

ЛЕКЦИЯ 6

“НИСХОДЯЩИЙ РАЗБОР. ПРИНЦИПЫ ПОСТРОЕНИЯ И АЛГОРИТМЫ”

ПЛАН

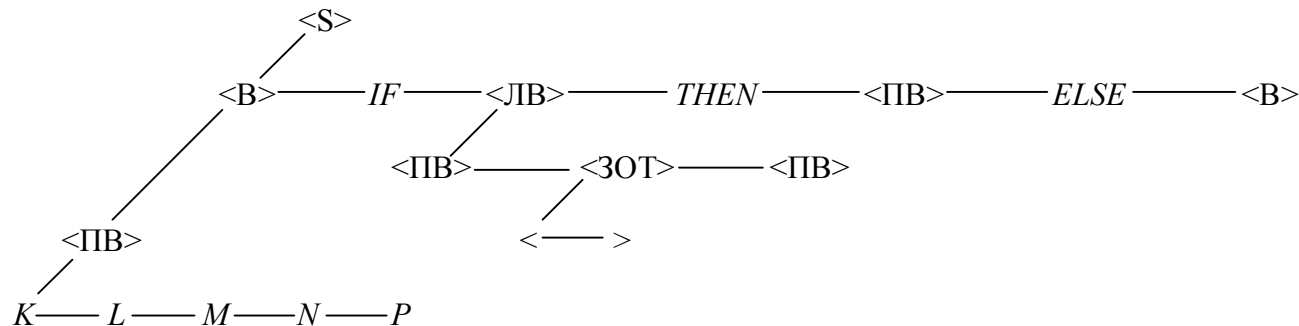
1. Задачи, стоящие на этапе синтаксическом анализа. Методы и способы синтаксического анализа.
2. Простой нисходящий разбор. Принцип построения и алгоритм. Пример разбора.
3. Прогнозирующий разбор: $LL(k)$ -грамматики и рекурсивный спуск.
4. МП-распознаватель для языка, порождаемого $LL(k)$ -грамматикой

Учебная грамматика $G[S]$

$\langle S \rangle ::= \langle B \rangle$
 $\langle B \rangle ::= \langle PB \rangle | \text{if} \langle LB \rangle \text{ then} \langle PB \rangle \text{ else} \langle B \rangle$
 $\langle PB \rangle ::= K | L | M | N | P$
 $\langle LB \rangle ::= \langle PB \rangle \langle ZOT \rangle \langle PB \rangle$
 $\langle ZOT \rangle ::= > | <$

Аксиома грамматики – S , B - выражение, PB - простое выражение, LB - логическое выражение, ZOT - знак отношения.

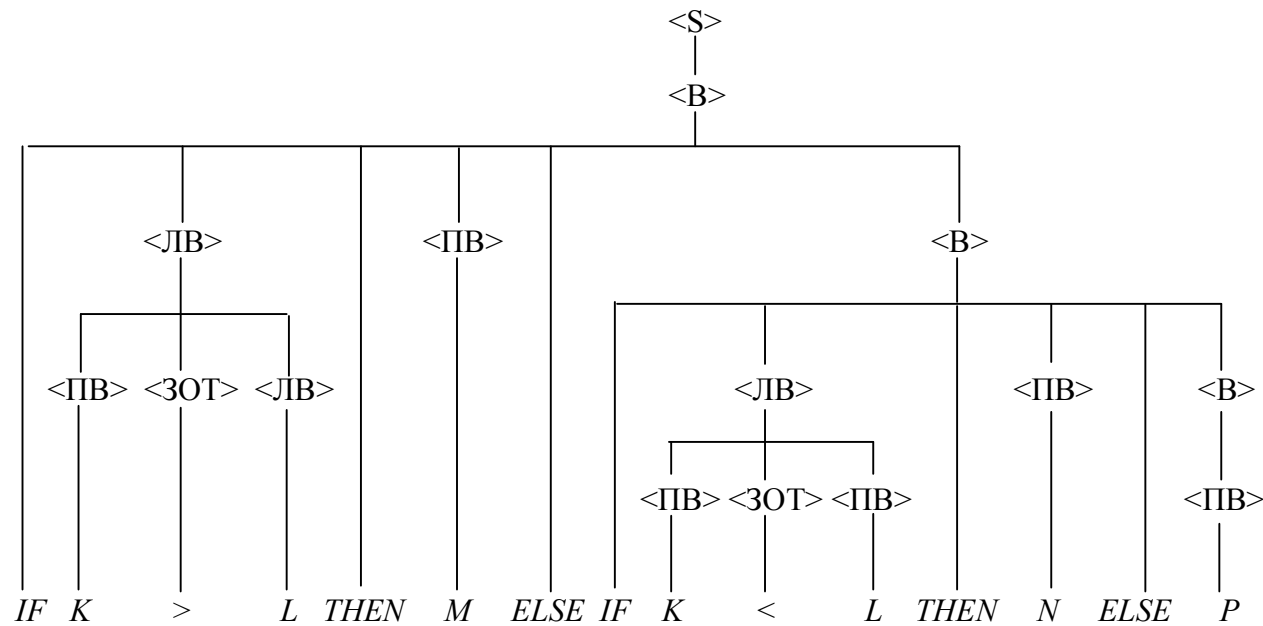
Синтаксический граф $G[S]$



Анализируемая строка

if K > L then M else if K < L then N else P.

Синтаксическое дерево разбираемой сентенциальной формы



Терминология:

1. Целью разбора на каждом этапе является *опознание* некоторого *нетерминального символа*.

2. Иерархия целей

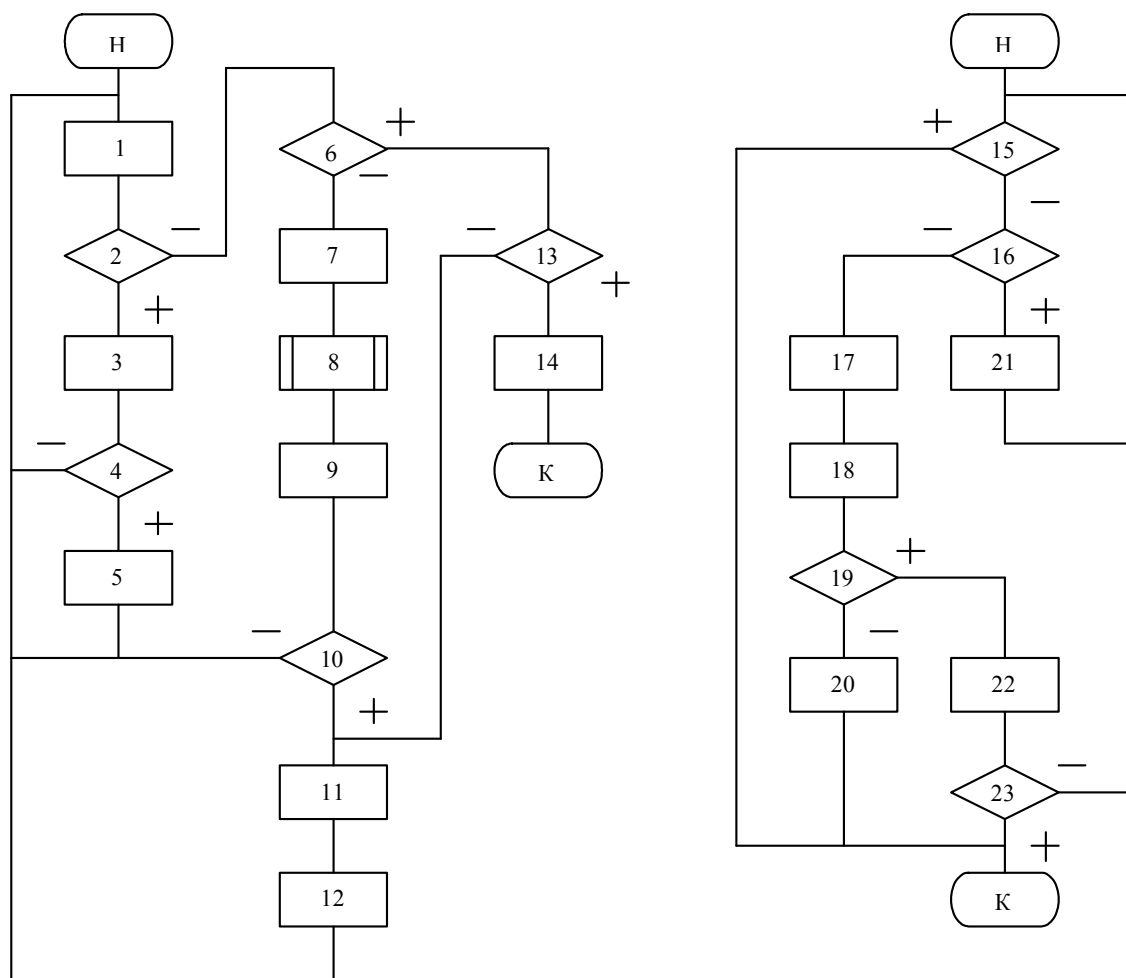
- *Достигнутая* цель или подцель.
- *Поставленная* цель или подцель, она называется тако же текущей.
- *Глобальная* (по отношению к поставленной) цель или подцель.

Табличное представление синтаксического графа

- Указатель на предшественника (отца) – строку таблицы, из которой была достигнута текущая.
- Наименование узла дерева – имя терминала или нетерминала, начало или конец правила.
- Наличие альтернативы у узла и ссылка на эту альтернативу.
- Тип узла – терминальный или нетерминальный.
- Ссылка на начало правила, определяющего нетерминальный узел.

№ элемента структуры	Предшественник	Имя узла дерева	Наличие альтернативы	Тип узла дерева	Начало правила
1	НП	S	—	Н	—
2	1	B	—	Н	4
3	—	КП	—	—	—
4	НП	B	—	Н	-
5	4	ПВ	7	Н	14
6	—	КП	—	—	—
7	4	IF	—	T	—
8	7	ЛВ	—	Н	25
9	8	THEN	—	T	—
10	9	ПВ	—	Н	14
11	10	ELSE	—	T	—
12	11	B	—	Н	4
13	—	КП	—	—	—
14	НП	ПВ	—	Н	—
15	14	K	17	T	—
16	—	КП	—	—	—
17	14	L	19	T	—
18	—	КП	—	—	—
19	14	M	21	T	—
20	—	КП	—	—	—
21	14	N	23	T	—
22	-	КП	—	—	—
23	14	P	—	T	—
24	—	КП	—	—	—
25	НП	ЛВ	—	Н	—
26	25	ПВ	—	Н	14
27	26	ЗОТ	—	Н	30
28	27	ПВ	—	Н	14
29	—	КП	—	—	—
30	НП	ЗОТ	—	Н	—
31	30	<	33	T	—
32	—	КП	—	—	—
33	30	>	—	T	—
34	—	КП	—	—	—

Алгоритм простого нисходящего разбора



ПОЯСНЕНИЯ К АЛГОРИТМУ

Левая часть схемы алгоритма.

1 – чтение символа из правил грамматики $S \leftarrow G(Z)$. 2 – проверка условия, что символ S – нетерминал? 3 – запись S в МПЦ. 4 – проверка, есть ли продукция с S в левой части? 5 – переход на начало соответствующего правила. 6, 10 – проверка на условие окончания – конец правила? 7 – чтение текущего символа из программы $T \leftarrow L\{G[z]\}$. 8 – осуществление попытки идентификации терминала S из грамматики и терминала T из кодов программы (блоки №№ 15 – 23). 9 – продолжение просмотра текущей поставленной цели. 11 – перепись последнего символа (на вершине) из МПЦ в МДЦ. 12 – возврат к глобальной подцели. 13 – конец анализируемой программы? 14 – перепись содержимого МПЦ в МДЦ.

Правая часть схемы алгоритма.

15 – терминалы из грамматики и программы совпали? 16 – есть ли альтернатива терминалу грамматики? 17 – вытолкнуть верхний элемент МПЦ. 18 – вернуться на глобальную подцель. 19 - Есть ли альтернатива глобальной подцели? 20 – генерация сообщения об ошибке. 21 – уход на альтернативу терминала. 22 – уход на альтернативу глобальной подцели. 23 – альтернатива – нетерминальный символ?

СОСТОЯНИЕ МАГАЗИНОВ

Такт	Анализируемый символ	Состояние МПЦ	Поступления в МДЦ
1		S B	
2	IF	S B ПВ	
3		S B	
4		S B ЛВ	
5	K	S B ЛВ ПВ	ПВ
6	>	S B ЛВ ЗОТ	ЗОТ
7	L	S B ЛВ ПВ	ПВ ЛВ
8	THEN	S B	
9	M	S B ПВ	ПВ
10	ELSE	S B B	
11	IF	S B B ПВ	
12		S B B	
13		S B B ЛВ	
14	K	S B B ЛВ ПВ	ПВ
15	<	S B B ЛВ ЗОТ	ЗОТ
16	L	S B B ЛВ ПВ	ПВ ЛВ
17	THEN	S B B	
18	N	S B B ПВ	ПВ
19	ELSE	S B B B	
20	P	S B B B ПВ	ПВ B
21	KT	S B B	B B S

ПРОГНОЗИРУЮЩИЙ НИСХОДЯЩИЙ РАЗБОР

$LL(k)$ – грамматикой называют такую грамматику, что для любой её сентенциальной формы $\omega A \gamma$, $\omega \in V_T^*$, $A \in V_N$, $\gamma \in V^*$, полученной в результате некоторого левостороннего вывода, для выбора продукции, имеющей в левой части нетерминал A , достаточно знать k очередных вводимых символов (символов входной, разбираемой строки).

Алгоритм определения $LL(k)$ -грамматики

1. Для каждой i -ой продукции грамматики необходимо найти головы цепочек из k символов, которые могут быть получены в процессе применения данного правила. Они являются одновременно и головами анализируемых цепочек. Множество цепочек, соответствующих i -ой продукции, обозначим C_i .

2. Если первый символ правой части продукции – нетерминальный, то ей ставится в соответствие объединённое множество, сконструированное объединением множеств C_i , построенных для продукций, имеющих в левых частях означенный нетерминал.

3. Если множества, соответствующие продукциям с одинаковыми левыми частями не пересекаются, то это $LL(k)$ -грамматика.

Проверку начинают с индекса $k = 1$. Увеличивают индекс k , если проверка окончилась неудачно. Увеличение индекса больше двух нецелесообразно

Пример. Исследование учебной грамматики на принадлежность к $LL(k)$ -грамматикам

Продукция грамматики			Множество цепочек
$\langle S \rangle$	$::=$	$\langle B \rangle$	$\{if, K, L, M, N, P\}$
$\langle B \rangle$	$::=$	$\langle ПВ \rangle$	$\{K, L, M, N, P\}$
$\langle B \rangle$	$::=$	$if \langle ЛВ \rangle then \langle ПВ \rangle else \langle B \rangle$	if
$\langle ПВ \rangle$	$::=$	K	$\{K\}$
$\langle ПВ \rangle$	$::=$	L	$\{L\}$
$\langle ПВ \rangle$	$::=$	M	$\{M\}$
$\langle ПВ \rangle$	$::=$	N	$\{N\}$
$\langle ПВ \rangle$	$::=$	P	$\{P\}$
$\langle ЛВ \rangle$	$::=$	$\langle ПВ \rangle \langle 3OT \rangle \langle ПВ \rangle$	$\{K, L, M, N, P\}$
$\langle 3OT \rangle$	$::=$	$>$	$\{>\}$
$\langle 3OT \rangle$	$::=$	$<$	$\{<\}$

СХЕМАТИЧНАЯ ПРОГРАММА РАСПОЗНАВАТЕЛЯ:

```
int      S_S(void),      S_B(void),      S_PB(void),      S_LV(void),
S_ZOT(void), S_k(void), S_l(void), S_m(void), S_n(void), S_p
(void), S_if(void), S_then(void), S_else(void), S_gt(void),
S_lt(void);
int code;                                /*текущий дескриптор лексемы*/
void main(void)
{
FILE *Prog_code;
if ((Prog_code=fopen("../", "r")) == NULL) printf("\n Ошибка
открытия")
        else { fscanf(Prog_code, "%d", &code);
if (S_S() == 0)  printf ("\n Синтаксическая ошибка");
        else printf ("\n OK");
}
```

```
int S_S (void)
{
    switch (code) {
        case "K":    return S_B(); break;
        case "L":    return S_B(); break;
        case "M":    return S_B(); break;
        case "N":    return S_B(); break;
        case "P":    return S_B(); break;
        case "if":   return S_B(); break;
        default: printf ("\n неверное начало оператора или конец
модуля");
                return 0;}
    }
```

```

int S_B (void)
{
    switch (code) {
        case "K": return S_PB(); break;
        case "L": return S_PB(); break;
        case "M": return S_PB(); break;
        case "N": return S_PB(); break;
        case "P": return S_PB(); break;
        case "if":    return
S_if()*S_LV()*S_then()*S_PB()*S_else()*S_B(); break;
        default:    printf ("\n неверное начало оператора или конец
модуля");
                    return 0;}
    }
...
int S_lt (void)
{
    switch (code) {
        case "<": return 1; break;
        default:    printf ("\n ожидался знак «меньше»");
                    return 0;}
    }
}

```

МП-распознаватель для $LL(1)$ -грамматики

Алгоритм построения

Оформлен в виде конструктивной теоремы.

Распознаватель с одним состоянием строится следующим образом

1. Входной алфавит $\Sigma = V_T \cup \{\blacktriangleleft\}$, \blacktriangleleft - конец входной строки.

2. Алфавит магазинных символов составляют

$$\Theta = V_N \cup \{t | t \in V_T \text{ и } (\mathbf{A} \rightarrow t\alpha) \notin R, \mathbf{A} \in V_N, \alpha \in V^*\} \cup \{\#\}.$$

С терминала t не начинается ни одна из правых частей правил, $\#$ - выталкиватель магазина.

3. Имеется одно состояние $Q = \{q\}$.

4. Начальное состояние магазина $Z_0 = \{\#S\}$, где S – аксиома грамматики.

5. Начальное состояние автомата $q_0 = q$.

6. Устройство управления $\text{Table}(Q, \Sigma)$ конструируется следующим образом.

6.1. Для продукции грамматики вида $\mathbf{A} \rightarrow t\alpha$, $t \in V_T$, $\alpha \in V^*$

$\text{Table}(\mathbf{A}, t) \Rightarrow \text{ЗАМЕНИТЬ } (\alpha^\nabla), \text{ СДВИГ}$, где

α^∇ - есть цепочка α , записанная в обратном порядке (задом на перёд).

6.2. Для правила $A \rightarrow t\alpha$, где $\alpha = X\beta$, $X \in V_N$, $\beta \in V^*$

Table (A , {ВЫБОР($A \rightarrow \alpha$)}) \Rightarrow ЗАМЕНИТЬ (α^∇), ДЕРЖАТЬ

Если $\alpha = \varepsilon$ - аннулирующая (пустая) цепочка, вместо ЗАМЕНИТЬ употребляется операция ВЫТОЛКНУТЬ.

Множество {ВЫБОР($A \rightarrow \alpha$)} = ПЕРВИЧНЫЙ(α), а если существует аннулирующая цепочка, то

{ВЫБОР($A \rightarrow \alpha$)} = ПЕРВИЧНЫЙ(α) \cup СЛЕДУЮЩИЙ(α),

где ПЕРВИЧНЫЙ(α) – цепочки единичной длины, полученные при исследовании грамматики на свойство $LL(1)$, СЛЕДУЮЩИЙ(α) – если голова правой части правила – нетерминальный символ, являющийся ε -порождающим, определяется по 2-м символом при исследовании на $LL(2)$.

6.3. Для правила $A \rightarrow t$, $A \in V_N$, $t \in V_T$

Table (A , t) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ

6.4. Для ситуации $b \in V_T \cap b \in \Theta$ (когда терминал является одновременно и магазинным символом)

Table (b , b) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ

6.5. Table ($\#$, \blacktriangleleft) \Rightarrow ДОПУСТИТЬ

6.6. Остальные элементы таблицы заполнить ОТВЕРГНУТЬ.

Формальная грамматика $G[S]$:

$$\begin{aligned}
 \langle S \rangle &::= \langle B \rangle & (1) \\
 \langle B \rangle &::= \langle PB \rangle | \text{if} \langle LB \rangle \text{ then} \langle PB \rangle \text{ else} \langle B \rangle & (2), (3) \\
 \langle PB \rangle &::= K | L | M | N | P & (4) - (8) \\
 \langle LB \rangle &::= \langle PB \rangle \langle ZOT \rangle \langle PB \rangle & (9) \\
 \langle ZOT \rangle &::= > | < & (10), (11)
 \end{aligned}$$

1. Алфавит входных символов $\Sigma = \{if, then, else, K, L, M, N, P, <, >, \blacktriangleleft\}$.

2. Алфавит магазинных символов

$$\Theta = \{S, B, PB, LB, ZOT, then, else, \#\}.$$

П.6.1. Для правила № 3 формальной грамматики имеем:

$$t = if, \alpha = \langle LB \rangle \text{ then} \langle PB \rangle \text{ else} \langle B \rangle,$$

поэтому $\text{Table}(B, if) \Rightarrow \text{ЗАМЕНИТЬ}(\langle B \rangle \text{ else} \langle PB \rangle \text{ then} \langle LB \rangle)$, СДВИГ.

П.6.2. Под него подпадают правила формальной грамматики (1), (2) и (9).

Для правила № 1: множество $\text{ВЫБОР}(S \rightarrow B) = \{if, K, L, M, N, P\}$, поэтому

$$\text{Table}(S, \{if, K, L, M, N, P\}) \Rightarrow \text{ЗАМЕНИТЬ}(B), \text{ДЕРЖАТЬ}.$$

Для правила № 2: множество $\text{ВЫБОР}(B \rightarrow PB) = \{K, L, M, N, P\}$, поэтому

$$\text{Table}(B, \{if, K, L, M, N, P\}) \Rightarrow \text{ЗАМЕНИТЬ}(PB), \text{ДЕРЖАТЬ}.$$

Для правила № 9: множество $\text{ВЫБОР}(\text{ЛВ} \rightarrow \langle \text{ПВ} \rangle \langle \text{ЗОТ} \rangle \langle \text{ПВ} \rangle) = \{K, L, M, N, P\}$, поэтому получаем

Table (**ЛВ**, {K, L, M, N, P}) \Rightarrow ЗАМЕНИТЬ ($\langle \text{ПВ} \rangle \langle \text{ЗОТ} \rangle \langle \text{ПВ} \rangle$), ДЕРЖАТЬ.

П.6.3. Ему соответствуют продукции грамматики (4) – (8), (10) и (11).

Для правил №№ (4) – (8):

Table (**ПВ**, {K, L, M, N, P}) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ.

Для правил №№ (10) и (11):

Table (**ЗОТ**, {<, >}) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ.

П.6.4. Под его действие подпадают пары (*then, then*) и (*else, else*):

Table (*then, then*) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ,

Table (*else, else*) \Rightarrow ВЫТОЛКНУТЬ, СДВИГ.

Представим устройство управления в табличном виде.

	<i>if</i>	<i>then</i>	<i>else</i>	<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>P</i>	>	<	◀
S	6.2.1			6.2.1	6.2.1	6.2.1	6.2.1	6.2.1			
B	6.1.3			6.2.3	6.2.3	6.2.3	6.2.3	6.2.3			
ПВ				6.3.4	6.3.5	6.3.6	6.3.7	6.3.8			
ЛВ				6.2.9	6.2.9	6.2.9	6.2.9	6.2.9			
ЗОТ									6.3.10	6.3.11	
<i>then</i>		6.4.3									
<i>else</i>			6.4.3								
#											Доп

В ячейках таблицы укажем код вида X.Y.Z, где X.Y – ссылка на номер пункта алгоритма построения МП-распознавателя, Z – номер продукции формальной грамматики, в соответствие которой поставлена команда, Доп – означает ДОПУСТИТЬ, пустые клеточки соответствуют ОТВЕРГНУТЬ.

Трасса разбора конструкции *if K > L then M else if K < L then N else P.*

№	Состояние стека	Вход	Правило
0	#< <i>S</i> >	<i>if</i>	6.2.1
1	#< <i>B</i> >	<i>if</i>	6.1.3
2	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ЛВ</i> >	K	6.2.9
3	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> > < <i>ЗОТ</i> > < <i>ПВ</i> >	K	6.3.4
4	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> > < <i>ЗОТ</i> >	>	6.3.10
5	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> >	L	6.3.5
6	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i>	<i>then</i>	6.4
7	#< <i>B</i> > <i>else</i> < <i>ПВ</i> >	M	6.3.6
8	#< <i>B</i> > <i>else</i>	<i>else</i>	6.4.3
9	#< <i>B</i> >	<i>if</i>	6.1.3
10	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ЛВ</i> >	K	6.2.9
11	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> > < <i>ЗОТ</i> > < <i>ПВ</i> >	K	6.3.4
12	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> > < <i>ЗОТ</i> >	<	6.3.11
13	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i> < <i>ПВ</i> >	L	6.3.5
14	#< <i>B</i> > <i>else</i> < <i>ПВ</i> > <i>then</i>	<i>then</i>	6.4.3
15	#< <i>B</i> > <i>else</i> < <i>ПВ</i> >	N	6.3.7
16	#< <i>B</i> > <i>else</i>	<i>else</i>	6.4.3
17	#< <i>B</i> >	P	6.2.3
18	#< <i>ПВ</i> >	P	6.3.8
19	#	◀	Допущено