编译原理与技术 H6

PB18111697 王章瀚

4.12

为文法

$$S \to (L)|a$$
$$L \to L, S|S$$

(b) 写出自上而下分析的栈操作代码, 它打印每个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

上一次作业中已经写出了消除了左递归的文法和翻译方案如下:

$Q \to S$ $\quad Q \to \{S.first = 1;\}S$

$S \to (L)$ $\quad S \to \{L.first = S.first;\}(L)\{S.last = L.last + 1;\}$

$S \to a$ $\quad S \to a\{S.last = S.first;\ print(S.first);\}$

$L \to SR$ $\quad L \to \{S.first = L.first\}S\{R.first = S.last + 1\}R\{L.last = R.last\}$

$R \to, SR_1$ $\quad R \to, \{S.first = R.first + 1;\}S\{R_1.first = S.last + 1;\}R_1\{R.last = R_1.last\}$

$R \to \epsilon$ $\quad R \to \{R.last = R.fisrt\}$

这样, 对于一个 LL(1) 文法来说, 只需要系统地引入非终结符即可在 LR 分析期间完成属性计算. 引入的符号及引入后对栈操作代码如下:

翻译方案	语义规则	操作代码
$Q \to AS$	$S.first = A.last$	$val[top-1] = val[top]$
$A \to \epsilon$	$A.last = 1$	$val[top] = 1$
$S \to (BL)$	$B.first = S.first + 1$ $L.first = B.last + 1$ $S.last = L.last + 1$	$val[top-3] = val[top-1] + 1$
$B \to \epsilon$	$B.last = B.first$	$val[top+1] = val[top-1] + 2$
$S \to a$	$S.last = S.first; print(S.first)$	$val[top] = val[top-1] + 1$
$L \to SR$	$S.first = L.first + 1$ $R.first = S.last + 1$ $L.last = R.last$	$val[top-1] = val[top]$
$R \to, DSR_1$	$D.first = R.first + 1$ $S.first = D.last + 1$ $R_1.first = S.last + 1$ $L.last = R_1.last$	$val[top-3] = val[top]$
$D \to \epsilon$	$D.last = L.first$	$val[top+1] = val[top-1] + 1$
$R \to \epsilon$	$R.last = R.first$	$val[top+1] = val[top]$

4.14

程序的文法如下:

$$P \to D$$
$$D \to D; D | id : T | proc\ id; D; S$$

(a). 写一个语法制导定义, 打印该程序一共声明了多少个 id.
即考虑文法 (这里的 T 和 S 就当成终结符了)

$$P \to D$$
$$D \to D; D$$
$$D \to id : T$$
$$D \to proc\ id; D; S$$

用 $count$ 表示直到当前符号包含的 id 个数. 则其语法制导定义如下:

产生式	语义规则
$P \to D$	$print(D.count)$
$D \to D_1; D_2$	$D.count = D_1.count + D_2.count$
$D \to id : T$	$D.count = 1$
$D \to proc\ id; D_1; S$	$D.count = 1 + D_1.count$

(b). 写一个翻译方案, 打印该程序每个变量 id 的嵌套深度. 例如, 当句型是 $a : T; proc\ b; ba : T; S$时, a和b的嵌套深度是 1, ba的嵌套深度是 2.
为了表示嵌套深度显然需要引入继承属性. 这里用 $depth$表示嵌套深度. 于是可以写出翻译方案如下:

$$P \to \{D.depth = 0;\}D$$
$$D \to \{D_1.depth = D.depth + 1;\}D_1; \{D_2.depth = D.depth + 1;\}D_2$$
$$D \to id : T\{print(D.depth);\}$$
$$D \to proc\ id\{print(D.depth);\}; \{D_1.depth = D.depth + 1;\}D_1; S$$