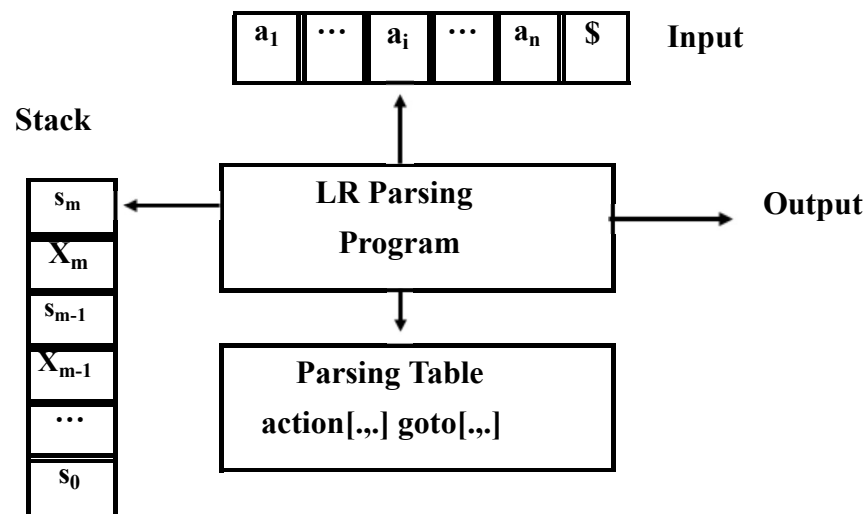


LR分析法

算法框架



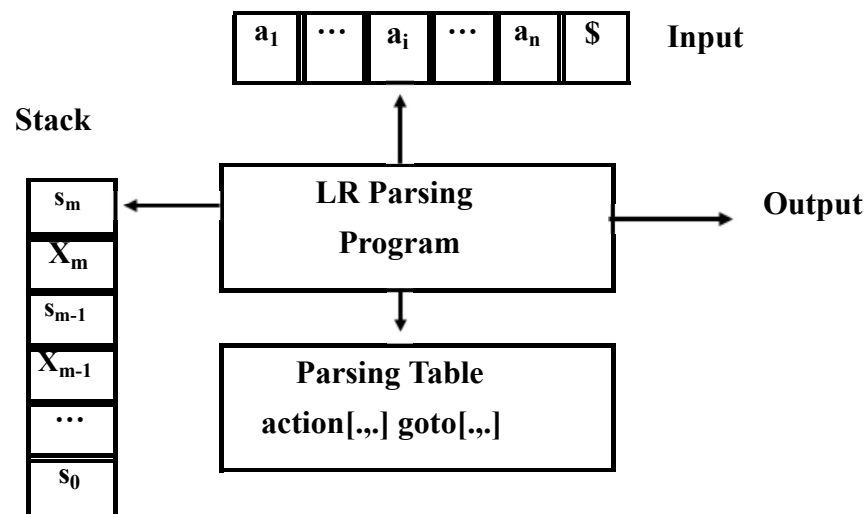
- 输入缓冲区：输入的单词串
- 栈=移进的输入单词归约出的终结符、非终结符串
- 栈+输入缓冲 形成 最右句型!
- 栈中状态 s_i 的作用?

算法思想：栈中串始终是活前缀

取栈中所有符号太低效

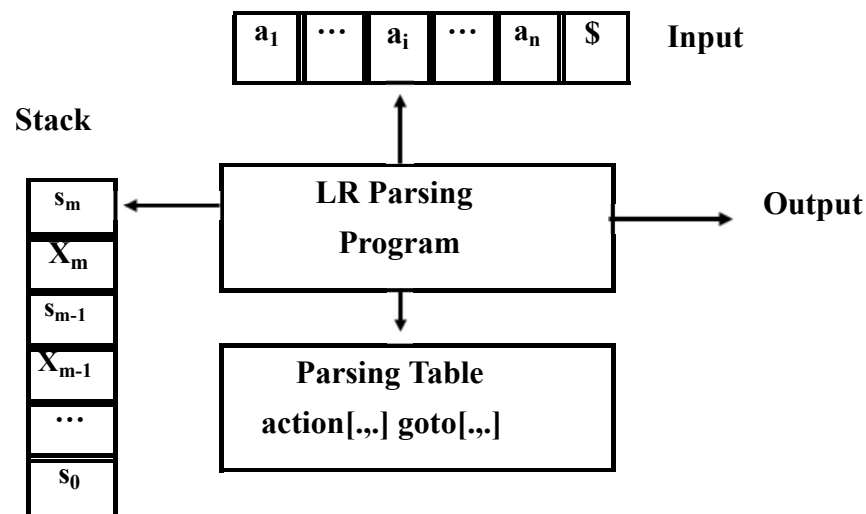
s_i 表示其下符号串，取栈顶即可

LR分析表



- 划分为action表和goto表
- action, 行: 状态 s_i , 列: 终结符
驱动算法执行的一般动作:
移进、归约、接受、失败
- goto, 行: 状态 s_i , 列: 非终结符
归约之后的处理: 下一状态
- 栈和输入的当前状态
称为configuration

LR分析算法执行



- $action[s_m, a_i]$ = 移进 s , 格局变化

$(s_0 X_1 s_1 \dots X_m s_m, a_i a_{i+1} \dots a_n \$)$

$(s_0 X_1 s_1 \dots X_m s_m a_i s, a_{i+1} \dots a_n \$)$

- $action[s_m, a_i]$ = 归约 $A \rightarrow X_{m-r+1} \dots X_m$

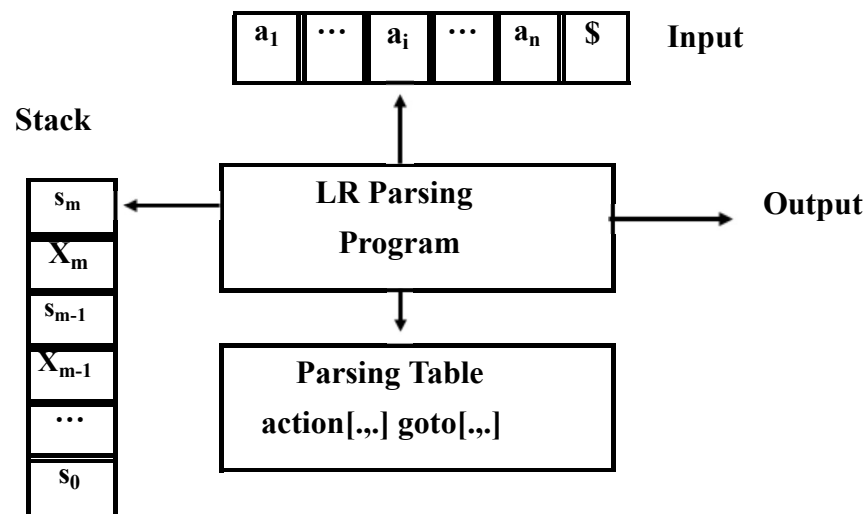
$(s_0 X_1 s_1 \dots X_m s_m, a_i a_{i+1} \dots a_n \$)$

$(s_0 X_1 s_1 \dots X_{m-r} s_{m-r} A s, a_i a_{i+1} \dots a_n \$)$

其中 $s = goto[s_{m-r}, A]$

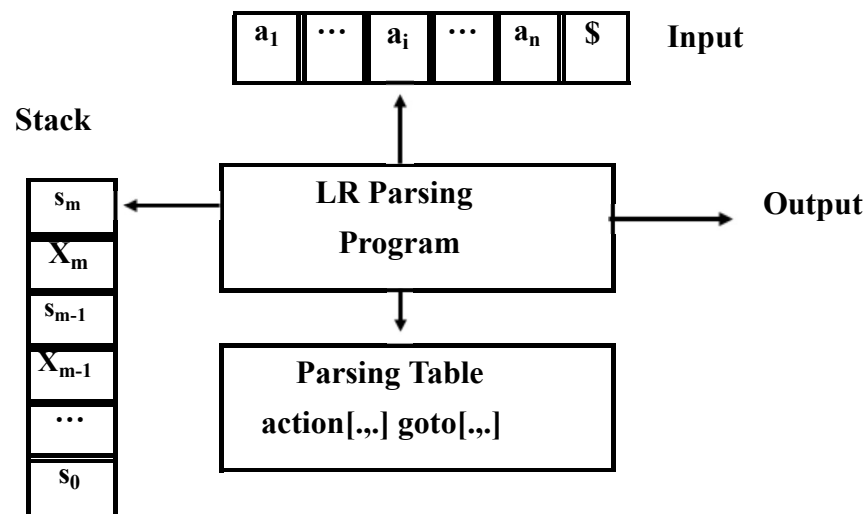
- $action[s_m, a_i]$ = 接受, 成功
- $action[s_m, a_i]$ = 错误, 做错误恢复
- 讲义P54、P55示例

LR分析表构造



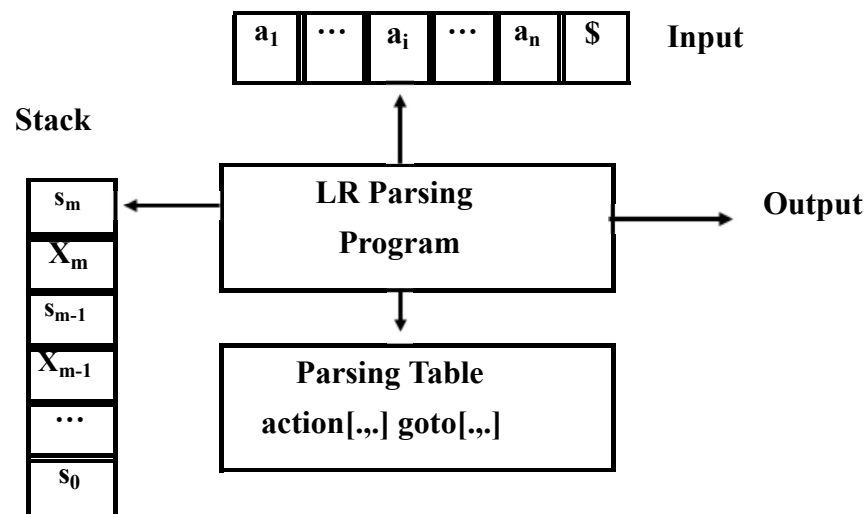
- 核心思想：保证栈中始终活前缀
- 活前缀为正规集 \mathcal{F}_R 构造DFA识别
- 活前缀无穷多，不可能穷举它们
 \mathcal{F}_R 找有穷的替代物来构造DFA
- LR(0)项目，产生式 $A \xrightarrow{F_0 R_0} XYZ \xrightarrow{F_0 R_0}$
 $A \xrightarrow{F_0 R_0} \cdot XYZ,$ $A \xrightarrow{F_0 R_0} X \cdot YZ,$
 $A \xrightarrow{F_0 R_0} XY \cdot Z,$ $A \xrightarrow{F_0 R_0} XYZ \cdot$
- 为何能替代活前缀？
 前缀 \mathcal{F}_R 句型 \mathcal{F}_R 产生式，“活”呢？

LR(0)项目



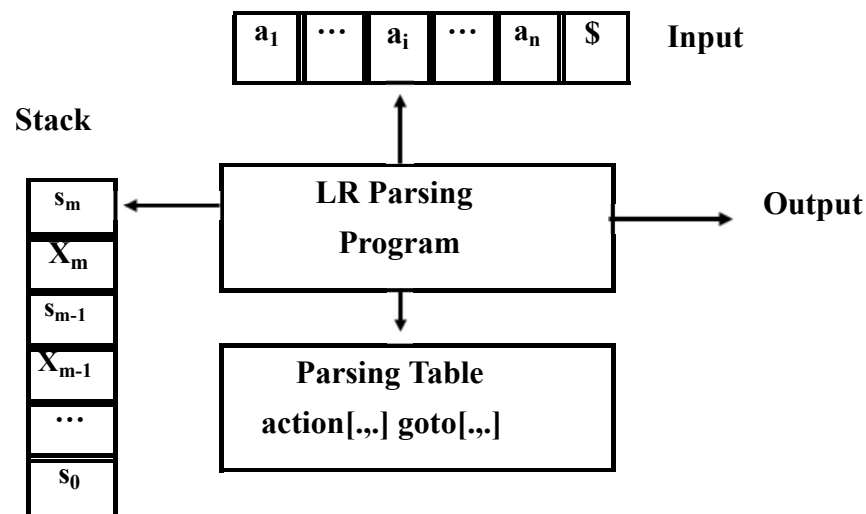
- $A \overset{\text{P.O.}}{\text{R.E.}} X \cdot YZ$, 点的含义?
 栈中已形成X, 期待继续形成YZ
 目标? 栈中形成XYZ后归约出A!
- 所以, 点=活前缀末尾!
- LR分析表构造——用LR(0)项目构造识别活前缀的DFA
- 天然有一个LR(0)项目构成的识别活前缀的NFA

LR(0)项目



- $A \overset{\text{P.O.}}{\text{R.E.}} X \cdot YZ$, 点的含义?
 栈中已形成X, 期待继续形成YZ
 目标? 栈中形成XYZ后归约出A!
- 所以, 点=活前缀末尾!
- LR分析表构造——用LR(0)项目构造识别活前缀的DFA
- 天然有一个LR(0)项目构成的识别活前缀的NFA

识别活前缀的NFA



- 每个LR(0)项目形成NFA一个状态
- 边（状态迁移）如何构造？

$A \overset{\text{FO}}{\underset{\text{RE}}{\boxed{\cdot}}} X \cdot YZ$ 到 $A \overset{\text{FO}}{\underset{\text{RE}}{\boxed{X \cdot}}} Y \cdot Z$ ，边上符号Y

——通过移进(和归约)形成了Y

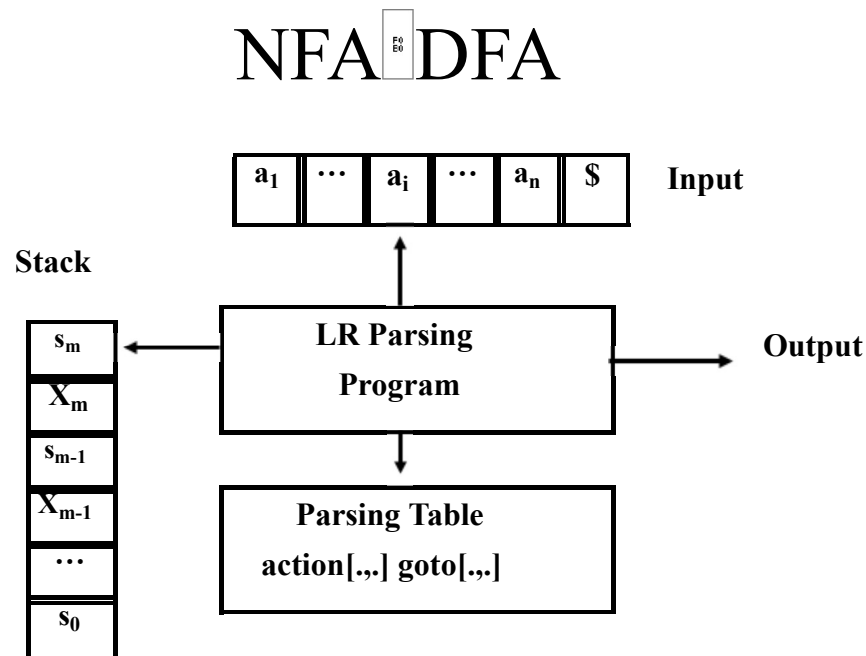
$A \overset{\text{FO}}{\underset{\text{RE}}{\boxed{a \cdot}}} Bb$ 到 $B \overset{\text{FO}}{\underset{\text{RE}}{\boxed{\cdot}}} g$ ，边上符号e

——为形成B，应先去形成g

- 讲义P60：LR分析——NFA运转

移进a：走a边——状态迁移 $d(T, a)$

归约： $A \overset{\text{FO}}{\underset{\text{RE}}{\boxed{a \cdot}}}$ 回退 $A \overset{\text{FO}}{\underset{\text{RE}}{\boxed{\cdot}}} a$ ，再走A边



- 构造LR(0)项目集规范族
项目集—NFA状态集，子集构造法！
- e-closure闭包——项目集I的闭包
closure(I): 若包含 $A \overset{\text{F0}}{\underset{\text{RE}}{\boxed{\cdot}}} a \cdot Bb$ ，则也加入 $B \overset{\text{F0}}{\underset{\text{RE}}{\boxed{\cdot}}} g$ ，因为两者间有e边！
- 状态迁移 $d(T, a)$ —— $\text{goto}(I, X)$
 $= \text{closure}(\{A \overset{\text{F0}}{\underset{\text{RE}}{\boxed{\cdot}}} \overset{\text{F0}}{\underset{\text{G1}}{\boxed{X}}} \cdot \overset{\text{F0}}{\underset{\text{G2}}{\boxed{\cdot}}} \mid A \overset{\text{F0}}{\underset{\text{RE}}{\boxed{\cdot}}} \overset{\text{F0}}{\underset{\text{G1}}{\boxed{X}}} \cdot \overset{\text{F0}}{\underset{\text{G2}}{\boxed{\cdot}}} \in I\})$
- 填分析表：a边填移进动作，A边填goto表， $A \overset{\text{F0}}{\underset{\text{RE}}{\boxed{\cdot}}} \overset{\text{F0}}{\underset{\text{G1}}{\boxed{\cdot}}}$ 填归约动作

