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 помощь 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 **A** = {a, b, c, +, !} содержит 5 букв,
    а алфавит **B** = {00, 01, 10, 11} содержит 4 буквы, каждая из которых состоит из двух символов.
- **Определение:**
  *цепочкой символов в алфавите **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 **V** - это подмножество цепочек конечной длины в этом алфавите.

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

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

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

Порождающая грамматика

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

Порождающая грамматика

- Пример грамматики:
  $G1 = (\{0,1\}, \{A,S\}, \mathbf{P}, S)$,

  - $VT = \{0,1\}$
  - $VN = \{A,S\}$
  - **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 $\alpha \to \beta$), 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
  $\gamma \to \delta$ содержится в **P**.
  - Например, цепочка 00A11 непосредственно выводима из 0A1 в G1.
- **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 $\gamma_0, \gamma_1, \ldots, \gamma_n$ $(n \geq 0)$, такие, что $\alpha = \gamma_0 \to \gamma_1 \to \ldots \to \gamma_n = \beta$.
- **Определение:** последовательность $\gamma_0, \gamma_1, \ldots, \gamma_n$ называется *выводом длины n*.
  - Например, $S \Rightarrow 000A111$ в грамматике G1 (см. пример выше), т.к. существует вывод $S \to 0A1 \to 00A11 \to 000A111$. Длина вывода равна 3.

Непосредственная выводимость

- Мы говорим, что $\alpha A\beta \to \alpha\gamma\beta$ (из $\alpha A\beta$ выводимо $\alpha\gamma\beta$), если $A \to \gamma$ правило грамматики.
- Пример: $S \to 01$; $S \to 0S1$.
- $\mathbf{S} \to 0\mathbf{S}1 \to 00\mathbf{S}11 \to 000111.$

Выводимость

- $\Rightarrow$ означает "выводится за ноль или более шагов"

- Базис: $\alpha \Rightarrow \alpha$ для самой цепочки α.
- Индукция: если $\alpha \Rightarrow \beta$ и $\beta \rightarrow \gamma$, то $\alpha \Rightarrow \gamma$.

Выводимость. Пример

- Пусть $S \to 01$; $S \to 0S1$ – правила грамматики.
- $S \to 0S1 \to 00S11 \to 000111$ – вывод в грамматике.
- Тогда $S \Rightarrow S$

  $S \Rightarrow 0S1$

  $S \Rightarrow 00S11$

  $S \Rightarrow 000111$

Пример:
CFG для $\{ 0^n 1^n \mid n \geq 1\}$

- Правила:

  S -> 01

  S -> 0S1

- Базис (основа): цепочка 01 принадлежит языку.
- Индукция: если w принадлежит языку, то и 0w1 принадлежит языку.

Основые определения (5) Язык. Сентенциальные формы

- **Определение:** *языком, порождаемым грамматикой* G = (VT, VN, P, S), называется множество **L**(G)={α ∈ **VT*** | S ⇒ α}.
  - Другими словами, L(G) - это все цепочки в алфавите VT, которые выводимы из S с помощью P.
  - Например, **L**(G1) = {0^n1^n | n>0}.

- **Определение:** цепочка a ∈ (**VT** ∪ **VN**)*, для которой S ⇒ α, 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 G1 и G2 называются *эквивалентными*, если **L**(G1) = **L**(G2).
  - Например, G1 = ({0,1}, {A,S}, **P1**, S) и G2 = ({0,1}, {S}, **P2**, S), где

    P1: $S \to 0A1$ **P2**: $S \to 0S1 \mid 01$

    $0A \to 00A1$

    $A \to \lambda$

    эквивалентны, т.к. обе порождают язык

    L(G1) = **L**(G2) = $\{0^n1^n \mid n>0\}$.

- **Определение:** грамматики G1 и G2 *почти эквивалентны*, если
  L(G1) ∪ {λ} = **L**(G2) ∪ {λ}.

Классификация грамматик и языков по Хомскому

- **ТИП 0:** Грамматика G = (**VT**, **VN**, **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 = (**VT**, **VN**, **P**, S) называется *неукорачивающей грамматикой*, если каждое правило из P имеет вид $\alpha \to \beta$, где $\alpha \in (\mathbf{VT} \cup \mathbf{VN})^+$, $\beta \in (\mathbf{VT} \cup \mathbf{VN})^+$ и $| \alpha | \le | \beta |$.
- Грамматика G = (**VT**, **VN**, **P**, S) называется *контекстно-зависимой (КЗ)*, если каждое правило из P имеет вид $\alpha \to \beta$, 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 = (**VT**, **VN**, **P**, S) называется *контекстно-свободной (КС)*, если каждое правило из **P** имеет вид $A \to \beta$, где $A \in$ **VN**, $\beta \in ($**VT** $\times$ **VN**$)^{+}$.
- Грамматика G = (**VT**, **VN**, **P**, S) называется *укорачивающей контекстно-свободной (УКС)*, если каждое правило из **P** имеет вид $A \to \beta$, где $A \in$ **VN**, $\beta \in ($**VT** $\times$ **VN**$)^{*}$.
- **ТИП 3:**
- Грамматика G = (**VT**, **VN**, **P**, S) называется *праволинейной*, если каждое правило из **P** имеет вид $A \to tB$ либо $A \to t$, где $A \in$ **VN**, $B \in$ **VN**, $t \in$ **VT**.
- Грамматика G = (**VT**, **VN**, **P**, S) называется *леволинейной*, если каждое правило из **P** имеет вид $A \to Bt$ либо $A \to t$, где $A \in$ **VN**, $B \in$ **VN**, $t \in$ **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 $A \to \lambda$, 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: $S \to 0A1$ **P2**: $S \to 0S1 \mid 01$

  $0A \to 00A1$

  $A \to \lambda$

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

$$\begin{aligned}
\mathbf{P}:\ & S \to aaCFD \\
& F \to AFB \mid AB \\
& AB \to bBA \\
& Ab \to bA \\
& AD \to D \\
& Cb \to bC \\
& CB \to C \\
& bCD \to \lambda
\end{aligned}$$

Пример.
Язык типа 1: L = {цепочки из 0 и 1 с одинаковым числом 0 и 1}

$$
\begin{aligned}
\mathbf{P}: & S \to ASB \mid AB \\
& AB \to BA \\
& A \to 0 \\
& B \to 1
\end{aligned}
$$

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

-

$$\mathbf{P}: S \to aQb \mid accb$$
$$Q \to cSc$$

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

$$
\begin{aligned}
\mathbf{P}: S &\to A\perp \mid B\perp \\
A &\to a \mid Ba \\
B &\to b \mid Bb \mid Ab
\end{aligned}
$$

Следующая тема:

«Проблема грамматического разбора. Распознаватели»