

HW5

肖桐 PB18000037

2020 年 10 月 24 日

解 1. 消除原文法的左递归后得到增广文法如下:

$$S' \rightarrow S$$

$$S \rightarrow (L) \mid a$$

$$L \rightarrow S, L \mid S$$

为非终结符号 L, S 增加一个综合属性 syn 和继承属性 inh , 为终结符 a 增加一个综合属性 syn .
其中属性 syn 表示由当前文法符号推出的字符串的最后一位字符, 所在当前句子中的位置.
属性 inh 表示由当前文法符号推出的字符串的第一个字符, 所在当前句子中的位置.
则可以得到以下语法制导定义:

表 1: 语法制导定义

产生式	语法规则
$S' \rightarrow S$	$S.inh = 1$ $S'.syn = S.syn$
$S \rightarrow (L)$	$L.inh = S.inh + 1$ $S.syn = L.syn + 1$
$S \rightarrow a$	$S.syn = a.syn$
$L \rightarrow S, L_1$	$L.syn = L_1.syn$ $S.inh = L.inh$ $L_1.inh = S.syn + 2$
$L \rightarrow S$	$L.syn = S.syn$ $S.inh = L.inh$

对应的翻译方案为: 初始化开始符号 $S.inh = 1$.

表 2: 翻译方案

$S' \rightarrow \{S.inh = 1; \} S \{S'.syn = S.syn; \}$
$S \rightarrow (\{L.inh = S.inh + 1; \} L \{S.syn = L.syn + 1; \})$
$S \rightarrow a \{S.syn = a.syn; \text{printf}(S.inh); \}$
$L \rightarrow \{S.inh = L.inh; \} S, \{L_1.inh = S.syn + 2; \} L_1 \{L.syn = L_1.syn; \}$
$L \rightarrow \{S.inh = L.inh; \} S \{L.syn = S.syn; \}$

预测翻译器:

```

1  int S'()
2  {
3      int S'.syn = S(1);
4      if (lookahead == '$')
5      {
6          return 1;    //accept
7      }
8      else
9      {
10         return 0;    //error
11     }
12 }
13
14 int S(int S.inh)
15 {
16     if (lookahead == 'a')
17     { //产生式 S -> a
18         match('a');
19         printf(a.syn); //S.inh = a.syn
20         return a.syn;
21     }
22     else if (lookahead == '(')
23     { //产生式 S -> (L)
24         match('(');
25         int S.syn = L(S.inh + 1) + 1;
26         match(')');
27         return S.syn;
28     }
29     else return 0; //error
30 }
31
32 int L(int L.inh)
33 {
34     int S.syn = S(L.inh)
35     if (lookahead == ',')
36     { //产生式 L -> S, L
37         match(',');
38         int L.syn = L(S.syn + 2);
39         return L.syn;
40     }
41     else
42     { //产生式 L -> S

```

```
43     return S.syn;  
44 }  
45 }
```