

编译原理 第四章 语法分析



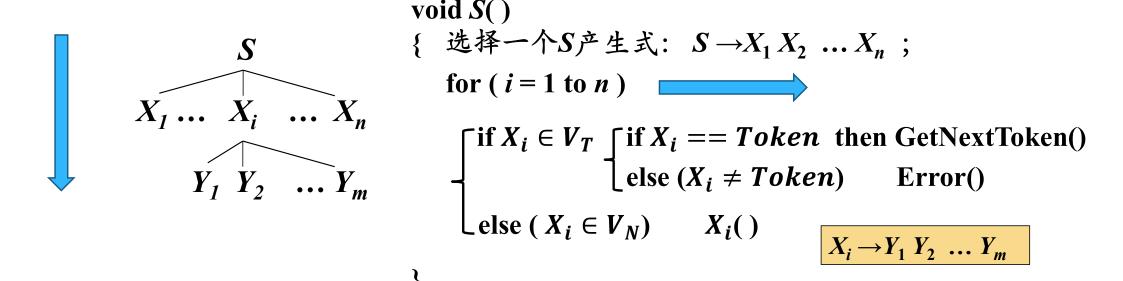
哈尔滨工业大学 陈鄞

第4讲(语法分析_1)要点

- ▶语法分析的任务: 识别句子, 确定句子的类型
- ▶对于不同类型的语句,后续的语义分析器将执行不同的语义动作
- ▶语法分析技术分类
 - ➤ top_down 前提: CFG无二义性
 - bottom_up

4.1top_down

▶通用形式: 递归下降分析 模拟了自顶向下建树过程, 最左推导



问题:假设 $A \rightarrow \alpha_1 |\alpha_2| \cdots |\alpha_n, 那么, 选哪一个?$ 需要回溯的分析(不确定的分析) 预测分析(确定的分析) 什么样的文法可以? LL(1)文法

 \triangleright SELECT($A \rightarrow \alpha$): 可以选用该产生式进行推导时对应的输入符号的集合 如何求?

 $\forall A \rightarrow \alpha_1 | \alpha_2 | \cdots | \alpha_n$ if $SELECT(A \rightarrow \alpha_1) \cap SELECT(A \rightarrow \alpha_2) \cap \cdots \cap SELECT(A \rightarrow \alpha_n) == \emptyset$ then 该文法称为LL(1)文法

 \triangleright SELECT($A \rightarrow \alpha$): 可以选用该产生式进行推导时对应的输入符号的集合 终结符集合

FOLLOW(A) $FIRST(\alpha)$

$$\alpha = X_1 X_2 \cdots X_k$$

ightharpoonup FIRST(lpha): 首终结符集。lpha能够推出的所有终结符串中位于串首的那些终结符构成的集合

 $= FIRST(X_1) \cup FIRST(X_2) \cup FIRST(X_3) \cup \cdots$

if
$$X_1 \Rightarrow^* \varepsilon$$
 if $X_2 \Rightarrow^* \varepsilon$

$$\alpha \Rightarrow^* \varepsilon \qquad \qquad \varepsilon \in FIRST(\alpha)$$

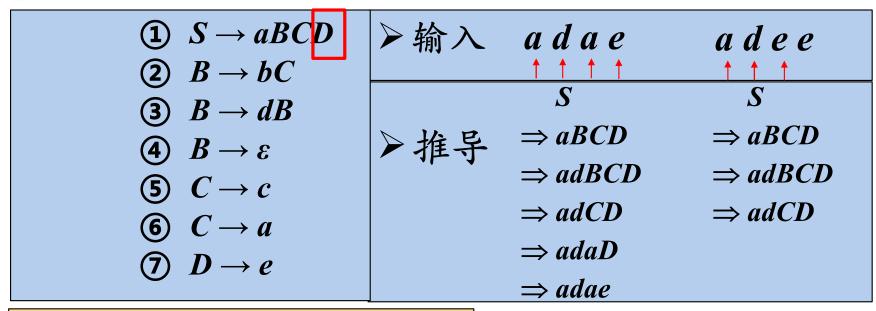
 \triangleright SELECT($A \rightarrow \alpha$): 可以选用该产生式进行推导时对应的输入符号的集合 终结符集合 FOLLOW(A) FIRST(α)

- \triangleright FIRST(α): 首终结符集。 α 能够推出的所有终结符串中位于串首的那些终结符构成的集合
- ► FOLLOW(A): 可以在某句型中紧跟在A后边的终结符构成的集合 计算方法见课件

$$SELECT(A \rightarrow \alpha) = \begin{cases} FIRST(\alpha) & \text{if } \varepsilon \notin FIRST(\alpha) \\ (FIRST(\alpha) - \{\varepsilon\}) \cup FOLLOW(A) & \text{if } \varepsilon \in FIRST(\alpha) \end{cases}$$

- ►SELECT(<u>A→α</u>): 可以选用该产生式进行推导时对应的输入符号的集合 相对于产生式而言 终结符集合
- ightarrow FIRST(lpha): 首终结符集。lpha能够推出的所有终结符串中位于串首的那些终结符构成的集合相对于串而言 可含 ϵ
- ▶ FOLLOW(A): 可以在某句型中紧跟在A后边的终结符构成的集合相对于非终结符而言

例



可以緊跟B后面出现的终结符:c、a

- ►什么时候使用*E*产生式?
 - \triangleright 如果当前某非终结符A与当前输入符a不匹配时,若存在 $A \rightarrow \varepsilon$,可以通过检查a是否可以出现在A的后面,来决定是否使用产生式 $A \rightarrow \varepsilon$ (若文法中无 $A \rightarrow \varepsilon$,则应报错)

FOLLOW(A)计算方法

- ▶不断应用下列规则,直到没有新的终结符可以被加入到任何FOLLOW集合中为止
 - ▶将\$放入FOLLOW(S)中,其中S是开始符号,\$是输入右端的结束标记
 - ▶如果存在一个产生式 $A \rightarrow \alpha B\beta$,那么 $FIRST(\beta)$ 中除ε之外的所有符号都在FOLLOW(B)中
 - ightharpoonup如果存在一个产生式 $A
 ightharpoonup \alpha B$,或存在产生式 $A
 ightharpoonup \alpha B$ β且 FIRST(β)包含 ε ,那么 FOLLOW(A)中的所有符号都在 FOLLOW(B)中 FOLLOW(A)
 ightharpoonup FOLLOW(B)

例

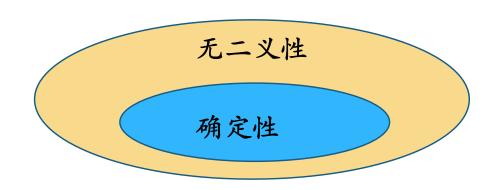
- \triangleright (1) $S \rightarrow Ax$
- \triangleright (2) $A \rightarrow bB$
- > (3) $B \rightarrow Cb$
- \triangleright (4) $C \rightarrow cBc$

$FOLLOW(A) \subseteq FOLLOW(B)$

C

小结:top down步骤

- ▶消除歧义 无二义性
- 文法改造
 - ▶消除左递归
 - > 提取左公因子
- ▶LL(1)文法判定 确定性



一 无二义性文法的判定

- ▶对于任意一个上下文无关文法,不存在一个算法, 判定它是无二义性的;但能给出一组充分条件, 满足这组充分条件的文法是无二义性的
 - ▶满足, 肯定无二义性
 - ▶不满足,也未必就是有二义性的

