Лекции по теории формальных языков

Лекция 3.
Автоматы с магазинной памятью.
Грамматики

Александр Сергеевич Герасимов
http://gas-teach.narod.ru

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета
Российской академии наук.
Весенний семестр 2010/11 учебного года

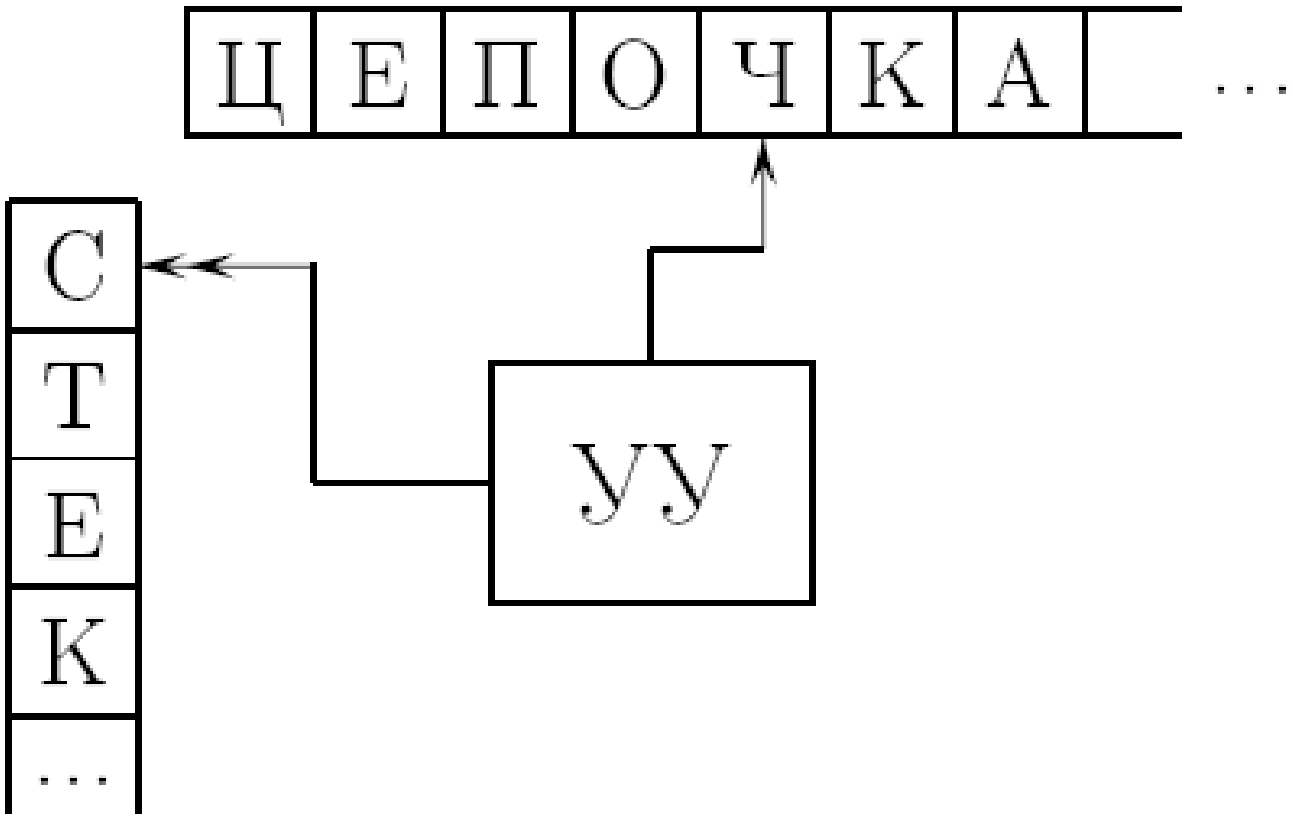
25 февраля 2011 г.

План

План

Автомат с магазинной памятью как физическое устройство

Рассмотрим более мощный, чем конечный автомат, распознаватель языка — *автомат с магазинной памятью* (*МП-автомат*, *МПА*, *магазинный автомат*, *pushdown automaton*, *PDA*).



Неформальное описание работы МПА как физического устройства

На каждом такте устройство управления (УУ) анализирует

- своё текущее состояние q,
- обозреваемый входной символ a,
- символ B на вершине стека

и принимает решение о том,

- в какое состояние q' перейти,
- какую цепочку записать в стек вместо символа B,
- следует ли сдвинуть указатель на следующую справа ячейку входной ленты ($\vec{}$) или оставить его на месте ($\_$).

Описанную работу на такте мы будем рассматривать как выполнение автоматом команды

$$(q, a, B) \to (q', \gamma, \vec{}) \quad \text{или} \quad (q, a, B) \to (q', \gamma, \_).$$

Автомат допускает цепочку, записанную на входной ленте, если он дошёл до её конца и оказался при этом в одном из заключительных состояний.

Определение МПА

Автоматом с магазинной памятью (*МПА*) называется семёрка $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \gamma_0)$, где

- Q — непустое конечное множество *состояний*,
- Σ — *входной* алфавит,
- Γ — *стековый* (или *магазинный*) алфавит,
- δ — конечное множество команд вида

$$(q, a, B) \to (q', \gamma, \overrightarrow{\ }) \quad \text{или} \quad (q, a, B) \to (q', \gamma, \_)$$

(где $q \in Q$, $a \in \Sigma$, $B \in \Gamma$, $q' \in Q$, $\gamma \in \Gamma^*$),
иначе говоря, δ — конечное подмножество множества $(Q \times \Sigma \times \Gamma) \times (Q \times \Gamma^* \times \{\overrightarrow{\ }, \_\})$,

- $q_0 \in Q$ — *начальное* состояние,
- $F \subseteq Q$ — множество *заключительных* состояний,
- $\gamma_0 \in \Gamma^*$ — *начальное содержимое стека* (*магазина*).

Соглашения о символах-ограничителях

- Для удобства будем считать, что входной и стековый алфавиты МПА содержат по специальному символу-ограничителю: $\dashv$ и ∇ соответственно.
- Символ конца входной строки $\dashv$ размещается на входной ленте непосредственно справа от входной цепочки (и не входит в неё).
- Символ дна стека ∇ размещается под нижним символом в стеке (и в стеке больше нигде и никогда не встречается). Когда стек пуст, его верхним символом является ∇, а при записи в стек символ дна остается на своём месте, т. е. при выполнении команды вида $(q, a, \nabla) \to (q', \gamma, ?)$ цепочка γ будет записана в стек сверху символа ∇.

Конфигурация МПА

Конфигурацией МПА $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, \mathrm{q}_0, F, \gamma_0)$ называется тройка $[\mathrm{q}, w, \gamma]$, где

- $\mathrm{q} \in Q$ — текущее состояние автомата,
- $w \in (\Sigma \setminus \{\dashv\})^*$ — суффикс входной цепочки, начинающийся с позиции указателя (и не включающий символа конца входной цепочки),
- $\gamma \in (\Gamma \setminus \{\nabla\})^*$ — содержимое стека (начиная с верхнего символа и не включая символа дна стека).

Переход МПА в следующую конфигурацию

Пусть дан МПА $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \gamma_0)$. Для любых $q, q' \in Q$, $a \in \Sigma \setminus \{\dashv\}$, $w \in (\Sigma \setminus \{\dashv\})^*$, $B \in \Gamma \setminus \{\nabla\}$, $\gamma, \gamma' \in (\Gamma \setminus \{\nabla\})^*$ будем писать

- $[q, aw, B\gamma] \models [q', w, \gamma'\gamma]$, если в автомате $\mathcal{M}$ имеется команда $(q, a, B) \to (q', \gamma', \vec{\ })$;
- $[q, aw, B\gamma] \models [q', aw, \gamma'\gamma]$, если в автомате $\mathcal{M}$ имеется команда $(q, a, B) \to (q', \gamma', \_)$;
- $[q, aw, \varepsilon] \models [q', w, \gamma']$, если в автомате $\mathcal{M}$ имеется команда $(q, a, \nabla) \to (q', \gamma', \vec{\ })$;
- $[q, aw, \varepsilon] \models [q', aw, \gamma']$, если в автомате $\mathcal{M}$ имеется команда $(q, a, \nabla) \to (q', \gamma', \_)$;
- $[q, \varepsilon, B\gamma] \models [q', \varepsilon, \gamma'\gamma]$, если в автомате $\mathcal{M}$ имеется команда $(q, \dashv, B) \to (q', \gamma', \_)$;
- $[q, \varepsilon, \varepsilon] \models [q', \varepsilon, \gamma']$, если в автомате $\mathcal{M}$ имеется команда $(q, \dashv, \nabla) \to (q', \gamma', \_)$;

и говорить, что конфигурация, указанная справа от знака $\models$, *следует* за конфигурацией, указанной слева от этого знака.

Язык, распознаваемый МПА

- Пусть дан МПА $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, \mathrm{q}_0, F, \gamma_0)$.
- Запись $[\mathrm{q}, w, \gamma] \models^* [\mathrm{q}', w', \gamma']$ будет означать, что автомат $\mathcal{M}$ может перейти из первой конфигурации во вторую за конечное (в том числе 0) число шагов.
- МПА $\mathcal{M}$ *распознаёт* (или *допускает*) цепочку $w \in (\Sigma \setminus \{\dashv\})^*$, если $[\mathrm{q}_0, w, \gamma_0] \models^* [\mathtt{f}, \varepsilon, \gamma]$ для некоторых $\mathtt{f} \in F$, $\gamma \in (\Gamma \setminus \{\nabla\})^*$.
- Множество всех цепочек, допускаемых МПА $\mathcal{M}$, называется *языком, распознаваемым* $\mathcal{M}$, и обозначается $L(\mathcal{M})$.
- Два МПА называются *эквивалентными*, если они распознают один и тот же язык.

Недетерминированные и детерминированные МПА

- Мы дали выше определение МПА, или недетерминированного МПА (НМПА). НМПА может иметь более одной команды с одной и той же левой частью (q, a, B).
- Определение детерминированного МПА (ДМПА) получаем из этого определения, дополнительно потребовав, что имеется не более одной команды с одной и той же левой частью (q, a, B), т. е. множество команд δ — частичная функция из $Q \times \Sigma \times \Gamma$ в $Q \times \Gamma^* \times \{\rightarrow, \_\}$.
- Если не будет указано иначе, далее под МПА будет пониматься ДМПА.

МПА, распознающий скобочный язык LB

- Определим МПА $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, \mathtt{q}, F, \varepsilon)$.
- $Q = \{\mathtt{q}, \mathtt{f}\}$, $\Sigma = \{[,]\}$, $\Gamma = \{[\}$, $F = \{\mathtt{f}\}$.
- Множество команд δ будем представлять *управляющей таблицей*, проиндексированной множествами Q, Σ и Γ: тройка индексов $(\mathtt{q}, \mathtt{a}, B)$, являющаяся левой частью команды, адресует в таблице клетку, представляющую правую часть этой команды. Не пишем в клетке состояние, если оно совпадает с текущим, и опускаем _.

		[	]	⊣
q	[	$[[, \rightarrow$	$\varepsilon, \rightarrow$	
	∇	$[, \rightarrow$		$\mathtt{f}, \varepsilon$
f	[			
	∇			

$L(\mathcal{M}) = LB$.

Команда допуска цепочки

- На предыдущем слайде можно было обойтись без состояния `f`: вместо перехода в состояние `f` в клетке $(\mathrm{q}, \dashv, \nabla)$ достаточно было поместить указание «допустить цепочку».
- Далее мы будем использовать в МПА специальную команду допуска цепочки, записываемую как $(\mathrm{q}, \mathrm{a}, B) \to \checkmark$.
- Любой МПА с командами допуска эквивалентен некоторому МПА без них (здесь под МПА понимаются как ДМПА, так и НМПА соответственно):
  - добавим новое состояние, объявив его заключительным,
  - в этом состоянии для любых символов на входе и на стеке выполняется только сдвиг по входной цепочке,
  - заменим все команды допуска на команды перехода в новое состояние.

МПА, распознающие язык $\{\mathtt{a}^n\mathtt{b}^n \mid n \geqslant 0\}$

1. В первом МПА состояние $\mathtt{q}$ — начальное и единственное заключительное. Этот МПА начинает работу с пустым стеком.

		$\mathtt{a}$	$\mathtt{b}$	$\dashv$
$\mathtt{q}$	$\mathtt{a}$	$\mathtt{aa}, \vec{}$	$\mathtt{r}, \varepsilon, \vec{}$	
	∇	$\mathtt{a}, \vec{}$		
$\mathtt{r}$	$\mathtt{a}$		$\varepsilon, \vec{}$	
	∇			$\mathtt{q}, \varepsilon$

2. Во втором МПА имеется лишь одно состояние, но введён новый стековый символ X. В начале работы в стеке находится символ X.

	$\mathtt{a}$	$\mathtt{b}$	$\dashv$
$\mathtt{a}$		$\varepsilon, \vec{}$	
X	$X\mathtt{a}, \vec{}$	ε	ε
∇			✓

Предложение о сравнении распознающих возможностей ДМПА и НМПА

Для произвольной цепочки $w = a_1 \dots a_n$ через $\overleftarrow{w}$ мы будем обозначать цепочку $a_n \dots a_1$.

Предложение

Если алфавит Σ содержит более одного символа, то язык $L = \{w\overleftarrow{w} \mid w \in \Sigma^*\}$ распознаётся некоторым НМПА, но не распознаётся никаким ДМПА.

Доказательство.

1. Определим НМПА $\mathcal{M}$, распознающий язык L.

Возьмём произвольный символ $X \notin \Sigma$. Положим, что $\Sigma \cup \{X\}$ — стековый алфавит, X — начальное содержимое стека. $\mathcal{M}$ имеет единственное состояние.

Предложение о сравнении распознающих возможностей ДМПА и НМПА: продолжение доказательства

Управляющая таблица НМПА $\mathcal{M}$ при $\Sigma = \{\mathtt{a}, \mathtt{b}\}$:

	$\mathtt{a}$	$\mathtt{b}$	$\dashv$
X	$X\mathtt{a}, \vec{}$ ε	$X\mathtt{b}, \vec{}$ ε	ε
$\mathtt{a}$	$\varepsilon, \vec{}$		
$\mathtt{b}$		$\varepsilon, \vec{}$	
∇			✓

$L(\mathcal{M}) = L$.

Предложение о сравнении распознающих возможностей ДМПА и НМПА: окончание доказательства

2. Покажем (нестрого), что никакой ДМПА не распознаёт язык L.

- Предположим, что некоторый ДМПА $\mathcal{M}'$ распознаёт L.
- Тогда для некоторых $w \in \Sigma^*$ и $a \in \Sigma$ автомат $\mathcal{M}'$, анализируя входную цепочку $waa\overleftarrow{w}$, после прочтения её префикса wa, начинает сравнивать прочитанную и непрочитанную части цепочки $waa\overleftarrow{w}$.
- Пусть $\mathcal{M}'$ после прочтения префикса wa цепочки $waa\overleftarrow{w}$ находится в состоянии q с символом B на вершине стека.
- ДМПА $\mathcal{M}'$ имеет единственную команду с левой частью (q, a, B). Эта команда указывает на начало сравнения прочитанной и непрочитанной частей цепочки $waa\overleftarrow{w}$.
- Но $\mathcal{M}'$ не сможет распознать цепочку $waaaa\overleftarrow{w} \in L$, так как начнёт сравнение сразу после прочтения префикса wa этой цепочки. Противоречие.

МПА, распознающий язык списков LL

- Алфавит языка LL: $\{\mathrm{a},\ ;,\ [,\ ]\}$.
  Определение языка LL:
  - $\mathrm{a} \in LL$;
  - если $u \in LL$ и $v \in LL$, то $[u] \in LL$ и $u; v \in LL$.
- МПА имеет единственное состояние, в начале работы в стеке находится символ L.

	a	;	[	]	$\dashv$
L	$M, \rightarrow$		$L]M, \rightarrow$		
M		$L, \rightarrow$		ε	ε
]				$\varepsilon, \rightarrow$	
∇					✓

МПА, распознающий язык списков LL: протокол обработки цепочки

Такт	Позиция указателя	Содержимое стека
1	$\diamond$a; [a; a; a] $\dashv$	$L\nabla$
2	a$\diamond$; [a; a; a] $\dashv$	$M\nabla$
3	a; $\diamond$[a; a; a] $\dashv$	$L\nabla$
4	a; [$\diamond$a; a; a] $\dashv$	$L]M\nabla$
5	a; [a$\diamond$; a; a] $\dashv$	$M]M\nabla$
6	a; [a; $\diamond$a; a] $\dashv$	$L]M\nabla$
7	a; [a; a$\diamond$; a] $\dashv$	$M]M\nabla$
8	a; [a; a; $\diamond$a] $\dashv$	$L]M\nabla$
9	a; [a; a; a$\diamond$] $\dashv$	$M]M\nabla$
10	a; [a; a; a$\diamond$] $\dashv$	$]M\nabla$
11	a; [a; a; a]$\diamond$ $\dashv$	$M\nabla$
12	a; [a; a; a]$\diamond$ $\dashv$	∇

МПА, распознающий язык арифметических выражений LA

- Алфавит языка LA: $\{\mathtt{x},\ +,\ *,\ (,\ )\}$.
  Определение языка LA:
  - $\mathtt{x} \in LA$;
  - если $u \in LA$ и $v \in LA$, то $(u) \in LA$, $u + v \in LA$ и $u * v \in LA$.
- МПА имеет единственное состояние, в начале работы в стеке находится символ E.

	$\mathtt{x}$	$+$	$*$	$($	$)$	$\dashv$
E	TE'			TE'		
E'		$TE', \rightarrow$			ε	ε
T	FT'			FT'		
T'		ε	$FT', \rightarrow$		ε	ε
F	$\varepsilon, \rightarrow$			$E), \rightarrow$		
$)$					$\varepsilon, \rightarrow$	
∇						$\checkmark$

1	$\diamond(\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$E\nabla$
2	$\diamond(\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$TE'\nabla$
3	$\diamond(\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$FT'E'\nabla$
4	$(\diamond\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$E)T'E'\nabla$
5	$(\diamond\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$TE')T'E'\nabla$
6	$(\diamond\mathrm{x}+\mathrm{x})*\mathrm{x}\dashv$	$FT'E')T'E'\nabla$
7	$(\mathrm{x}\diamond+\mathrm{x})*\mathrm{x}\dashv$	$T'E')T'E'\nabla$
8	$(\mathrm{x}\diamond+\mathrm{x})*\mathrm{x}\dashv$	$E')T'E'\nabla$
9	$(\mathrm{x}+\diamond\mathrm{x})*\mathrm{x}\dashv$	$TE')T'E'\nabla$
10	$(\mathrm{x}+\diamond\mathrm{x})*\mathrm{x}\dashv$	$FT'E')T'E'\nabla$
11	$(\mathrm{x}+\mathrm{x}\diamond)*\mathrm{x}\dashv$	$T'E')T'E'\nabla$
12	$(\mathrm{x}+\mathrm{x}\diamond)*\mathrm{x}\dashv$	$E')T'E'\nabla$
13	$(\mathrm{x}+\mathrm{x}\diamond)*\mathrm{x}\dashv$	$)T'E'\nabla$
14	$(\mathrm{x}+\mathrm{x})\diamond*\mathrm{x}\dashv$	$T'E'\nabla$
15	$(\mathrm{x}+\mathrm{x})*\diamond\mathrm{x}\dashv$	$FT'E'\nabla$
16	$(\mathrm{x}+\mathrm{x})*\mathrm{x}\diamond\dashv$	$T'E'\nabla$
17	$(\mathrm{x}+\mathrm{x})*\mathrm{x}\diamond\dashv$	$E'\nabla$
18	$(\mathrm{x}+\mathrm{x})*\mathrm{x}\diamond\dashv$	∇

План

Определение грамматики

Грамматики (и особенно контекстно-свободные грамматики) – основной способ задания формальных языков.

Грамматикой называется четвёрка $G = (\Sigma, \Gamma, P, S)$, где

- Σ — *основной* (или *терминальный*) алфавит, элементы которого называют *терминалами* (или *терминальными символами*);
- Γ — *вспомогательный* (или *нетерминальный*) алфавит, элементы которого называют *нетерминалами* (*нетерминальными* или *вспомогательными символами*); $\Sigma \cap \Gamma = \emptyset$;
- $S \in \Gamma$ — выделенный нетерминал, называемый *аксиомой* (или *начальным нетерминалом*);
- P — конечное множество слов вида $\alpha \to \beta$ (где $\alpha \in (\Sigma \cup \Gamma)^* \Gamma (\Sigma \cup \Gamma)^*$, $\beta \in (\Sigma \cup \Gamma)^*$), каждое такое слово называют *правилом вывода* (просто *правилом* или *продукцией*); $\to \notin \Sigma \cup \Gamma$.

Пример. Грамматика $G_1 = (\Sigma, \Gamma, P, S)$: $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, $\Gamma = \{S, D\}$, S — аксиома, $P = \{S \to \mathtt{a}SD\mathtt{a},\ S \to \mathtt{aba},\ \mathtt{a}D \to D\mathtt{a},\ \mathtt{b}D \to \mathtt{bb}\}$.

Непосредственная выводимость в грамматике

Пусть $G = (\Sigma, \Gamma, P, S)$ — грамматика, $\gamma, \delta \in (\Sigma \cup \Gamma)^*$.

- Говорят, что цепочка δ *непосредственно выводима* в G из цепочки γ (и обозначают это через $\gamma \Rightarrow_G \delta$ или через $\gamma \Rightarrow \delta$), если существуют такие цепочки $\alpha, \beta, \gamma_1, \gamma_2 \in (\Sigma \cup \Gamma)^*$, что $\gamma = \gamma_1\alpha\gamma_2$, $\delta = \gamma_1\beta\gamma_2$ и $\alpha \to \beta \in P$.
- Пример непосредственной выводимости в G_1:
  - $\underbrace{\mathrm{a}}_{\gamma_1} S \underbrace{D\mathrm{a}}_{\gamma_2} \Rightarrow \underbrace{\mathrm{a}}_{\gamma_1}\mathrm{aba}\underbrace{D\mathrm{a}}_{\gamma_2}$ (правило $S \to \mathrm{aba} \in P$),
  - $\mathrm{a}D\mathrm{b}S\mathrm{a} \Rightarrow D\mathrm{ab}S\mathrm{a}$ ($\mathrm{a}D \to D\mathrm{a} \in P$),
  - $S \Rightarrow \mathrm{aba}$ ($S \to \mathrm{aba} \in P$).

Выводимость в грамматике

- Говорят, что цепочка δ *выводима* в G из цепочки γ (и обозначают это через $\gamma \Rightarrow_G^* \delta$ или через $\gamma \Rightarrow^* \delta$), если существуют такие число $n \geqslant 0$ и цепочки $\eta_0, \eta_1, \ldots, \eta_n$, что

$$\gamma = \eta_0 \Rightarrow \eta_1 \Rightarrow \eta_2 \Rightarrow \ldots \Rightarrow \eta_n = \delta.$$

  Последовательность цепочек $\eta_0, \eta_1, \ldots, \eta_n$ называется *выводом* (в G цепочки δ из цепочки γ), а число n — *длиной* этого вывода.
- Отношение выводимости $\Rightarrow_G^*$ есть рефлексивно-транзитивное замыкание отношения непосредственной выводимости $\Rightarrow_G$. Транзитивное замыкание отношения непосредственной выводимости $\Rightarrow_G$ мы будем обозначать через $\Rightarrow_G^+$ (или через $\Rightarrow^+$).
- Пример. Рассматриваем грамматику G_1 с правилами $P = \{S \to \mathrm{a}S D\mathrm{a},\ S \to \mathrm{aba},\ \mathrm{a}D \to D\mathrm{a},\ \mathrm{b}D \to \mathrm{bb}\}$.
  $S \Rightarrow_{G_1}^* \mathrm{a}^2\mathrm{b}^2\mathrm{a}^2$, поскольку

$$S \Rightarrow \mathrm{a}S D\mathrm{a} \Rightarrow \mathrm{aaba}D\mathrm{a} \Rightarrow \mathrm{aab}D\mathrm{aa} \Rightarrow \mathrm{aabbaa}\ .$$

Язык, порождаемый грамматикой

- *Языком, порождаемым грамматикой* $G = (\Sigma, \Gamma, P, S)$, называется множество $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.
- Грамматики G и G' называются *эквивалентными*, если $L(G) = L(G')$.
- Пример. $\mathrm{aba} \in L(G_1)$, $\mathrm{a}^2\mathrm{b}^2\mathrm{a}^2 \in L(G_1)$. Можно показать, что $L(G_1) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n \mid n \geqslant 1\}$.
  Установим, что $L(G_1) \supseteq \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n \mid n \geqslant 1\}$:
  - применим $(n-1)$ раз правило $S \to \mathrm{a}S D\mathrm{a}$, а затем правило $S \to \mathrm{aba}$:

    $$S \Rightarrow \mathrm{a}SD\mathrm{a} \Rightarrow \mathrm{aa}SD\mathrm{a}D\mathrm{a} \Rightarrow \ldots \Rightarrow \mathrm{a}^{n-1}S(D\mathrm{a})^{n-1} \Rightarrow \mathrm{a}^n\mathrm{ba}(D\mathrm{a})^{n-1};$$

  - далее, пока это возможно, применяем правило $\mathrm{a}D \to D\mathrm{a}$:

    $$\mathrm{a}^n\mathrm{ba}(D\mathrm{a})^{n-1} \Rightarrow^* \mathrm{a}^n\mathrm{b}D^{n-1}\mathrm{a}^n;$$

  - наконец, $(n-1)$ раз применяем правило $\mathrm{b}D \to \mathrm{bb}$:

    $$\mathrm{a}^n\mathrm{b}D^{n-1}\mathrm{a}^n \Rightarrow^* \mathrm{a}^n\mathrm{b}^n\mathrm{a}^n.$$

Соглашение об альтернативах

- Правила $\alpha \to \beta_1$, $\alpha \to \beta_2$, $\ldots$, $\alpha \to \beta_n$ (называемые альтернативами) будем записывать в виде $\alpha \to \beta_1 | \beta_2 | \ldots | \beta_n$.
- Пример. Грамматика $G_2 = (\Sigma, \Gamma, P, S)$, порождающая все непустые цепочки из строчных латинских букв и десятичных цифр, начинающиеся с буквы:
  - $\Sigma = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, 0, 1, \ldots, 9\}$,
  - $\Gamma = \{\langle \mathtt{имя} \rangle, \langle \mathtt{буква} \rangle, \langle \mathtt{цифра} \rangle\}$,
  - $\langle \mathtt{имя} \rangle$ — аксиома,
  - множество правил P:
    $\langle \mathtt{имя} \rangle = \langle \mathtt{буква} \rangle | \langle \mathtt{имя} \rangle \langle \mathtt{буква} \rangle | \langle \mathtt{имя} \rangle \langle \mathtt{цифра} \rangle$,
    $\langle \mathtt{буква} \rangle = \mathtt{a} | \mathtt{b} | \ldots | \mathtt{z}$,
    $\langle \mathtt{цифра} \rangle = 0 | 1 | \ldots | 9$.

Соглашения о символах и цепочках

Грамматику можно задавать только множеством правил вывода, если

- терминалы обозначать строчными буквами из начала латинского алфавита ($\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \ldots$);
- нетерминалы — заглавными буквами из начала латинского алфавита ($A, B, C, D, E, F, \ldots, S, T$) или словами в угловых скобках (например, $\langle \mathtt{имя} \rangle, \langle \mathtt{буква} \rangle, \langle \mathtt{цифра} \rangle$);
- символы из объединения терминального и нетерминального алфавитов — заглавными буквами из конца латинского алфавита (U, V, W, X, Y, Z);
- цепочки терминалов — строчными буквами из конца латинского алфавита (u, v, w, x, y, z);
- цепочки символов из объединения терминального и нетерминального алфавитов — строчными греческими буквами ($\alpha, \beta, \gamma, \ldots, \omega$);
- в левой части первого правила вывода стоит аксиома.

При этом можно добавлять индексы.

Грамматика, порождающая язык $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geqslant 1\}$

- Применим $(n-1)$ раз правило $S \to \mathtt{a}SBC$, а затем правило $S \to \mathtt{ab}C$:

$$S \Rightarrow \mathtt{a}SBC \Rightarrow \mathtt{aa}SBCBC \Rightarrow \ldots \Rightarrow \mathtt{a}^{n-1}S(BC)^{n-1} \Rightarrow \mathtt{a}^n\mathtt{b}C(BC)^{n-1}.$$

- Далее, пока это возможно, применяем правило $CB \to BC$:

$$\mathtt{a}^n\mathtt{b}C(BC)^{n-1} \Rightarrow^* \mathtt{a}^n\mathtt{b}B^{n-1}C^n.$$

- Теперь $(n-1)$ раз применяем правило $\mathtt{b}B \to \mathtt{bb}$:

$$\mathtt{a}^n\mathtt{b}B^{n-1}C^n \Rightarrow^* \mathtt{a}^n\mathtt{b}^nC^n.$$

- Наконец, применяем правило $\mathtt{b}C \to \mathtt{bc}$, а затем $(n-1)$ раз правило $\mathtt{c}C \to \mathtt{cc}$:

$$\mathtt{a}^n\mathtt{b}^nC^n \Rightarrow \mathtt{a}^n\mathtt{b}^n\mathtt{c}C^{n-1} \Rightarrow^* \mathtt{a}^n\mathtt{b}^n\mathtt{c}^n.$$

Грамматика, порождающая язык $L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid$ равны числа вхождений $\mathtt{a}$ и $\mathtt{b}$ в $w\}$

- Правила $S \to ASB|\varepsilon$ позволяют получить цепочки вида $A^n B^n$ $(n \geqslant 0)$.
- С помощью правила $AB \to BA$ можно переставлять буквы A и B.
- Наконец, правила $A \to \mathtt{a}$, $B \to \mathtt{b}$ превращают цепочки нетерминалов в цепочки терминалов языка L.

Вторая грамматика, порождающая язык $L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid$ равны числа вхождений $\mathtt{a}$ и $\mathtt{b}$ в $w\}$

- Обозначим через $\#_{\mathtt{c}}(z)$ число вхождений символа $\mathtt{c}$ в цепочку z.
- Если $\mathtt{a}u \in L$, то $\#_{\mathtt{a}}(u) + 1 = \#_{\mathtt{b}}(u)$.
  Пусть B — нетерминал, из которого выводятся все цепочки u такие, что $\#_{\mathtt{a}}(u) + 1 = \#_{\mathtt{b}}(u)$.
  Имеем правило $S \to \mathtt{a}B$.
- Аналогично, пусть A — нетерминал, из которого выводятся все цепочки v такие, что $\#_{\mathtt{a}}(v) = \#_{\mathtt{b}}(v) + 1$. Имеем правило $S \to \mathtt{b}A$.
- Рассмотрим цепочку u, выводимую из B.
  - Если $u = \mathtt{a}x$, то $\#_{\mathtt{a}}(x) + 2 = \#_{\mathtt{b}}(x)$. Добавим правило $B \to \mathtt{a}BB$.
  - Если $u = \mathtt{b}x$, то $\#_{\mathtt{a}}(x) = \#_{\mathtt{b}}(x)$. Добавим правило $B \to \mathtt{b}S$.
- Аналогично будем иметь правила $A \to \mathtt{b}AA|\mathtt{a}S$.
- Наконец, добавим правило $S \to \varepsilon$.

Грамматика, порождающая язык $\{wcw \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$

- С помощью правил $S \to \mathtt{a}AS$, $S \to \mathtt{b}BS$ и $S \to \mathtt{c}$ породим цепочку вида $V\mathtt{c}$, где в V за каждым $\mathtt{a}$ стоит A и за каждым $\mathtt{b}$ стоит B.
- Затем переместим A и B направо (сохранив их порядок относительно друг друга) правилами $A\mathtt{a} \to \mathtt{a}A$, $A\mathtt{b} \to \mathtt{b}A$, $B\mathtt{a} \to \mathtt{a}B$, $B\mathtt{b} \to \mathtt{b}B$. Получим цепочку вида $wW\mathtt{c}$, где W есть цепочка w, но записанная заглавными буквами.
- Наконец, превратим A и B в $\mathtt{a}$ и $\mathtt{b}$ соответственно и поставим их после символа $\mathtt{c}$ посредством правил $A\mathtt{c} \to \mathtt{ca}$, $B\mathtt{c} \to \mathtt{cb}$.

Классы грамматик и языков: иерархия Хомского

Грамматика $G = (\Sigma, \Gamma, P, S)$ называется

(0) грамматикой *общего вида* (*грамматикой типа 0* или грамматикой *без ограничений*);

(1) *контекстно-зависимой* (*грамматикой типа 1* или *неукорачивающей* грамматикой), если каждое её правило имеет вид
- $\alpha A \beta \to \alpha \gamma \beta$, где $\gamma \neq \varepsilon$, или
- $S \to \varepsilon$, но тогда S не входит в правую часть никакого правила;

(2) *контекстно-свободной* (*грамматикой типа 2* или *бесконтекстной* грамматикой), если каждое её правило имеет вид $A \to \alpha$;

(3) *праволинейной* (*грамматикой типа 3*, *автоматной* или *регулярной* грамматикой), если каждое её правило имеет вид $A \to \mathrm{a}B$, $A \to \mathrm{a}$ или $A \to \varepsilon$.

Языку приписывается тип грамматики, которой он порождается. Например, контекстно-свободные грамматики порождают так называемые контекстно-свободные языки (языки типа 2).

Соотношения между классами языков

Пусть $\mathbb{L}_i$ — класс языков типа i $(i = 0, 1, 2, 3)$.

$$\mathbb{L}_3 \subsetneq \mathbb{L}_2 \subsetneq \mathbb{L}_1 \subsetneq \text{класс разрешимых языков} \subsetneq$$
$$\mathbb{L}_0 = \text{класс перечислимых языков.}$$

- Язык L в алфавите Σ называется *разрешимым*, если существует алгоритм (например, машина Тьюринга), который для любой цепочки $w \in \Sigma^*$ выдаёт 1, если $w \in L$, и выдаёт 0, если $w \notin L$.
- Язык L в алфавите Σ называется *перечислимым*, если существует алгоритм, который для любой цепочки $w \in \Sigma^*$ выдаёт 1, если $w \in L$, и не заканчивает работу в противном случае.

(Подробнее о разрешимых и перечислимых множествах можно прочесть, например, в книге: Герасимов А. С. Курс математической логики и теории вычислимости: Учебное пособие. СПб.: Изд-во «ЛЕМА», 2011; электронный вариант книги — на `http://gas-teach.narod.ru/cmlct`.)

Связь классов языков и распознавателей

- $\mathbb{L}_2$ совпадает с классом языков, распознаваемых НМПА (увидим в дальнейшем).
- $\mathbb{L}_3$ совпадает с классом языков, распознаваемых конечными автоматами (упражнение).

Практически всегда для описания синтаксиса языков программирования используются контекстно-свободные грамматики, которые мы и будем изучать.

Литература

Основная литература

- Замятин А. П., Шур А. М. Языки, грамматики, распознаватели: Учебное пособие. Екатеринбург : Изд-во Урал. ун-та, 2007 (электронный вариант книги — на `http://elar.usu.ru`, поиск).

Дополнительная литература

- Ахо А., Лам М., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструментарий. М.: ООО "И.Д. Вильямс", 2008.
- Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.
- Мартыненко Б. К. Языки и трансляции: Учеб. пособие. СПб.: Издательство С.-Петербургского университета, 2004 (электронный вариант книги — на `http://www.math.spbu.ru/user/mbk`).