Представления регулярных языков. Критерий регулярности

Теория формальных языков

2022 г.

Недетерминированные КА

Определение

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

- Q — множество состояний;
- Σ — алфавит терминалов;
- δ — 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 $\mathscr{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$.

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

Недетерминированные КА

Определение

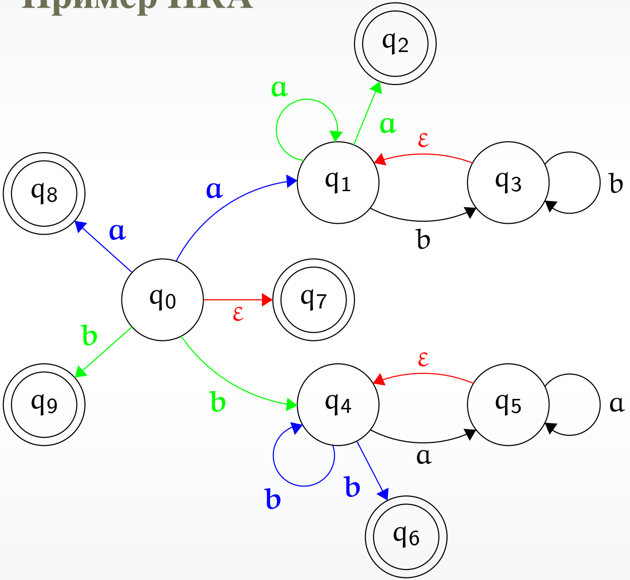
Недетерминированный конечный автомат (NFA) — это пятёрка $\mathscr{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$.

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

Определение

Язык $\mathscr{L}$, распознаваемый НКА $\mathscr{A}$ — 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 $\mathscr{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$, где:

- Q — множество состояний;
- Σ — алфавит терминалов;
- δ — 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.

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

Детерминированный КА

Определение

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

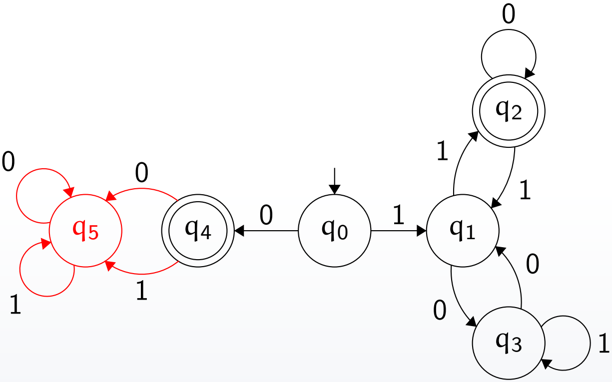
- Q — множество состояний;
- Σ — алфавит терминалов;
- δ — множество правил перехода вида $\langle q_i, a_i, q_j \rangle$, где $q_i, q_j \in Q$, $a_i \in \Sigma$, причём $\forall q_i, a_i \exists q_j (\langle q_i, a_i, q_j \rangle \in \delta \;\&\; \forall q_k (\langle q_i, a_i, q_k \rangle \in \delta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

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

Язык $\mathscr{L}$, распознаваемый $\mathscr{A}$ — 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 $\mathscr{A}$ к $D(\mathscr{A})$

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

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

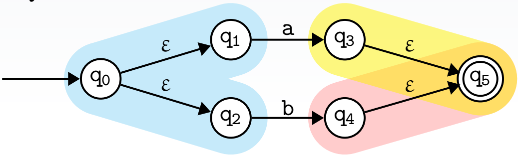
Удаление ε-переходов

- Один из шагов детерминизации.
- Может пониматься в разных смыслах: удаление только переходов без изменения числа состояний, и построение состояний, замкнутых относительно ε-переходов (то есть аналогично детерминизации, но только по ε-переходам).

Результаты этих преобразований будут различны, причём первое сохраняет недетерминированные переходы, а второе может их детерминизировать.

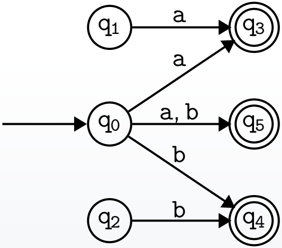
Удаление ε-переходов

Рассмотрим следующий автомат.

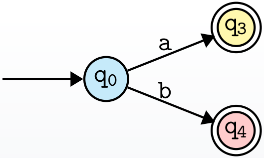


ε-замыкание q_0 — это $\{q_0, q_1, q_2\}$. ε-замыкание q_3 — это $\{q_3, q_5\}$. ε-замыкание q_4 — это $\{q_4, q_5\}$. Остальные состояния ε-замкнуты собой.

Результат первого преобразования:

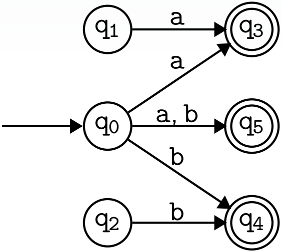


Результат второго преобразования:

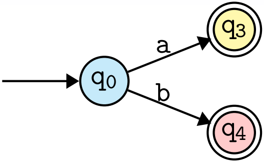


Удаление ε-переходов

Результат первого преобразования:



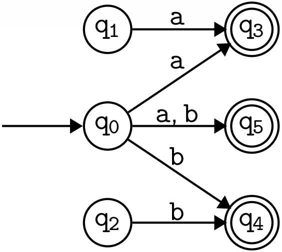
Результат второго преобразования:



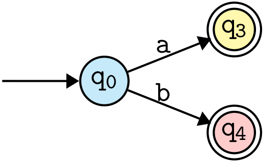
В первом случае q_3 и q_4 стали финальными, потому что из них есть путь по ε-переходам в финальное состояние. Дополнительно добавились переход из q_0 в q_5 по $\mathtt{a}$ (поскольку такой путь есть из ε-достижимого из q_0 состояния q_1) и аналогичный переход в q_5 по $\mathtt{b}$. После чего все ε-переходы были удалены. Для завершения построения, следует ещё удалить недостижимые состояния q_1 и q_2. Автомат остался недетерминированным.

Удаление ε-переходов

Результат первого преобразования:

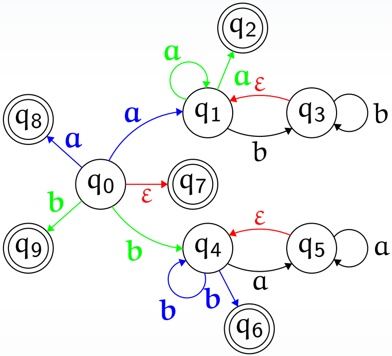


Результат второго преобразования:



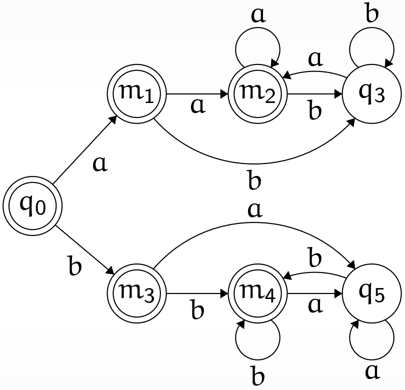
Во втором случае ε-замыкания состояний исходного автомата сразу же рассматривались как состояния нового автомата. Это привело к тому, что удалось сэкономить одно состояние, и результат оказался детерминированным.

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 $h(a_1 \circ a_2 \circ \cdots \circ a_n) = h(a_1) \circ h(a_2) \circ \cdots \circ h(a_n)$. Здесь $\circ$ —конкатенация.

Утверждение

Пусть $\mathscr{L}$ — регулярный язык над Σ. Тогда регулярны:

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

Замыкания регулярных языков

Утверждение

Пусть $\mathscr{L}$ — регулярный язык над Σ. Тогда регулярны:

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

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

Замыкания регулярных языков

Утверждение

Пусть $\mathscr{L}$ — регулярный язык над Σ. Тогда регулярны:

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

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

Замыкания регулярных языков

Утверждение

Пусть $\mathscr{L}$ — регулярный язык над Σ. Тогда регулярны:

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

Рассмотрим DFA $\mathscr{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$, распознающий $\mathscr{L}$.
Построим $\mathscr{A}' = \langle Q, \Sigma, q_0, F, \delta' \rangle$ такой, что
$\langle q_i, a, q_j \rangle \in \delta' \Leftrightarrow q_i \xrightarrow{h(a)} q_j$ в исходном автомате $\mathscr{A}$.

Примеры

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

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

Примеры

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

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

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

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

Эквивалентность слов в DFA

Пусть дан DFA $\mathscr{A}$. Положим
$w_1 \equiv_{\mathscr{A}} w_2 \Leftrightarrow \exists \mathsf{q}_\mathsf{i}(\mathsf{q}_0 \xrightarrow{w_1} \mathsf{q}_\mathsf{i} \;\&\; \mathsf{q}_0 \xrightarrow{w_2} \mathsf{q}_\mathsf{i})$.

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

Эквивалентность слов в DFA

Пусть дан DFA $\mathscr{A}$. Положим
$w_1 \equiv_{\mathscr{A}} w_2 \Leftrightarrow \exists q_i (q_0 \xrightarrow{w_1} q_i \ \& \ q_0 \xrightarrow{w_2} q_i)$.

Если $w_1 \equiv_{\mathscr{A}} w_2$, тогда $\forall z (w_1 z \in \mathscr{L}(\mathscr{A}) \Leftrightarrow w_2 z \in \mathscr{L}(\mathscr{A}))$.
Рассмотрим более общее отношение. Положим

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

Теорема Майхилла-Нероуда

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

Критерий регулярности языка

Теорема Майхилла-Нероуда

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

Критерий регулярности языка

Теорема Майхилла-Нероуда

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

$\Rightarrow$: Пусть $\mathscr{L}$ регулярен. Тогда он порождается некоторым DFA $\mathscr{A}$ с конечным числом состояний N. Значит, множество $\left\{\mathsf{q}_\mathsf{i} \mid \mathsf{q}_0 \xrightarrow{w} \mathsf{q}_\mathsf{i}\right\}$ конечно, а для каждых двух w_1, w_2 таких, что $\mathsf{q}_0 \xrightarrow{w_1} \mathsf{q}_\mathsf{i}$ и $\mathsf{q}_0 \xrightarrow{w_2} \mathsf{q}_\mathsf{i}$, выполняется $w_1 \equiv_{\mathscr{L}} w_2$.

Критерий регулярности языка

Теорема Майхилла-Нероуда

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

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

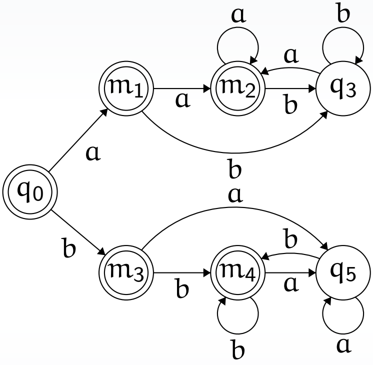
- Начальным состоянием объявим класс эквивалентности A_0 такой, что $\varepsilon \in A_0$.
- Конечными объявим такие A_j, что $\forall w \in A_j (w \in \mathscr{L})$.
- Если $w \in A_i$, $w\ a_k \in A_j$, тогда добавляем в δ правило $\langle A_i, a_k, A_j \rangle$. $\forall w_1, w_2 \in A_i$, $w_1 a_k$ 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 $\exists a (q_i \xrightarrow{a} q_1' \mathbin{\&} q_j \xrightarrow{a} q_2' \mathbin{\&} \{q_1', q_2'\}$ — помеченная пара$)$.
4. Продолжаем шаг 3, пока не будет появляться новых помеченных пар.

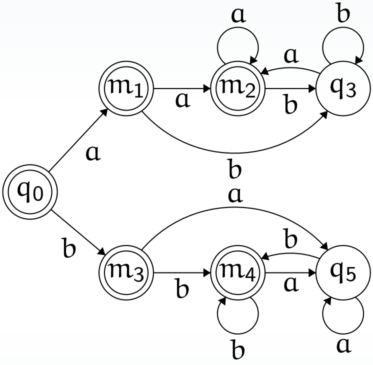
Пары, оставшиеся непомеченными, можно объединить.

Пример минимизации



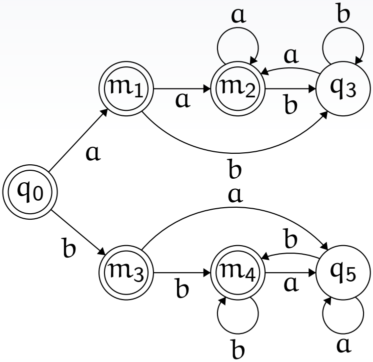
m_1						
m_2						
q_3						
m_3						
m_4						
q_5						
	q_0	m_1	m_2	q_3	m_3	m_4

Пример минимизации



m_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

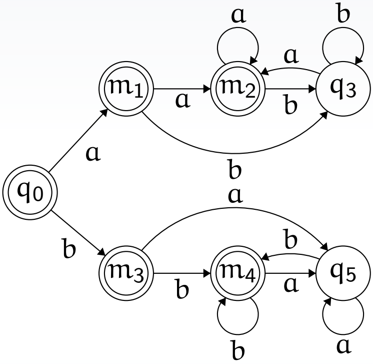


	q_0	m_1	m_2	q_3	m_3	m_4
m_1						
m_2						
q_3	✓	✓	✓			
m_3				✓		
m_4				✓		
q_5	✓	✓	✓		✓	✓

$q_0 \xrightarrow{a} m_1, m_1 \xrightarrow{a} m_2$ $\quad$ $q_0 \xrightarrow{b} m_3, m_1 \xrightarrow{b} q_3$ $\quad$ $\{m_1, m_2\} \xrightarrow{a} m_2$ $\quad$ $\{m_1, m_2\} \xrightarrow{b} q_3$

$q_0 \xrightarrow{a} m_1, m_2 \xrightarrow{a} m_2$ $\quad$ $q_0 \xrightarrow{b} m_3, m_2 \xrightarrow{b} q_3$

Пример минимизации

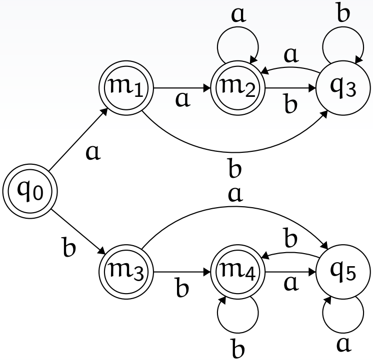


m_1	✓					
m_2	✓					
q_3	✓	✓	✓			
m_3				✓		
m_4				✓		
q_5	✓	✓	✓		✓	✓
	q_0	m_1	m_2	q_3	m_3	m_4

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

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

Пример минимизации

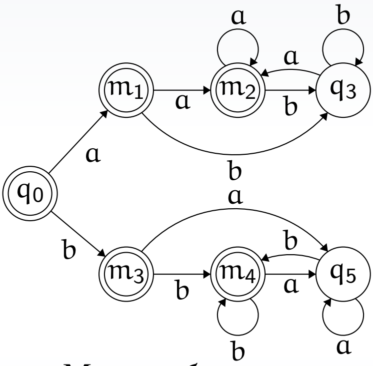


m_1	✓					
m_2	✓					
q_3	✓	✓	✓			
m_3	✓	✓	✓	✓		
m_4	✓	✓	✓	✓		
q_5	✓	✓	✓		✓	✓
	q_0	m_1	m_2	q_3	m_3	m_4

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

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

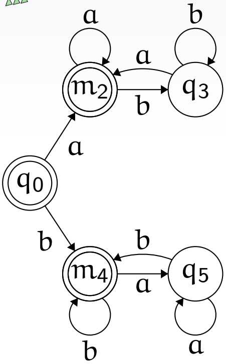
Пример минимизации



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_1	✓					
m_2	✓					
q_3	✓	✓	✓			
m_3	✓	✓	✓	✓		
m_4	✓	✓	✓	✓		
q_5	✓	✓	✓	✓	✓	✓
	q_0	m_1	m_2	q_3	m_3	m_4

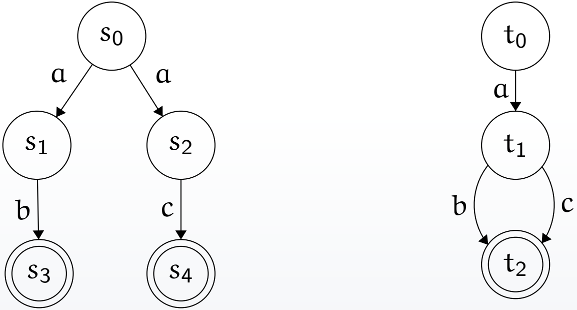
Меньше чем пятью состояниями не обойтись. Рассмотрим слова ε, a, b, ab, ba. Каждые два из них различаются по $\equiv_{\mathscr{L}}$ при выборе одного из трёх z: ε, a или b.

Бисимуляция

Скажем, что состояния s_1, s_2 системы переходов $\mathscr{A}$ 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 M.– N. и производных

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

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

Связь М.– N. 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.– N.

Задача

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

Стандартный подход

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

Диагональная конструкция (условие $w_i +\!\!+ z_i \in \mathscr{L}$) — 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 «$\in \mathscr{L}$», $-$ читаем как «$\notin \mathscr{L}$».

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

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 $w_i \in \{\varepsilon, a, b, ab, ba\}$.

	ε	a	b
ε	+	+	+
a	+	+	—
b	+	—	+
ab	—	+	—
ba	—	—	+

В этой таблице все строчки различны, значит, выбранные w_i действительно лежат в различных классах эквивалентности, и DFA, распознающий язык $\mathscr{L}$, 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

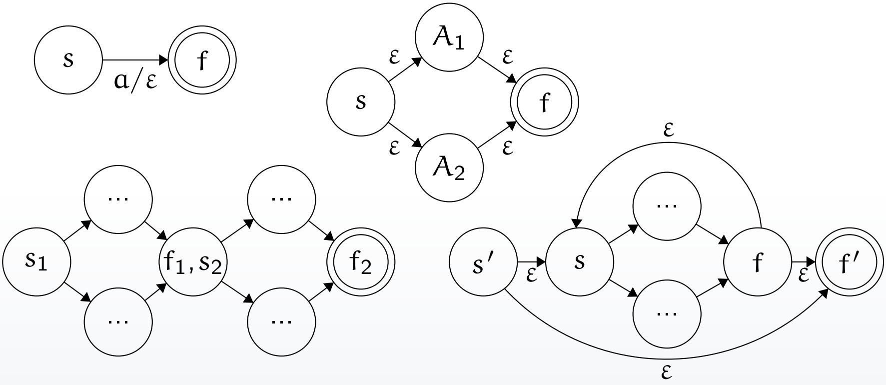
Пусть $\mathscr{A}$ — NFA. Тогда $\mathtt{det}(\mathtt{reverse}(\mathtt{det}(\mathtt{reverse}(\mathscr{A}))))$ — минимальный DFA, эквивалентный $\mathscr{A}$.

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

- Обращение автомата
- Детерминизация
- Удаление ε-правил
- Минимизация
- Разметка

Автомат Томпсона

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



Несколько конструкций

- Автомат Глушкова: $\mathrm{rmeps}(\mathrm{Th}(R))$;
- Автомат Антимирова: $\mathrm{rmeps}(\mathrm{deannote}(\mathrm{minimize}(\mathrm{rmeps}(\mathrm{annote\_eps}(\mathrm{Th}(R))))))$;
- Автомат Илия–Ю: $\mathrm{deannote}(\mathrm{minimize}(\mathrm{rmeps}(\mathrm{annote}(\mathrm{Th}(R)))))$.