

Построение синтаксического анализатора на основании LR грамматики

Порядок выполнения

1. Построить грамматику с предшествованием
2. Определить отношения предшествования
3. Изобразить восходящую схему разбора
4. Выполнить свертку заданного примера
5. Разработать синтаксический анализатор на базе yacc
6. Разработать лексический анализатор на базе lex
7. Выполнить синтаксический анализ заданного примера

Состав отчета

- Титульный лист (фамилия, группа, номер варианта, наименование работы, задание)

Варианты заданий В задании указано содержательное описание грамматики и простейший пример для облегчения понимания.

1. конструкция for языка c++.

```
for (
    a=0, b=76;
    i<10 && b>0 ;
    ++i , b=i+1
){
    b++;
    i++;
}
```

Listing 1: for

2. конструкция if языка c++.

```
if ( a==0 && b<=0)
{
    a=b; b++;
}
else
{
```

```
a++;  
b+=2;  
}
```

Listing 2: if

3. конструкция switch языка c++.

```
switch(a){  
    case 1: a=1;  
    case 3:  
    case 2: a++; break;  
    default: a=0;  
}
```

Listing 3: switch

4. тэг img языка разметки HTML:

```

```

Listing 4: img

5. Объявление функции в c++:

```
int funcname(int a, char b, float c = 0.1);
```

Listing 5: func

6. тэг table языка разметки HTML:

```
<table>  
    <tr>  
        <th>заголовок 1</th>  
        <th>заголовок 2</th>  
    </tr>  
    <tr>  
        <td>ячейка 1</td>  
        <td>ячейка 2</td>  
    </tr>  
    <tr>  
        <td>ячейка 3</td>  
        <td>ячейка 4</td>  
    </tr>  
</table>
```

Listing 6: table

7. тэг ul (нечислованный список) языка разметки HTML:

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ul>
```

Listing 7: ul

8. конструкция while языка c++.

```
while( a>0 || b<76){
    b++;
    a++;
}
```

Listing 8: while

9. конструкция do-while языка c++.

```
do{
    b++;
    a++;
}while( a>0 || b<76);
```

Listing 9: do-while

10. конструкция insert языка запросов SQL:

```
insert into tablename(pk,column1,column2)
values (1, 2, 'varchar');
```

Listing 10: insert

11. конструкция select языка запросов SQL:

```
select pk, column1, column2
from tablename
where column1 = 2
and column2 like 'xxx%';
```

Listing 11: select

12. конструкция delete языка запросов SQL:

```
delete from tablename
where column1 = 1
or column2 like 'xxx%';
```

Listing 12: delete

13. тэг form языка разметки HTML:

```
<form method="post" action="/registration/">
  <input type="text" name="login" />
  <input type="text" name="email" />
  <input type="password" name="password" />
  <input type="submit" />
</form>
```

Listing 13: form

14. конструкция if языка Pascal:

```
if (a <= 10) or (b >= 2) then
begin
  a := 10;
  b := 10;
end else
  a := b;
```

Listing 14: if

15. конструкция for языка Pascal:

```
for i:= 0 to 10 step 2 do
begin
  a := i;
end;
```

Listing 15: for

Пример реализации Построим грамматику

$S \rightarrow sXfX$

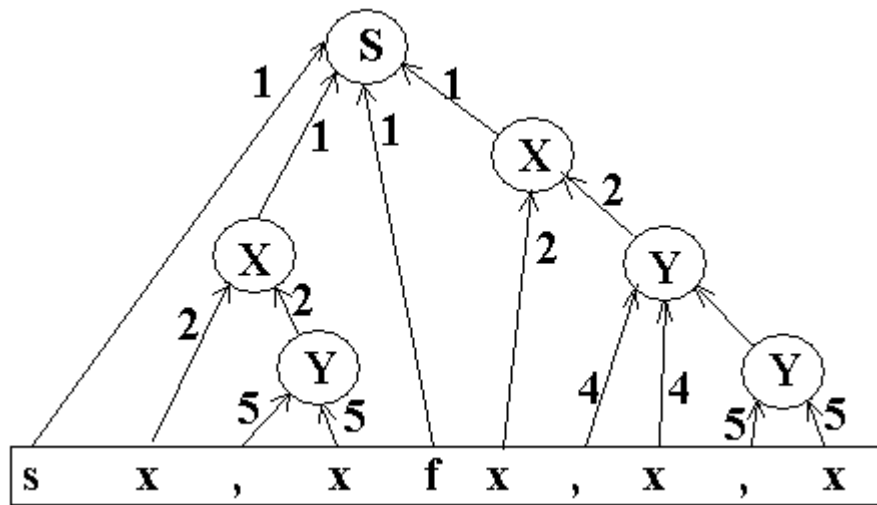
$X \rightarrow xY$

$X \rightarrow x$

$Y \rightarrow ,xY$

$Y \rightarrow ,x$

	s	X	Y	x	f	$,$
s		=		<		
X					=	
Y					>	
x			=		>	<
f		=		<		
$,$				=		



Обозначим условно ^ как начало, а \$ как конец строки.
 $sx,xfx \Rightarrow$
 $\langle s \langle x \langle , \doteq x \rangle f \langle x \rangle \Rightarrow$
 $\langle s \langle x \doteq Y \rangle f \doteq X \rangle \Rightarrow$
 $\langle s \langle X \doteq f \doteq X \rangle \Rightarrow \langle S \rangle$

```

%%
x      { return x; }
f      { return f; }
,      { return comma; }
.      { return yytext; }
%%
  
```

Listing 16: lab4.l

```

%start start

%token f x comma

%%

start:  X f X { printf("%s\n", "OK"); }
        ;
X :     x
        | x Y
        ;
Y :     comma x
        | comma x Y
        ;
  
```

```
%%  
/* start of programs */  
#include "lex.yy.c"  
  
main() { return yyparse();}  
  
yyerror(char *s) { fprintf(stderr,"%s\n",s); }
```

Listing 17: lab4.y

Выполняем в shell следующие команды:

```
flex lab4.l  
yacc lab4.y  
gcc -o lab4 y.tab.c -ll  
./lab4
```