

# Представления регулярных языков. Критерий регулярности



---

Теория формальных языков  
2022 г.



# Недетерминированные КА

## Определение

Недетерминированный конечный автомат (NFA) — это пятёрка  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ , где:

- $Q$  — множество состояний;
- $\Sigma$  — алфавит терминалов;
- $\delta$  — множество правил перехода вида  $\langle q_i, (a_i | \varepsilon), M_i \rangle$ , где  $q_i \in Q$ ,  $a_i \in \Sigma$ ,  $M_i \in 2^Q$ ;
- $q_0 \in Q$  — начальное состояние;
- $F \subseteq Q$  — множество конечных состояний.

Сокращаем:  $\langle q_1, a, q_2 \rangle \in \delta \Leftrightarrow \langle q_1, a, M \rangle \in \delta \ \& \ q_2 \in M$ .

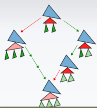


## Недетерминированные КА

### Определение

Недетерминированный конечный автомат (NFA) — это пятёрка  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ .

- $q \xrightarrow{\varepsilon} q' \Leftrightarrow (q = q') \vee \exists p_1, \dots, p_k (\langle q, \varepsilon, p_1 \rangle \in \delta \ \& \ \langle p_k, \varepsilon, q' \rangle \in \delta \ \& \ \forall i, 1 \leq i < k \langle p_i, \varepsilon, p_{i+1} \rangle \in \delta)$ .
- $q \xrightarrow{a} q' \Leftrightarrow \exists p, p' (q \xrightarrow{\varepsilon} p \ \& \ \langle p, a, p' \rangle \in \delta \ \& \ p' \xrightarrow{\varepsilon} q')$ .
- $q \xrightarrow{a_1 \dots a_k} q' \Leftrightarrow \exists p_1, \dots, p_{k-1} (q \xrightarrow{a_1} p_1 \ \& \ p_{k-1} \xrightarrow{a_k} q' \ \& \ \forall i, 1 \leq i < k-1 (p_i \xrightarrow{a_{i+1}} p_{i+1}))$ .



## Недетерминированные КА

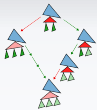
### Определение

Недетерминированный конечный автомат (NFA) — это пятёрка  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ .

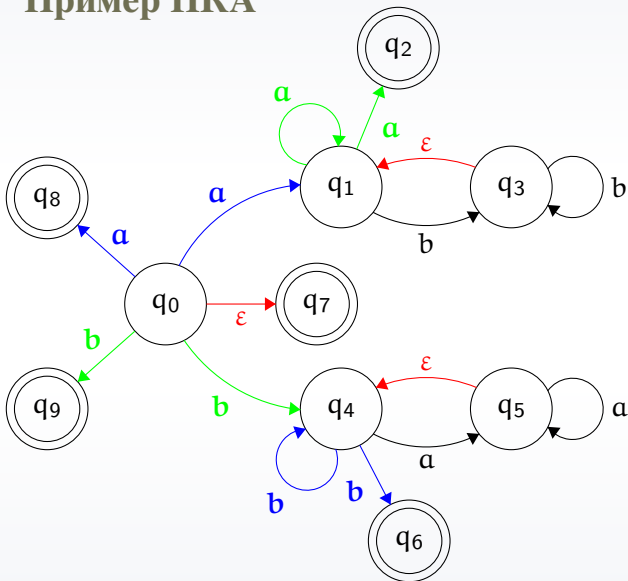
- $q \xrightarrow{\varepsilon} q' \Leftrightarrow (q = q') \vee \exists p_1, \dots, p_k (\langle q, \varepsilon, p_1 \rangle \in \delta \ \& \ \langle p_k, \varepsilon, q' \rangle \in \delta \ \& \ \forall i, 1 \leq i < k \langle p_i, \varepsilon, p_{i+1} \rangle \in \delta)$ .
- $q \xrightarrow{a} q' \Leftrightarrow \exists p, p' (q \xrightarrow{\varepsilon} p \ \& \ \langle p, a, p' \rangle \in \delta \ \& \ p' \xrightarrow{\varepsilon} q')$ .
- $q \xrightarrow{a_1 \dots a_k} q' \Leftrightarrow \exists p_1, \dots, p_{k-1} (q \xrightarrow{a_1} p_1 \ \& \ p_{k-1} \xrightarrow{a_k} q' \ \& \ \forall i, 1 \leq i < k - 1 (p_i \xrightarrow{a_{i+1}} p_{i+1}))$ .

### Определение

Язык  $\mathcal{L}$ , распознаваемый НКА  $\mathcal{A}$  — это множество слов  $\{w \mid \exists q \in F (q_0 \xrightarrow{w} q)\}$ .



## Пример НКА





# Детерминированный КА

## Определение

Детерминированный конечный автомат (DFA) — это пятёрка  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ , где:

- $Q$  — множество состояний;
- $\Sigma$  — алфавит терминалов;
- $\delta$  — множество правил перехода вида  $\langle q_i, a_i, q_j \rangle$ , где  $q_i, q_j \in Q, a_i \in \Sigma$ , причём  $\forall q_i, a_i \exists q_j (\langle q_i, a_i, q_j \rangle \in \delta \ \& \ \forall q_k (\langle q_i, a_i, q_k \rangle \in \delta \Rightarrow q_k = q_j))$ ;
- $q_0 \in Q$  — начальное состояние;
- $F \subseteq Q$  — множество конечных состояний.

$\varepsilon$ -переходов нет  $\Rightarrow q \xrightarrow{a} q' \Leftrightarrow \langle q, a, q' \rangle \in \delta$ .



# Детерминированный КА

## Определение

Детерминированный конечный автомат (DFA) — это пятёрка

$\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ , где:

- $Q$  — множество состояний;
- $\Sigma$  — алфавит терминалов;
- $\delta$  — множество правил перехода вида  $\langle q_i, a_i, q_j \rangle$ , где  $q_i, q_j \in Q, a_i \in \Sigma$ , причём  $\forall q_i, a_i \exists q_j (\langle q_i, a_i, q_j \rangle \in \delta \ \& \ \forall q_k (\langle q_i, a_i, q_k \rangle \in \delta \Rightarrow q_k = q_j))$ ;
- $q_0 \in Q$  — начальное состояние;
- $F \subseteq Q$  — множество конечных состояний.

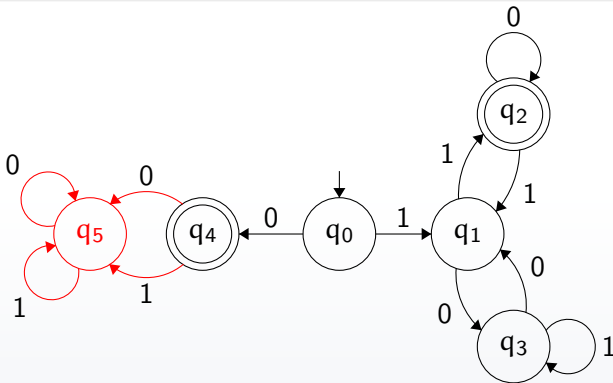
$\varepsilon$ -переходов нет  $\Rightarrow q \xrightarrow{a} q' \Leftrightarrow \langle q, a, q' \rangle \in \delta$ .

Язык  $\mathcal{L}$ , распознаваемый  $\mathcal{A}$  — это множество слов  $\{w \mid \exists q \in F (q_0 \xrightarrow{w} q)\}$ .

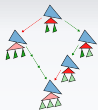


## Sink/trap state (состояние–ловушка)

«Ловушка» — не конечное состояние с переходами лишь в себя. Нужны для корректного задания DFA, но иногда по умолчанию не описываются.







## Детерминизация NFA

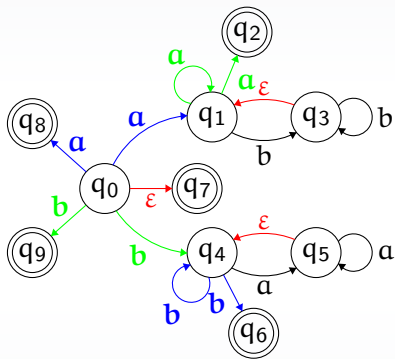
От  $\mathcal{A}$  к  $D(\mathcal{A})$

Состояния DFA  $D(\mathcal{A})$  — это состояния  $m_i \in 2^Q$ , где  $Q$  — состояния NFA  $\mathcal{A}$ .

- $m_0 = \{q_i \mid q_0 \xrightarrow{\varepsilon} q_i\};$
- $m_i \in F_D \Leftrightarrow \exists q_i, q_j \{q_i \in m_i \ \& \ q_j \in F(\mathcal{A}) \ \& \ q_i \xrightarrow{\varepsilon} q_j\};$
- $\langle m, a, m' \rangle \in \delta_D \Leftrightarrow m' = \{q_i \mid \exists q_j \in m (q_j \xrightarrow{a} q_i)\}.$



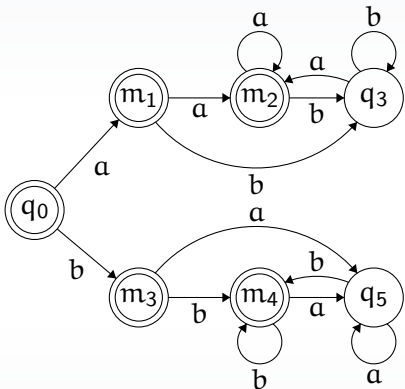
## Пример детерминизации



- $\{q_0\} \xrightarrow{a} \{q_1, q_8\},$   
 $\{q_0\} \xrightarrow{b} \{q_4, q_9\};$
- $\{q_1, q_8\} \xrightarrow{a} \{q_1, q_2\},$   
 $\{q_1, q_8\} \xrightarrow{b} \{q_3\}; \{q_1, q_8\} \sim m_1.$
- $\{q_1, q_2\} \xrightarrow{a} \{q_1, q_2\},$   
 $\{q_1, q_2\} \xrightarrow{b} \{q_3\}; \{q_1, q_2\} \sim m_2.$
- $\{q_3\} \xrightarrow{a} \{q_1, q_2\}, \{q_3\} \xrightarrow{b} \{q_3\};$
- $\{q_4, q_9\} \xrightarrow{b} \{q_4, q_6\},$   
 $\{q_4, q_9\} \xrightarrow{a} \{q_5\}; \{q_4, q_9\} \sim m_3;$
- $\{q_4, q_6\} \xrightarrow{b} \{q_4, q_6\},$   
 $\{q_4, q_6\} \xrightarrow{a} \{q_5\}; \{q_4, q_6\} \sim m_4.$
- $\{q_5\} \xrightarrow{b} \{q_4, q_6\}, \{q_5\} \xrightarrow{a} \{q_5\}.$



## Пример детерминизации



- $\{q_0\} \xrightarrow{a} \{q_1, q_8\},$   
 $\{q_0\} \xrightarrow{b} \{q_4, q_9\};$
- $\{q_1, q_8\} \xrightarrow{a} \{q_1, q_2\},$   
 $\{q_1, q_8\} \xrightarrow{b} \{q_3\}; \{q_1, q_8\} \sim m_1.$
- $\{q_1, q_2\} \xrightarrow{a} \{q_1, q_2\},$   
 $\{q_1, q_2\} \xrightarrow{b} \{q_3\}; \{q_1, q_2\} \sim m_2.$
- $\{q_3\} \xrightarrow{a} \{q_1, q_2\}, \{q_3\} \xrightarrow{b} \{q_3\};$
- $\{q_4, q_9\} \xrightarrow{b} \{q_4, q_6\},$   
 $\{q_4, q_9\} \xrightarrow{a} \{q_5\}; \{q_4, q_9\} \sim m_3;$
- $\{q_4, q_6\} \xrightarrow{b} \{q_4, q_6\},$   
 $\{q_4, q_6\} \xrightarrow{a} \{q_5\}; \{q_4, q_6\} \sim m_4.$
- $\{q_5\} \xrightarrow{b} \{q_4, q_6\}, \{q_5\} \xrightarrow{a} \{q_5\}.$



## Замыкания регулярных языков

Гомоморфизм над свободной полугруппой (множеством слов) полностью определяется значениями на буквах, поскольку по определению  $h(a_1 \circ a_2 \circ \dots \circ a_n) = h(a_1) \circ h(a_2) \circ \dots \circ h(a_n)$ .  
Здесь  $\circ$  — конкатенация.

### Утверждение

Пусть  $\mathcal{L}$  — регулярный язык над  $\Sigma$ . Тогда регулярны:

- язык  $\Sigma^* \setminus \mathcal{L}$ ;
- для любого гомоморфизма  $h$  язык  $\{h(w) \mid w \in \mathcal{L}\}$ ;
- для любого гомоморфизма  $h$  язык  $\{w \mid h(w) \in \mathcal{L}\}$ .



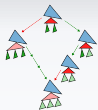
## Замыкания регулярных языков

### Утверждение

Пусть  $\mathcal{L}$  — регулярный язык над  $\Sigma$ . Тогда регулярны:

- язык  $\Sigma^* \setminus \mathcal{L}$ ;
- для любого гомоморфизма  $h$  язык  $\{h(w) \mid w \in \mathcal{L}\}$ ;
- для любого гомоморфизма  $h$  язык  $\{w \mid h(w) \in \mathcal{L}\}$ .

Рассмотрим DFA  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ , распознающий  $\mathcal{L}$ . Построим  $\mathcal{A}' = \langle Q, \Sigma, q_0, Q \setminus F, \delta \rangle$ . Тогда  $w \notin \mathcal{L} \Leftrightarrow w \in \mathcal{L}(\mathcal{A}')$ .



## Замыкания регулярных языков

### Утверждение

Пусть  $\mathcal{L}$  — регулярный язык над  $\Sigma$ . Тогда регулярны:

- язык  $\Sigma^* \setminus \mathcal{L}$ ;
- для любого гомоморфизма  $h$  язык  $\{h(w) \mid w \in \mathcal{L}\}$ ;
- для любого гомоморфизма  $h$  язык  $\{w \mid h(w) \in \mathcal{L}\}$ .

Рассмотрим регулярное выражение  $R$  такое, что  $\mathcal{L}(R) = \mathcal{L}$ .  
Заменим в нём все  $a_i \in \Sigma$  на  $h(a_i)$ . Полученное таким образом выражение  $R'$  также регулярно, причём  $\mathcal{L}(R') = h(\mathcal{L})$ .



## Замыкания регулярных языков

### Утверждение

Пусть  $\mathcal{L}$  — регулярный язык над  $\Sigma$ . Тогда регулярны:

- язык  $\Sigma^* \setminus \mathcal{L}$ ;
- для любого гомоморфизма  $h$  язык  $\{h(w) \mid w \in \mathcal{L}\}$ ;
- для любого гомоморфизма  $h$  язык  $\{w \mid h(w) \in \mathcal{L}\}$ .

Рассмотрим DFA  $\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$ , распознающий  $\mathcal{L}$ . Построим  $\mathcal{A}' = \langle Q, \Sigma, q_0, F, \delta' \rangle$  такой, что

$\langle q_i, a, q_j \rangle \in \delta' \Leftrightarrow q_i \xrightarrow{h(a)} q_j$  в исходном автомате  $\mathcal{A}$ .



## Примеры

Рассмотрим язык  $\mathcal{L}' = \{a^n b^m \mid n \neq m\}$ .

Предположим,  $\mathcal{L}'$  регулярен. Тогда  $a^* b^* \setminus \mathcal{L}' = \{a^n b^n\}$  также регулярен, а мы знаем, что это не так.  $\perp$





## Примеры

Рассмотрим язык  $\mathcal{L}' = \{a^n b^m \mid n \neq m\}$ .

Предположим,  $\mathcal{L}'$  регулярен. Тогда  $a^*b^* \setminus \mathcal{L}' = \{a^n b^n\}$  также регулярен, а мы знаем, что это не так.  $\perp$

Рассмотрим язык  $\mathcal{L}^f = \{(abaabb)^n b^n\}$ .

Попытка доказать его нерегулярность леммой о накачке породит перебор по накачиваемым строкам  $(abaabb)^+$ ,  $(abaabb)^*a$ ,  $(abaabb)^*ab$ ,  $(abaabb)^*aba$ ,  $(abaabb)^*abaa, \dots$ . Рассмотрим гомоморфизм  $h(a) = abaabb$ ,  $h(b) = b$ .  $h^{-1}(\mathcal{L}^f) = \{a^n b^n\}$ , который был бы регулярен, если бы  $\mathcal{L}^f$  был регулярен.  $\perp$



## Эквивалентность слов в DFA

Пусть дан DFA  $\mathcal{A}$ . Положим

$$w_1 \equiv_{\mathcal{A}} w_2 \Leftrightarrow \exists q_i (q_0 \xrightarrow{w_1} q_i \ \& \ q_0 \xrightarrow{w_2} q_i).$$

Если  $w_1 \equiv_{\mathcal{A}} w_2$ , тогда  $\forall z (w_1 z \in \mathcal{L}(\mathcal{A}) \Leftrightarrow w_2 z \in \mathcal{L}(\mathcal{A}))$ .



## Эквивалентность слов в DFA

Пусть дан DFA  $\mathcal{A}$ . Положим

$$w_1 \equiv_{\mathcal{A}} w_2 \Leftrightarrow \exists q_i (q_0 \xrightarrow{w_1} q_i \ \& \ q_0 \xrightarrow{w_2} q_i).$$

Если  $w_1 \equiv_{\mathcal{A}} w_2$ , тогда  $\forall z (w_1 z \in \mathcal{L}(\mathcal{A}) \Leftrightarrow w_2 z \in \mathcal{L}(\mathcal{A}))$ .

Рассмотрим более общее отношение. Положим

$w_1 \equiv_{\mathcal{L}} w_2 \Leftrightarrow \forall z (w_1 z \in \mathcal{L} \Leftrightarrow w_2 z \in \mathcal{L})$ . Это отношение разбивает  $\mathcal{L}$  на классы эквивалентности.

### Теорема Майхилла-Нероуда

Язык  $\mathcal{L}$  регулярен тогда и только тогда, когда множество его классов эквивалентности по  $\equiv_{\mathcal{L}}$  конечно.



## Критерий регулярности языка

### Теорема Майхилла-Нероуда

Язык  $\mathcal{L}$  регулярен тогда и только тогда, когда множество классов эквивалентности по  $\equiv_{\mathcal{L}}$  конечно.



## Критерий регулярности языка

### Теорема Майхилла-Нероуда

Язык  $\mathcal{L}$  регулярен тогда и только тогда, когда множество классов эквивалентности по  $\equiv_{\mathcal{L}}$  конечно.

$\Rightarrow$ : Пусть  $\mathcal{L}$  регулярен. Тогда он порождается некоторым DFA  $\mathcal{A}$  с конечным числом состояний  $N$ . Значит, множество  $\{q_i \mid q_0 \xrightarrow{w} q_i\}$  конечно, а для любых двух  $w_1, w_2$  таких, что  $q_0 \xrightarrow{w_1} q_i$  и  $q_0 \xrightarrow{w_2} q_i$ , выполняется  $w_1 \equiv_{\mathcal{L}} w_2$ .



## Критерий регулярности языка

### Теорема Майхилла-Нероуда

Язык  $\mathcal{L}$  регулярен тогда и только тогда, когда множество классов эквивалентности по  $\equiv_{\mathcal{L}}$  конечно.

$\Leftarrow$ : Пусть все слова в  $\Sigma^*$  принадлежат  $N$  классам эквивалентности  $A_1, \dots, A_n$  по  $\equiv_{\mathcal{L}}$ . Построим по ним DFA  $\mathcal{A}$ , распознающий  $\mathcal{L}$ . Классы  $A_i$  объявим состояниями.

- Начальным состоянием объявим класс эквивалентности  $A_0$  такой, что  $\varepsilon \in A_0$ .
- Конечными объявим такие  $A_j$ , что  $\forall w \in A_j (w \in \mathcal{L})$ .
- Если  $w \in A_i, w a_k \in A_j$ , тогда добавляем в  $\delta$  правило  $\langle A_i, a_k, A_j \rangle$ .  $\forall w_1, w_2 \in A_i, w_1 a_k$  и  $w_2 a_k$  всегда принадлежат одному и тому же  $A_j$ .



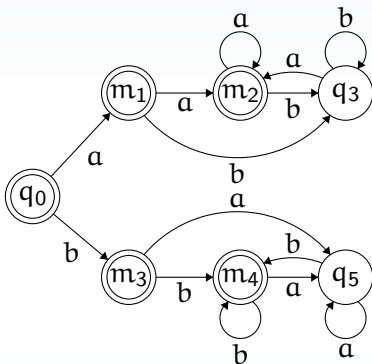
## Минимизация DFA

- 1 Построим таблицу всех двухэлементных множеств  $\{q_i, q_j\}$ ,  $q_i, q_j \in Q$ .
- 2 Пометим все множества  $\{q_i, q_j\}$  такие, что одно из  $q_i, q_j$  из  $F$ , а второе нет.
- 3 Пометим все множества  $\{q_i, q_j\}$  такие, что  $\exists a (q_i \xrightarrow{a} q'_1 \ \& \ q_j \xrightarrow{a} q'_2 \ \& \ \{q'_1, q'_2\} \text{ — помеченная пара})$ .
- 4 Продолжаем шаг 3, пока не будет появляться новых помеченных пар.

Пары, оставшиеся непомеченными, можно объединить.



## Пример минимизации

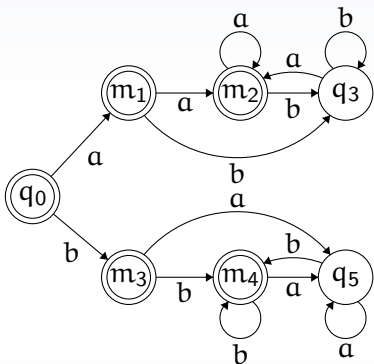


m <sub>1</sub>						
m <sub>2</sub>						
q <sub>3</sub>						
m <sub>3</sub>						
m <sub>4</sub>						
q <sub>5</sub>						
	q <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	q <sub>3</sub>	m <sub>3</sub>	m <sub>4</sub>





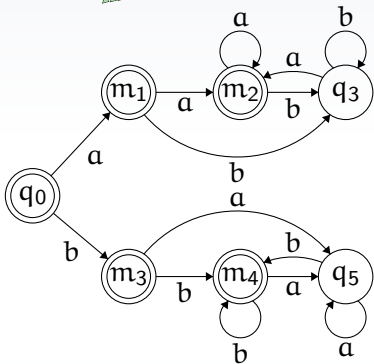
## Пример минимизации



$m_1$						
$m_2$						
$q_3$	✓	✓	✓			
$m_3$				✓		
$m_4$				✓		
$q_5$	✓	✓	✓		✓	✓
	$q_0$	$m_1$	$m_2$	$q_3$	$m_3$	$m_4$



## Пример минимизации



m <sub>1</sub>						
m <sub>2</sub>						
q <sub>3</sub>	✓	✓	✓			
m <sub>3</sub>				✓		
m <sub>4</sub>				✓		
q <sub>5</sub>	✓	✓	✓		✓	✓
	q <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	q <sub>3</sub>	m <sub>3</sub>	m <sub>4</sub>

$q_0 \xrightarrow{a} m_1, m_1 \xrightarrow{a} m_2$

$q_0 \xrightarrow{b} m_3, m_1 \xrightarrow{b} q_3$

$\{m_1, m_2\} \xrightarrow{a} m_2$

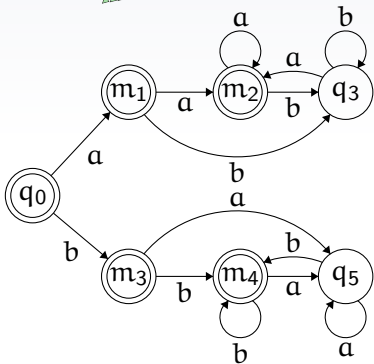
$\{m_1, m_2\} \xrightarrow{b} q_3$

$q_0 \xrightarrow{a} m_1, m_2 \xrightarrow{a} m_2$

$q_0 \xrightarrow{b} m_3, m_2 \xrightarrow{b} q_3$



## Пример минимизации

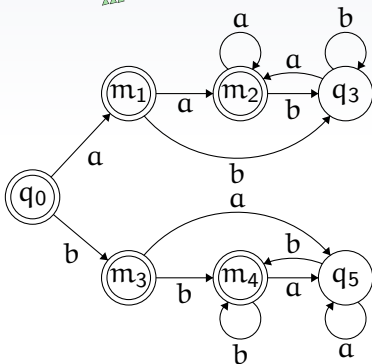


m <sub>1</sub>	✓					
m <sub>2</sub>	✓					
q <sub>3</sub>	✓	✓	✓			
m <sub>3</sub>				✓		
m <sub>4</sub>				✓		
q <sub>5</sub>	✓	✓	✓		✓	✓
	q <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	q <sub>3</sub>	m <sub>3</sub>	m <sub>4</sub>

$q_0 \xrightarrow{a} m_1, m_3 \xrightarrow{a} q_5$      $q_0 \xrightarrow{a} m_1, m_4 \xrightarrow{a} q_5$      $m_1 \xrightarrow{a} m_2, m_3 \xrightarrow{a} q_5$   
 $m_2 \xrightarrow{a} m_2, m_3 \xrightarrow{a} q_5$      $m_1 \xrightarrow{a} m_2, m_4 \xrightarrow{a} q_5$      $m_2 \xrightarrow{a} m_2, m_4 \xrightarrow{a} q_5$



## Пример минимизации



m <sub>1</sub>	✓					
m <sub>2</sub>	✓					
q <sub>3</sub>	✓	✓	✓			
m <sub>3</sub>	✓	✓	✓	✓		
m <sub>4</sub>	✓	✓	✓	✓		
q <sub>5</sub>	✓	✓	✓		✓	✓
	q <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	q <sub>3</sub>	m <sub>3</sub>	m <sub>4</sub>

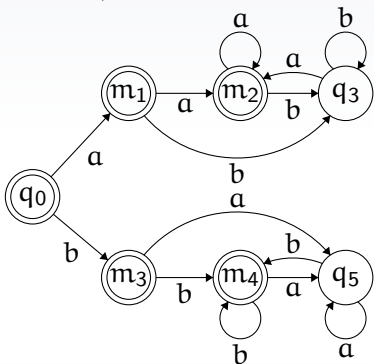
$\{m_3, m_4\} \xrightarrow{a} q_5$

$\{m_3, m_4\} \xrightarrow{b} m_4$

$q_3 \xrightarrow{a} m_2, q_5 \xrightarrow{a} m_4$



## Пример минимизации

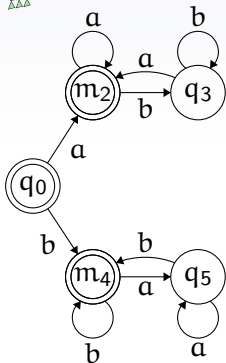


$m_1$	✓					
$m_2$	✓					
$q_3$	✓	✓	✓			
$m_3$	✓	✓	✓	✓		
$m_4$	✓	✓	✓	✓		
$q_5$	✓	✓	✓	✓	✓	✓
	$q_0$	$m_1$	$m_2$	$q_3$	$m_3$	$m_4$

Можно объединить состояния  $m_1$  и  $m_2$  и состояния  $m_3$  и  $m_4$ .

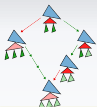


## Пример минимизации



m <sub>1</sub>	✓					
m <sub>2</sub>	✓					
q <sub>3</sub>	✓	✓	✓			
m <sub>3</sub>	✓	✓	✓	✓		
m <sub>4</sub>	✓	✓	✓	✓		
q <sub>5</sub>	✓	✓	✓	✓	✓	✓
	q <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	q <sub>3</sub>	m <sub>3</sub>	m <sub>4</sub>

Меньше чем пятью состояниями не обойтись. Рассмотрим слова  $\varepsilon$ ,  $a$ ,  $b$ ,  $ab$ ,  $ba$ . Каждые два из них различаются по  $\equiv_{\mathcal{L}}$  при выборе одного из трёх  $z$ :  $\varepsilon$ ,  $a$  или  $b$ .

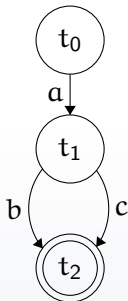
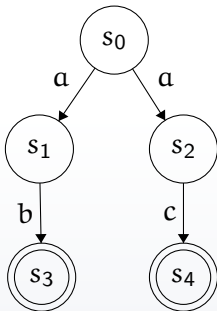


## Бисимуляция

Скажем, что состояния  $s_1, s_2$  системы переходов  $\mathcal{A}$  находятся в отношении бисимуляции ( $s_1 \sim s_2$ ), если выполняются условия:

- $\forall t_1, a(s_1 \xrightarrow{a} t_1 \Rightarrow \exists t_2(s_2 \xrightarrow{a} t_2 \ \& \ t_1 \sim t_2));$
- $\forall t_2, a(s_2 \xrightarrow{a} t_2 \Rightarrow \exists t_1(s_1 \xrightarrow{a} t_1 \ \& \ t_1 \sim t_2)).$

Бисимуляция — более сильное свойство, чем эквивалентность!



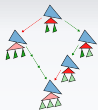


## Связь М.–Н. и производных

Пусть  $w^{-1}U$  — это производная  $U$  по  $w$ , т.е.  $\{v \mid wv \in U\}$ .  
Тогда выполнено  $x \equiv_U y \Leftrightarrow x^{-1}U = y^{-1}U$ .

- Количество производных (как языков) регулярного языка конечно.
- Конструкция Брзозовки порождает минимальный DFA.





## Связь М.–Н. и производных

Пусть  $w^{-1}U$  — это производная  $U$  по  $w$ , т.е.  $\{v \mid wv \in U\}$ .  
Тогда выполнено  $x \equiv_U y \Leftrightarrow x^{-1}U = y^{-1}U$ .

- Количество производных (как языков) регулярного языка конечно.
- Конструкция Брзозовки порождает минимальный DFA.

Но проблема с правилами переписывания (ACI):

- $(w_1 \mid w_2) \mid w_3 = w_1 \mid (w_2 \mid w_3)$
- $w_1 \mid w_2 = w_2 \mid w_1$
- $w \mid w = w$



## Применение теоремы М.–Н.

### Задача

Дан язык  $\mathcal{L}$ . Показать, что он не регулярен, пользуясь теоремой Майхилла–Нероуда.

### Стандартный подход

- 1 Подобрать бесконечную последовательность префиксов  $w_1, \dots, w_n, \dots$
- 2 Подобрать бесконечную последовательность суффиксов  $z_1, \dots, z_n, \dots$ , такую, что  $w_i ++ z_i \in \mathcal{L}$ .
- 3 Доказать, что в таблице конкатенаций все строки различны (значит,  $\forall i, j \exists k (w_i z_k \in \mathcal{L} \ \& \ w_j z_k \notin \mathcal{L})$ ).

Диагональная конструкция (условие  $w_i ++ z_i \in \mathcal{L}$ ) — одна из многих возможных, обычно она довольно удобна.



## Диагональная конструкция

Рассмотрим язык  $L = \{a^n b^n\}$ . Положим  $w_i = a^i$ ,  $z_i = b^i$ . Тогда таблица конкатенаций  $w_i, z_j$  будет выглядеть следующим образом. Здесь  $+$  — это то же, что « $\in \mathcal{L}$ », — читаем как « $\notin \mathcal{L}$ ».

	$z_1 = b$	$z_2 = b^2$	$z_3 = b^3$	$\dots$	$z_n = b^n$	$\dots$
$w_1 = a$	+	—	—		—	
$w_2 = a^2$	—	+	—		—	
$w_3 = a^3$	—	—	+		—	
$\dots$			$\dots$			
$w^n = a^n$	—	—	—		+	
$\dots$						



## Доказательство минимальности

Так же можно обосновывать минимальность DFA. Рассмотрим минимальный автомат из примера выше. Его язык — слова в  $\{a, b\}^*$ , начинающиеся и заканчивающиеся одной и той же буквой. Построим таблицу классов эквивалентности по  $w_i \in \{\varepsilon, a, b, ab, ba\}$ .

	$\varepsilon$	$a$	$b$
$\varepsilon$	+	+	+
$a$	+	+	—
$b$	+	—	+
$ab$	—	+	—
$ba$	—	—	+

В этой таблице все строчки различны, значит, выбранные  $w_i$  действительно лежат в различных классах эквивалентности, и DFA, распознающий язык  $\mathcal{L}$ , не может иметь меньше пяти состояний.

При доказательстве минимальности DFA достаточно подобрать  $\lceil \log_2 n \rceil + 1$  различающих суффиксов  $z_i$ , где  $n$  — число состояний автомата.



## О порождении новых алгоритмов

Пусть  $\mathcal{A}$  — NFA. Тогда  $\text{det}(\text{reverse}(\text{det}(\text{reverse}(\mathcal{A}))))$  — минимальный DFA, эквивалентный  $\mathcal{A}$ .

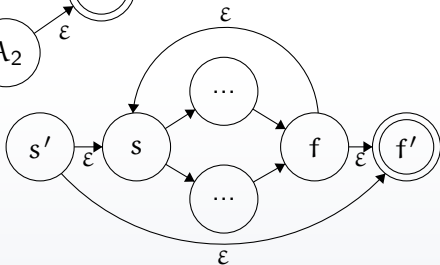
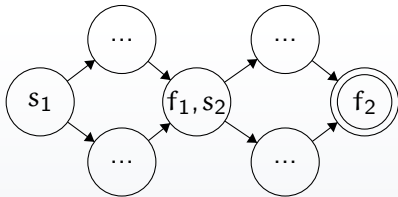
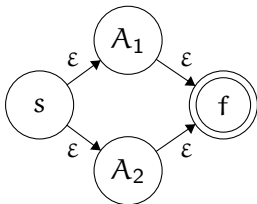
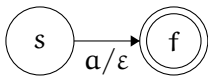
Многие алгоритмы для порождения малых (не минимальных) NFA являются комбинациями нескольких базовых операций.

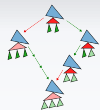
- Обращение автомата
- Детерминизация
- Удаление  $\varepsilon$ -правил
- Минимизация
- Разметка



## Автомат Томпсона

- Единственное начальное состояние
- Единственное конечное состояние
- Не больше двух переходов из каждого состояния





## Несколько конструкций

- Автомат Глушкова:  $\text{rmeps}(\text{Th}(R))$ ;
- Автомат Антимирова:  
 $\text{rmeps}(\text{deannotate}(\text{minimize}(\text{rmeps}(\text{annotate\_eps}(\text{Th}(R)))))$ ;
- Автомат Илия–Ю:  
 $\text{deannotate}(\text{minimize}(\text{rmeps}(\text{annotate}(\text{Th}(R)))))$ .