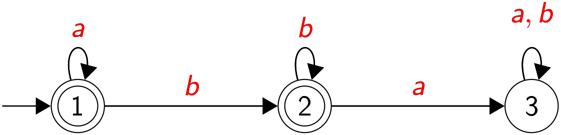
Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \Sigma, \delta)$



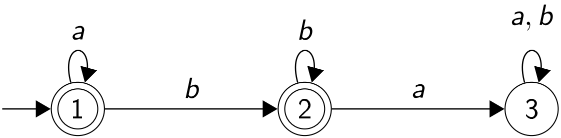
Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



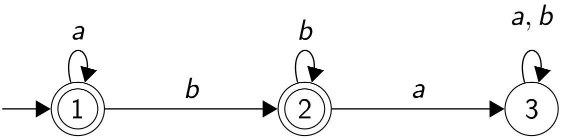
Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



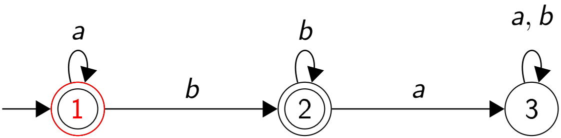
Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



$\delta(1, a)$

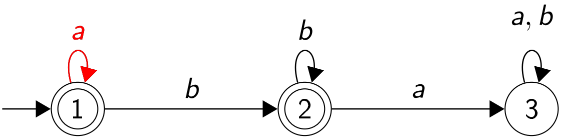
DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$



$\delta(1, a)$

Automaton

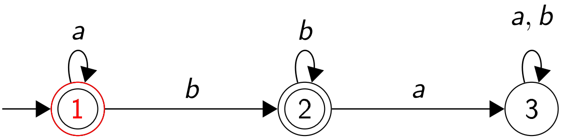
DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$



$\delta(1, a)$

Automaton

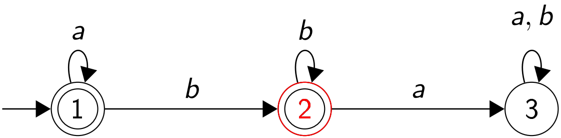
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



$$\delta(1, a) = 1$$

Automaton

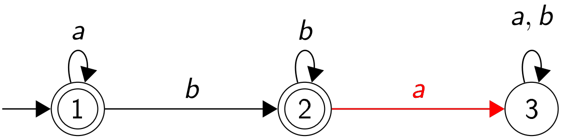
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



$\delta(2, a)$

Automaton

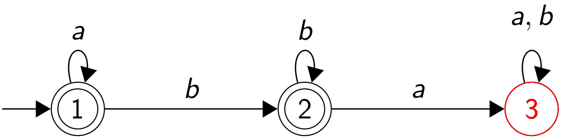
DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$



$\delta(2, a)$

Automaton

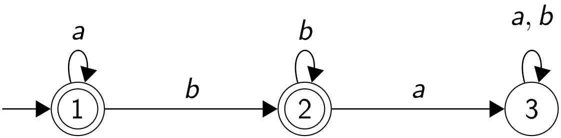
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



$$\delta(2, a) = 3$$

Automaton

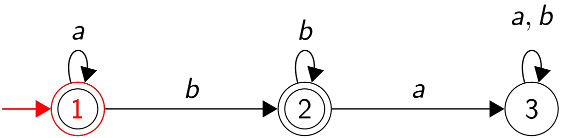
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



$aaab$

Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

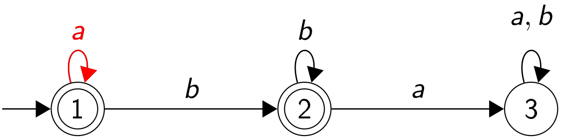


$aaab$

1

Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$

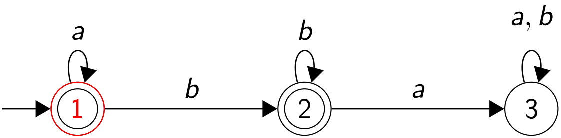


$aaab$

$1 \xrightarrow{a}$

Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

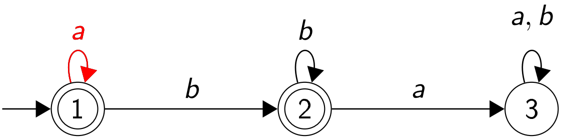


a *aab*

$1 \xrightarrow{a} 1$

Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$

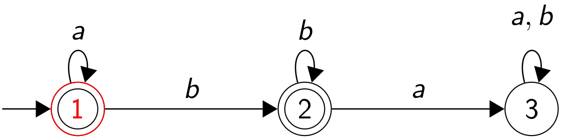


$a \quad aab$

$1 \xrightarrow{a} 1 \xrightarrow{a}$

Automaton

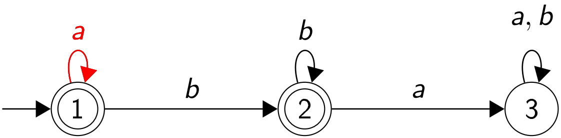
DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$



$aa \quad ab$

$1 \xrightarrow{a} 1 \xrightarrow{a} 1$

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$

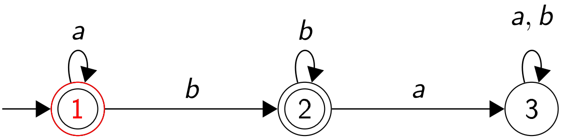


$aa \quad ab$

$1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{a}$

Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

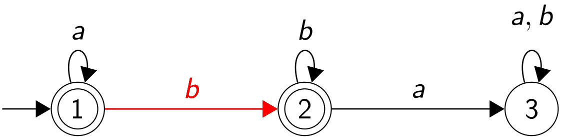


$aaa \quad b$

$1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{a} 1$

Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$

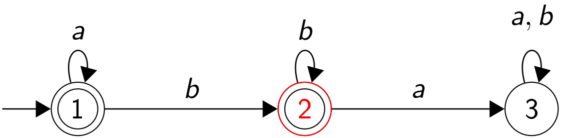


$aaa \quad b$

$1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b}$

Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

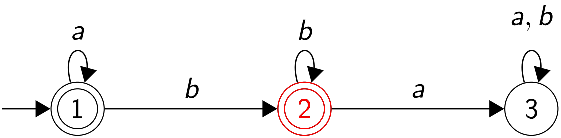


$aaab$

$1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b} 2$

Automaton

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$



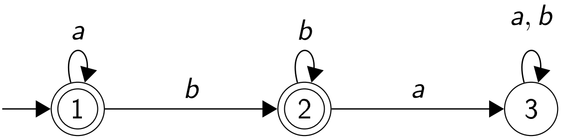
aaab

$1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \qquad \delta(1, aaab) = 2$

Final $\rightsquigarrow$ Accepted word

Automaton

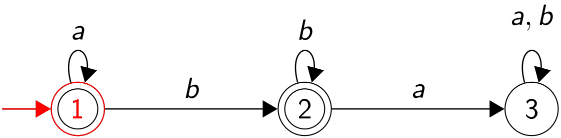
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



ba

Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$

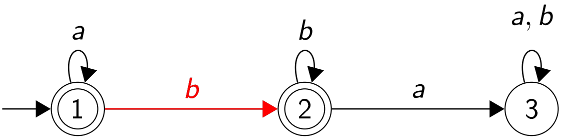


ba

1

Automaton

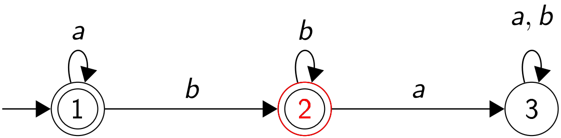
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$



ba

$1 \xrightarrow{b}$

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

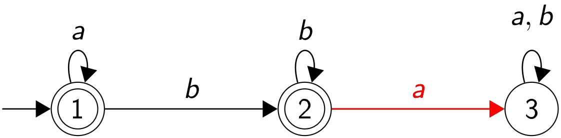


$b \quad a$

$1 \xrightarrow{b} 2$

Automaton

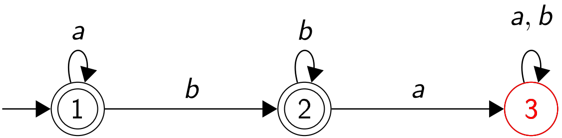
DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a, b\}, \delta)$



$b \quad a$

$1 \xrightarrow{b} 2 \xrightarrow{a}$

DFA : $\mathscr{A} = (\{1, 2, 3\}, 1, \{1, 2\}, \{a, b\}, \delta)$

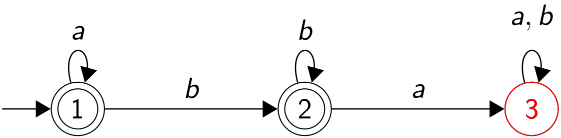


ba

$1 \xrightarrow{b} 2 \xrightarrow{a} 3$

Automaton

DFA : $\mathscr{A} = (\{1,2,3\}, 1, \{1,2\}, \{a,b\}, \delta)$



ba

$1 \xrightarrow{b} 2 \xrightarrow{a} 3 \qquad \delta(1, ba) = 3$

Not final $\rightsquigarrow$ Non-accepted word

Automata and languages I

Definition

For each automaton $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$, the *language accepted by* $\mathscr{A}$ is the set

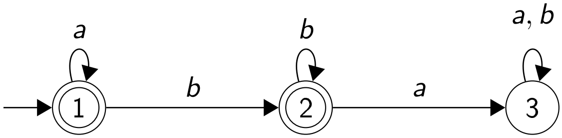
$$L(\mathscr{A}) := \{w \in \Sigma^* : \delta(q_0, w) \in F\}.$$

Automata and languages I

Definition

For each automaton $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$, the *language accepted by* $\mathscr{A}$ is the set

$$L(\mathscr{A}) := \{w \in \Sigma^* : \delta(q_0, w) \in F\}.$$

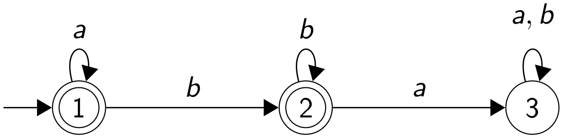


Automata and languages I

Definition

For each automaton $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$, the *language accepted by* $\mathscr{A}$ is the set

$$L(\mathscr{A}) := \{w \in \Sigma^* : \delta(q_0, w) \in F\}.$$



$L(\mathscr{A}) = a^* b^*$

Automata and languages II

Let Σ be an alphabet.

Recall

A language over Σ is a subset of Σ^*.

Automata and languages II

Let Σ be an alphabet.

Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

Automata and languages II

Let Σ be an alphabet.

Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

$a^* b^*$

Automata and languages II

Let Σ be an alphabet.

Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

$a^* b^*, \{aa, b, ca\}$

Automata and languages II

Let Σ be an alphabet.

Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

$a^* b^*, \{aa, b, ca\}, \{u \in \{0,1\} : |u|_1 \in 2\,\mathbb{N}\}$

Automata and languages II

Let Σ be an alphabet.

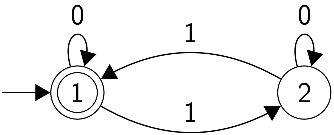
Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

$a^* b^*, \{aa, b, ca\}, \{u \in \{0,1\} : |u|_1 \in 2\,\mathbb{N}\}$



Automata and languages II

Let Σ be an alphabet.

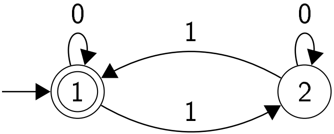
Recall

A language over Σ is a subset of Σ^*.

Definition

A *regular language* is a language accepted by a DFA.

$a^* b^*, \{aa, b, ca\}, \{u \in \{0,1\} : |u|_1 \in 2\mathbb{N}\}$



$\{a^n b^n : n \in \mathbb{N}\}$

DFA vs NDFA

Deterministic Finite Automaton – Non-Deterministic Finite Automaton

DFA vs NDFA

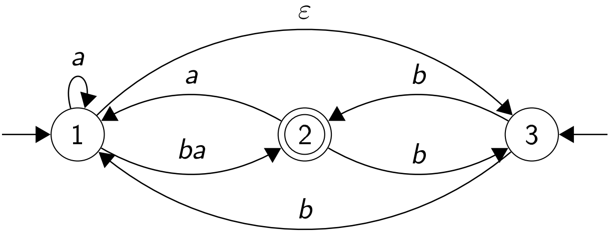
Deterministic Finite Automaton – Non-Deterministic Finite Automaton

	DFA	NDFA
Initial state	q_0	$I \subseteq Q$, $\sharp I \geq 1$
Transitions	Function on Σ	Relation on Σ^*

DFA vs NDFA

Deterministic Finite Automaton – Non-Deterministic Finite Automaton

	DFA	NDFA
Initial state	q_0	$I \subseteq Q$, $\sharp I \geq 1$
Transitions	Function on Σ	Relation on Σ^*

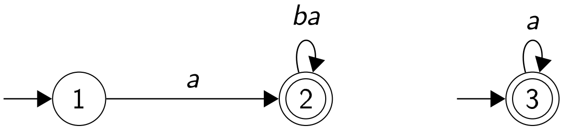


Why NDFA ?

$a(ba)^* \cup a^*$

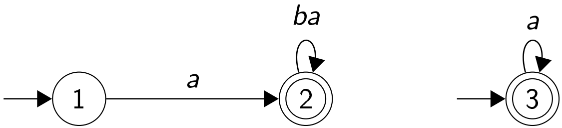
Why NDFA ?

$a(ba)^* \cup a^*$



Why NDFA ?

$a(ba)^* \cup a^*$

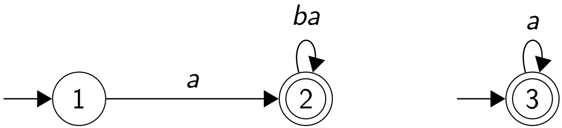


Proposition [Rabin-Scott]

Every language accepted by a NDFA is accepted by a DFA.

Why NDFA ?

$a(ba)^* \cup a^*$



Proposition [Rabin-Scott]

Every language accepted by a NDFA is accepted by a DFA.

Existence of an algorithm

Product of automata

Given two regular languages L and M, is the language $L \cap M$ also regular ?

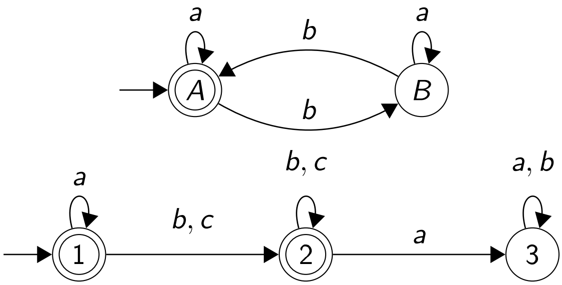
Product of automata

Given two regular languages L and M, is the language $L \cap M$ also regular ?

Product of automata

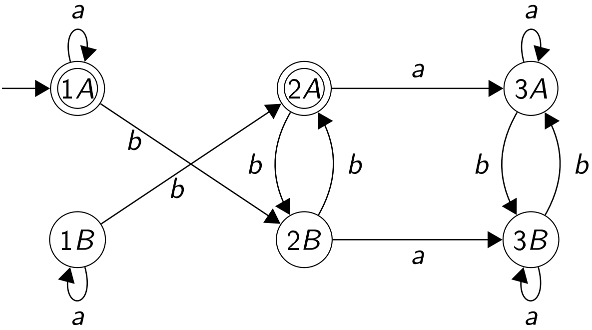
Given two regular languages L and M, is the language $L \cap M$ also regular ?

Product of automata



Given two regular languages L and M, is the language $L \cap M$ also regular ?

Product of automata



Complete automaton

Definition

An automaton $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ is *complete* if $\forall q \in Q, \forall \sigma \in \Sigma$,

$$\delta(q, \sigma)$$

is defined.

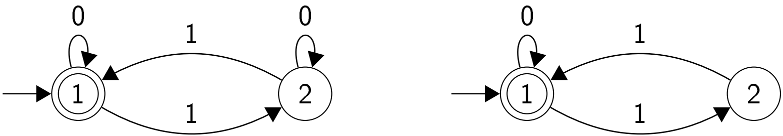
Complete automaton

Definition

An automaton $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ is *complete* if $\forall q \in Q, \forall \sigma \in \Sigma$,

$$\delta(q, \sigma)$$

is defined.



Accessible automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

A state $q \in Q$ is *accessible* if $\exists w \in \Sigma^*$ s.t.

$$\delta(q_0, w) = q.$$

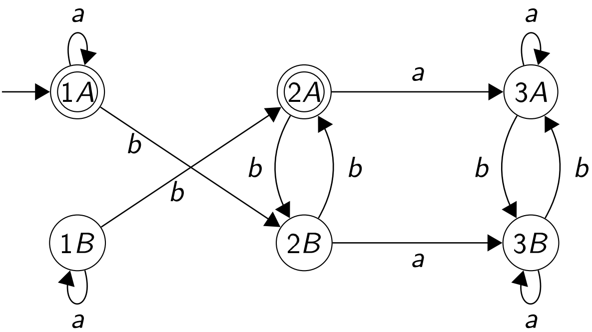
Accessible automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

A state $q \in Q$ is *accessible* if $\exists w \in \Sigma^*$ s.t.

$$\delta(q_0, w) = q.$$



Reduced automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

Two states $q, p \in Q$ are *distinguished* if $\exists w \in \Sigma^*$ s.t.

$$(\delta(q,w) \in F \text{ and } \delta(p,w) \notin F) \quad \text{or} \quad (\delta(q,w) \notin F \text{ and } \delta(p,w) \in F)\,.$$

Reduced automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

Two states $q, p \in Q$ are *distinguished* if $\exists w \in \Sigma^*$ s.t.

$$(\delta(q, w) \in F \text{ and } \delta(p, w) \notin F) \quad \text{or} \quad (\delta(q, w) \notin F \text{ and } \delta(p, w) \in F).$$

The automaton $\mathscr{A}$ is *reduced* if all its states are two by two distinguished.

Reduced automaton

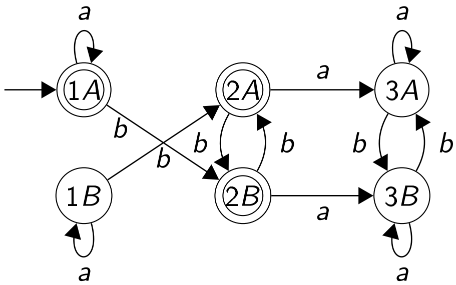
Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

Two states $q, p \in Q$ are *distinguished* if $\exists w \in \Sigma^*$ s.t.

$$(\delta(q, w) \in F \text{ and } \delta(p, w) \notin F) \quad \text{or} \quad (\delta(q, w) \notin F \text{ and } \delta(p, w) \in F)\,.$$

The automaton $\mathscr{A}$ is *reduced* if all its states are two by two distinguished.



Coaccessible automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

A state $q \in Q$ is *coaccessible* if $\exists w \in \Sigma^*$ s.t.

$$\delta(q, w) \in F.$$

Coaccessible automaton

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

A state $q \in Q$ is *coaccessible* if $\exists w \in \Sigma^*$ s.t.

$$\delta(q, w) \in F.$$

The automaton $\mathscr{A}$ is *coaccessible* if all its states are coaccessible.

Coaccessible automaton

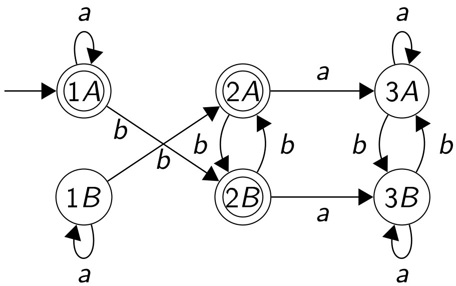
Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

A state $q \in Q$ is *coaccessible* if $\exists w \in \Sigma^*$ s.t.

$$\delta(q, w) \in F.$$

The automaton $\mathscr{A}$ is *coaccessible* if all its states are coaccessible.



Automaton with disjoint states

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

Two states $q, p \in Q$ are *disjoint* if $\forall w \in \Sigma^*$,

$$\delta(q, w) \in F \Rightarrow \delta(p, w) \notin F \text{ and } \delta(p, w) \in F \Rightarrow \delta(q, w) \notin F.$$

Automaton with disjoint states

Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

Two states $q, p \in Q$ are *disjoint* if $\forall w \in \Sigma^*$,

$$\delta(q, w) \in F \Rightarrow \delta(p, w) \notin F \text{ and } \delta(p, w) \in F \Rightarrow \delta(q, w) \notin F.$$

The automaton $\mathscr{A}$ *has disjoint states* if all its states are two by two disjoint.

Automaton with disjoint states

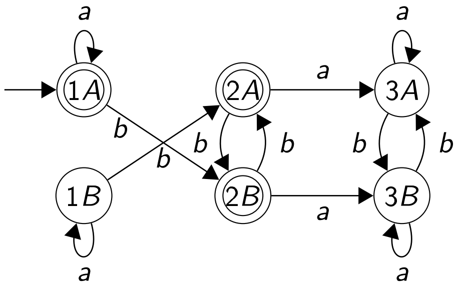
Let $\mathscr{A} = (Q, q_0, F, \Sigma, \delta)$ be an automaton.

Definition

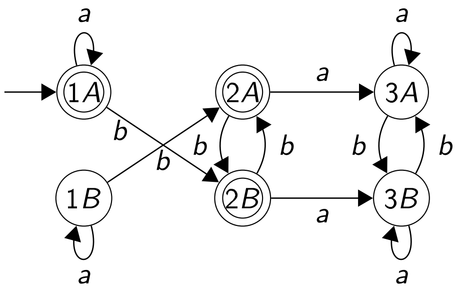
Two states $q, p \in Q$ are *disjoint* if $\forall w \in \Sigma^*$,

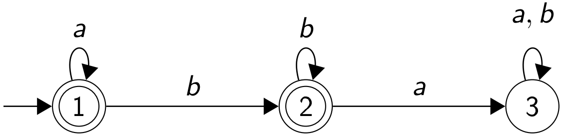
$$\delta(q, w) \in F \Rightarrow \delta(p, w) \notin F \text{ and } \delta(p, w) \in F \Rightarrow \delta(q, w) \notin F.$$

The automaton $\mathscr{A}$ *has disjoint states* if all its states are two by two disjoint.



Coaccessible + with disjoint states $\Rightarrow$ reduced


a
a
1A
2A
3A
b
b
b
b
b
b
1B
2B
3B
a
a
a


a
b
a, b
b
a
1
2
3

Minimal automaton II

Theorem

For any regular language L, there exists a unique (up to isomorphism) minimal automaton accepting L.

Minimal automaton II

Theorem

For any regular language L, there exists a unique (up to isomorphism) minimal automaton accepting L.

Theorem

An automaton is minimal if and only if it is accessible and reduced.

Minimal automaton II

Theorem

For any regular language L, there exists a unique (up to isomorphism) minimal automaton accepting L.

Theorem

An automaton is minimal if and only if it is accessible and reduced.

One algorithm :

Minimal automaton II

Theorem

For any regular language L, there exists a unique (up to isomorphism) minimal automaton accepting L.

Theorem

An automaton is minimal if and only if it is accessible and reduced.

One algorithm :

1. Eject non accessible states

Minimal automaton II

Theorem

For any regular language L, there exists a unique (up to isomorphism) minimal automaton accepting L.

Theorem

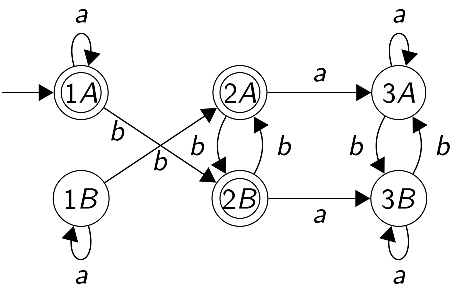
An automaton is minimal if and only if it is accessible and reduced.

One algorithm :

1. Eject non accessible states
2. Look for undistinguished states

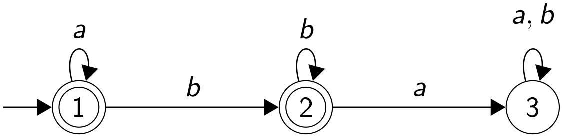
Example of minimization

1. Eject non accessible states
2. Look for undistinguished states



Example of minimization

1. Eject non accessible states
2. Look for undistinguished states



State complexity

Definition

The *state complexity* of a regular language is the number of states of its minimal automaton.

State complexity

Definition

The *state complexity* of a regular language is the number of states of its minimal automaton.

The state complexity of a^*b^* is 3.

Theorem [Alexeev, 2004]

The state complexity of the language $0^* \operatorname{rep}_b(m\mathbb{N})$ is

$$\min_{N \geq 0} \left\{ \frac{m}{\gcd(m, b^N)} + \sum_{n=0}^{N-1} \frac{b^n}{\gcd(b^n, m)} \right\}$$

The Thue-Morse set

$$\mathscr{T} = \{n \in \mathbb{N} : |\operatorname{rep}_2(n)|_1 \in 2\,\mathbb{N}\}$$

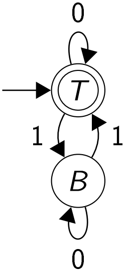
The Thue-Morse set

$$\mathscr{T} = \{n \in \mathbb{N} : |\operatorname{rep}_2(n)|_1 \in 2\,\mathbb{N}\}$$

Language to be studied : $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$

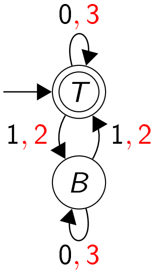
$$\mathscr{T} = \{n \in \mathbb{N} : |\operatorname{rep}_2(n)|_1 \in 2\,\mathbb{N}\}$$

Language to be studied : $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$



$$\mathscr{T} = \{n \in \mathbb{N} : |\operatorname{rep}_2(n)|_1 \in 2\,\mathbb{N}\}$$

Language to be studied : $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$



The result

$$\mathscr{T} = \{n \in \mathbb{N} : |\operatorname{rep}_2(n)|_1 \in 2\,\mathbb{N}\}$$

Theorem

Let $m \in \mathbb{N}$ and $p \in \mathbb{N}_{\geq 1}$.
Then the state complexity of the language $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$ is

$$2k + \left\lceil \frac{z}{p} \right\rceil$$

if $m = k2^z$ with k odd.

The method

Automaton	Language accepted
$\mathscr{A}_{\mathscr{T},2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\mathscr{T} \times \mathbb{N})$

The method

Automaton	Language accepted
$\mathscr{A}_{\mathscr{T},2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\mathscr{T} \times \mathbb{N})$
$\mathscr{A}_{m,2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\{(n, mn) : n \in \mathbb{N}\})$

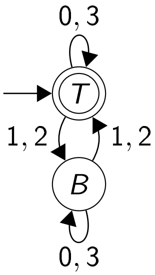
The method

Automaton	Language accepted
$\mathscr{A}_{\mathscr{T},2^p}$	$(0,0)^* \, \mathrm{rep}_{2^p}\,(\mathscr{T} \times \mathbb{N})$
$\mathscr{A}_{m,2^p}$	$(0,0)^* \, \mathrm{rep}_{2^p}\,(\{(n, mn) : n \in \mathbb{N}\})$
$\mathscr{A}_{\mathscr{T},2^p} \times \mathscr{A}_{m,2^p}$	$(0,0)^* \, \mathrm{rep}_{2^p}\,(\{(t, mt) : t \in \mathscr{T}\})$

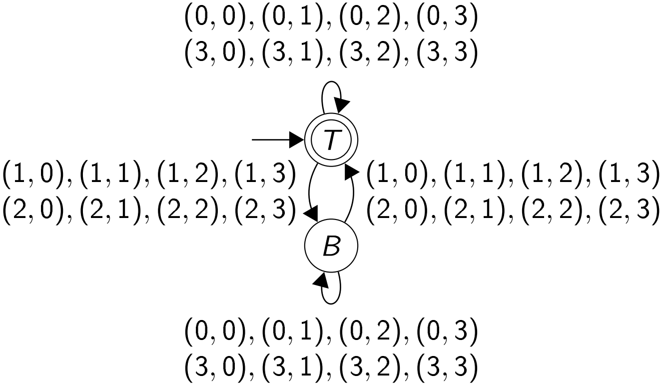
The method

Automaton	Language accepted
$\mathscr{A}_{\mathscr{T},2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\mathscr{T} \times \mathbb{N})$
$\mathscr{A}_{m,2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\{(n,mn) : n \in \mathbb{N}\})$
$\mathscr{A}_{\mathscr{T},2^p} \times \mathscr{A}_{m,2^p}$	$(0,0)^* \operatorname{rep}_{2^p}(\{(t,mt) : t \in \mathscr{T}\})$
$\pi\left(\mathscr{A}_{\mathscr{T},2^p} \times \mathscr{A}_{m,2^p}\right)$	$0^* \operatorname{rep}_{2^p}(m\mathscr{T})$

The automaton $\mathscr{A}_{\mathscr{T},2^p}$: $(0,0)^* \operatorname{rep}_{2^p}(\mathscr{T} \times \mathbb{N})$

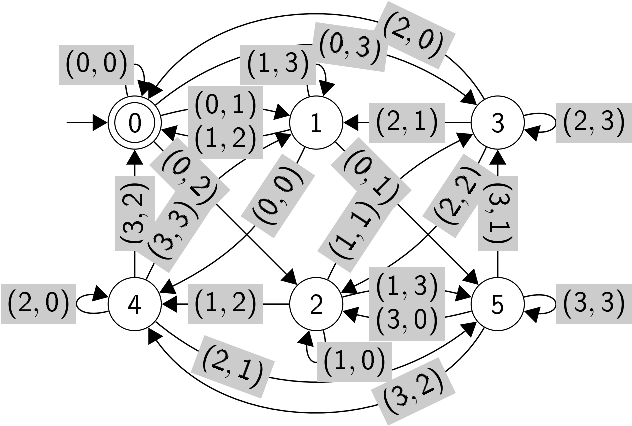


The automaton $\mathscr{A}_{\mathscr{T},2^p}$: $(0,0)^* \operatorname{rep}_{2^p}(\mathscr{T} \times \mathbb{N})$



$$\delta_{\mathscr{T},2^p}(X,(d,e)) = \begin{cases} X \text{ if } d \in \mathscr{T} \\ \overline{X} \text{ otherwise} \end{cases}$$

The automaton $\mathscr{A}_{m,b}$: $(0,0)^* \operatorname{rep}_b(\{(n, mn) : n \in \mathbb{N}\})$



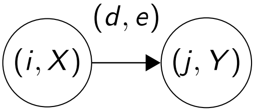
$$\delta_{m,b}(i, (d, e)) = j \qquad \Leftrightarrow \qquad bi + e = md + j$$

$\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p}$: $(0,0)^* \operatorname{rep}_{2^p}(\{(t, mt) : t \in \mathscr{T}\})$

$$(0, T), \ldots, (m-1, T) \qquad (0, B), \ldots (m-1, B)$$

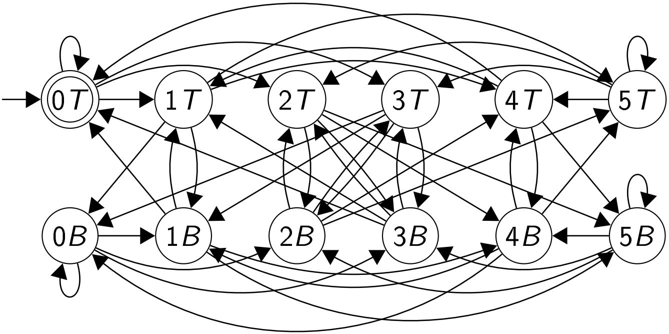
$\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p} : (0,0)^* \operatorname{rep}_{2^p}(\{(t, mt) : t \in \mathscr{T}\})$

$(0, T), \ldots, (m-1, T) \qquad (0, B), \ldots (m-1, B)$

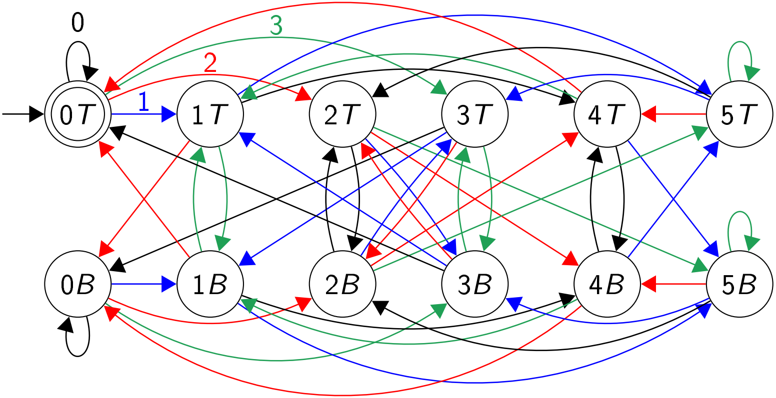


$$2^p i + e = md + j$$

$$Y = \begin{cases} X \text{ if } d \in \mathscr{T} \\ \overline{X} \text{ otherwise} \end{cases}$$



The automaton $\pi(\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p})$: $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$



Proposition

The automaton $\pi\left(\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p}\right)$ is

- deterministic
- accessible
- coaccessible

Proposition

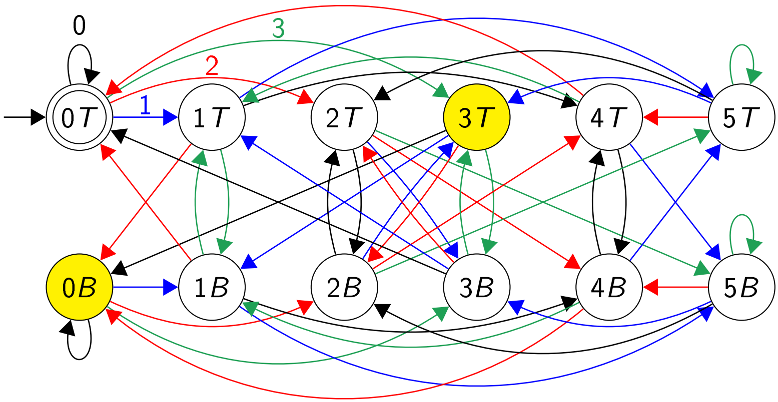
The automaton $\pi\left(\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p}\right)$ is

- deterministic
- accessible
- coaccessible

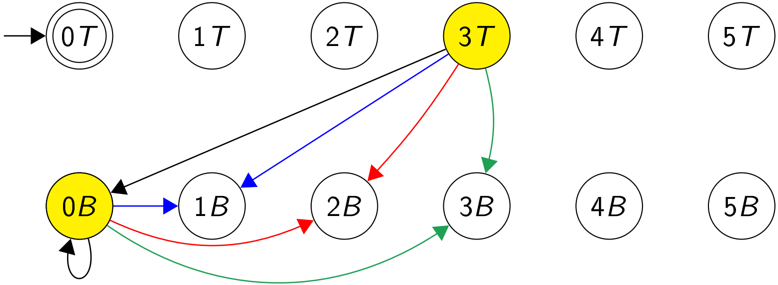
Proposition

In the automaton $\pi\left(\mathscr{A}_{m,2^p} \times \mathscr{A}_{\mathscr{T},2^p}\right)$, the states (i, T) and (i, B) are disjoint for all $i \in \{0, \ldots, m-1\}$.

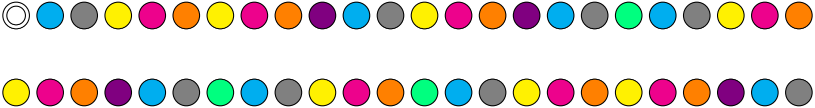
The automaton $\pi(\mathscr{A}_{6,4} \times \mathscr{A}_{\mathscr{T},4})$



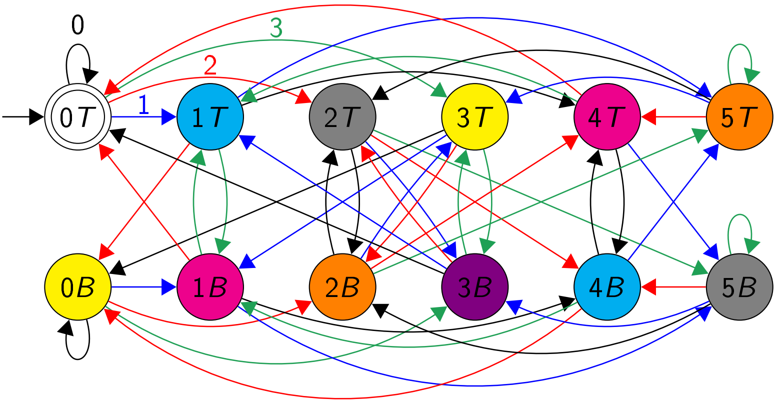
The automaton $\pi(\mathscr{A}_{6,4} \times \mathscr{A}_{\mathscr{T},4})$



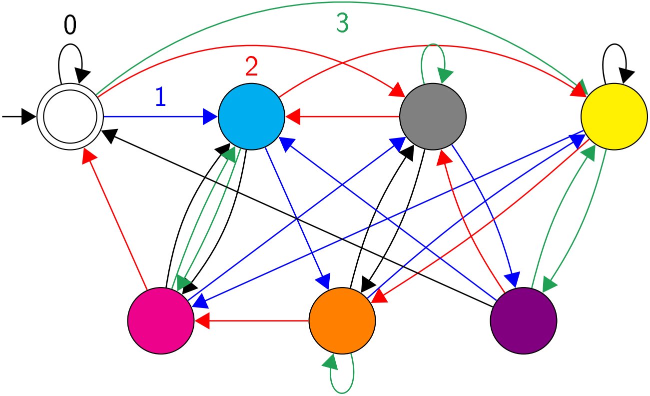
The automaton $\pi(\mathscr{A}_{24,4} \times \mathscr{A}_{\mathscr{T},4})$



Automaton recognizing $6\mathscr{T}$ in base 4



Automaton recognizing $6\mathscr{T}$ in base 4



Theorem

Let $m \in \mathbb{N}$ and $p \in \mathbb{N}_{\geq 1}$.
Then the state complexity of the language $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$ is

$$2k + \left\lceil \frac{z}{p} \right\rceil$$

if $m = k2^z$ with k odd.

Theorem

Let $m \in \mathbb{N}$ and $p \in \mathbb{N}_{\geq 1}$.
Then the state complexity of the language $0^* \operatorname{rep}_{2^p}(m\mathscr{T})$ is

$$2k + \left\lceil \frac{z}{p} \right\rceil$$

if $m = k2^z$ with k odd.

$2 \times 3 + \left\lceil \frac{1}{2} \right\rceil = 7$

The automaton $\pi\left(\mathscr{A}_{24,4} \times \mathscr{A}_{\mathscr{T},4}\right)$