

4.12 为文法

 $S \rightarrow (L) \mid a$ $L \rightarrow L_1 S \mid S$ (b) 写出自下而上的分析的栈操作代码, 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。

由作业15可知:

为文法符号 S 和 L 各一个继承属性 in 和一个综合属性 out , in 表示在句子中该文法符号推出的字符序列的前面已经有多少个字符; out 表示该文法符号推出的字符序列的最后一个字符在句子中是第几个字符。

语法制导定义:

 $S' \rightarrow S$ $S.in = 0$ // 初始化, 计数的起点 $S \rightarrow (L)$ $L.in = S.in + 1, S.out = L.out + 1$ $S \rightarrow a$ $S.out = S.in + 1, \text{print}(S.out)$ $L \rightarrow L_1 S$ $L.in = L_1.in, S.in = L_1.out + 1, L.out = S.out$ $L \rightarrow S$ $S.in = L.in, L.out = S.out$

翻译方案:

① $S' \rightarrow \{ S.in = 0 ; \}$ S ② $S \rightarrow \{ L.in = S.in + 1 ; \}$ $(L) \{ S.out = L.out + 1 ; \}$ ③ $S \rightarrow a \{ S.out = S.in + 1 ; \text{print}(S.out) ; \}$ ④ $L \rightarrow \{ L_1.in = L.in ; \}$ $L_1, \{ S.in = L_1.out + 1 ; \}$ $S \{ L.out = S.out ; \}$ ⑤ $L \rightarrow \{ S.in = L.in ; \}$ $S \{ L.out = S.out ; \}$

对语法制导定义加入标记非终结符 Q, M, N , 使所有继承属性均由复写规则设置, 且可静态确定其在栈中的位置. (i 为继承属性, s 为综合属性)

产生式	语义规则
$S' \rightarrow QS$	$S.in = Q.S$
$Q \rightarrow \varepsilon$	<u>$Q.S = 0$</u> — 一切的源头

$S \rightarrow (ML)$	$M.i = S.in, L.in = M.S, S.out = L.out + 1$
$M \rightarrow \varepsilon$	$M.S = M.i + 1$

$S \rightarrow a$	$S.out = S.in + 1, \text{print}(S.out)$
-------------------	---

$L \rightarrow L_1 N S$	$L.in = L.in, N.i = L_1.out, S.in = N.S, L.out = S.out$
$N \rightarrow \varepsilon$	$N.S = N.i + 1$

$L \rightarrow S$	$S.in = L.in, L.out = S.out$
-------------------	------------------------------

由上述语法制导定义写出自下而上分析的栈操作代码.

产生式	语义规则
-----	------

$S' \rightarrow QS$	
---------------------	--

$Q \rightarrow \varepsilon$	$\text{stack}[\text{top}+1].val = 0;$
-----------------------------	---------------------------------------

$S \rightarrow (ML)$	$\text{stack}[\text{top}-3].val = \text{stack}[\text{top}-1].val + 1;$
----------------------	--

$M \rightarrow \varepsilon$	$\text{stack}[\text{top}+1].val = \text{stack}[\text{top}-1].val + 1;$
-----------------------------	--

$S \rightarrow a$	$\text{stack}[\text{top}].val = \text{stack}[\text{top}-1].val + 1; \text{print}(\text{stack}[\text{top}].val)$
-------------------	---

$L \rightarrow L_1 N S$	$\text{stack}[\text{top}-3].val = \text{stack}[\text{top}].val;$
-------------------------	--

$N \rightarrow \varepsilon$	$\text{stack}[\text{top}+1].val = \text{stack}[\text{top}-1].val + 1;$
-----------------------------	--

$L \rightarrow S$	
-------------------	--

自上而下依次写出每个

stack 代码即可, 同样要写不想递归的中间码

4.14 程序的文法如下:

eg. 语句 $a:T; \text{proc } b; c:T; S$ 时
 a 和 b 嵌套深度是 1, c 是 2

$P \rightarrow D$

$D \rightarrow D; D \mid \text{id} : T \mid \text{proc id}; D; S$

(a) 写一个语法制导定义, 打印该程序一共声明了多少个 id。

(b) 写一个翻译方案, 打印该程序每个变量 id 的嵌套深度。

(a) 用继承属性 i 表示 D 中声明的 id 的个数
语法规则
 $P \rightarrow D$ $\text{print}(D.i)$

$D \rightarrow D_1; D_2$ $D.i = D_1.i + D_2.i$

$D \rightarrow \text{id} : T$ $D.i = 1$

$D \rightarrow \text{proc id}; D_1; S$ $D.i = D_1.i + 1$

(b). 用继承属性 l 表示 D 中声明的变量 id 的嵌套深度。

① 语法制导定义:

语法规则
 $P \rightarrow D$ $D.l = 1$

$D \rightarrow D_1; D_2$ $D_1.l = D.l, D_2.l = D.l$

$D \rightarrow \text{id} : T$ $\text{print}(\text{id.name}, D.l)$

$D \rightarrow \text{proc id}; D_1; S$ $D_1.l = D.l + 1, \text{print}(\text{id.name}, D.l)$

② 翻译树

$P \rightarrow D \quad \{D.L = 1;\}$

$D \rightarrow \begin{matrix} D_1; \\ D_2 \end{matrix} \quad \begin{matrix} \{D_1.L = D.L;\} \\ \{D_2.L = D.L;\} \end{matrix}$

$D \rightarrow id:T \quad \{\text{print}(id.name, D.L);\}$

$D \rightarrow \text{proc } id; \quad \{D_1.L = D.L + 1; \text{print}(id.name, D.L);\}$
 $D_1; S$