

Теория автоматов и формальных языков

Регулярные языки

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

сентября 2016г.

Регулярная грамматика

Праволинейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow aB$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Левوليнейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow Ba$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Регулярная грамматика

Праволинейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow aB$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Левوليнейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow Ba$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Теорема

Пусть L — формальный язык.

$\exists G_r$ — праволинейная грамматика, т.ч. $L = L(G_r) \Leftrightarrow \exists G_l$ — левوليнейная грамматика, т.ч. $L = L(G_l)$

Регулярная грамматика

Праволинейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow aB$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Левوليнейная грамматика — грамматика, все правила которой имеют следующий вид:

- $A \rightarrow Ba$ или $A \rightarrow a$, где $A, B \in V_N, a \in V_T$

Теорема

Пусть L — формальный язык.

$\exists G_r$ — праволинейная грамматика, т.ч. $L = L(G_r) \Leftrightarrow \exists G_l$ — левوليнейная грамматика, т.ч. $L = L(G_l)$

Регулярная грамматика — праволинейная или левوليнейная грамматика

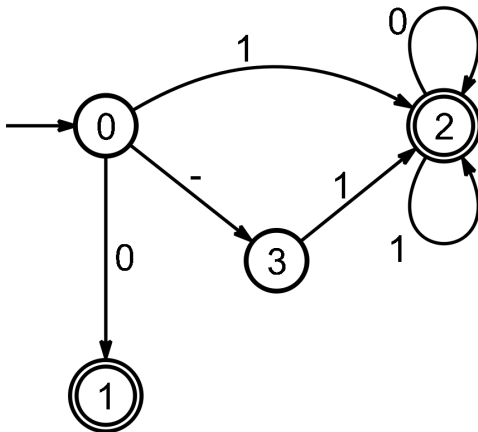
Конечный автомат — $\langle Q, \Sigma, \delta, q_0, F \rangle$

- $Q \neq \emptyset$ — конечное множество состояний
- Σ — Конечный входной алфавит
- δ — отображение типа $Q \times \Sigma \rightarrow Q$
- $q_0 \in Q$ — начальное состояние
- $F \subseteq Q$ — множество конечных состояний

Пример конечного автомата

$Q = \{0, 1, 2, 3\}, \Sigma = \{0, 1, -\}, q_0 = 0, F = \{1, 2\}$

$\delta(0, 0) = 1; \delta(0, 1) = 2; \delta(0, -) = 3; \delta(3, 1) = 2; \delta(2, 0) = 2; \delta(2, 1) = 2$



- Обобщаем функцию перехода:
 - ▶ $\delta'(q, \varepsilon) = q$
 - ▶ $\delta'(q, xa) = \delta(\delta'(q, x), a)$, где $x \in \Sigma^*$, $a \in \Sigma$
- Цепочка ω **распознается** конечным автоматом $\langle Q, \Sigma, \delta, q_0, F \rangle$, если $\exists p \in F. \delta'(q_0, \omega) = p$
- **Язык, распознаваемый конечным автоматом:**
 $\{\omega \in \Sigma^* \mid \exists p \in F. \delta'(q_0, \omega) = p\}$

Минимальный конечный автомат

- Конечные автоматы A_1 и A_2 эквивалентны, если распознают один и тот же язык
- **Минимальный конечный автомат** — автомат, имеющий наименьшее число состояний, распознающий тот же язык, что и данный

Классы эквивалентности

Отношение эквивалентности — рефлексивное, симметричное, транзитивное отношение

- $xRx; xRy \Leftrightarrow yRx; xRy, yRz \Rightarrow xRz$

Теорема

$\forall R$ — отношение эквивалентности на множестве S

Можно разбить S на k непересекающихся подмножеств $I_1 \dots I_k$, т.ч.

$$aRb \Leftrightarrow a, b \in I_j$$

Множества $I_1 \dots I_k$ называются **классами эквивалентности**

Эквивалентные состояния

- $\omega \in \Sigma^*$ различает состояния q_i и q_j , если
$$\delta'(q_i, \omega) = t_1, \delta'(q_j, \omega) = t_2 \Rightarrow (t_1 \notin F \Leftrightarrow t_2 \in F)$$
- q_i и q_j эквивалентны ($q_i \sim q_j$), если
$$\forall \omega \in \Sigma^*. \delta'(q_i, \omega) = t_1, \delta'(q_j, \omega) = t_2 \Rightarrow (t_1 \in F \Leftrightarrow t_2 \in F)$$
 - ▶ Является отношением эквивалентности

Лемма

$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle, p_1, p_2, q_1, q_2 \in Q, q_i = \delta(p_i, c)$
 $\omega \in \Sigma^*$ различает q_1 и q_2 . Тогда $c\omega$ различает p_1 и p_2

Доказательство

$$\delta'(p_i, c\omega) = \delta'(\delta(p_i, c), \omega) = \delta'(q_i, \omega) = t_i$$





















