

实验 非递归预测分析

一、实验目的与任务

设计一个非递归预测分析器，实现对表达式语言的分析，理解自上而下语法分析方法的基本思想，掌握设计非递归预测分析器的基本方法。

二、实验要求

建立文法及其 LL(1)分析表表示的数据结构，设计并实现相应的预测分析器，对源程序经词法分析后生成的二式代码流进行预测分析，如果输入串是文法定义的句子则输出“是”，否则输出“否”。

三、文法描述及其 LL(1)分析表

表达式语言(XL) 的语法规则如下：

1. 程序 \rightarrow 表达式；
2. |表达式； 程序
3. 表达式 \rightarrow 表达式 + 项
4. |项
5. 项 \rightarrow 项 * 因式
6. |因式
7. 因式 \rightarrow num_or_id
8. |(表达式)

将该语言的文法转换为如下的 LL(1)文法：

- | | | | |
|---|---------------------------------|----|-----------------------------------|
| 1 | prgm \rightarrow expr;prgm' | 8 | term \rightarrow factor term' |
| 2 | prgm' \rightarrow prgm | 9 | term' \rightarrow *factor term' |
| 3 | prgm' $\rightarrow \epsilon$ | 10 | term' $\rightarrow \epsilon$ |
| 4 | expr \rightarrow term expr' | 11 | factor \rightarrow (expr) |
| 5 | expr $\rightarrow \epsilon$ | 12 | factor \rightarrow num |
| 6 | expr' \rightarrow +term expr' | 13 | system_goal \rightarrow prgm |
| 7 | expr' $\rightarrow \epsilon$ | | |

该 LL(1)文法的 LL(1)分析表如下：

T							
N	Num	+	*	()	;	#
prgm	1			1		1	
prgm'	2			2		2	3
expr	4			4	5	5	
expr'		6			7	7	
term	8			8			
term'		10	9		10	10	
factor	12			11			
system_goal	13			13		13	

对文法中每个文法符号指定一个常数值，符号编码表如下：

文法符号	常数值	备注
(4	终结符 (#为输入结束标志)
Num	6	
+	2	
)	5	
;	1	
*	3	
#	0	
Expr	258	非终结符
expr'	260	
term	259	
term'	262	
factor	261	
prgm	256	
prgm'	257	
system_goal	263	

四、 文法及其 LL(1)分析表的数据结构

文法的产生式可用数组 `Yy_pushtab[]` 存放。数组的第一个下标是产生式号，第一个产生式的序号为 0；每列按逆序存放该产生式右部各符号的常数值，并以 0 结束。对于该表达式语言 XL 的 LL(1)分析表，可用数组 `Yy_d[]` 存放。第一个下标是非终结符数值，第二个下标是终结符数值，数组元素的值为：0(表示接受)，1(表示产生式号)，-1(表示语法错)。

数组 `Yy_d[]` 的具体内容及表示如下：

0	1	2	3	4	5	6
#	:	+	*	()	Num
-1	0	-1	-1	0	-1	0
2	1	-1	-1	1	-1	1
-1	4	-1	-1	3	4	3
-1	-1	-1	-1	7	-1	7
-1	6	5	-1	-1	6	-1
-1	-1	-1	-1	10	-1	11
-1	9	9	8	-1	9	-1
-1	12	-1	-1	12	-1	12

prgm 256
 prgm' 257
 expr 258
 term 259
 expr' 260
 factor 261
 term' 262
 system_goal 263

数组 Yy_pushtab[]的具体内容及表示如下：

0	257, 1, 258, 0 <u>prgm</u> ; <u>expr</u>	Yyp00
1	256, 0 <u>prgm</u>	Yyp01
2	0	Yyp02
3	260, 259, 0 <u>expr</u> term	Yyp03
4	0	Yyp04
5	260, 259, 2, 0 <u>expr</u> term +	Yyp05
6	0	Yyp06
7	262, 261, 0 term factor	Yyp07
8	262, 261, 2, 0 term factor *	Yyp08
9	0	Yyp09
10	5, 258, 4, 0) <u>expr</u> (Yyp10
11	6, 0 <u>Num</u>	Yyp11
12	256, 0 <u>prgm</u>	Yyp12

五、预测分析器总控程序结构

预测分析器总控程序使用上面的两个表 Yy_pushtab、Yy_d 和一个分析栈(元素类型为 int) ， 其结构如下：

初始化; /* 把开始符号的常数值压入分析站, 输入指向第一个输入符号 */

```
while(分析栈非空) {
    if(栈顶常数表示一个终结符)
        if(该常数与输入符号的常数不等)
            报语法错;
        else {
            把一个数从栈顶弹出;
            advance 读下一输入符号;
        }
    else { /* 栈顶的常数表示一个非终结符 */
        what_to_do=Yy_d[栈顶常数][当前输入符号的常数];
```

```

        if(what_to_do == -1)
            报语法错;
        else {
            把栈顶元素弹出栈;
            把 Yy_pushtab[what_to_do]中列出的全部常数压入分析栈;
        }
    }
}

```

请实现该程序。在程序中添加输出栈内容的功能，以便和手工模拟分析过程作比较。

六、预测分析控制程序的测试用例

用预测分析器和手工模拟两种方式对文法的句子 1+2； 进行分析。综合分析过程可用下表表示。

栈(符号)	栈(数值)	输入串	What_to_do
system_goal	263	1+2; #	12
prgm	256	1+2; #	0
prgm'; expr	257 1 258	1+2; #	3
prgm'; expr' term	257 1 260 259	1+2; #	7
prgm' ; expr' term'	257 1 260 262 261	1+2; #	11
factor	257 1 260 262 6	1+2; #	
prgm' ; expr' term'	257 1 260 262	+2; #	9
Num	257 1 260	+2; #	5
prgm'; expr' term'	257 1 260 259 2	+2; #	
prgm'; expr'	257 1 260 259	2; #	7
prgm'; expr' term +	257 1 260 262 261	2; #	11
prgm'; expr' term	257 1 260 262 6	2; #	
prgm' ; expr' term'	257 1 260 262	; #	9
factor	257 1 260	; #	6
prgm' ; expr' term'	257 1 260 262	; #	
Num	257	#	2
prgm'; expr' term'			
prgm'; expr'			
prgm';			
prgm'			

思考：请考虑如何设计并实现 LL(1)分析表的自动生成程序。