

Homework 5

PB17000297 罗晏宸

October 15 2019

1 Exercise 4.12

文法如下：

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid S \end{aligned}$$

(a) 写一个翻译方案，它输出每个 a 的嵌套深度。例如，对于句子 $(a, (a, a))$ ，输出的结果是 1 2 2。

(b) 写一个翻译方案，它打印出每个 a 在句子中是第几个字符。例如，当句子是 $(a, (a, (a, a), (a)))$ 时，打印的结果是 2 5 8 10 14。

解

(a) 用继承属性 **depth** 表示嵌套深度，则翻译方案如下：

$$\begin{array}{ll}
 S' \rightarrow & \{ S.\text{depth} = 0; \} \\
 & S \\
 S \rightarrow & \{ L.\text{depth} = S.\text{depth} + 1; \} \\
 & (L) \\
 S \rightarrow a & \{ \text{print}(S.\text{depth}); \} \\
 L \rightarrow & \{ L_1.\text{depth} = L.\text{depth}; \} \\
 & L_1, \\
 & S \\
 L \rightarrow & \{ S.\text{depth} = L.\text{depth}; \} \\
 & S
 \end{array}$$

(b) 用继承属性 **before** 表示句子中在文法符号前的字符，用综合属性 **out** 表示文法符号推出的字符总数，则翻译方案如下：

$$\begin{array}{ll}
 S' \rightarrow & \{ S.\text{before} = 0; \} \\
 & S \\
 S \rightarrow & \{ L.\text{before} = S.\text{before} + 1; \} \\
 & (L) \\
 S \rightarrow a & \{ S.\text{out} = 1; \text{print}(S.\text{before} + 1); \} \\
 L \rightarrow & \{ L_1.\text{before} = L.\text{before}; \} \\
 & L_1, \\
 & S \\
 L \rightarrow & \{ S.\text{before} = L.\text{before}; \} \\
 & S
 \end{array}$$

2 Non-textbook Exercise

(1)

给出习题 4.12(a) 和 (b) 的翻译方案所对应的属性栈代码。

解 对应的属性栈代码如下：

$S' \rightarrow$ { $S.depth = 0;$ }
 S
 $S \rightarrow$ { $L.depth = S.depth + 1;$ }
 (L)
 $S \rightarrow a$ { $print(S.depth);$ }
 $L \rightarrow$ { $L_1.depth = L.depth;$ }
 $L_1,$ { $S.depth = L.depth;$ }
 S
 $L \rightarrow$ { $S.depth = L.depth;$ }
 S

```

stack[top].depth = 0;

stack[top - 1].depth = stack[top - 2].depth + 1;

print(stack[top].depth);
// stack[top - 2].depth = stack[top - 2].depth;
stack[top].depth = stack[top - 2].depth;

// stack[top].depth = stack[top].depth;

```

$S' \rightarrow$ { $S.before = 0;$ }
 S
 $S \rightarrow$ { $L.before = S.before + 1;$ }
 (L) { $S.out = L.out + 2;$ }
 $S \rightarrow a$ { $S.out = 1; print(S.before + 1);$ }
 $L \rightarrow$ { $L_1.before = L.before;$ }
 $L_1,$ { $S.before = L.before + L_1.out + 1;$ }
 S { $L.out = L_1.out + S.out + 1;$ }
 $L \rightarrow$ { $S.before = L.before;$ }
 S { $L.out = S.out;$ }

```

stack[top].depth = 0;

stack[top - 1].before = stack[top - 2].before + 1;

stack[top - 2].out = stack[top - 1].out + 2;

stack[top].out = 1; print(stack[top].before + 1);

// stack[top - 2].before = stack[top - 2].before;

stack[top].before = stack[top - 2].before + stack[top - 2].out + 1;

stack[top - 2].out = stack[top - 2].out + stack[top].out + 1;

// stack[top].before = stack[top].before;

// stack[top].out = stack[top].out;

```

(2)

给出习题 4.12(a) 和 (b) 的翻译方案所对应的 YACC 语义代码。

解

(a) 翻译方案对应的语义代码如下：

```
1 Start      :      { $1.depth = 0; }
2           S
3           ;
4 S          :      { $2.depth = $.depth + 1; }
5           ' ( ' L ' ) '
6           ;
7 S          : a      { printf("%d\n", $.depth); }
8           ;
9 L          :      { $1.depth = $.depth; }
10          L_1 ' , ' { $3.depth = $.depth; }
11          S
12          ;
13 L_1       :      { $1.depth = $.depth; }
14          S
15          ;
```

(b) 翻译方案对应的语义代码如下：

```
1 Start      :      { $1.before = 0; }
2           S
3           ;
4 S          :      { $2.before = $.before + 1; }
5           ' ( ' L ' ) ' { $.out = $2.out + 2; }
6           ;
7 S          : a      {
8                       $.out = 1;
9                       printf("%d\n", $.before + 1);
10                      }
```

```

11      ;
12 L      :      { $1.before = $$ before; }
13      L_1 ' , ' { $$ before = $$ before + $1.out + 1; }
14      S      { $$ out = $1.out + $3.out + 1; }
15      ;
16 L_1     :      { $1.before = $$ before; }
17      S      { $$ out = $1.out; }
18      ;

```

(3)

针对以下文法

$$\begin{aligned}
 E &\rightarrow E \text{ '}' E \\
 &| E \text{ '<' } E \\
 &| \text{ number}
 \end{aligned}$$

设计语法制导定义，使之能计算诸如 $1 < 2 < 3$ 的表达式值为 True；而计算表达式 $1 < 5 > 3$ 的值也为 True。

解 用综合属性 *left* 和 *right* 表示 *E* 推出的字符序列中最左和最右的数字，综合属性 *bool* 表示 *E* 推出的表达式的布尔值，语法制导的定义如下：

产生式	语义规则
$E' \rightarrow E$	$print(E.bool);$
$E \rightarrow E_1 \text{ '}' E_2$	$E.left = E_1.left; \quad E.right = E_2.right;$ $E_1.bool = (E_1.right > E_2.left) ? \text{true} : \text{false}$
$E \rightarrow E_1 \text{ '<' } E_2$	$E.left = E_1.left; \quad E.right = E_2.right;$ $E_1.bool = (E_1.right < E_2.left) ? \text{true} : \text{false}$
$E \rightarrow \text{ number}$	$E.left = \text{number.val}; \quad E.right = \text{number.val}; \quad E.bool = \text{true};$