# ЛЕКЦИЯ 1
# "ОБОБЩЕННАЯ СХЕМА ТРАНСЛЯЦИИ ПРОГРАММ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ. ЭЛЕМЕНТЫ ТЕОРИИ ФОРМАЛЬНЫХ ГРАММАТИК"

## ПЛАН

1. Основные определения
2. Процесс трансляции и его логическая организация.
3. Виды информационных таблиц, используемых транслятором.
4. Схемы трансляции модулей: Схема метода срезов, прямой метод трансляции.
5. Понятие цепочки.
6. Определение формальной грамматики и способы ее задания.
7. Порождение цепочек. Сентенциальные формы, фразы, рекурсия.
8. Понятие языка. Классификация языков и формальных грамматик по Хомскому.

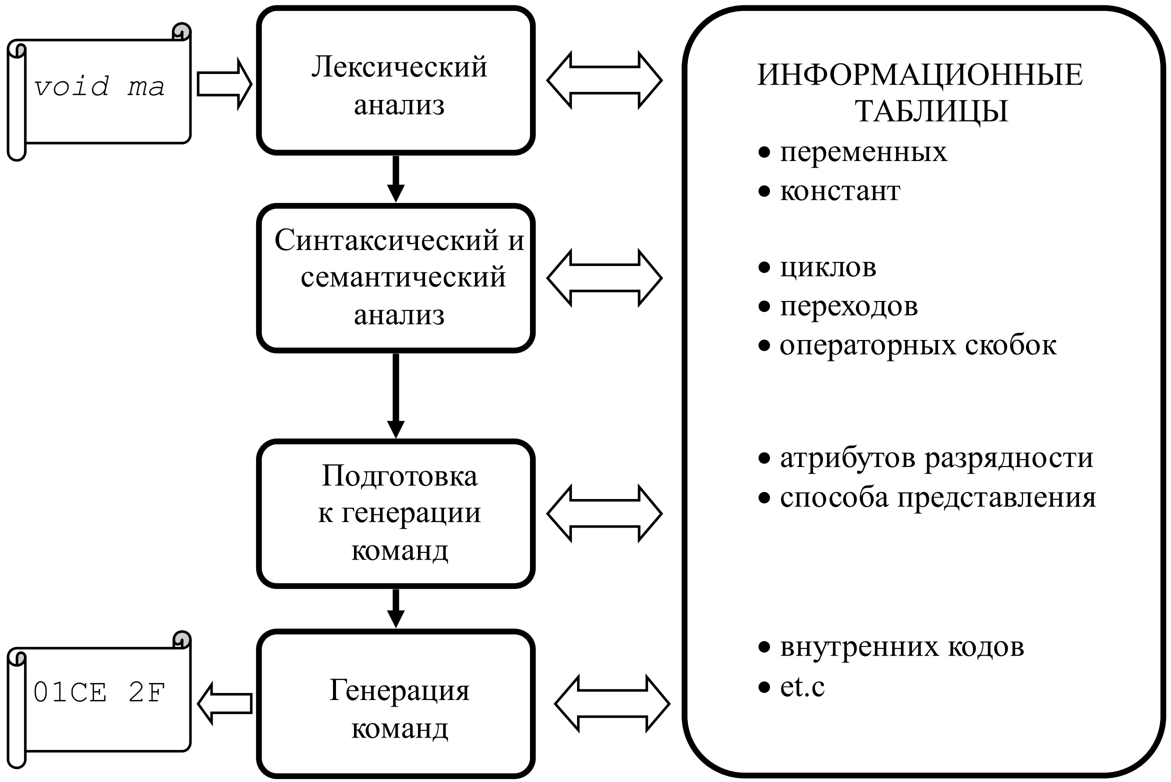



Рисунок 1 – Составные части компилятора [1]

1. Грис Д. Конструирование компиляторов для ЦВМ. – М.: Мир, 1975. – 544 с.

## Определения и понятия трансляции

**Составной блок (СБ)** – совокупность, состоящая из описательной части, размещаемой вначале, и операторной, следующей за описательной.

Предполагается, что ***составной блок*** в своих частях может содержать составные блоки соподчинённых уровней, а так же ***минимальные составные блоки*** и блоки.

**Минимальный составной блок (МСБ)** – составной блок, содержащий в операторной части ***только*** блоки.

**Блок (Б)** – языковая конструкция, не содержащая описательной части.

**Конструкция –** может быть составным блоком, минимальным составным блоком, блоком или любым их допустимым сочетанием.

$$S(n, m) = D(n, m) \bigstar P(n, m),$$

где $n$ – номер уровня конструкции; $m$ – номер конструкции в уровне; $D$ – описательная часть конструкции; $\bigstar$ – знак композиции; $P$ – операторная часть.

**Утверждение 1**. Операторная часть блока $S(n+1, m+i)$ является одновременно и операторной частью минимального составного блока $S(n, m)$.

**Утверждение 2**. Любой блок $S(n, m)$ приводится к ***квазиблоку*** путём отбрасывания описательной части.

**Лемма**. Минимальный составной блок можно свести к блоку.

**Теорема**. Составной блок любого уровня можно свести к блоку.

**Срез** – участок программы, ограниченный слева блочным началом, а справа – первым встреченным блочным концом.

```
BEGIN                                 {1.1}
…
    BEGIN                             {2.1}
    …
        BEGIN                         {3.1}
        …
        END                           {3.1}
    …
        BEGIN                         {3.2}
        …
        END                           {3.2}
    …
    END                               {2.1}
    …
    BEGIN                             {2.2}
    …
        BEGIN                         {3.1}
        …
            BEGIN                     {4.1}
            …
            END                       {4.1}
        …
        END                           {3.1}
    …
        BEGIN                         {3.2}
        …
        END                           {3.2}
    …
    END                               {2.2}
…
    BEGIN                             {2.3}
    …
    END                               {2.3}
…
END                                   {1.1}
```

1.1 – 2.1 – 3.1

1.1 – 2.1 – 3.2

1.1 – 2.2 – 3.1 – 4.1

1.1 – 2.2 – 3.2

1.1 – 2.3

Возникают следующие пять срезов:

1) 1.1 – 2.1 – 3.1;
2) 1.1 – 2.1 – 3.2;
3) 1.1 – 2.2 – 3.1 – 4.1;
4) 1.1 – 2.2 – 3.2;
5) 1.1 – 2.3.

| № | Срез | Результат трансляции |
|---|---|---|
| 1 | 1.1 – 2.1 – 3.1 | (3.1) |
| 2 | 1.1 – 2.1 – 3.2 | (3.1), (3.2) |
| 3 | | (3.1), (3.2), (2.1) |
| 4 | 1.1 – 2.2 – 3.1 – 4.1 | (3.1), (3.2), (2.1), (4.1) |
| 5 | | (3.1), (3.2), (2.1), (4.1), $(3.1)^*$ |
| 6 | 1.1 – 2.2 – 3.2 | (3.1), (3.2), (2.1), (4.1), $(3.1)^*$, $(3.2)^*$ |
| 7 | | (3.1), (3.2), (2.1), (4.1), $(3.1)^*$, $(3.2)^*$, (2.2) |
| 8 | 1.1 – 2.3 | (3.1), (3.2), (2.1), (4.1), $(3.1)^*$, $(3.2)^*$, (2.2), (2.3) |
| 9 | | (3.1), (3.2), (2.1), (4.1), $(3.1)^*$, $(3.2)^*$, (2.2), (2.3), (1.1) |

Звёздочками обозначены блоки с одинаковой нумерацией, но относящиеся к разным срезам.

“Глокая куздра штеко будланула бокра и курдячит бокрёнка…”
Л.В. Успенский, “Слово о словах”

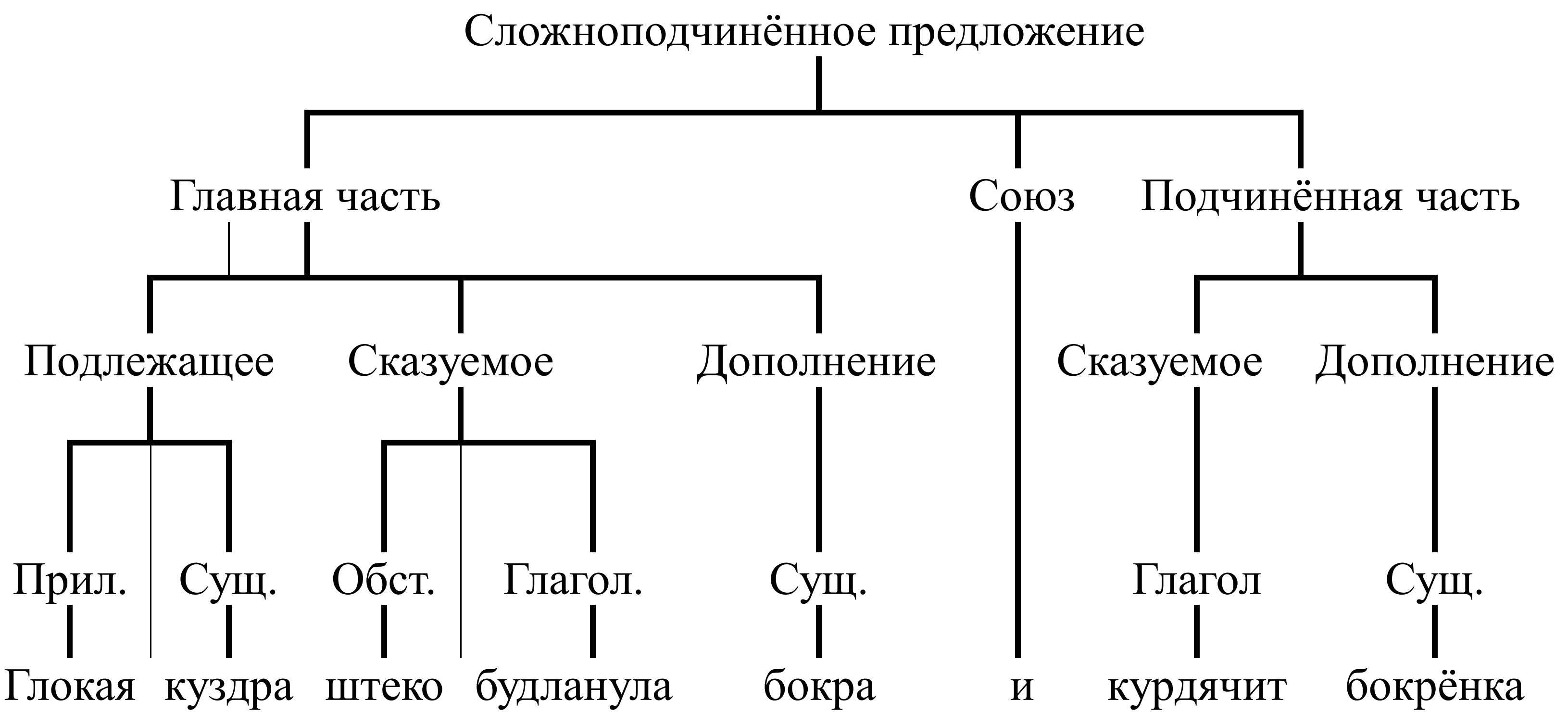



Рисунок 2– Разбор предложения “по частям”

*Операция конкатенации* определяется следующим образом: если $x$ и $y$ цепочки, то их возможные конкатенации суть $xy$ и $yx$.

*Степени* цепочек и *произведения* и *степени* множеств:

$$A \cdot B = \{xy \mid x \in A, y \in B\},$$

$$w^0 = \varepsilon;\ w^n = w^{n-1} \cdot w,$$

$$W^0 = \varepsilon;\ W^n = W^{n-1} \cdot W.$$

У цепочек различаются ***головы*** и ***хвосты***, этот термин употребляется, когда не важна остальная часть цепочки, при этом $\Lambda x = x\Lambda = x$.

Пусть $\boldsymbol{\omega = xy}$, тогда $\boldsymbol{x}$ – голова и правильная голова, если $\boldsymbol{y \neq \varepsilon}$; $\boldsymbol{y}$ – хвост и правильный хвост, если $\boldsymbol{x \neq \varepsilon}$.

*Пример*. Пусть $\omega$ = *ПИВО*. В этом случае {$\boldsymbol{\varepsilon}$, *П, ПИ, ПИВ, ПИВО*} – суть головы, при этом {$\boldsymbol{\varepsilon}$, *П, ПИ, ПИВ*} – являются правильными головами.

На базе операции возведения множества в степень определяют полную (*) и усечённую (+) итерации множеств по следующему алгоритму:

$$\boldsymbol{A^+ = A \cup A^2 \cup A^3 \cup \ldots \cup A^n \cup A^{n+1}.}$$
$$\boldsymbol{A^* = A^0 \cup A^+}.$$

В современной литературе применяются следующая терминология.

$A^*$ – называется итерацией Клини или просто итерацией.

$A^+$ – называется позитивной итерацией.

***Продукция*** или ***правило подстановки*** есть упорядоченная пара **(*U, x*)**, записываемая в форме $\boldsymbol{U} \mathbin{::=} x$ или $\boldsymbol{U} \rightarrow x$, в которой $\boldsymbol{U}$ – символ, $x$ – непустая цепочка символов.

Знаки **{::= , →}** трактуются “определяется как” или “заменяется на”.

Словарь грамматики есть объединение словарей терминальных и нетерминальных символов

$$V = V_N \cup V_T.$$

***Формальная грамматика G***[***S***] (или просто грамматика) есть конечное непустое множество, задаваемое упорядоченной четвёркой:

$$(V_T, V_N, S, R),$$

в которой

- $\boldsymbol{V_T}$ – словарь терминальных символов;
- $\boldsymbol{V_N}$ – словарь нетерминальных символов;
- $\boldsymbol{S}$ – нетерминальный символ, который должен появиться в левой части хотя бы одного правила (он называется аксиомой или помеченным символом);
- $\boldsymbol{R}$ – счётное множество правил грамматики.

Замечание: Строго говоря, продукция определяется как конечное множество отображений вида

$$(V_N \cup V_T)^* \times V_N \rightarrow (V_N \cup V_T)^* \times (V_N \cup V_T)^*$$

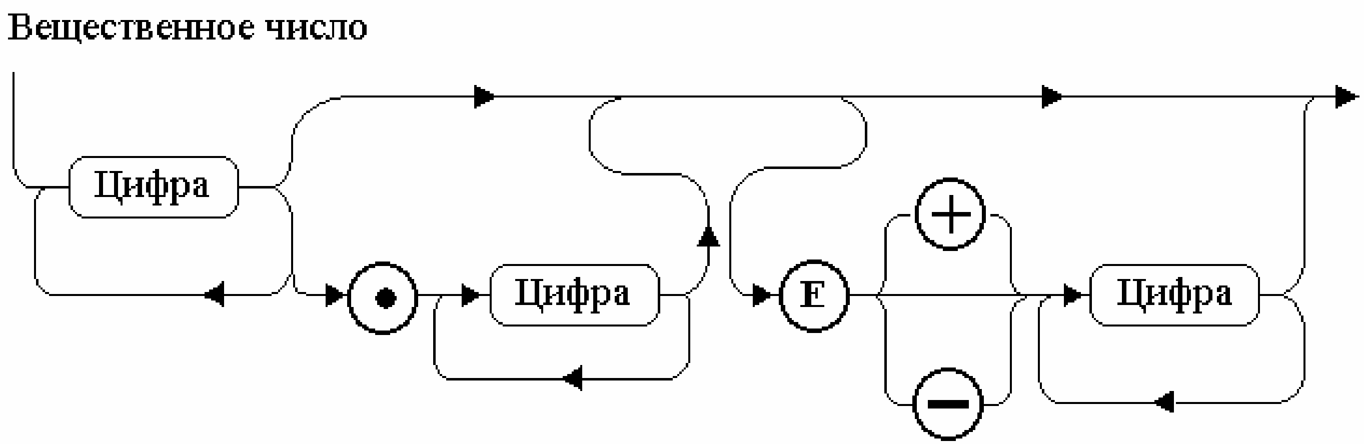
Способы задания формальной грамматики

1. Овал – диаграммы.

2. Форма Бэкуса-Наура (≡ нормализованная форма Бэкуса ≡ НФБ ≡ НБФ). (J.W. Backus, P. Naur).

Нетерминалы записываются прописными буквами, либо в угловых скобках **<…>**, а терминальным символам соответствуют маленькие буквы при написании. Используется символ | - разделитель правых частей правил с одинаковой левой частью, символы $\{\ldots\}_{минимум}^{максимум}$ - повторитель конструкции, заключенной в фигурные скобки, нижний индекс – минимально возможное число повторений, как правило – это единица или ноль. Верхний индекс – максимальное число повторений. Обычно это бесконечность.

3. Древовидные и табличные структуры



Грамматика G[Bin]

| | Развернутая запись | Компактная запись |
|---|---|---|
| (1) | **<Bin> ::= <Bn>** | |
| (2) | **<Bn> ::= <dig><Bn>** | **<Bn> ::= <dig><Bn>\|<dig>** |
| (3) | **<Bn> ::= <dig>** | |
| (4) | **<dig> ::=** 0 | **<dig> ::=** 0\|1 |
| (5) | **<dig> ::=** 1 | |

Грамматика G[<ПРЕДЛОЖЕНИЕ>]

Правила: *R* = {

<ПРЕДЛОЖЕНИЕ>→<ПОДЛЕЖАЩЕЕ><СКАЗУЕМОЕ><ДОПОЛНЕНИЕ>●

<ПОДЛЕЖАЩЕЕ> → <ПРИЛАГАТЕЛЬНОЕ> <СУЩЕСТВИТЕЛЬНОЕ>

<СКАЗУЕМОЕ> → *ЗАСЛОНЯЕТ*

<ДОПОЛНЕНИЕ> → <ПРИЛАГАТЕЛЬНОЕ> <СУЩЕСТВИТЕЛЬНОЕ>

<ПРИЛАГАТЕЛЬНОЕ> → *СТАРЫЙ*

<СУЩЕСТВИТЕЛЬНОЕ> → *ДОМ*| *ДУБ*

*}*

$V_T$ = {*СТАРЫЙ, ДОМ, ДУБ, ЗАСЛОНЯЕТ,* ●};

$V_N$ = {ПРЕДЛОЖЕНИЕ, ПОДЛЕЖАЩЕЕ, СКАЗУЕМОЕ, ДОПОЛНЕНИЕ, ПРИЛАГАТЕЛЬНОЕ, СУЩЕСТВИТЕЛЬНОЕ};

*S* = ПРЕДЛОЖЕНИЕ

## Порождение цепочек

Пусть $G[S]$ –формальная грамматика.

1. Цепочка $v$ ***непосредственно порождает*** цепочку $w$, (обозначается $v \Rightarrow w$), если цепочки представимы в виде $v = xUy$, $w=xuy$, и существует правило $\boldsymbol{U} \to u$.

2. Цепочка $v$ ***порождает*** цепочку $w$ (записывается $v \Rightarrow^{+} w$, или $v \Rightarrow^{*} w$), если существует последовательность ***непосредственных*** выводов

$$v = u_0 \Rightarrow u_1 \Rightarrow \ldots \Rightarrow u_n = w,\ \ n > 1.$$

Последовательность ***непосредственных*** выводов называется ***выводом длины n***, а цепочка $w$ является ***словом*** для $v$.

Пример порождения цепочек для грамматики G[**Bin**].

***<Bin>*** $\Rightarrow^{(1)}$ ***<Bn>***,
***<Bn>*** $\Rightarrow^{(2)}$ ***<dig><Bn>,***
***<dig><Bn>*** $\Rightarrow^{(2)}$ ***<dig><dig><Bn>,***
***<dig><dig><Bn>*** $\Rightarrow^{(3)}$ ***<dig><dig><dig>,***
***<dig><dig><dig>*** $\Rightarrow^{(4)}$ ***<dig><dig>0,***
***<dig><dig>0*** $\Rightarrow^{(5)}$ $\Rightarrow^{(5)}$ ***<dig>***10***,***
***<dig>***10 $\Rightarrow^{(5)}$110.

***<dig><Bn>*** $\Rightarrow^{+}$ ***<dig>***10

Цепочка $x$ называется ***сентенциальной формой*** грамматики G[Z], если она выводима из аксиомы $Z$ (обозначается $Z \Rightarrow^{*} x$).

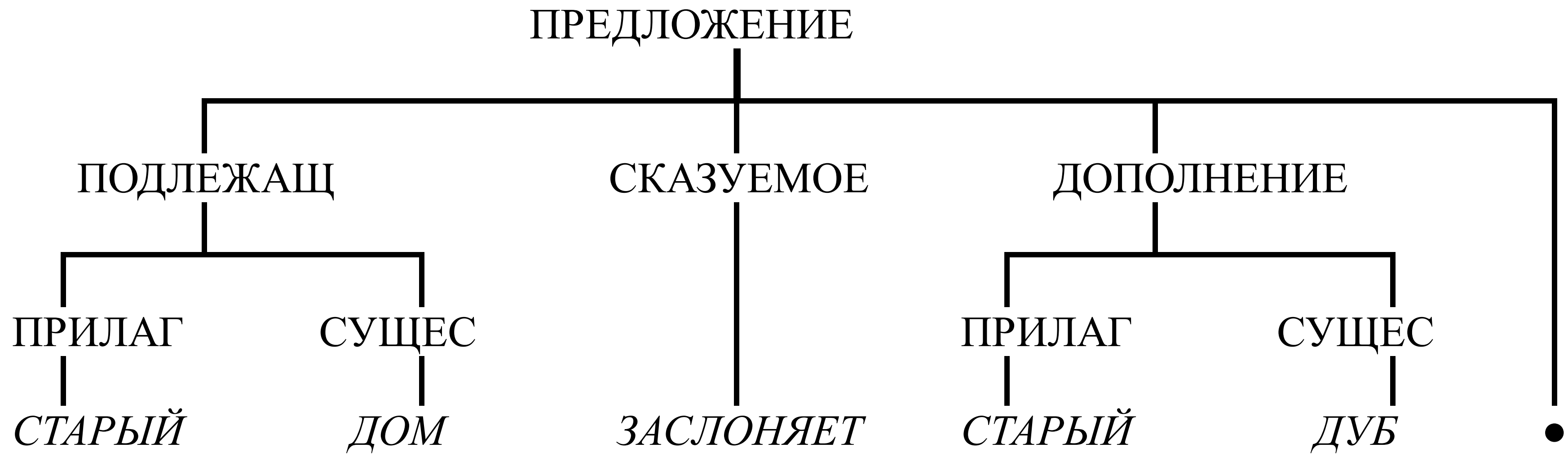
Пример: ***<Bin>*** $\Rightarrow^{*}$ 110*,* ***<Bin>*** $\Rightarrow^{*}$ ***<dig><dig>***0.

***Предложением*** называется сентенциальная форма, состоящая ***только*** из терминальных символов.

Пример: ***<Bin>*** $\Rightarrow^{*}$ 1, ***<Bin>*** $\Rightarrow^{*}$ 111, ***<Bin>*** $\Rightarrow^{*}$ 110110 и т.д.

Язык *L{G[Z]}* – это множество предложений грамматики ***G[Z]*** или подмножество множества всех терминальных цепочек, таких что $L\{G[Z]\} = \{x|\ Z \Rightarrow^{*} x,\ x \in V_T\}$.

Пример: Для грамматики “Russian Basic” язык представлен множеством из четырёх предложений *L{G[*ПРЕДЛОЖЕНИЕ]} = { *СТАРЫЙ ДОМ ЗАСЛОНЯЕТ СТАРЫЙ ДУБ*●; *СТАРЫЙ ДОМ ЗАСЛОНЯЕТ СТАРЫЙ ДОМ*●; *СТАРЫЙ ДУБ ЗАСЛОНЯЕТ СТАРЫЙ ДОМ*●; *СТАРЫЙ ДУБ ЗАСЛОНЯЕТ СТАРЫЙ ДУБ*●}



Пусть $G[Z]$ – грамматика, а $w=xuy$ – сентенциальная форма этой грамматики. Цепочка $u$ называется ***фразой*** для нетерминала $U$, если

$$Z \Rightarrow^* xUy,$$
$$U \Rightarrow^+ u$$

и ***простой фразой***, если $U \Rightarrow u$.

## РЕКУРСИЯ В ПРОДУКЦИЯХ ГРАММАТИКИ

1. Если $U \Rightarrow^+ xUy$ – грамматика рекурсивна (центрально рекурсивна) по отношению к $U$, в частности, правило вида $A \to \alpha A\beta$ – рекурсия и называется ***прямым самовставлением***.

2. Для ситуации $U \Rightarrow^+ Uxy$ – имеет место ***левая рекурсия***, а когда $A \to A\alpha$ - ***прямая левая рекурсия***.

3. Когда $U \Rightarrow^+ xyU$ – возникает ***правая рекурсия***, и в случае $A \to \alpha A$ - ***прямая правая рекурсия***.

## КЛАССЫ (ТИПЫ) ЯЗЫКОВ И ГРАММАТИК

Классификация предложена Хомским (Chomsky N).

Продукции грамматики есть конечное множество отображений (алфавитный оператор)

$$(V_N \cup V_T)^* \times V_N \to (V_N \cup V_T)^* \times (V_N \cup V_T)^*.$$

**Класс 0.** Грамматика фазовой структуры – ГФС. Правила грамматики такого класса имеют вид

$$u ::= v, \text{ где } u \in V^+, v \in V^*.$$

При этом, ***левая часть не может состоять только из терминальных символов***, так как при этом бы была утрачена фазовость. Последний термин, применительно к выводу, означает, что он возможен только при наличии переменных символов.

**Класс 1.** Получается из ГФС наложением на левую часть продукций ограничений вида

$$xUy ::= xuy, xUz ::= xvz, \text{ где } u, v \in V^+, x,y \in V^*, U \in V_N.$$

Эти грамматики называют ***контекстно-чувствительными*** ≡ ***контекстно-зависимыми*** ≡ ***неукорачивающей грамматикой***≡***непосредственных составляющих*** (НС – грамматики).

**Класс 2.** Образуется на базе первого путём ограничения левых частей продукций до одиночного нетерминала.

$$U ::= u, \text{ где } u \in V^*, U \in V_N.$$

Такие грамматики именуются ***контекстно-свободными*** (КС – грамматики) или ***бесконтекстными*** грамматиками.

**Класс 3.** Автоматные (регулярные) грамматики

$$U ::= u, U ::= Tu, \; U ::= uT, \text{ где } u \in V_T, U, T \in V_N.$$