# ЛЕКЦИЯ 2
# “ЭКВИВАЛЕНТНЫЕ ПРЕОБРАЗОВАНИЯ ГРАММАТИК. СИНТАКСИЧЕСКИЕ ДЕРЕВЬЯ И ИХ СВОЙСТВА.”

## ПЛАН

1. Эквивалентные преобразования грамматик.
2. Получение ε-свободной грамматики.
3. Преобразование грамматик в нормальную форму Хомского.
4. Преобразование грамматик в нормальную форму Грейбах.
5. Синтаксические деревья и их свойства.
6. Неоднозначность грамматик: установление неоднозначности и коррекция её при помощи синтаксических деревьев.
7. Двоичные деревья и правила их построения.
8. Скелеты деревьев трансляции.

## ЭКВИВАЛЕНТНЫЕ ГРАММАТИКИ

Пример 1.

| | |
|---|---|
| $G_1[S]$ | $G_2[S]$ |
| $\mathbf{S} \to \mathbf{A} \mid \mathbf{B}$ | $\boldsymbol{S} \to a\ \mathbf{A} \mid b\ \mathbf{B} \mid b \mid a$ |
| $\mathbf{A} \to a\ \mathbf{A} \mid a$ | $\mathbf{A} \to a\ \mathbf{A} \mid a$ |
| $\mathbf{B} \to b\ \mathbf{B} \mid b$ | $\mathbf{B} \to b\ \mathbf{B} \mid b$ |

Язык: $L = \{a^n, b^n; n \geq 1\}$.

Пример 2.

| | |
|---|---|
| $G_1[S]$ | $G_2[S]$ |
| $\boldsymbol{S} \to a\ \mathbf{A} \mid b\ \mathbf{B}$ | $\boldsymbol{S} \to \mathbf{A}\ b$ |
| $\mathbf{A} \to a$ | $\mathbf{A} \to a$ |
| $\mathbf{B} \to b$ | |

Язык $L = \{ ab \}$.

## ПОЛУЧЕНИЕ $\varepsilon$-СВОБОДНОЙ ГРАММАТИКИ

Правило вида **B** $\to \varepsilon$ называется ***ε-продукцией***. Грамматика, ***не имеющая*** ε-продукций, называется ***ε-свободной*** (e-free).

**Теорема**. Для любого контекстно-свободного языка ***L*** существует $\varepsilon$-свободная КС-грамматика ***G[S]***, такая, что ***L{G[S]}*** = ***L*** \ ***{ε}***.

Цель преобразования формально записывается так

$$S \underset{G}{\Rightarrow}^{*} x \quad \Leftrightarrow \quad S \underset{G'}{\Rightarrow}^{*} x, \quad \text{для} \quad \forall x \in V_T^{+}, \tag{1}$$

$$x \in L\{G[S]\} \setminus \{\varepsilon\} \Leftrightarrow x \in L\{G`[S]\}. \tag{2}$$

## АЛГОРИТМ ПРЕОБРАЗОВАНИЯ

Пусть $G[S] = (V_T, V_N, S, R)$ — КС-грамматика, имеющая $\varepsilon$-продукции.

1. Объединим ***все*** $\varepsilon$-продукции (непосредственного вывода) из исходного множества правил $R$ в множество $R_\varepsilon$.

2. Отыщем все нетерминальные символы $N_i \in V_N$, такие что $N_i \Rightarrow^{+} \varepsilon$ или $N_i \Rightarrow^{*} \varepsilon$. Указанные символы будем называть ***ε-порождающими***.

3. Каждой продукции $p \in R$, в ***правой*** части которой находятся ε-порождающие символы, поставим в соответствие продукцию, в правой части которой, по сравнению с исходной, опущены один или более ε-порождающих символов. Получим множество $R_0$.

4. Конструируем новое множество продукций грамматики по правилу

$$R` = \{R \setminus R_\varepsilon\} \cup \{R_0\}.$$

Пример.

$S \to [E] | \mathrm{E},$

$E \to D | D+E | D–E,$

$D \to F | D*F | D/F,$

$F \to a | b | c | \varepsilon.$

1. Множество $\mathrm{R}_\varepsilon = \{F \to \varepsilon\}$.
2. Поиск $\varepsilon$-порождающих символов $\mathrm{F} \Rightarrow \varepsilon$, $\mathrm{S} \Rightarrow^* \varepsilon$, $\mathrm{E} \Rightarrow^+ \varepsilon$, $\mathrm{D} \Rightarrow^+ \varepsilon$.
3 Построение множества $R_0$.

$\mathrm{S} \to [\ ],$

$\mathrm{E} \to \mathrm{D}+ | +\mathrm{E} | \mathrm{D}– | –\mathrm{E} | + | –,$

$\mathrm{D} \to *\mathrm{F} | \mathrm{D}* | \mathrm{D}/ | /\mathrm{F} | / | *.$

4. Окончательный набор продукций для $G`[S]$.

$\mathrm{S} \to [\mathrm{E}] | \mathrm{E} | [],$

$\mathrm{E} \to \mathrm{D} | \mathrm{D}+\mathrm{E} | \mathrm{D}–\mathrm{E} | \mathrm{D}+ | +\mathrm{E} | \mathrm{D}– | –\mathrm{E} | + | –,$

$\mathrm{D} \to \mathrm{F} | \mathrm{D}*\mathrm{F} | \mathrm{D}/\mathrm{F} | *\mathrm{F} | \mathrm{D}* | \mathrm{D}/ | /\mathrm{F} | / | *,$

$\mathrm{F} \to a | b | c.$

# ПРЕОБРАЗОВАНИЕ ГРАММАТИКИ В НОРМАЛЬНУЮ ФОРМУ ХОМСКОГО (CHOMSKY NORMAL FORM)

Продукция КС-грамматики $G[S] = (V_T, V_N, S, R)$ называется ***релевантной*** тогда и только тогда, когда существует вывод некоторой цепочки $x \in L\{G[S]\}$, в которой эта продукция используется.

То есть, если $A ::= \alpha,\ \alpha \in V^*,\ A \in V_N$ – релевантная продукция, то

$$\forall\, x \in \{L(G[S]) \mid S \Rightarrow^* \beta A \gamma \Rightarrow \beta \alpha \gamma \Rightarrow^+ x\}.$$

**Теорема.** Любой $\varepsilon$-свободный КС-язык может быть порождён некоторой КС-грамматикой в нормальной форме Хомского, продукции которой имеют вид**:** $A \rightarrow BC$ или $A \rightarrow \alpha$, где $A$, $B$, $C \in V_N$, $\alpha \in V_T$.

## АЛГОРИТМ ПРЕОБРАЗОВАНИЯ

1. Среди продукций грамматики отыщем все ***первичные продукции*** вида

$$A_i \rightarrow B_j,\ A_i,\ B_j \in V_N.$$

правая часть которых содержит единственный нетерминальный символ.

При этом всё множество продукций с нетерминалом $A_i$ в левой части разбивается на подмножества: $U\{A_i\}$ – первичных продукций и $N\{A_i\}$ – непервичных продукций.

Для $\forall A_i$, где $U\{A_i\} \neq \{\}$, поставим в соответствие множество правил

$$\{A_i \Rightarrow^+ \alpha \mid A_i \rightarrow B_j,\ B_j \Rightarrow^+ \alpha,\ \alpha \in V^+,\ B_j \in N(B_j)\}.$$

В результате не останется первичных продукций.

2. Отыщем все ***вторичные*** продукции. Правые части таких продукций представляют собой ***смешанные цепочки*** терминальных и нетерминальных символов.

Каждый терминальный символ такой цепочки $\eta$ заменяется нетерминалом $\mathbf{A}_\eta$. Одновременно множество продукций пополняется правилом $A_\eta \to \eta$, определяющим данный нетерминал.

Для примера, пусть правая часть вторичной продукции $x_1x_2...x_m,\ x_i \in (V_T \cup V_N), i = 1,\overline{m}$. Тогда после преобразования она примет вид $y_1y_2...y_m,\ y_i \in V_N,\ i = 1,\overline{m}$ при соответствии между цепочками

$$y_i = \begin{cases} x_i, & x_i \in V_N, \\ X_{x_i}, & X_{x_i} \in V_N, x_i \in V_T, \\ R^* = R \cup \left\{ X_{x_i} ::= x_i \right\} & \end{cases}$$

Будут устранены все вторичные продукции.

3. В модифицированном, в ходе п.п. 2 и 3 настоящего алгоритма, множестве продукций, отыщем ***третичные*** продукции, в правых частях которых содержится ***не менее трёх*** нетерминалов подряд.

Для каждой найденной третичной продукции строится разложение. Если исходная третичная продукция имеет вид:

$$A_r \to S_1S_2...S_m,\ S_i \in V_N,\ i = 1,\overline{m},\ m \geq 2\ .$$

Её разложение будет выглядеть так:

$$A_r \to \ S_1 N_{B1},$$

$$N_{B1} \to S_2 N_{B2},$$

$$\ldots$$

$$N_{Bm-1} \to S_{m-1} S_m.$$

Нетерминалы $N_{Bi}$**,** $i$=1,$m$–1 суть новые, нетерминалы, ***не содержащиеся более ни в одной продукции***, добавленные к словарю $V_N$ в ходе разложения.

Пример.

S→ABA|A,

A→*a*|*a*A|B,

B→*b*|*b*B.

1. Первичные продукции суть

S→A,

A→B.

Имеем следующие множества первичных и непервичных продукций.

U(A)={A→B}, N(A)={A→*a*A|*a*},

U(B)={}, N(B)={B→*b*B|*b*},

U(S)={S→A}. N(S)={S→ABA}.

Правилу A→B поставим в соответствие заменяющие правила A→*b*B|*b*, а продукции S→A – правила S→*a*A|*b*B|*a*|*b*.

Получим новый набор правил изменённой грамматики.

$S \to ABA|aA|bB|a|b,$

$A \to aA|bB|a|b,$

$B \to b|bB.$

2 Ко вторичным продукциям относятся все продукции с правыми частями $a$A и $b$B. Введением правил: C→$a$, D→$b$ эти правые части преобразуются к виду **CA** и **DB.** В результате:

$S \to ABA|CA|DB|a|b,$

$A \to CA|DB|a|b,$

$B \to DB|b,$

$C \to a,$

$D \to b.$

3. Единственная третичная продукция: S→ABA. Заменим её продукциями S→AE, E→BA**.** Получим грамматику в нормальной форме Хомского:

$S \to AE|CA|DB|a|b,$

$A \to CA|DB|a|b,$

$B \to DB|b,$

$E \to BA,$

$C \to a,$

$D \to b.$

ИССЛЕДОВАНИЕ НА РЕЛЕВАНТННОСТЬ

***Свойство А***. Если все нетерминальные символы правой части правила продуктивны, то продуктивен и символ, стоящий в левой части правила.

**Алгоритм поиска непродуктивных нетерминалов**

1. Составить список нетерминалов, для которых найдётся хотя бы одна продукция, правая часть которого состоит только из терминалов.

2. Если найдено такое правило, что все нетерминальные символы, стоящие в его правой части, уже помещены в список, то добавить в список нетерминал, стоящий в левой части данного правила.

3. Если список не пополняется, то эти нетерминальные символы продуктивны, остальные элементы $V_N$ – не продуктивны.

***Свойство Б***. Если все нетерминальный символ в левой части правила является достижимым, то достижимы и все символы, стоящие в правой части этого правила.

**Алгоритм поиска недостижимых нетерминалов**

1. Составить множество, первоначально, содержащее только аксиому грамматики.

2. Если найдено правило, нетерминал из левой части которого уже принадлежит множеству, то поместить во множество нетерминалы из правой части правила, которые в нём отсутствуют.

3. Если множество не пополняется, то нетерминалы, его составляющие, являются достижимыми, а прочие нетерминалы из $V_N$ – нет.

## ПРЕОБРАЗОВАНИЕ ГРАММАТИКИ В НОРМАЛЬНУЮ ФОРМУ ГРЕЙБАХ

КС-грамматика, все продукции которой имеют вид

$$A \to a\beta,\ a \in V_T,\ \beta \in (V_N \cup V_T)^*$$

называется представленной в форме Грейбах (Greibach S.A. normal form).

**Теорема.** Для каждого ε-свободного КС-языка существует КС-грамматика в нормальной форме Грейбах, такая, что $L\{G_0[S]\} = L\{G_G[S`]\}$.

**Теорема 1**. Если $A\to\alpha B\gamma$, $A, B\in V_N$, $\alpha,\gamma\in V^*$ правило КС – грамматики, а $B\to\beta_1|\beta_2|\ldots|\beta_m$ – все продукции данной грамматики с нетерминалом $B$ в своих левых частях, то указанные продукции можно заменить правилами вида $A\to\alpha\beta_1\gamma|\alpha\beta_2\gamma|\ldots|\alpha\beta_m\gamma$, причём произведённая замена никак не отражается на языке, порождаемом этой грамматикой.

**Теорема 2**. Если продукции некоторой контекстно-свободной грамматики являются рекурсивными слева продукциями $A\to A\alpha_1|A\alpha_2|...A\alpha_m$, $A \in V_N$, $\alpha_i \in (V \setminus \mathrm{A})^*$, а остальные продукции с нетерминалом $A$ в левой части суть $A\to\beta_1|\beta_2|\ldots|\beta_k$, то новая грамматика, эквивалентная исходной, может быть построена путём добавления нетеминала $A^\nabla$ к множеству $\boldsymbol{V_N}$ и заменой этих продукций исходной грамматики продукциями вида

$$A\to\beta_1|\beta_2|\ldots|\beta_k|\beta_1 A^\nabla|\beta_2 A^\nabla\ldots|\beta_k A^\nabla \text{ и}$$
$$A^\nabla\to\alpha_1|\alpha_2|\ldots|\alpha_m|\beta_1 A^\nabla|\beta_2 A^\nabla\ldots|\beta_k A^\nabla.$$

## Синтаксические деревья и графы

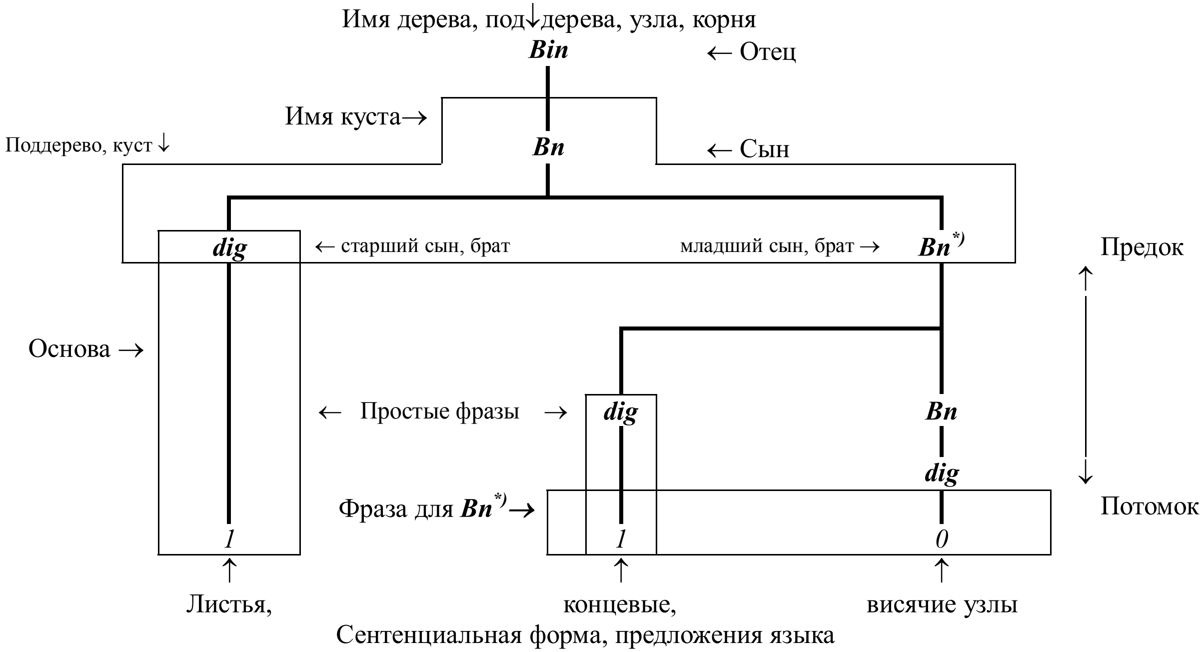



Рисунок 1 – Синтаксическое дерево разбора предложения 110 грамматики $G[Bin]$ (иллюстрация терминологии)

# СВОЙСТВА СИНТАКСИЧЕСКИХ ДЕРЕВЬЕВ

1. Для каждого синтаксического дерева существует, по крайней мере, один вывод.

2. Для каждого вывода существует ***соответствующее*** синтаксическое дерево.

3. Куст дерева указывает непосредственный вывод, в котором имя куста заменяется узлами куста. То есть, среди продукций грамматики существует ***правило***, ***левая*** часть которого – ***имя куста***, а ***правая*** – цепочка из узлов куста, например $<Bn> \to <dig><Bn>$.

4. Концевые узлы дерева образуют сентенциальную форму на каждом шаге вывода или приведения $<Bin> \Rightarrow^{*} 110$, $<Bin> \Rightarrow^{*} <dig>10$ и т.д

5. Если $U$ – корень поддерева для сентенциальной формы $w=xuy$, где $u$ образует цепочку концевых узлов этого поддерева, то $u$ – фраза сентенциальной формы $w$ для $U$. Если поддерево представлено единственным кустом, то это простая фраза, например

$w = 110$, $u \sim 10$, $U \sim <Bn>$ – фраза, $u \sim 1$, $U \sim <dig>$ – простая фраза.

6. Синтаксическое дерево не отражает порядок использования правил. Для синтаксического анализа это несущественно, главное, что дерево построено с их использованием.

## ИССЛЕДОВАНИЕ ГРАММАТИК НА ОДНОЗНАЧНОСТЬ

Предложение является ***синтаксически неоднозначным***, если для его вывода существует ***несколько*** деревьев. Грамматика ***неоднозначна***, если допускает ***неоднозначные*** предложения.

Примеры. 1. “They are flying planes”. 2. “Комиссия по изучению четвертичного периода АН СССР”.

3. Грамматика G[V].

$$V::=V+V|V-V,$$
$$V::=V*V|V/V,$$
$$V::=\langle iden\rangle|\langle data\rangle|(V)$$

Порождает сентенциальные формы:

$V \Rightarrow^{*} \langle iden\rangle + \langle data\rangle * \langle iden\rangle$ и $V \Rightarrow^{*} \langle iden\rangle *(\langle iden\rangle - \langle iden\rangle)$, которые соответствуют в программах на алгоритмических языках конструкции выражений, например

$$x + 2.0*y \text{ или } alpha*(beta - gamma).$$

Сентенциальной форме $\langle iden\rangle + \langle data\rangle * \langle iden\rangle$.

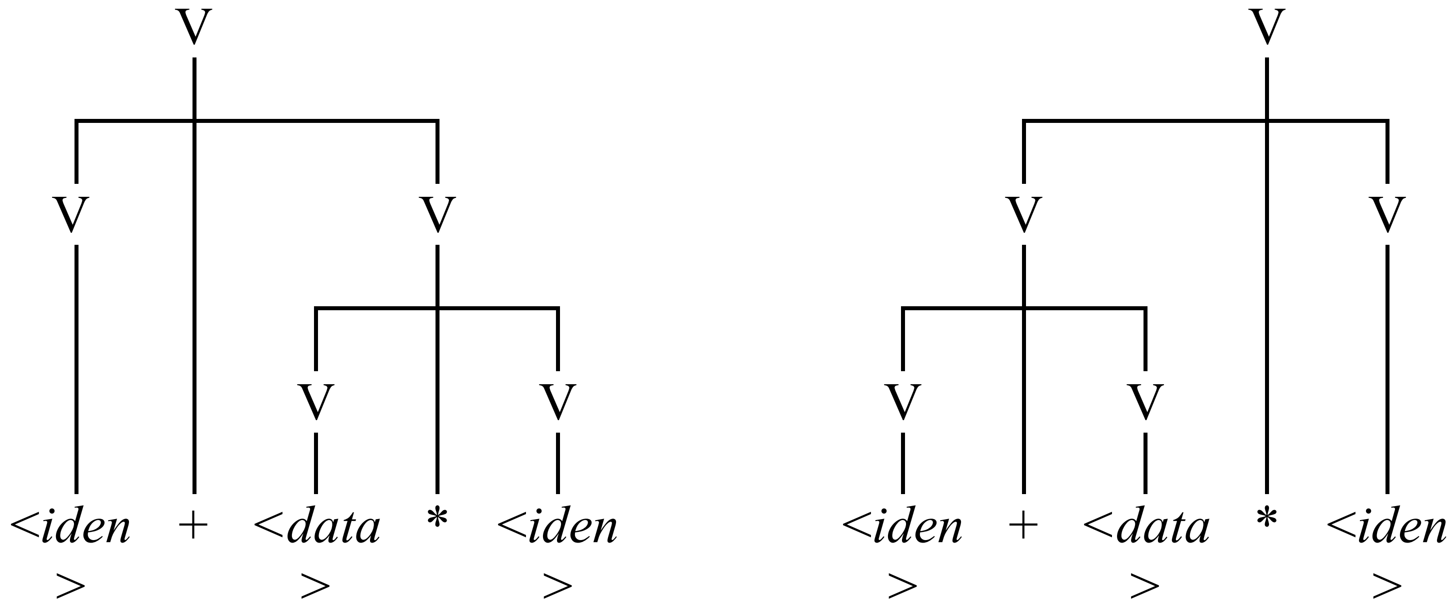


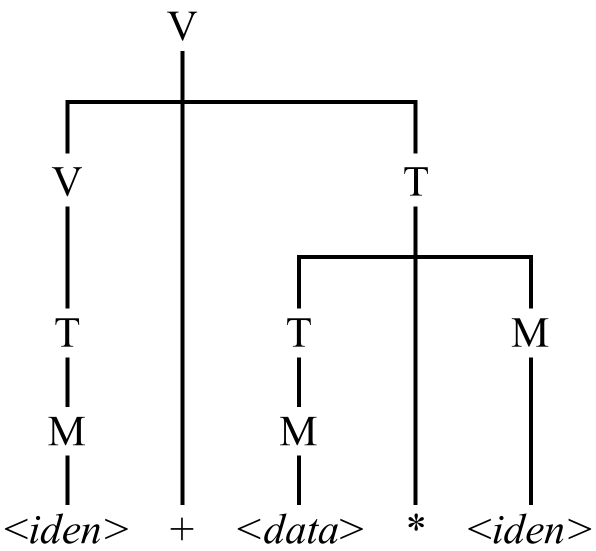

Рисунок 2 – Неоднозначные синтаксические деревья грамматики G[V]

## ПРЕОБРАЗОВАННАЯ ГРАММАТИКА

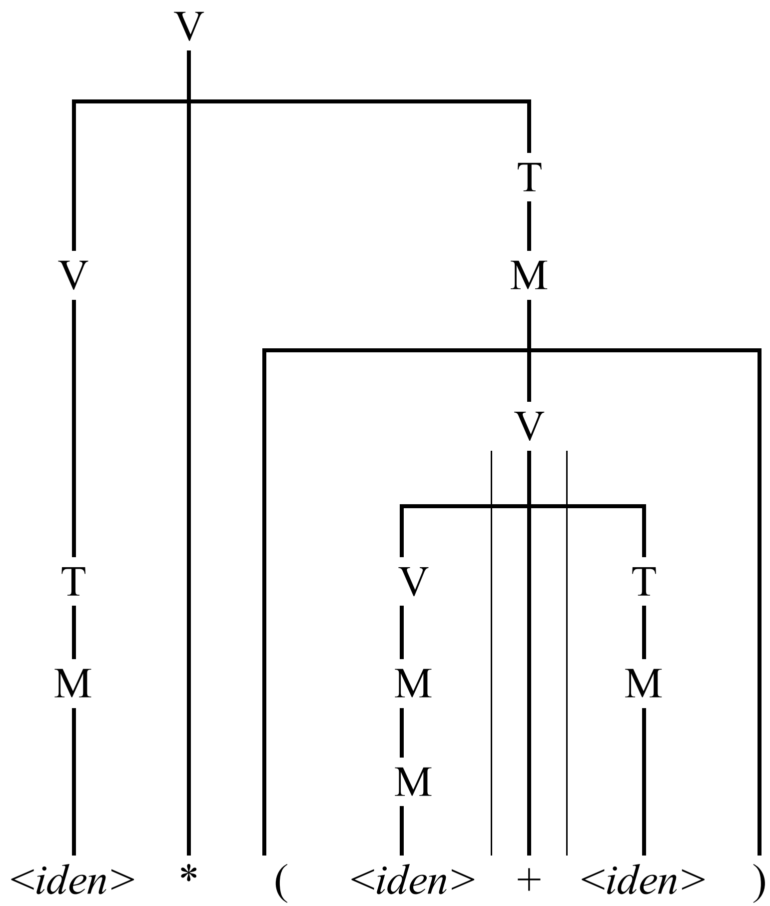
V::=T|V+T|V–T,

T::=M|V*M|V/M,

M::=<*iden*>|<*data*>|(V).




а)




б)

# ПОСТРОЕНИЕ ДВОИЧНЫХ ДЕРЕВЬЕВ ПО ПРОИЗВОЛЬНОМУ ДЕРЕВУ

**Правила преобразования**

Пусть ***T*** – произвольное дерево, а ***B(T)*** - соответствующее ему двоичное дерево.

1. Вершина ***в*** является левым потомком (страшим сыном) вершины ***а*** дерева ***B(T)*** тогда и только тогда, когда ***в*** самый левый потомок ***а*** в ***T***.

От отца дуга влево ведёт к старшему сыну. Рисунок 3, а.

2. Вершина ***в*** является правым потомком (младшим сыном) вершины ***а*** дерева ***B(T)*** тогда и только тогда, когда ***в*** является ближайшем из правых братьев ***а*** дерева ***T***.

Правая дуга ведёт от старших братьев к младшим. Рисунок 3, б.

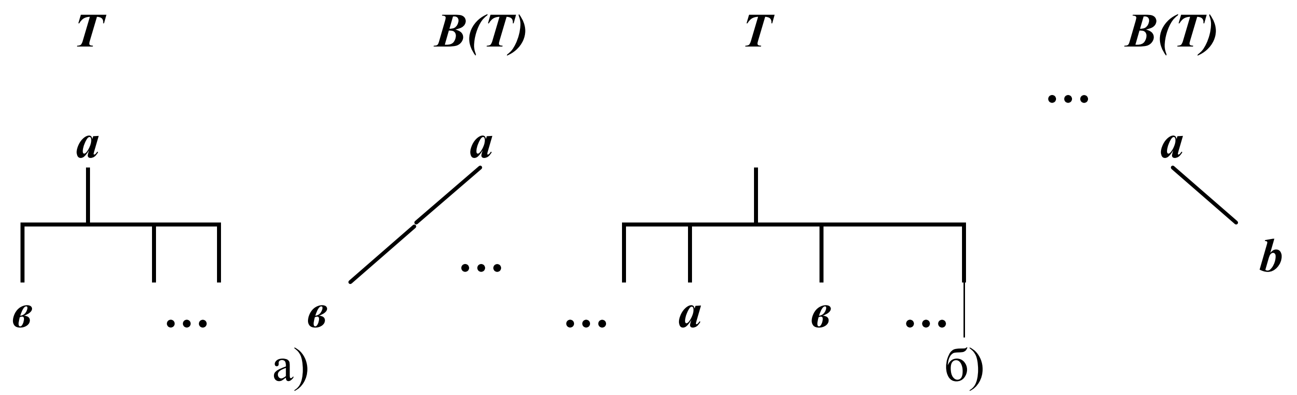



Рисунок 3 – Иллюстрация правил преобразования

# СВОЙСТВА ДВОИЧНЫХ ДЕРЕВЬЕВ

***Лево-(Право) -стороння цепочка*** двоичного дерева есть последовательность ***вершин***
$a_1, a_2, a_3, \ , a_i, \ldots, a_n, (n > 1)$, где $a_i$ является левым (правым) потомком $a_{i-1}$.

Цепочка является ***максимальной***, если содержится сама в себе. Все остальные её подцепочки имеют ***правильные*** головы и хвосты.

Для произвольных деревьев левосторонние и правосторонние цепочки определяются аналогично.

**Лемма 1.** Левосторонние цепочки в ***B(T)*** являются левосторонними и в ***T*** и наоборот.

***Лемма 2.*** Максимальная правосторонняя цепочка узлов в ***B(T)*** является ***аргументом*** некоторого правила подстановки ***P***. Предок левой вершины в цепочке есть корень **P**.

***Лемма 3.*** Максимальная левосторонняя цепочка узлов в ***B(T)*** содержит только один терминальный символ, являющийся концевым (последней вершиной) в этой цепочке.

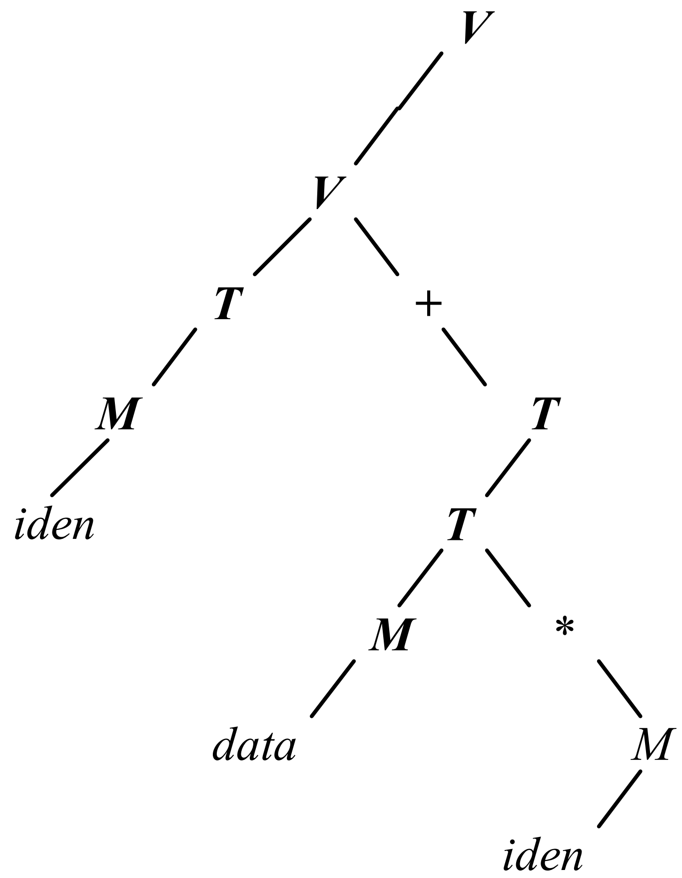


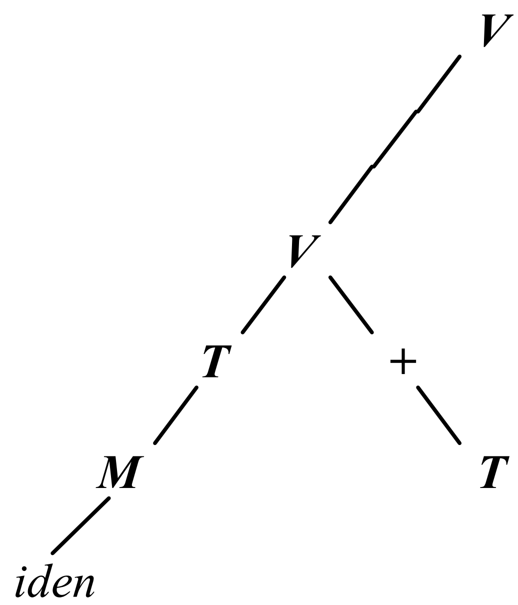

Рисунок 4 – Двоичное дерево сентенциальной формы <*iden*> + <*data*> * <*iden*>

## СКЕЛЕТЫ ДЕРЕВЬЕВ ТРАНСЛЯЦИИ

Пусть имеется двоичное синтаксическое дерево D: начиная с аксиомы ***S,*** заканчивая предложением языка $\boldsymbol{x_1 x_2}$ $\boldsymbol{x_n}$**.**, как это показано на рисунке 4.

**Определение 1**. Максимальная левосторонняя цепочка от ***S*** к $\boldsymbol{x_n}$ называется ***позвоночником*** этого дерева.

**Определение 2**. Позвоночник, вместе с максимальными правосторонними цепочками, выходящими из него, называется скелетом дерева подстановок.



РЕЗЮМЕ

Тактика трансляции такова.

Для грамматики $G[S]$, по заданной входной цепочке $x$, транслятор находит, если это возможно, некоторый вывод цепочки $x$ из аксиомы $S$

$$S \Rightarrow \alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n \equiv x$$

и строит вывод

$$S \Rightarrow (\alpha_0, \beta_0) \Rightarrow (\alpha_1, \beta_1) \Rightarrow (\alpha_2, \beta_2) \ \ldots \Rightarrow (\alpha_n, \beta_n) \equiv (x, y).$$

То есть, ***каждая цепочка*** $\beta_m$ ***порождает цепочку*** $\beta_{m+1}$ с помощью соответствующего правила, применяемого при переходе от $\alpha_m$ к $\alpha_{m+1}$, а цепочка $y$, представляющая собой результат трансляции, служит выходом для цепочки $x$.

Пусть $V_T$ — входной алфавит, а $\Delta$ — выходной алфавит.

***Переводом*** с языка $L_1 \subseteq V_T^*$ на язык $L_2 \subseteq \Delta^*$ называется отображение $T$ из $V_T^*$ в $\Delta^*$, для которого $L_1$ – область определения, а $L_1$ – множество значений.