

自顶向下方法

一、递归下降分析

问：为什么采用最左推导？

因为对输入串的扫描是自左至右进行的，只有使用最左推导，才能保证按扫描的顺序匹配输入串。

二、递归调用预测分析

三、非递归预测分析

例：为如下文法构造预测分析程序转换图

$$\begin{aligned} E &\rightarrow E+T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid id \end{aligned} \quad (\text{文法 4.3})$$

消除文法中存在的左递归，得到

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \epsilon \\ F &\rightarrow (E) \mid id \end{aligned} \quad (\text{文法 4.4})$$

LL (1) 文法的定义

如果一个文法的预测分析表 M 不含多重定义的表项，则称该文法为 LL(1)文法。

LL(1)的含义：

- 第一个 L 表示从左至右扫描输入符号串
- 第二个 L 表示生成输入串的一个最左推导
- 1 表示在决定分析程序的每步动作时，向前看一个符号

LL (1) 方法的判断

一个文法是 LL(1)文法，当且仅当它的每一个产生式 $A\alpha \rightarrow \beta$ ，满足：

- $FIRST(\alpha) \cap FIRST(\beta) = \phi$ 并且
- 若 β 推导出 ϵ ，则 $FIRST(\alpha) \cap FOLLOW(A) = \phi$

LR(k)的含义：

- L 表示自左至右扫描输入符号串
- R 表示为输入符号串构造一个最右推导的逆过程
- k 表示为作出分析决定而向前看的输入符号的个数。

若用算法 4.5 构造出来的分析表不含有冲突，则该分析表称为该文法的 SLR(1)分析表
具有 SLR(1)分析表的文法称为 SLR(1)文法

如果在执行上述算法的过程中，始终没有向前看任何输入符号，则构造的 SLR 分析表称为 LR(0)分析表

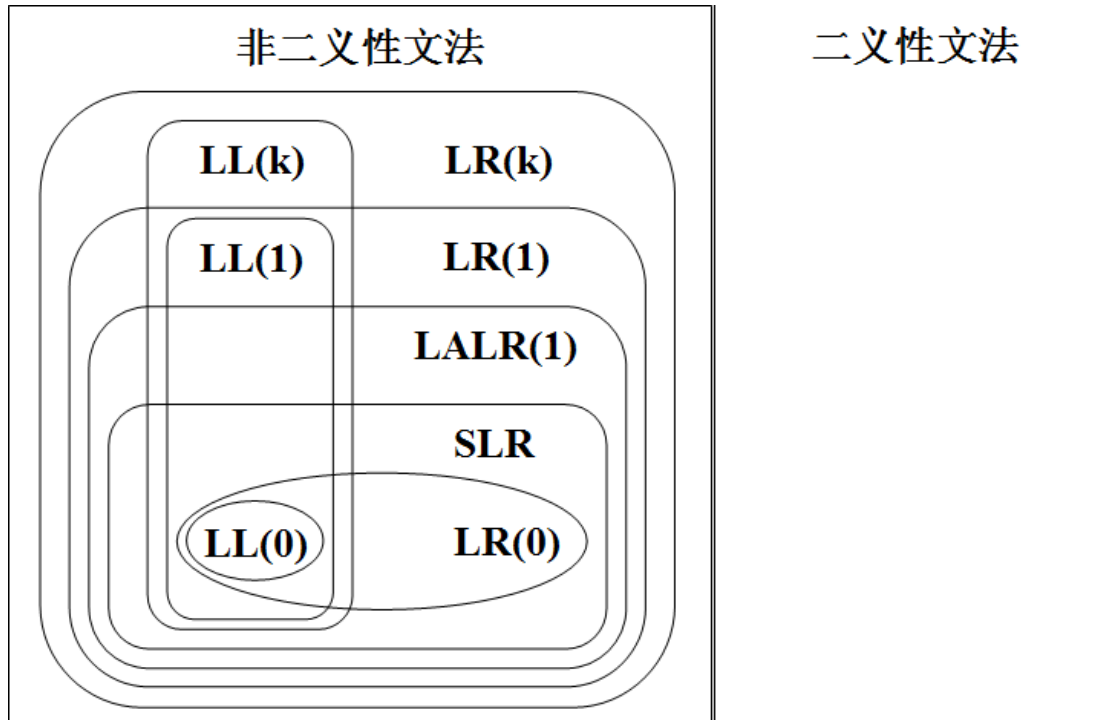
具有 LR(0)分析表的文法称为 LR(0)文法。

一个文法是 LR(0)文法，当且仅当该文法的每个活前缀的有效项目集中：

- 要么所有元素都是移进-待约项目
- 要么只含有唯一的归约项目

具有如下产生式的文法是一个 **LR(0)** 文法。

$$A \rightarrow (A) | a$$



n 任何二义性文法决不是 LR 文法，因而也不是 SLR 或 LALR 文法。