Лекция 4. 06 Марта 2025

Способы задания языков:

- Описательный. К сожалению не для всех типов грамматик его можно осуществить. Но для грамматик типа 3 есть способ описания в виде регулярных выражений. Он основан на определении праволинейного языка.

$PM \leftrightarrow PB$ Если G — праволинейная грамматика, то $L(G)$ — ...

Регулярные множества и праволинейныйе грамматики

Язык определяется праволинейнй грамматикой т.и.т.т., когда он является регулярным множеством.

- Лемма 3. Множества $\emptyset, \{\varepsilon\}$, и$a$ для всех $a \in \Sigma$ являтся праволинейными языками.

- Док-во:

1. $G = ((\{S\}), \Sigma, P, S)$ — праволинейная грамматика, для которой $L(G) =$
2. $G(\{S\}, \Sigma, \{S \to \varepsilon\}, S)$ — праволинейная грамматика для ...
3. $G_{\text{п}} = (\{S\}.\Sigma, \{s \to a\}, S\})$ — праволинейная грамматика, для которой $L(G) = \{a\}$

- Лемма 4. Если PиQ — праволинейные языки, то языки

1. $P \cup Q$,
2. PQ,
3. P^*

тоже праволинейные.

- Док-во конструктивное — мы строим эти грамматики:

Так как P и Q — праволинейные, то $\exists$ праволинейные грамматики $G_P = (N_1, \Sigma, P_1, S_1)$ и $G_Q = (N_2, \Sigma, P_2, S_2)$, для которых $L(G_P) = P$ и $L(G_Q) = Q$ Считаем что $N_1 \cap N_2 = \emptyset$.

1. $G_3 = (N_1 \cup N_2, \Sigma, P_1 \cup P_2 \cup \{S_3 \to S_1 \mid S_2\}, S_3)$ — праволинейная $L(G_3) = L(G_P) \cup L(G_Q)$, так как для каждого вывода $S_3 \Rightarrow w$

т.к. для каждого вывода $S_3 \underset{G_3}{\overset{+}{\Rightarrow}} w$ и обратно.

Так как G_3 — праволинейная грамматка, то $L(G_3)$ — праволинейный язык

2. $G_4 = (N_1 \cup N_2, \Sigma, P_4, S_1)$ — праволинейная.

   P_4:
   1. Если $A \to xB$ есть в P_1, то $A \to xB$ принадлежит P_4.
   2. Если $A \to x$ есть в P_1, то $A \to xS_2$ принадлежит P_4.
   3. Все правила из P_2 принадлежит P_4.

Отметим, что если $S_1 \underset{G_P}{\overset{+}{\Rightarrow}} w$, то $S_1 \underset{G_4}{\overset{+}{\Rightarrow}} wS_w$, а если $S_2 \underset{G_Q}{\overset{+}{\Rightarrow}} x$, то $S_2 \underset{G_4}{\overset{+}{\Rightarrow}} x$.

Таким образом $L_{G_P} L_{G_Q} \subseteq L(G_4)$.

$S_1 = W_1 \dots W_n \underbrace{A}_{\text{нетерминальный символ}}$

W_A

W

Мы должны из строки W_Z породить грамматики $L(G_P)$ $L(G_Q)$

W_A

WS_2

Пусть $S_1 \overset{+}{\underset{G_4}{\Rightarrow}} w$. Т.к. в G_4 нет правил $A \to x$, попавших из P_1, то этот вывод можно сделать так: $S_1 \overset{+}{\underset{G_4}{\Rightarrow}} xS_2 \overset{+}{\underset{G_4}{\Rightarrow}}$, где $w = xy$ и все правила в выводе $S \overset{+}{\underset{G_4}{\Rightarrow}} wS_2$ попали в P_4 по (1) и (2).

Следовательно, должны быть выводы $\left(S_1 \Rightarrow \overset{+}{G}_P\right)$ и $S_2 \Rightarrow \overset{+}{G}_Q y$.

Отсюда $L(G_4) \subseteq L_{G_P} L(G_Q)$

3. Пусть $G_5 = (N_1 \cup \{S_5\}, \Sigma, P_5, S_5)$ такая, что $S_5 \in N_1$ и P_5:
   1. $A \to xB$ есть в $P_1 \Rightarrow (A \to xB) \in P_5$
   2. $(A \to x) \in P_1 \Rightarrow (A \to xS_5), (A \to x) \in P_5$
   3. $(S_5 \to S_1 \mid \varepsilon) \in P_5$

Очевидно, что

$$S_5 \overset{+}{\underset{G_5}{\Rightarrow}} x_1 S_5 \overset{+}{\underset{G_5}{\Rightarrow}} x_1 x_2 S_5 \overset{+}{\underset{G_5}{\Rightarrow}} \dots \overset{+}{\underset{G_5}{\Rightarrow}} x_1 x_2 \dots x_{n-1} x_n$$

$\Leftrightarrow$, когда $S_1 \overset{+}{\underset{G_P}{\Rightarrow}} x_1, S_1 \overset{+}{\underset{G_P}{\Rightarrow}} x_2, \dots, S_1 \overset{+}{\underset{G_P}{\Rightarrow}} x_n$

Отсюда следует, что $L(G_4) = (L(G_P)) *$

Теорема.

Язык является регулярным множеством $\Leftrightarrow$ он праволинейный.

- Док-во:

Необходимость:

Следует из лемм 3 и 4, индукцией по числу шагов построерния регулярного нмножества, гед одни шаг — это применение одного из правил, определяющих регулярные множества.

Достаточность:

Пусть $G = (N, \Sigma, P, S)$ — праволинейная грамматика и $N = \{A_1, A_2, ..., A_n\}$.

Можно построить старндартную систему уравнений с регулярными коффициентами, неизвестными которой являются нетерминалы из N.

Уравнение для A_i будет иметь вид: $A_i = a_{i0} + a_{i1}A_1 + ...a_{in}$, где

1. $a_{i0} = w_1 + ... + w_k$, если $A_I \to w_1 \mid ... \mid w_k$ — все правила с левой частью A_i и правой частью, состоящей только из терминалов ($k = 0 \Rightarrow a_{i0} = 0$)
2. Для $j > 0 a_{ij} = x_1 + ... + x_m$, если $A_i \to x_1 A_j \mid ... \mid x_m A_j$ — все правила с левой частью A_i и правой частью, оканчивающуюся на A_j (если $m = 0$, то $a_{ij} = 0$)/

Решая эту систему уравнений получаем решенеие f для $N = \{A_1, A_2, ..., A_n\}$

Для S получаем PB $f(S)$, которое определяет язык $L(G)$.

Но алгоритм строит $f(S)$ как язык, обозначаемый некоторым PB.

Таким образом, $L(G)$ — регулярное множество.

Пример на доске:

Система уравнений с регулярными коэффициентами:

$S \to aA + bB + \varepsilon$

Конкатенация регулярных выражений

$A \to b + aA$

$B \to bB + \underbrace{aA + bA}_{\text{одинаковый нетерминал}} + b$

$\Downarrow$

$B \to bB + (a+b)A + b$

Пример построения грамматики по PB

Рассмотрим регулярное выражение $(0 + 1) * 2$. Построим праволинейную грамматику, определяющую тот же язык:

- Для языков $\{0\}, \{1\}, \{2\}$ определим грамматики (согласно леммам):

$$G_0 = (\{S_i\}, \{0, 1, 2\}, \{S_i\} \to 0\}, S_i)$$

$$G_1 = (\{S_i\}, \{0, 1, 2\}, \{S_i\} \to 1\}, S_i)$$

$$G_2 = (\{S_i\}, \{0, 1, 2\}, \{S_i\} \to 2\}, S_i)$$

- Для РВ $0 + 1$ по грамматикам G_0 и G_1 строим грамматику

$$\begin{gathered} P_3 = \{S_3 - S_0, \\ S_3 - S_1, \\ S_0 - 0, \\ S_1 - 1, \} \end{gathered}$$

- Для PB $(0 + 1)*$ по грамматике G_3 строим грамматику $G_4 = (\{S_4, S_3, S_0, S_1\}, \{0, 1, 2\}, P_4, S_4)$, где P_4
- $$\begin{gathered} S_4 \to S_3 \mid \varepsilon \\ S_3 \to S_0 \mid S_1 \\ S_0 \to 0S_4 | \ 0 \\ S_1 \to 1S_4 \mid 1 \end{gathered}$$
- Для $PB (0 + 1)$*2. Построим праволинейную грамматику

$G_5 = \{\{S_4, S_3, S_0, S_1, S_2\}, \{0, 1, 2\}, P_5,$ что-то$\}$

где P_5:

определяющую тот же язык.

- Пусть дана грамматика, определяемая правилами:

$$S \to 0A \mid 1B \mid \varepsilon A \to 0A \mid 1B \mid 2B \to 0B \mid 1A$$

- рассмотрим систему уравнений:

$$\begin{gathered} S = 0A + 1B - \varepsilon \\ A = 0A + 1B - 2 \\ B = 0B - 1A \end{gathered}$$

- Выразим $B = 0 * 1A$ и подставим его значение в 1-е и 2-е уравнения:

$$\begin{gathered} S = 0A + 10\text{*}1A - \varepsilon \\ A = 0A + 10\text{*}1A - 2 \\ B = 0\text{*}1A \end{gathered}$$

- Выразим $A = \{0 + 10 * 1\} * 2$ и подставим его в первое уравнение:

$$\begin{gathered} S = 0(0 + 10\text{*}1)\text{*}2 + 10\text{*}1(0 + 10\text{*}1)\text{*}2 + \varepsilon \\ A = (0 - 10\text{*}1)\text{*}2 \\ B = 0\text{*}1A \end{gathered}$$

Выражение ля S можно преобразовать к виду $S = (0 - 10 * 1) + 2 + \varepsilon$

Это РВ будет определять то же регулярное множество (регулярный язык), что и данная праволинейная грамматика.

Конечные автоматы и регулярные множества.

Наибольшую популярность получили именно конечные автоматы, потому что они служат не только для описания регулярных языков, но и для описания модели алгоритма, работающего в режиме дискретных шагов, — именно так работают почти все алгоритмы, ведь это заложено в определении.

Конечный автомат — это:

- Формальная система
- Помнит только конечное количество информации
- Информация представляется его **состояниями**
- Состояние изменяется под воздействием **входов**
- Правила, которые говорят, как изменять состояние под воздействием входов, называются переходами

Что является ключевой особенностью? Объект, названный нами автоматом, короче давайте рисовать.

Этот автомат умеет работать с некоторыми символами, поступающими на вход.

$$\underbrace{a(t) \longrightarrow}_{\text{вход}} \square_{S(t)}$$

$a_i \in \Sigma, s(t+1) = \delta(s(t), a(t))$ — функция перехода

Допустимые входы

- Дана последовательность входов (входная строка).
- Начать в начальном состоянии и следовать по переходу по каждому очередному символу входной строки.
- Вход принимается, если вы перенеслись в финальное (принимающее) состояние после чтения всех входных символов.

Язык автомата

- Множество строк, принимаемых автоматом A, являются *языком* автомата A.
- Обзонгачается $L(A)$
- Различные множества финальных состояний $\to$ Разные языки.

Детерминированный конечный автомат

- Формализм для определения языков, состоящих из:
  - Конечного множества состояний (Обозначается обычно Q)
  - Входной алфавит (Σ, обычно)
  - Функция переходов (δ, обычно)
  - Начальное состояние (q_0 в Q, обычно)
  - Финальные состояния ($F \subseteq Q$, обычно)

Примечание: на зачёте, отвечая на вопрос про детерминированный конечный автомат, нужно не только перечислить эти компоненты, но и дать определение функции переходов, иначе ответ будет неполный

$$A = (Q, \Sigma, \delta, q_0, F)$$
$$\delta : Q \times \Sigma \to Q$$

Функция переходов

- Имеет два аргумента: состояние и входной символ.

- $\delta(q, a) =$ состояние, в которое КДА переходит, если он в состоянии q получает на вход символ a.
- Замечание: всегда есть следующее состояние — добавим *Мёртвое состояние* если нет переходов (Пример далее)
- Форма задания: таблица переходов, граф.
- Функцию пеехoдов можно доопределить для слов:

$$\delta^\star(q, a) = \delta(q, a) \text{ если } |a| = 1$$
$$\delta^\star(q, aw) = \delta(\delta^\star(q, w), a)$$

Начальное состояние указывается стрелочкой $\to$, а конечное (заключительное) — звёздочкой $\star$

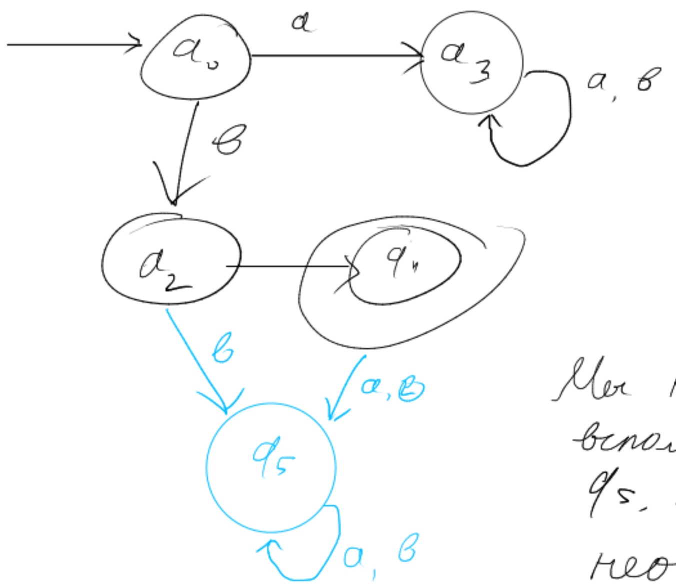
Пример:

Таблица переходов:

	a_0	a_1	$\dots$	a_k
$\to {}^* a_0$				
a_1				
${}^* a_2$				
$\vdots$				
${}^* q_1$				

звёздочкой помечаются начальные состояния

Ориентированный граф:



Мы также придумали вспомогательный переход q_5, куда идут все необработанные слова

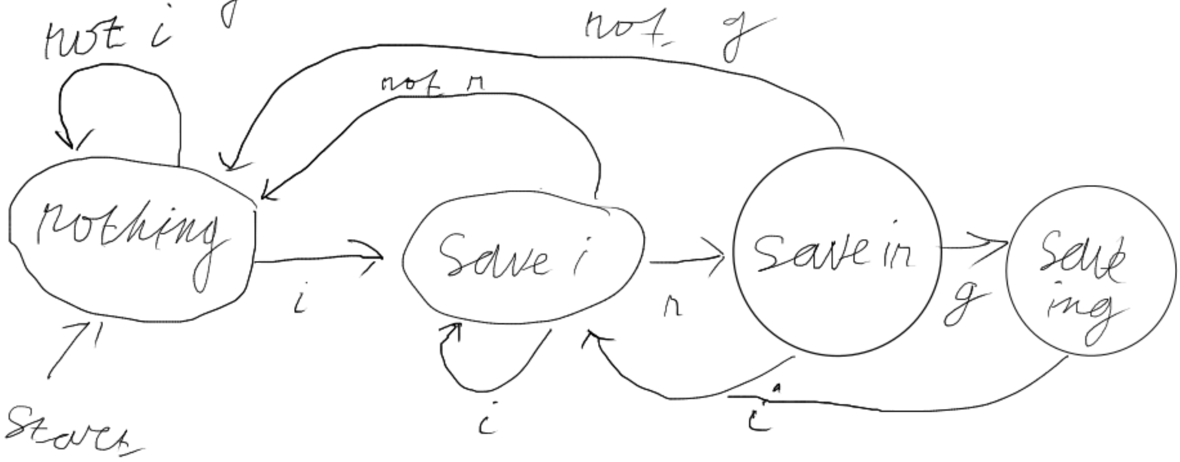
Это защищает от зависания автомата.

Расширенная дельта

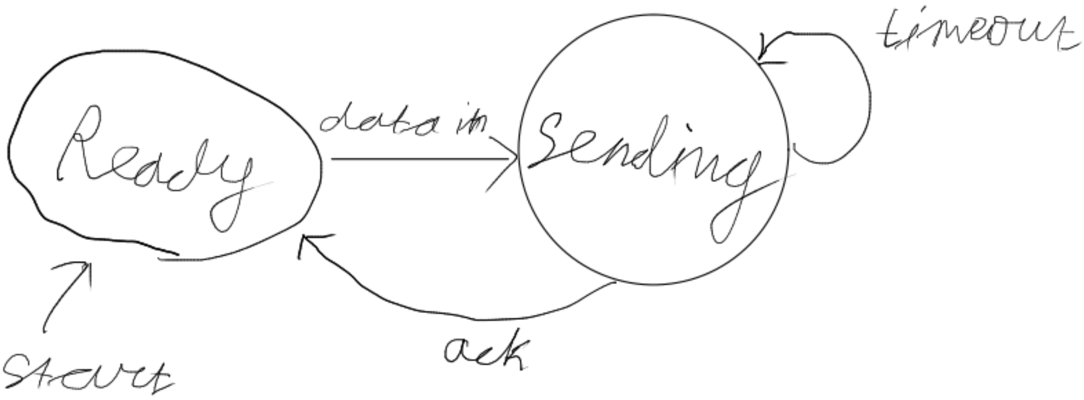
- Мы не будем различать обозначения δ и $\delta^\star$
- Причина: $\delta^\star(q, a) = \delta(\delta^\star(q, \in), a) = \delta(q, a)$ — Расширенная дельта — $\delta^\star$

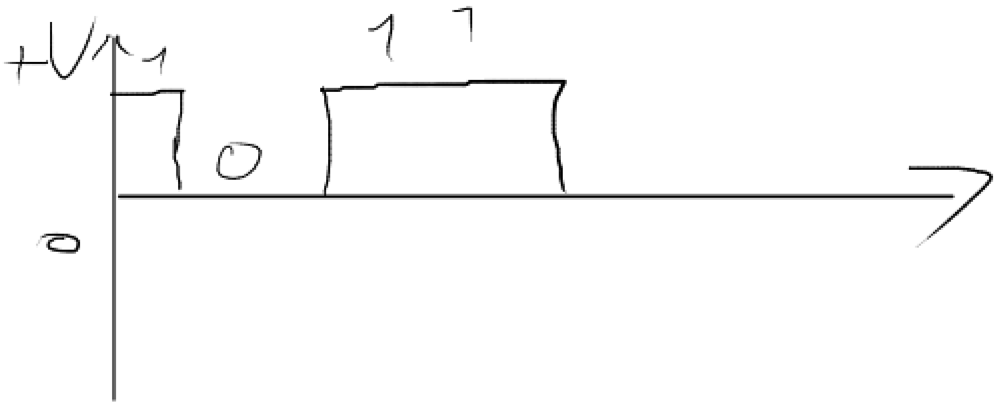
Пример: Распознавание строк, оканчивающихся на “ing”

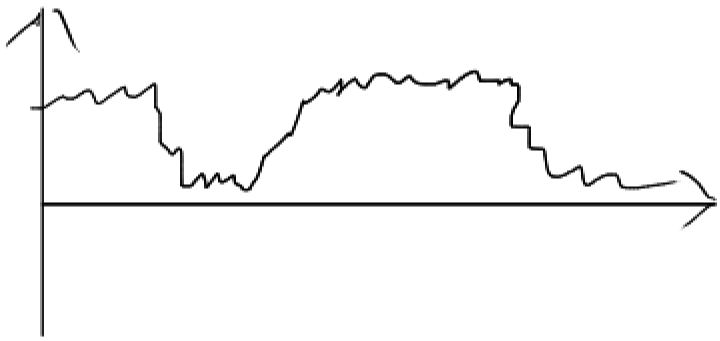
Определение слова, оканчивающегося на -ing:



Пример: Протокол для пересылки данных







Пример: Строки без 11

- Строка еще не иммет 11. Не оканчивается 1.
- Строка еще не имеет 11. Но оканчивается на 1.
- Найдены две 1 идущие подряд

соглашение: Строки и символы.

- $\ldots w, x, y, z$ — строки
- $a, b, c, \ldots$ —- одиночные входные символы

Язык КДА

- Автоматы всех видов определяют языки.
- Если A — автомат, $L(A)$ — его язык.
- Для КДА A, $L(A)$ есть множество строк, помечающих пути из начального состояния в финальное
- Формально: $L(A)$

Доказательство равенства множеств

Часто нам нужно доказать, что два описания множеств определят фактически одно и то же множество.

В нашем случае одно множество “КДА”

Док-во

- В общем, чтобы доказать $S = T$, нам нужно доказать две части: $S \subseteq T \wedge T \subseteq S$. То есть
- Если w есть в S, то $w \in T$
- $w \in T \Rightarrow w \in S$
- В нашем случае:

$S = L(A)$, т.е., quote.angle.l.double S — есть язык нашего автомата A quote.angle.r.double

$T =$ в словах языка нет двух последовательных единиц

Часть 1: $S \subseteq T$

-
- Доказательство индукцией по..

Индуктивная гипотеза

$$\delta(A, w) = A \Rightarrow \text{последовательные } 1 \notin w \wedge w \text{ не заканчивается на } 1$$
$$\delta(A, w) = B \Rightarrow$$

- Базис: $|w| = 0$; т.е. $w = \varepsilon$
  - (1) выполняется, т.к. ∊ не имеет 1 совсем
- (2) выполняется т.к. $\delta(a, b)$ не является B

Индуктивный шаг

- Пусть (1) и (2) выполняются для строк короче чем w ....
- По предположению ...
- Так как w не пуста, можно записать $w = xa$

Индуктивный шаг — 2

- Нужно доказать (1)

Индуктивный шаг — 3

- Так как $\delta(A, w) =$

Часть 2: $T \subseteq S$

- Теперь мы должны доказать: если w не имеет 11, то принимается
- От противного: если