Теория формальных языков и грамматик. Определения 1.

Цепочка символов в алфавите V - любая конечная последовательность символов этого алфавита.

Пустая цепочка (ε) - *ц*епочка, которая не содержит ни одного символа.

Если α и β - цепочки, то цепочка $\alpha\beta$ **- *конкатенация* цепочек α и β.**

Например, если **α = ab и β = cd, то $\alpha\beta$ = abcd,**
$\alpha\varepsilon = \varepsilon\alpha = \alpha$.

Обращение (или реверс) цепочки α - цепочка, символы которой записаны в обратном порядке, обозначается как α^R.

Например, если **α = abcdef, то α^R = fedcba, $\varepsilon = \varepsilon^R$.**

n-ая степенью цепочки α (α^n) – конкатенация n цепочек α;

$$\alpha^0 = \varepsilon; \quad \alpha^n = \alpha\,\alpha^{n-1} = \alpha^{n-1}\,\alpha.$$

Длина цепочки **-** количество составляющих ее символов.

Например, если α = abcdefg, то длина α равна 7.
Длину цепочки α обозначается **$|\alpha|$** . **$|\varepsilon| = 0$**

Определения 2.

Язык **в алфавите V - это подмножество цепочек конечной длины в этом алфавите.**

V^* - множество, содержащее все цепочки конечной длины в алфавите V, включая пустую цепочку ε.

Например, если V = { 0, 1 }, то

$V^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, ...\}$.

V^+ - множество, содержащее все цепочки конечной длины в алфавите V, исключая пустую цепочку ε.

$V^* = V^+ \cup \{ \varepsilon \}$.

Порождающая грамматика

Порождающая грамматика G - это четверка

$$\mathbf{G = (T, N, P, S)} \quad , \quad \text{где}$$

T – непустое множество ***терминальных*** *символов*
(алфавит терминалов),

N – непустое множество ***нетерминальных*** *символов*
(алфавит нетерминалов), не пересекающийся с T,

P - конечное подмножество множества $(T \cup N)^+ \times (T \cup N)^*$.

Элемент (α, β) множества P называется ***правилом вывода*** и записывается в виде

$$\alpha \rightarrow \beta,$$

причем α содержит **хотя бы один нетерминальный символ**.

S - ***начальный символ*** ***(цель)*** грамматики, $\mathbf{S} \in \mathbf{N}$.

Декартовым произведением $A \times B$ множеств A и B называется множество $\{ (a,b) \mid a \in A, b \in B\}$.

Соглашения

1) Большие латинские буквы будут обозначать нетерминальные символы.

2) **S** - будет обозначать начальный символ (цель) грамматики.

3) Маленькие греческие буквы будут обозначать цепочки символов.

4) Все остальные символы (маленькие латинские буквы, знаки операций и пр.) будем считать терминальными символами.

5) для записи правил вывода с одинаковыми левыми частями

$\alpha \to \beta 1 \quad \alpha \to \beta 2 \quad ... \quad \alpha \to \beta n$

будем пользоваться сокращенной записью

$\alpha \to \beta 1 \mid \beta 2 \mid ... \mid \beta n.$

Каждое βi , i = 1, 2, ... ,n , будем называть ***альтернативой*** правила вывода из цепочки α.

Пример грамматики:

G1 = ({0,1}, {A,S}, P, S), где P состоит из правил:

$S \to 0A1$

$0A \to 00A1$

$A \to \varepsilon$

Определения 3.

Цепочка $\beta \in (T \cup N)^*$ ***непосредственно выводима*** из цепочки $\alpha \in (T \cup N)^+$ в грамматике $G = (T, N, P, S)$, обозначается: $\alpha \rightarrow \beta$,
если $\alpha = \xi_1 \gamma \xi_2$, $\beta = \xi_1 \delta \xi_2$, где $\xi_1, \xi_2, \delta \in (T \cup N)^*$, $\gamma \in (T \cup N)^+$
и правило вывода $\gamma \rightarrow \delta$ содержится в P.

Цепочка $\beta \in (T \cup N)^*$ ***выводима*** из цепочки $\alpha \in (T \cup N)^+$ в грамматике $G = (T, N, P, S)$, обозначается $\alpha \Rightarrow \beta$,
если существуют цепочки $\gamma 0, \gamma 1, \ldots, \gamma n$ $(n >= 0)$, такие, что

$$\alpha = \gamma 0 \rightarrow \gamma 1 \rightarrow \ldots \rightarrow \gamma n = \beta.$$

Последовательность $\gamma 0, \gamma 1, \ldots, \gamma n$ называется ***выводом длины n*.**

Язык, порождаемый грамматикой **$G = (T, N, P, S)$ - $L(G) = \{\alpha \in T^* \mid S \Rightarrow \alpha\}$.**

Сентенциальная форма в грамматике $G = (T, N, P, S)$ - цепочка $\alpha \in (T \cup N)^*$, для которой $S \Rightarrow \alpha$.

Язык, порождаемый грамматикой - множество терминальных сентенциальных форм.

Определения 4.

Грамматики G1 и G2 называются ***эквивалентными***,
если $L(G1) = L(G2)$.

Например, $G1 = (\{0,1\}, \{A,S\}, P1, S)$ и $G2 = (\{0,1\}, \{S\}, P2, S)$

P1: $S \to 0A1$
$0A \to 00A1$
$A \to \varepsilon$

P2: $S \to 0S1 \mid 01$

эквивалентны, т.к. обе порождают язык $L(G1) = L(G2) = \{ 0^n 1^n \mid n > 0\}$.

Грамматики G1 и G2 ***почти эквивалентны***,
если $L(G1) \cup \{\varepsilon\} = L(G2) \cup \{\varepsilon\}$.

Например, $G1 = (\{0,1\}, \{A,S\}, P1, S)$ и $G2 = (\{0,1\}, \{S\}, P2, S)$

P1: $S \to 0A1$
$0A \to 00A1$
$A \to \varepsilon$

P2: $S \to 0S1 \mid \varepsilon$

почти эквивалентны, так как

$L(G1) = \{ 0^n 1^n \mid n > 0 \}$, а $L(G2) = \{ 0^n 1^n \mid n >= 0 \}$.

Классификация грамматик и языков по Хомскому

ТИП 0:

Грамматика G = (T, N, P, S) - *грамматика типа 0*, если на ее правила вывода не накладывается никаких ограничений.

ТИП 1:

Грамматика G = (T, N, P, S) - ***неукорачивающая*** *грамматикой*, если каждое правило из P имеет вид

$\alpha \rightarrow \beta$, где $\alpha \in (T \cup N)^+$, $\beta \in (T \cup N)^+$ и $|\alpha| <= |\beta|$.

Исключение - в неукорачивающей грамматике допускается ***наличие правила $S \rightarrow \varepsilon$, при условии, что S (начальный символ) не встречается в правых частях правил.***

Грамматика G = (T, N, P, S) - ***контекстно-зависимая*** *(КЗ)*, если каждое правило из P имеет вид

$\alpha \rightarrow \beta$, где $\alpha = \xi1\ A\ \xi2$; $\beta = \xi1\ \gamma\ \xi2$; $A \in N$; $\gamma \in (T \cup N)+$; $\xi1, \xi2 \in (T \cup N)^*$.

В КЗ-грамматике допускается ***Исключение.***

Грамматику типа 1 можно определить как неукорачивающую либо как контекстно-зависимую.

Классификация грамматик и языков по Хомскому

ТИП 2:

Грамматика G = (T, N, P, S) - ***контекстно-свободная*** *(КС)*, если каждое правило из P имеет вид $A \to \beta$, где $A \in N$, $\beta \in (T \cup N)^*$.

Грамматика G = (T, N, P, S) - ***неукорачивающая контекстно-свободная*** *(НКС)*, если каждое правило из P имеет вид $A \to \beta$, где $A \in N$, $\beta \in (T \cup N)^+$.

В неукорачивающей КС-грамматике допускается ***Исключение.***

Грамматику типа 2 можно определить как контекстно-свободную либо как неукорачивающую контекстно-свободную.

ТИП 3:

Грамматика G = (T, N, P, S) - ***праволинейная***, если каждое правило из P имеет вид имеет вид: $A \to wB$ либо $A \to w$, где $A, B \in N$, $w \in T^*$.

Грамматика G = (T, N, P, S) - ***леволинейная*****,** если каждое правило из P имеет вид: $A \to Bw$ либо $A \to w$, где $A, B \in N$, $w \in T^*$.

*Грамматику типа 3 (**регулярную**, P-грамматику)* можно определить как праволинейную либо как леволинейную.

Автоматная грамматика - праволинейная (леволинейная) грамматика, такая, что каждое правило с непустой правой частью имеет вид: $A \to a$ либо $A \to aB$ (для леволинейной, $A \to a$ либо $A \to Ba$), где $A, B \in N$, $a \in T$.

Соотношения между типами грамматик

неук. Р $\subset$ неук. КС $\subset$ КЗ $\subset$ Тип 0

(1) Любая регулярная грамматика является КС-грамматикой.

(2) Любая неукорачивающая КС-грамматика является КЗ-грамматикой. $\rightarrow$,

(3) Любая неукорачивающая грамматика является грамматикой типа 0.

Язык L(G) является ***языком типа k*** по Хомскому, если его можно описать грамматикой типа k, где k - максимально возможный номер типа грамматики по Хомскому.

Соотношения между типами языков

Р $\subset$ КС $\subset$ КЗ $\subset$ Тип 0

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

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

(3) Каждый КЗ-язык является языком типа 0, но существуют языки типа 0, которые не являются КЗ-языками
(например: язык, состоящий из записей самоприменимых алгоритмов Маркова в некотором алфавите).

(4) Кроме того, существуют языки, которые вообще нельзя описать с помощью порождающих грамматик. Такие языки не являются рекурсивно перечислимым множеством.

Проблема, можно ли язык, описанный грамматикой типа **k**, описать грамматикой типа **k + 1** (k = 0, 1, 2), является **алгоритмически неразрешимой**.

Диаграмма Венна для классов языков