

3.4 LR(1)文法分析实验设计思想及算法

3.4.1 实验基本思路^[2]

LR 文法的每个项目的一般形式是 $[A \rightarrow \alpha \cdot \beta, a_1a_2...a_k]$ ，此处， $A \rightarrow \alpha \cdot \beta$ 是一个 LR(0)项目，每一个 a 都是终结符。这样的项目称为一个 LR(k)项目。项目中的 $a_1a_2...a_k$ 称为它的向前搜索字符串（或展望串）。向前搜索字符串仅对规约项目 $[A \rightarrow \alpha \cdot \beta, a_1a_2...a_k]$ 有意义。对于任何移进或待约项目 $[A \rightarrow \alpha \cdot \beta, a_1a_2...a_k]$ ， $\beta \neq \epsilon$ ，搜索字符串 $a_1a_2...a_k$ 没有作用。规约项目 $[A \rightarrow \alpha \cdot, a_1a_2...a_k]$ 意味着：当它所属的状态呈现在栈顶且后续的 k 个输入符号为 $a_1a_2...a_k$ 时，才可以把栈顶上的 α 规约为 A 。我们只对 $k \leq 1$ 的情形感兴趣，因为，对多数程序语言的语法来说，向前搜索（展望）一个符号就多半可以确定“移进”或“规约”。

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 LR(1)文法分析，并进行了简单的测试。

3.4.2 算法流程

LR 文法分析的架构如图 17 所示。

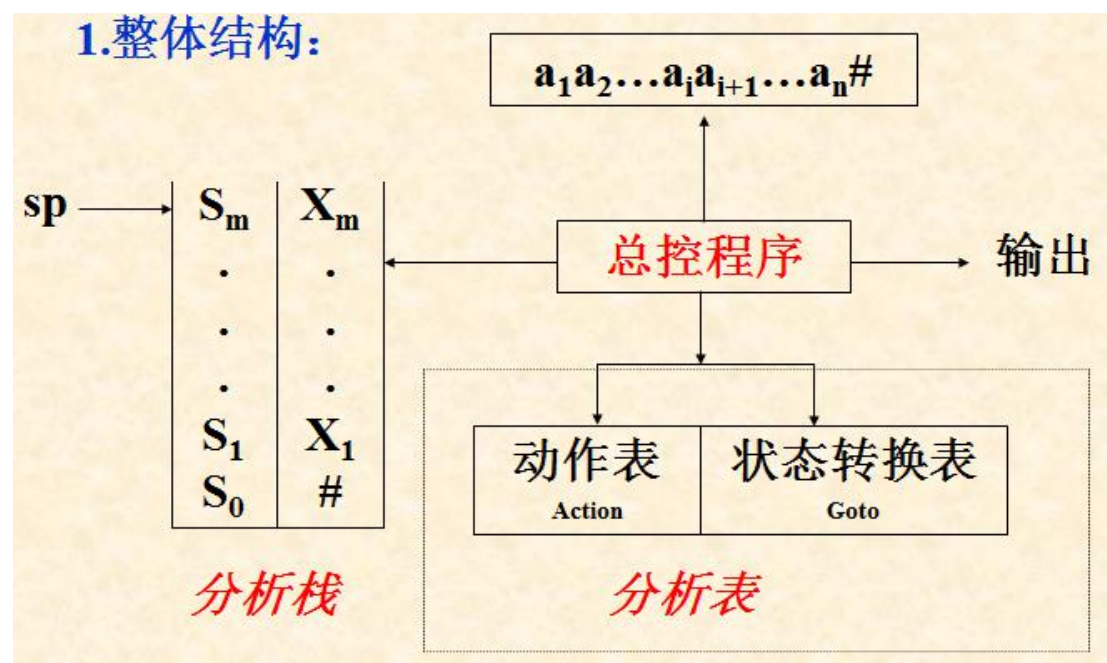


Figure 17 LR 文法分析架构