
Лекция 02.
Задача описания входного языка



Основной аппарат

- Формальные языки и грамматики

математические модели,
использующие представление текстов
в виде **цепочек символов**



Замечания (1)

- Для описания языков программирования используются контекстно-свободные грамматики (КСГ).
- КСГ – мощный аппарат, но не может определить все возможные языки.
- Эффективны для описания вложенных структур, например, скобок и блоков в языках программирования.



Замечания (2)

- Основная идея заключается в использовании «переменных» для определения «множеств» цепочек символов.
- Эти переменные определены рекурсивно (с помощью рекурсивных «правил вывода»).
- Рекурсивные правила для переменной («продукции») включают в себя только конкатенацию.
- Альтернативные правила для переменной позволяют объединять цепочки.



Основные определения (1)

Алфавит

□ **Определение:**

алфавит - это конечное множество символов.

- Например, алфавит $\mathbf{A} = \{a, b, c, +, !\}$ содержит 5 букв, а алфавит $\mathbf{B} = \{00, 01, 10, 11\}$ содержит 4 буквы, каждая из которых состоит из двух символов.

□ **Определение:**

цепочкой символов в алфавите \mathbf{V} называется любая конечная последовательность символов этого алфавита.

□ **Определение:**

цепочка, которая не содержит ни одного символа, называется *пустой цепочкой*. Для ее обозначения будем использовать символ λ .



Основные определения (2)

Операции над цепочками

□ **Определение:**

если a и b - цепочки, то цепочка ab называется *конкатенацией* (или *сцеплением*) цепочек a и b .

- Например, если $a = ab$ и $b = cd$, то $ab = abcd$.
- Для любой цепочки a всегда $a\lambda = \lambda a = a$.

□ **Определение:**

обращением (или *реверсом*) цепочки a называется цепочка, символы которой записаны в обратном порядке.

Обращение цепочки a будем обозначать a^R .

- Например, если $a = abcdef$, то $a^R = fedcba$.
- Для пустой цепочки: $\lambda = \lambda^R$.

□ **Определение:** n -ой степенью цепочки a (будем обозначать a^n) называется конкатенация n цепочек a .

- $a^0 = \lambda$; $a^n = aa^{n-1} = a^{n-1}a$.

□ **Определение:** *длина цепочки* - это число составляющих ее символов.



Основные определения (3)

Язык. Итерация

- **Определение:** язык в алфавите \mathbf{V} - это подмножество цепочек конечной длины в этом алфавите.

- **Определение:** обозначим через \mathbf{V}^* множество, содержащее все цепочки в алфавите \mathbf{V} , включая пустую цепочку λ .
 - Например, если $\mathbf{V}=\{0,1\}$,
то $\mathbf{V}^* = \{\lambda, 0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$.

- **Определение:** обозначим через \mathbf{V}^+ множество, содержащее все цепочки в алфавите \mathbf{V} , исключая пустую цепочку λ .
- Следовательно, $\mathbf{V}^* = \mathbf{V}^+ \cup \{\lambda\}$.

- **Определение:** декартовым произведением $\mathbf{A} \times \mathbf{B}$ множеств ~~\mathbf{A} и \mathbf{B}~~ называется множество ~~$\{(a,b) \mid a \in \mathbf{A}, b \in \mathbf{B}\}$~~ .



Порождающая грамматика

- **Определение:** порождающая грамматика **G** - это четверка $(\mathbf{VT}, \mathbf{VN}, \mathbf{P}, \mathbf{S})$, где
 - **VT** - алфавит терминальных символов (терминалов),
 - **VN** - алфавит нетерминальных символов (нетерминалов, «переменных»), не пересекающийся с VT,
 - **P** - конечное подмножество множества $(\mathbf{VT} \cup \mathbf{VN})^+ \rightarrow (\mathbf{VT} \cup \mathbf{VN})^*$; элемент (α, β) множества P называется **правилом вывода** и записывается в виде $\alpha \rightarrow \beta$,
 - **S** - начальный символ (цель, аксиома) грамматики, $S \in \mathbf{VN}$.
- Для записи правил вывода с одинаковыми левыми частями $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$ будем пользоваться сокращенной записью $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$.
- Каждое $\beta_i, i = 1, 2, \dots, n$, будем называть *альтернативой* правила вывода из цепочки α .



Порождающая грамматика

□ Пример грамматики:

$$G1 = (\{0,1\}, \{A,S\}, \mathbf{P}, S),$$

- $V_T = \{0,1\}$
- $V_N = \{A,S\}$
- \mathbf{P} состоит из правил
 $S \rightarrow 0A1$
 $0A \rightarrow 00A1$
 $A \rightarrow \lambda$



Основные определения (4)

Выводимость. Выводы

- **Определение:** цепочка $\beta \in (\mathbf{VT} \cup \mathbf{VN})^*$ непосредственно выводима из цепочки $\alpha \in (\mathbf{VT} \cup \mathbf{VN})^+$ в грамматике $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ (обозначим $\alpha \rightarrow \beta$), если $\alpha = \xi_1 \gamma \xi_2$, $\beta = \xi_1 \delta \xi_2$, где $\xi_1, \xi_2, \delta \in (\mathbf{VT} \cup \mathbf{VN})^*$, $\gamma \in (\mathbf{VT} \cup \mathbf{VN})^+$ и правило вывода $\gamma \rightarrow \delta$ содержится в \mathbf{P} .
 - Например, цепочка 00A11 непосредственно выводима из 0A1 в G_1 .
- **Определение:** цепочка $\beta \in (\mathbf{VT} \cup \mathbf{VN})^*$ выводима из цепочки $\alpha \in (\mathbf{VT} \cup \mathbf{VN})^+$ в грамматике $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ (обозначим $\alpha \Rightarrow \beta$), если существуют цепочки $\gamma_0, \gamma_1, \dots, \gamma_n$ ($n \geq 0$), такие, что $\alpha = \gamma_0 \rightarrow \gamma_1 \rightarrow \dots \rightarrow \gamma_n = \beta$.
- **Определение:** последовательность $\gamma_0, \gamma_1, \dots, \gamma_n$ называется *выводом длины n* .
 - Например, $S \Rightarrow 000A111$ в грамматике G_1 (см. пример выше), т.к. существует вывод $S \rightarrow 0A1 \rightarrow 00A11 \rightarrow 000A111$. Длина вывода равна 3.



Непосредственная выводимость

□ Мы говорим, что $\alpha A \beta \rightarrow \alpha \gamma \beta$
(из $\alpha A \beta$ выводимо $\alpha \gamma \beta$),
если $A \rightarrow \gamma$ правило грамматики.

□ Пример: $S \rightarrow 01$; $S \rightarrow 0S1$.

□ **S** \rightarrow 0**S**1 \rightarrow 00**S**11 \rightarrow 000111.

The diagram shows the derivation of the string 000111 from the start symbol S. The string S is highlighted in red in each step. Colored circles and arrows illustrate the replacement of S with 0S1, 00S11, and 000111.



Выводимость

- \Rightarrow означает
“выводится за ноль или более шагов”
- **Базис:**
 $a \Rightarrow a$ для самой цепочки a .
- **Индукция:**
если $a \Rightarrow \beta$ и $\beta \rightarrow \gamma$, то $a \Rightarrow \gamma$.



Выводимость. Пример

- Пусть $S \rightarrow 01$; $S \rightarrow 0S1$ – правила грамматики.
- $S \rightarrow 0S1 \rightarrow 00S11 \rightarrow 0001111$ – вывод в грамматике.
- Тогда $S \Rightarrow S$
 - $S \Rightarrow 0S1$
 - $S \Rightarrow 00S11$
 - $S \Rightarrow 0001111$



Пример:

CFG для $\{ 0^n 1^n \mid n \geq 1 \}$

□ Правила:

$S \rightarrow 01$

$S \rightarrow 0S1$

□ **Базис (основа)**: цепочка 01 принадлежит языку.

□ **Индукция**: если w принадлежит языку, то и $0w1$ принадлежит языку.



Основные определения (5)

Язык. Сентенциальные формы

- **Определение:** языком, порождаемым грамматикой $G = (VT, VN, P, S)$, называется множество $L(G) = \{a \in VT^* \mid S \Rightarrow a\}$.
 - Другими словами, $L(G)$ - это все цепочки в алфавите VT , которые выводимы из S с помощью P .
 - Например, $L(G_1) = \{0^n 1^n \mid n > 0\}$.

- **Определение:** цепочка $a \in (VT \cup VN)^*$, для которой $S \Rightarrow a$, называется *сентенциальной формой* в грамматике $G = (VT, VN, P, S)$.
 - Таким образом, язык, порождаемый грамматикой, можно определить как множество терминальных сентенциальных форм.



Основные определения (6)

Эквивалентные грамматики

- **Определение:** грамматики G_1 и G_2 называются *эквивалентными*, если $L(G_1) = L(G_2)$.
 - Например, $G_1 = (\{0,1\}, \{A,S\}, \mathbf{P}_1, S)$ и $G_2 = (\{0,1\}, \{S\}, \mathbf{P}_2, S)$, где
 $\mathbf{P}_1: \quad S \rightarrow 0A1 \quad \mathbf{P}_2: \quad S \rightarrow 0S1 \mid 01$
 $\quad \quad 0A \rightarrow 00A1$
 $\quad \quad A \rightarrow \lambda$
эквивалентны, т.к. обе порождают язык
 $L(G_1) = L(G_2) = \{0^n 1^n \mid n > 0\}$.

- **Определение:** грамматики G_1 и G_2 *почти эквивалентны*, если
 $L(G_1) \cup \{\lambda\} = L(G_2) \cup \{\lambda\}$.



Классификация грамматик и языков по Хомскому

- **ТИП 0:** Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *грамматикой типа 0*, если на правила вывода не накладывается никаких ограничений (кроме тех, которые указаны в определении грамматики).
- **ТИП 1:**
- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *неукорачивающей грамматикой*, если каждое правило из \mathbf{P} имеет вид $\alpha \rightarrow \beta$, где $\alpha \in (\mathbf{VT} \cup \mathbf{VN})^+$, $\beta \in (\mathbf{VT} \cup \mathbf{VN})^+$ и $|\alpha| \leq |\beta|$.
- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *контекстно-зависимой (КЗ)*, если каждое правило из \mathbf{P} имеет вид $\alpha \rightarrow \beta$, где $\alpha = \xi_1 A \xi_2$; $\beta = \xi_1 \gamma \xi_2$; $A \in \mathbf{VN}$; $\gamma \in (\mathbf{VT} \cup \mathbf{VN})^+$; $\xi_1, \xi_2 \in (\mathbf{VT} \cup \mathbf{VN})^*$.



Классификация грамматик и языков по Хомскому

□ ТИП 2:

- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *контекстно-свободной (КС)*, если каждое правило из \mathbf{P} имеет вид $A \rightarrow \beta$, где $A \in \mathbf{VN}$, $\beta \in (\mathbf{VT} \times \mathbf{VN})^+$.
- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *укорачивающей контекстно-свободной (УКС)*, если каждое правило из \mathbf{P} имеет вид $A \rightarrow \beta$, где $A \in \mathbf{VN}$, $\beta \in (\mathbf{VT} \times \mathbf{VN})^*$.

□

□ ТИП 3:

- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *праволинейной*, если каждое правило из \mathbf{P} имеет вид $A \rightarrow tB$ либо $A \rightarrow t$, где $A \in \mathbf{VN}$, $B \in \mathbf{VN}$, $t \in \mathbf{VT}$.
- Грамматика $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$ называется *леволинейной*, если каждое правило из \mathbf{P} имеет вид $A \rightarrow Bt$ либо $A \rightarrow t$, где $A \in \mathbf{VN}$, $B \in \mathbf{VN}$, $t \in \mathbf{VT}$.



Соотношения между типами грамматик

- любая регулярная грамматика является КС-грамматикой;
- любая регулярная грамматика является УКС-грамматикой;
- любая КС-грамматика является КЗ-грамматикой;
- любая КС-грамматика является неукорачивающей грамматикой;
- любая КЗ-грамматика является грамматикой типа 0.
- любая неукорачивающая грамматика является грамматикой типа 0.

- **Замечание:** УКС-грамматика, содержащая правила вида $A \rightarrow \lambda$, не является КЗ-грамматикой и не является неукорачивающей грамматикой



Соотношения между типами языков

- каждый регулярный язык является КС-языком, но существуют КС-языки, которые не являются регулярными (например, $L = \{a^n b^n \mid n > 0\}$).
- каждый КС-язык является КЗ-языком, но существуют КЗ-языки, которые не являются КС-языками (например, $L = \{a^n b^n c^n \mid n > 0\}$).
- каждый КЗ-язык является языком типа 0.

- **Например**, КЗ-грамматика $G1 = (\{0,1\}, \{A,S\}, P1, S)$ и КС-грамматика $G2 = (\{0,1\}, \{S\}, P2, S)$, где

$$P1: \quad S \rightarrow 0A1 \quad P2: \quad S \rightarrow 0S1 \mid 01$$

$$0A \rightarrow 00A1$$

$$A \rightarrow \lambda$$

описывают один и тот же язык $L = L(G1) = L(G2) = \{0^n 1^n \mid n > 0\}$. Язык L будет КС-языком,



Пример.

Язык типа 0: $L = \{a^2 b^{n^2-1} \mid n \geq 1\}$

P: $S \rightarrow aaCFD$

$F \rightarrow AFB \mid AB$

$AB \rightarrow bBA$

$Ab \rightarrow bA$

$AD \rightarrow D$

$Cb \rightarrow bC$

$CB \rightarrow C$

$bCD \rightarrow \lambda$



Пример.

Язык типа 1: $L = \{\text{цепочки из } 0 \text{ и } 1 \text{ с одинаковым числом } 0 \text{ и } 1\}$

P: $S \rightarrow ASB \mid AB$

$AB \rightarrow BA$

$A \rightarrow 0$

$B \rightarrow 1$



Пример.

Язык типа 2: $L = \{(ac)^n (cb)^n \mid n > 0\}$



P: $S \rightarrow aQb \mid accb$

$Q \rightarrow cSc$



Пример.

Язык типа 3: $L = \{\omega \perp \mid \omega \in \{a,b\}^+, \text{ где нет двух рядом стоящих } a\}$

P: $S \rightarrow A\perp \mid B\perp$

$A \rightarrow a \mid Ba$

$B \rightarrow b \mid Bb \mid Ab$



Следующая тема:

«Проблема грамматического разбора.
Распознаватели»

