

自顶向下的语法分析

从文法的开始符号出发，反复使用文法的产生式，寻找与输入符号串匹配的推导。

带回溯的自顶向下分析技术效率很低，代价极高，只有理论意义。

回溯产生的原因与解决办法

- 消除左递归

若 $A \rightarrow A\alpha|\beta$ ，其中 β 不以 A 开头

则修改为：

$$A \rightarrow \beta A'$$
$$A' \rightarrow \alpha A'|\epsilon$$

可以理解为，从原推导式得到句子为 β 开头后跟若干个 α 。那么改为强制得到一个 β 在最前面，然后通过右递归/空串来表示若干个 α

例：

消除直接左递归： $P \rightarrow PaPb|BaP$

只需要对引起直接左递归的 P 进行替换即可。

$$P \rightarrow BaPP'$$
$$P' \rightarrow aPbP'|\epsilon$$

- 提取左因子

若 $A \rightarrow \alpha\beta_1|\alpha\beta_2$

改写成： $A \rightarrow \alpha A'$

$$A' \rightarrow \beta_1|\beta_2$$

递归下降分析法

在不含左递归和每个非终结符的所有候选式的终结首符集都两两不相交条件下，构造一个不带回溯的自上而下分析程序，该分析程序由一组递归过程组成，每个过程对应文法的一个非终结符。

预测分析表

使用一张分析表和一个栈，同样可以实现递归下降分析。这种方法实现的语法分析程序叫预测分析程序。

矩阵 $M[A, a]$ ，行标是非终结符 V_n ，列标是终结符 V_t 或串终结符 $\#$ ，矩阵元素 $M[A, a]$ 是存放 A 的一个候选式，指出当前栈顶符号为 A 且读入符号为 a 时应选的候选式，或存放错误标志。

过程：

1. $\#$ 和 S 入栈，读入输入串的第一个单词
2. 根据栈顶文法符号 X 和当前单词 a ，做不同处理
 - 若 $X = a = \#$ ，分析成功，停止。
 - 若 $X = a \neq \#$ ，把 X 从栈顶弹出，再读入下一个符号，转2
 - 若 $X \in V_n$ ，查分析表 M

- 若 $M[X, a] = X \rightarrow UVW$, 则将X弹出栈, 将UVW逆序压入栈。(U在栈顶), 转2
- 若 $M[X, a] = error$, 转出错处理
- 若 $M[X, a] = X \rightarrow \varepsilon$, 将X弹出栈, 转2

FIRST/FOLLOW

- FIRST集
 - 针对终结符、非终结符、候选式
 - 步骤
 - 以产生式 $A \rightarrow \alpha BC | BC$ 为例:
 1. 看产生式右侧第一个字符, 终结符 α , 直接放入 $FIRST(A)$
 2. 看产生式右侧第一个字符, 非终结符B, 将 $FIRST(B)$ 放入 $FIRST(A)$
 3. 如果 $B \rightarrow \varepsilon$, 将 $FIRST(C)$ 放入 $FIRST(A)$
- FOLLOW集
 - 只针对非终结符
 - 步骤
 1. 把#放进开始符号的FOLLOW集
 2. 非终结符后紧跟终结符, 则将该终结符放入非终结符的FOLLOW集
 3. 循环扫描: 对于 $A \rightarrow \alpha BC$
 1. $FIRST(C)$ 放入 $FOLLOW(B)$
 2. $FOLLOW(A)$ 放入 $FOLLOW(C)$
 3. 如果存在 $C \rightarrow \varepsilon$, 则将 $FOLLOW(A)$ 放入 $FOLLOW(B)$
- 根据FF构造预测分析表

对每个产生式:

 - 对每个终结符号 $a \in FIRST(\alpha)$, 把 $A \rightarrow \alpha$ 添加至 $M[A, a]$ 中
 - 若 $a \xRightarrow{*} \varepsilon$, 对任何 $b \in FOLLOW(A)$, 把 $A \rightarrow \alpha$ 添加至 $M[A, b]$ 中
 - 所有未定义的添加错误标记

LL(1)分析法

自左而右, 最左推导, 向前看1个符号。

若一个文法的分析表不含多重定义入口, 则被称为LL(1)文法。

充要条件

一个文法G是LL(1)的, 当且仅当对于G的每一个非终结符号A的任何两个不同产生式 $A \rightarrow \alpha | \beta$, 满足:

- $FIRST(\alpha) \cap FIRST(\beta) = \emptyset$
- 假若 $\beta \xRightarrow{*} \varepsilon$, 那么 $FIRST(\alpha) \cap FOLLOW(A) = \emptyset$

错误检测

- 栈顶的终结符和当前输入符号不匹配
- 栈顶非终结符与当前输入符号在预测表对应项中的信息为空

相关结论

LL1能判文法不能判语言

- (1) 任何LL(1)文法都是无二义性的；
- (2) 左递归文法必然不是LL(1)文法；
- (3) 存在一种算法，它能判断任意的文法是否为LL(1)文法；
- (4) 存在一种算法，它能判定任意两个LL(1)文法是否产生相同的语言；
- (5) 不存在这样的算法，它能判定任意的上下文无关语言是否为LL(1)语言；
- (6) 非LL(1)语言 (即不能由任何LL(1)文法产生的上下文无关语言) 是存在的