1. **Постановка задачи: описание языка с примерами**

   Выражений из оператора 'while'. Пример: while(++a<10) while (a<b) print(b++);

   Требуется разработать грамматику оператора while, в теле которого находится либо ещё один оператор while, либо некоторая функция печати print().

   expr – выражение сравнения значений двух переменных оператором <.

   int – произвольное целое число.

   text – произвольная буква английского алфавита.

   Нетерминалы: *temp, expr, value*
   Терминалы: *while , print , int , text , (,) , ; , < , ++*

2. **Разработать грамматику G по описанию (БНФ)**

$$temp ::= while(expr)temp \mid print(value);$$
$$expr ::= value < value \mid value$$
$$value ::= ++text \mid text \mid text++ \mid int$$

3. **Определить класс и однозначность языка и грамматики**

   Грамматика G является контекстно-свободной (класс языка тип II), по определению:

$$U ::= u$$
$$U \in V - T, \qquad u \in V^*$$

   Т. е. слева один нетерминальный символ, а в правой части – произвольная цепочка. Тип 0 и I не подходят, поскольку в тех классах возможно присутствие терминальных символов в левой части правил. Тип III не

подходит, поскольку грамматика не подходит по условиям к правой части.

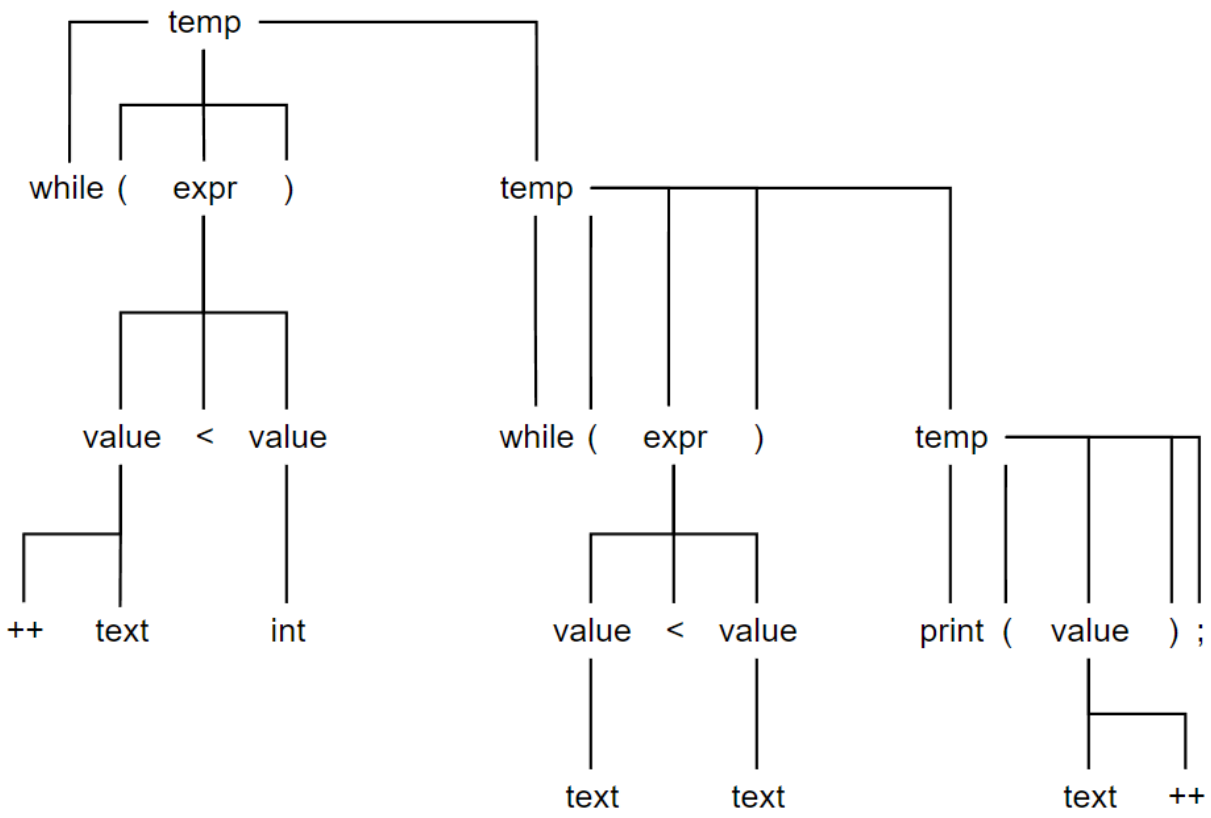
Класс языка определяется классом грамматики большего порядка по классификации по Хомскому, которая может его породить. Так как регулярные выражения не могут решить проблему баланса скобок, язык не может быть регулярным и относится к классу контекстно-свободных. Грамматика является однозначной, т.к. таблицы SLR и LL можно построить без конфликтов и существует единственное синтаксическое дерево разбора для произвольных предложений. Контекстно-свободный язык называется однозначным, если для него существует однозначная грамматика.

4. **Построить синтаксическое дерево и вывод произвольного предложения**

   Пример: $while(++a<10)\ while\ (a<b)\ print(b++);$

   Запишем пример в контексте грамматики:

   $while(++text<int)\ while\ (text<text)\ print(text++);$



Вывод:

$temp \rightarrow$

$\rightarrow while(expr)\ temp \rightarrow$

$\rightarrow while(value < value)\ temp \rightarrow$

$\rightarrow while(++text < value)\ temp \rightarrow$

$\rightarrow while(++text < int)\ temp \rightarrow$

$\rightarrow\ while(++text < int)\ while(expr)\ temp \rightarrow$

$\rightarrow\ while(++text < int)\ while(value < value)\ temp \rightarrow$

$\rightarrow\ while(++text < int)\ while(text < value)\ temp \rightarrow$

$\rightarrow\ while(++text < int)\ while(text < text)\ temp \rightarrow$

$\rightarrow\ while(++text < int)\ while(text < text)\ print(value); \rightarrow$

$\rightarrow\ while(++text < int)\ while(text < text)\ print(text++);$

5. **Произвести 2 последовательных редукции основы в произвольной сентенциальной форме**

Пусть *G[Z]* – грамматика. Цепочка *x* называется сентенциальной формой, если *x* выводима из начального символа Z ($Z \to* x$). Предложение – сентенциальная форма, состоящая только из терминальных символов.
Пусть *G[Z]* – грамматика и *w=xuy* – сентенциальная форма. Тогда *u* называется фразой сентенциальной формы *w* для нетерминального символа *U*, если $Z \to* xUy$ и $U \to +u$, *u* называется простой фразой, если $Z \to* xUy$ и $U \to u$.
Основой всякой сентенциальной формы называется самая левая простая фраза.

Пример: $while(0)\ print(1);$

Запишем в контексте грамматики: $while(int)\ print(int);$

$v = xUy,\ \ w = xuy,\ \ v => w$

1. $w =\ while(int)\ print(int);$
   $x = while($
   $u = int$
   $y =)\ print(numint);$
   $U = value$
   $v =\ while(value)\ print(int);$

2. $w =\ while(value)\ print(int);$
   $x = while($
   $u = value$
   $y =\ print(int);$
   $U = expr$
   $v =\ while(expr)\ print(int);$

   $while(int)\ print(int); \to while(value)\ print(int); \to\ while(expr)\ print(int);$

6. Разработать анализатор: классический сканер и рекурсивный нисходящий предикативный распознаватель

Грамматика, пригодная для предиктивного анализа:

(0) $temp' ::= temp$

(1) $temp ::= while(expr)\ temp$

(2) $temp ::= print(param);$

(3) $expr ::= param\ compar$

(4) $compar ::= < param$

(5) $compar ::= > param$

(6) $compar ::= \varepsilon$

(7) $param ::= plus\ var$

(8) $param ::= var\ plus$

(9) $param ::= num$

(10) $plus ::= ++$

(11) $plus ::= \varepsilon$

(12) $var ::= i$

7. Построить таблицу SLR(1), привести пример разбора по ней

Нетерминалы: *temp, expr, param, compar, plus, var*

Терминалы: *while, (,), print, ;, <, >, ε, num, ++, i, ε*

I0:

temp' = • *temp*

temp :: = • **while (** *expr* **)** *temp*

temp ::= • **print (** *param* **);**

expr ::= • *param compar*

compar ::= • < *param*

compar ::= • > *param*

compar ::= • ε

param ::= • ++ *var*

param ::= • *var plus*

param ::= • **num**

plus ::= • ++

plus ::= • ε

var ::= • **i**

I1 = goto (I0, temp)

temp' = temp •

I2 = goto (I0, while) = I25

temp :: = **while** • **(** *expr* **)** *temp*

I3 = goto (I0, print) = I25

temp ::= **print** • **(** *param* **);**

I4 = goto (I0, param) = I12

expr ::= *param* • *compar*

compar ::= • < *param*

compar ::= • > *param*

compar ::= • ε

I5 = goto (I0, <) = I4

compar ::= < • *param*

param ::= • ++*var*

param ::= • *var plus*

param ::= • **num**

var ::= • **i**

I6 = goto (I0, >) = I4

compar ::= > • *param*

param ::= • ++*var*

param ::= • *var plus*

param ::= • **num**

var ::= • **i**

I7 = goto (I0, ε)

compar ::= ε •

plus ::= ε •

I8 = goto (I0, ++)

param ::= ++ • *var*

var ::= • **i**

plus ::= ++ •

I9 = goto (I0, var) = I5, I6, I12, I13

param ::= *var* • *plus*

plus ::= • ++

plus ::= • ε

I10 = goto (I0, num) = I5, I6, I12, I13

param ::= **num** •

I11 = goto (I0, i) = I5, I6, I8, I12, I13, I17

var ::= **i** •

I12 = goto (I2, "(")

temp :: = **while (** • *expr* **)** *temp*

expr ::= • *param compar*

param ::= • ++*var*

param ::= • *var plus*

param ::= • **num**

var ::= • **i**

I13 = goto (I3,))

temp ::= **print (** • *param* **);**

param ::= • ++*var*

param ::= • *var plus*

param ::= • **num**

var ::= • **i**

I14 = goto (I4, compar)

expr ::= *param compar* •

I15 = goto (I4, ε)

compar ::= ε •

I16 = goto (I5, param)

compar ::= < *param* •

I17 = goto (I5, ++) = I6, I12, I13

param ::= ++ • *var*

var ::= • **i**

I18 = goto (I6, param)

compar ::= > *param* •

I19 = goto (I8, var) = I17

param ::= ++ *var* •

I20 = goto (I9, plus)

param ::= *var plus* •

I21 = goto (I9, ++)

plus ::= ++ •

I22 = goto (I9, ε)

plus ::= ε •

I23 = goto (I12, expr)

temp :: = **while (** *expr* • **)** *temp*

I24 = goto (I13, param)

temp ::= **print (** *param* • **);**

I25 = goto (I23, expr)

temp :: = **while (** *expr* **)** • *temp*

temp :: = • **while (** *expr* **)** *temp*

temp ::= • **print (** *param* **);**

I26 = goto (I24, “)”)

temp ::= **print (** *param* **)** • **;**

I27 = goto (I25, temp)

temp :: = **while (** *expr* **)** *temp* •

I28 = goto (I26, “;”)

temp ::= **print (** *param* **) ;**

	FIRST	**FOLLOW**
temp'	while print	$
temp	while print	$
expr	++ num	)
value	++ num i	< >)
compar	< > ε	)
plus	ε	< >)
var	i	++) < >

	while	**(**	**)**	**v**	**^**	**++**	**;**	**i**	**print**	**num**	**$**	**temp**	**expr**	**param**	**compar**	**plus**	**var**
0	S2			S5	S6	S8		S11	S3	S10		1		4			9
1											acc						
2		S12															
3		S13															
4			R3	S5	S6										14		
5			R10	R10	R10	S17		S11		S10				16			9
6			R10	R10	R10	S17		S11		S10				18			9
7			R6	R11	R11												
8			R10	R10	R10			S11									19
9			R8	R8	R8	S21										20	
10			R9	R9	R9												
11			R12	R12	R12	R12											
12			R10	R10	R10	S17		S11		S10			23	4			9
13			R10	R10	R10	S17		S11		S10				24		20	9
14			R3														
15			R6														
16			R4														
17			R10	R10	R10	R10		S11									17
18			R5														

19			R7	R7	R7												
20			R8	R8	R8												
21			R10	R10	R10												
22			R11	R11	R11												
23			S25										25				
24			S26				S28										
25	S2								S3			27					
26							S28										
27											R1						
28											R2						

SLR(1)

Пример: $while\ (\ i\ )\ print\ (i++);$

Стек	**Вход**	**Действие**
0	while (i) print (i++);$	Перенос while (s2)
0 while 2	(i) print (i++);$	Перенос ((s12)
0 while 2 (12	i) print (i++); $	Перенос i (s11)
0 while 2 (12 i 11	) print (i++);$	Свёртка (R12) *var* ::= **i**
0 while 2 (12 var 8	) print (i++);$	Свёртка (R8) *param* ::= *var plus*
0 while 2 (12 param 3	) print (i++);$	Свёртка (R3) *expr* ::= *param compar*
0 while 2 (12 expr 23	) print (i++);$	Перенос) (s25)
0 while 2 (12 expr 23) 25	print (i++);$	Перенос print (s3)
0 while 2 (12 expr 23) 25 print 3	(i++);$	Перенос ((s13)

0 while 2 (12 expr 23) 25 print 3 (13	i++);$	Перенос i (s11)
0 while 2 (12 expr 23) 25 print 3 (13 i 11	++);$	Свёртка (R12) *var* ::= **i**
0 while 2 (12 expr 23) 25 print 3 (13 var 13	++);$	Перенос ++ (S17)
0 while 2 (12 expr 23) 25 print 3 (13 var 13 ++ 17	);$	Свёртка (R10) *plus* ::= **++**
0 while 2 (12 expr 23) 25 print 3 (13 var 13 plus 20	);$	Свёртка (R8) param ::= var plus
0 while 2 (12 expr 23) 25 print 3 (13 param 24	);$	Перенос) (S26)
0 while 2 (12 expr 23) 25 print 3 (13 param 24) 26	;$	Перенос ; (S28)
0 while 2 (12 expr 23) 25 print 3 (13 param 24) 26 ; 28	$	Свёртка (R2) temp ::= print (param);
0 while 2 (12 expr 23) 26 temp 27	$	Свёртка (R1) temp :: = while (expr) temp
0 temp 1	$	access

8. Построить таблицу LL(1), привести пример разбора по ней

Грамматика:

(0) $temp' ::= temp$

(1) $temp ::= while(expr)\ temp$

(2) $temp ::= print(param);$

(3) $expr ::= param\ compar$

(4) $compar ::= < param$

(5) $compar ::= > param$

(6) $compar ::= \varepsilon$

(7) $param ::= ++ var$

(8) $param ::= var\ plus$

(9) $param ::= num$

(10) $plus ::= ++$

(11) $plus ::= \varepsilon$

(12) $var ::= i$

	FIRST	FOLLOW
temp'	while print	$
temp	while print	$
expr	++ num	)
value	++ num i	< >)
compar	< > ε	)
plus	ε	< >)
var	i	++) < >

	while	**print**	**num**	**i**	++	<	>
temp'	0	0					
temp	1	2					
expr			3	3	3		
value			9	8	7		
compar						4	5
plus					10	11	
var				12			

LL(1)

Пример: $while\ (\ i < num)\ print\ (i++);$

Стек	**Вход**	**Выход**
$ temp’	while (i < num) print (i++);$	*temp’ = temp (0)*
$ temp	while (i < num) print (i++);$	*temp* :: = **while (** *expr* **)** *temp (1)*
$ temp) expr (while	while (i < num) print (i++);$	Съели while
$ temp) expr (	(i < num) print (i++);$	Съели (
$ temp) expr	i < num) print (i++);$	*expr* ::= *param compar (3)*
$ temp) compar param	i < num) print (i++);$	*param* ::= *var plus (8)*
$ temp) compar plus var	i < num) print (i++);$	*var* ::= **i** (12)
$ temp) compar plus i	i < num) print (i++);$	Съели i
$ temp) compar plus	< num) print (i++);$	*plus ::=* ε (11)
$ temp) compar	< num) print (i++);$	*compar* ::= **<** *param (4)*
$ temp) param <	< num) print (i++);$	Съели <
$ temp) param	num) print (i++);$	*param* ::= **num** (9)
$ temp) num	num) print (i++);$	Съели num
$ temp)	) print (i++);$	Съели)
$ temp	print (i++);$	*temp* ::= **print (** *param* **);** (2)
$;) param (print	print (i++);$	Съели print
$;) param (	(i++);$	Съели (
$;) param	i++);$	*param* ::= *var plus (8)*
$;) plus var	i++);$	*var* ::= **i** (12)
$;) plus i	i++);$	Съели i
$;) plus	++);$	*plus* ::= **++** (++)
$;) ++	++);$	Съели ++
$;)	);$	Съели)
$;	;$	Съели ;
$	$	Съели $
	Access	