Методы трансляции Лаб.раб. Кузнецов Д.Б.

Лабораторная 1

Построение лексического анализатора на основании автоматной грамматики

Состав работ

1. Построить автоматную грамматику
2. Построить автомат
3. Привести к детерминированному автомату
4. Реализовать автомат программно на языке Си
5. Написать отчет, содержащий результаты выполнения п. п. 1-4

Варианты

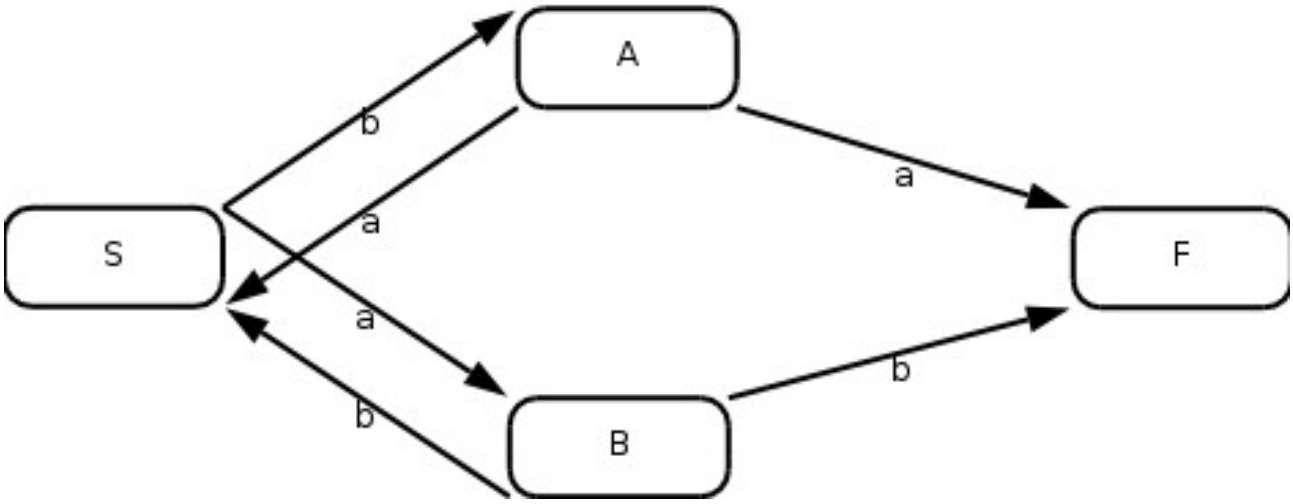
N	Содержательное описание грамматики	Примеры
1	Символы a,b стоят парами, порядок в паре не важен	ababbaabbaba
2	Одинаковые символы стоят парами	aabbaabbbbaa
3	В начале строки a, далее - вперемешку	ababbbabb
4	Первый символ — не важен, далее одни a	baaaaaaaaa aaaaaaaaaa
5	Либо одни a, либо одни b	aaaaaaaaaa bbbbbbbbb
6	В конце строки b, начало — не важно	ababbbabb
7	Одинаковые символы не должны стоять рядом	abababababab bababababa
8	В строке должна встретиться хотя бы одна буква a	bbbbbabbb aaaaaaaaaa
9	Предпоследним символом строки должна быть b, остальные — не важно	abbabaabb
10	Вторым символом строки должна быть a, остальные — не важно	baaaaabbb

Пример выполнения

1. Грамматика

S → aB
S → bA
B → bF
A → aF
B → bS
A → aS
F — конечное состояние

2. Автомат



	-	-	-	+
	S	A	B	F
a	B	S,F		
b	A		S,F	

3. Детерминированный автомат

	-	-	-	+	-	+
	S	A	B	F	{}	{S,F}
a	B	{S,F}	{}	{}	{}	B
b	A	{}	{S,F}	{}	{}	A

4. Программа

```
#define S  0
#define A  1
#define B  2
#define F  3
#define H  4 /* {S,F} */
#define W  5 /* {} */

#define P  1 /* + */
#define M  0 /* - */

#define _a 0 /* a */
#define _b 1 /* b */

struct action
{
 int sos; /* sostoyanie */
```

```
 int out; /* output */
}


main()
{
 int table[2][6];
 table[_a][S] = B;
 table[_b][S] = A;
 table[_a][A] = H;
 table[_b][A] = W;
 table[_a][B] = W;
 table[_b][B] = H;
 table[_a][F] = W;
 table[_b][F] = W;
 table[_a][W] = W;
 table[_b][W] = W;
 table[_a][H] = B;
 table[_b][H] = A;

 int output[6];
 output[S] = M;
 output[A] = M;
 output[B] = M;
 output[F] = P;
 output[W] = M;
 output[H] = P;


 int input[]={_b,_a,_a,_b};
 int n = 4, i;
 int isos = S;

 for( i=0; i<n; i++)
 {
  isos=table[input[i]][isos];
 }

 printf("%d\n", output[isos]);
}
```

Контрольные вопросы

1. Что такое автомат?
2. Какие автоматы называются детерминированными?
3. Какие признаки у автоматной грамматики?
4. Какой автомат строится по автоматной грамматике, Милли или Мура? Обоснуйте.
5. Что это?
6. Какой элемент автомата реализует массив table?
7. Из каких элементов состоит автоматов?

Лабораторная 2

Построение лексического анализатора с использованием lex

Состав работ

1. Построить регулярное выражение
2. Составить файл для lex
3. Получить программу на Си
4. Откомпилировать и запустить
5. Написать отчет, содержащий результаты выполнения п. п. 1-2, результат выполнения п. 4

Варианты

N	Содержательное описание грамматики	Примеры
1	Символы a,b стоят парами, порядок в паре не важен	ababbaabbaba
2	Одинаковые символы стоят парами	aabbaabbbbaa
3	В начале строки a, далее - вперемешку	ababbbabb
4	Первый символ — не важен, далее одни a	baaaaaaaaa aaaaaaaaaa
5	Либо одни a, либо одни b	aaaaaaaaaa bbbbbbbbb
6	В конце строки b, начало — не важно	ababbbabb
7	Одинаковые символы не должны стоять рядом	abababababab bababababa
8	В строке должна встретиться хотя бы одна буква a	bbbbbabbb aaaaaaaaaa
9	Предпоследним символом строки должна быть b, остальные — не важно	abbabaabb
10	Вторым символом строки должна быть a, остальные — не важно	baaaaabbb

Пример выполнения

1. Регулярное выражение

(ab|ba)+

2. Файл для lex

```
%%
(ab|ba)+ { printf(«+»);}
```

```
.* { printf(«-»); }
%%
yyerror(char *str)
{ printf(str); }
main()
{ yylex(); }
```

3. Получение программы на Си

```
flex -o l2.c l2.l
```

l2.c – имя выходного файла

l2.l – имя входного файла

4. Компиляция и запуск

```
cc -o l2 l2.c -lfl
./l2
ababbababa
+
bbbaa
-
```

Контрольные вопросы

1. Для какой грамматики применимы регулярные выражения?
2. Как в регулярным выражением задать
   - одиночный символ
   - повторение предшествующего символа
   - альтернативу
   - диапазон символов
   - число повторений предшествующего символа
3. Какие разделы могут быть в lex-файле?
4. Что делает lex?
5. Что это?

Лабораторная 3

Построение синтаксического анализатора на основании LL(1) грамматики

Состав работ

```
1. Построить LL(1) грамматику
2. Определить множества выбора для каждого правила
3. Изобразить низходящую схему разбора
4. Разработать лексический анализатор на базе lex
5. Построить МП-автомат по грамматике
6. Реализовать МП-автомат на Си
7. Реализовать синтаксический анализ методом рекурсивного спуска
```

Варианты

```
1. for
for(i=0,kjfg=76;i>hyht&&ti>tth7;i++,lf--);

2. if
if(d<65)
{
o;
o;
o;
}
elseif(ueh>6)
{
o;
o;
}
elseif(ffg=rt)
{
o;
o;
o;
o;
}
else
{
o;
}

3. switch
switch(jj)
{
case 77: o;o;o;
case 2:
case 'u': o;o;o;
default: o;o;o;
}

4. url
jjjt://rhh.drthdrh.drhttrh.rdhrdth.rt
```

5. tag
```
<img src="/images/a.gif" width="66" fgfgh="ghfgh">
```

6. pro func
```
int funct(int trhgm, char gdg, float rr);
```

7. kom shell
```
cp -r -t -g  -i -- dg/dfg/fdg dfg/dfg/fg fdg/dfg/dfg
```

8. html
```
<html>
<head>
<title>
df dfgdfg dfg
</title>
</head>
<body>
gggegre eg ergeej
</body>
</html>
```

9. table
```
<table>
<tr>
<td>xxfg gf  fxx</td>
<td>xxx uu</td>
<td>adx xg xhxxx</td>
</tr>
<tr>
<td>fdfd</td>
</tr>
</table>
```

10. li
```
<ul>
<li>xkl 8y   jkh xxx
<li>xxx jhg78    h  g
<li>xxxx xxxx xxx
</ul>
```

11. while
```
while(x<89)
{
o;
o;
o;
}
```

12. insert
```
insert into mytab values (97,76786,'xkjhhjg')
```

13. create table
```
create table mytab (int xhg, int uuux, float hxu);
```

14. :=
```
x1=gx+877+jx-kx+gx-ygx-xhh;
```

15. cc
```
cc -o file file.c ytdfd.c hgvhjf.c -lxj -lgjx -ljx
```

Контрольные вопросы

1. Какое место занимает синтаксический анализатор в компиляторе?
2. Какие грамматики применимы для синтаксического анализа?
3. Для каких грамматик применим МП-автомат?
4. Для каких грамматик применим метод рекурсивного спуска?
5. Какие ограничения на правила накладывают LL(1) грамматики?
6. Как определить множество выбора для правила LL(1) грамматики?
7. Чем МП-автомат отличается от автомата Мура?
8. Каким должен быть МП-автомат для работы с LL(1) грамматикой?
9. Какие символы становятся стековыми в МП-автомат для работы с LL(1) грамматикой?
10. Каков алгоритм работы МП-автомата?

Лабораторная 4

Построение синтаксического анализатора на основании LR грамматики

Состав работ

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

Варианты

1. for
```
for(i=0,kjfg=76;i>hyht&&ti>tth7;i++,lf--);
```

2. if
```
if(d<65)
{
o;
o;
o;
}
elseif(ueh>6)
{
o;
o;
}
elseif(ffg=rt)
{
o;
o;
o;
o;
}
else
{
o;
}
```

3. switch
```
switch(jj)
{
case 77: o;o;o;
case 2:
case 'u': o;o;o;
default: o;o;o;
}
```

4. url
```
jjjt://rhh.drthdrh.drhttrh.rdhrdth.rt
```

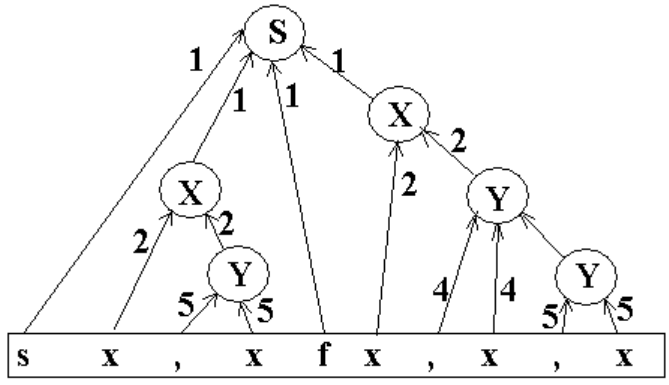
```
5. tag
<img src="/images/a.gif" width="66" fgfgh="ghfgh">

6. pro func
int funct(int trhgm, char gdg, float rr);

7. kom shell
cp -r -t -g  -i -- dg/dfg/fdg dfg/dfg/fg fdg/dfg/dfg

8. html
<html>
<head>
<title>
df dfgdfg dfg
</title>
</head>
<body>
gggegre eg ergeej
</body>
</html>

9. table
<table>
<tr>
<td>xxfg gf  fxx</td>
<td>xxx uu</td>
<td>adx xg xhxxx</td>
</tr>
<tr>
<td>fdfd</td>
</tr>
</table>

10. li
<ul>
<li>xkl 8y   jkh xxx
<li>xxx jhg78    h  g
<li>xxxx xxxx xxx
</ul>

11. while
while(x<89)
{
o;
o;
o;
}

12. insert
insert into mytab values (97,76786,'xkjhhjg')
13. create table
create table mytab (int xhg, int uuux, float hxu);
14. :=
x1=gx+877+jx-kx+gx-ygx-xhh;
15. cc
cc -o file file.c ytdfd.c hgvhjf.c -lxj -lgjx -ljx
```

Пример выполнения

1) $S \rightarrow sXfX$
2) $X \rightarrow xY$

3) X → x
4) Y → ,xY
5) Y → ,x

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



^ s x , x f x $ (^ - условно начало строки, $ - конец) расставим знаки в цепочке символов.
=> ^ < s < x < , ≐x > f < x > $ =>
=> ^ < s < x ≐Y > f ≐X > $ => ^ < s ≐X ≐f ≐X > $ = > ^ < S > $

##########prim1.y#################
%token S X F Z
%%
es: S iks F iks {printf(“OK!”);} //первое правило
;
iks:X
| X igrek
;
igrek: Z X
| Z X igrek
;
%%
yyerror() { printf(" Ошибка! "); }
main() {
yyparse(); } //функция, которая получается из раздела правил

```
#####################################

    ########prim1.l#################
    %{
    #include "prim1_y.h"       //там лексемы будут определены как макросы
    %}
    %%
    s               { return( S ); }
    x               { return( X ); }
    f               { return( F ); }
    [,]             { return( Z ); }
    .               { return( yytext[0] ); }
    %%
    ############################
```

Контрольные вопросы

1. Какое место занимает синтаксический анализатор в компиляторе?
2. Какие грамматики применимы для синтаксического анализа?
3. Что такое грамматика с предшествованием?
4. Из каких разделов состоит программа на уасс?
5. Для какого типа грамматики используют программы на уасс?
6. Приведите пример грамматики, не являющейся грамматикой с предшествованием
7. В каких случаях в грамматике с предшествованием между символами ставится отношение >?
8. В каких случаях в грамматике с предшествованием между символами ставится отношение <?
9. В каких случаях в грамматике с предшествованием между символами ставится отношение =?