

## 2.4 LL(1)文法分析实验设计思想及算法

### 2.4.1 实验基本思路

首先给出 LL(1)文法的定义<sup>[2]</sup>:

一个文法  $G$  被称为 LL(1)文法, 如果它满足以下条件:

(1) 文法不含左递归。

(2) 对于文法中每一个非终结符  $A$  的各个产生式的候选首字符集两两不相交。即, 若

$$A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$$

$$\text{则 } \text{FIRST}(\alpha_i) \cap \text{FIRST}(\alpha_j) = \Phi \quad (i \neq j)$$

(3) 对文法中的每个非终结符  $A$ , 若它存在某个候选首字符集包含  $\epsilon$ , 则

$$\text{FIRST}(\alpha_i) \cap \text{FOLLOW}(A) = \Phi \quad (i = 1, 2, \dots, n)$$

这里, LL(1)的第一个 L 表示从左向右扫描输入串, 第二个 L 表示最左推导, 1 表示分析时每一步只需向前查看一个符号。

对于一个 LL(1)文法, 可以对其输入串进行有效的无回溯的自上而下分析。假设要用非终结符  $A$  进行匹配, 面临的输入符号为  $a$ ,  $A$  的所有产生式为:

$$A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$$

(1) 若  $a \in \text{FIRST}(\alpha_i)$ , 则指派  $\alpha_i$  去执行匹配任务。

(2) 若  $a$  不属于任何一个候选首符集, 则:

①若  $\epsilon$  属于某个  $\text{FIRST}(\alpha_i)$  且  $a \in \text{FOLLOW}(A)$ , 则让  $A$  与  $\epsilon$  自动匹配;

②否则,  $a$  的出现是一种错误。

综述, 本实验对教材上的描述的几个算法进行了实现, 成功达成了 LL(1)文法分析, 并进行了简单的测试。

### 2.4.2 算法流程

LL(1)文法分析的架构如图 8 所示。