



□ 自顶向下 (Top-down)

❖ 针对输入串，从文法的开始符号出发，尝试根据产生式规则**推导 (derive)**出该输入串。

❖ 分析树的构造方法

➤ 从根部开始

□ 自底向上 (Bottom-up)

❖ 针对输入串，尝试根据产生式规则**归约 (reduce)**到文法的开始符号。

❖ 分析树的构造方法：

➤ 从叶子开始



□递归下降的预测分析

- ❖ 包括一个输入缓冲区和向前看指针 *lookahead*, 自左向右扫描输入串
- ❖ 设计一个辅助过程 *match()*, 将 *lookahead* 指向的位置与产生式迭代生成的终结符进行匹配, 如匹配, 将 *lookahead* 挪到下一个位置
- ❖ 为每一个非终结符写一个分析过程
 - 该过程可以调用其他非终结符的过程及 *match*
 - 这些过程可能是递归的



□可能进入无限循环

□考虑以下文法

$$S \rightarrow Sa / b$$

□该文法是左递归的(left-recursive)

□自顶向下分析方法无法处理左递归

❖考虑输入文法符号串为baaaaa

❖最左推导如下：

➤ $S \Rightarrow Sa \Rightarrow Saa \Rightarrow Saaa \Rightarrow Saaaa \dots$

➤输入缓冲区lookahead指针纹丝未动



□文法左递归

$$A \Rightarrow^+ A \alpha$$

□直接左递归

$A \rightarrow A \alpha \mid \beta$, 其中 α, β 不以 A 开头

❖ 串的特点 $\beta \alpha \dots \alpha$

□消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \epsilon$$



□直接左递归

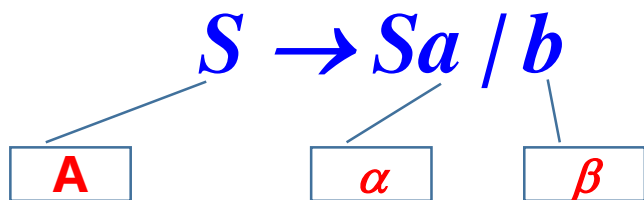
$A \rightarrow A\alpha \mid \beta$, 其中 α , β 不以 A 开头

□消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \varepsilon$$

□考虑之前的文法



$$S \rightarrow bS'$$

$$S' \rightarrow aS' \mid \varepsilon$$

baaaaaa推导:

$$S \Rightarrow bS' \Rightarrow baS' \Rightarrow baaS' \Rightarrow baaaS' \Rightarrow baaaaS' \Rightarrow baaaaaS'$$

输入缓冲区指针不停地移动



□例 算术表达文法

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid \text{id}$$

$$(T + T \dots + T)$$

$$(F * F \dots * F)$$

□消除左递归后文法

$$E \rightarrow TE'$$

$$E' \rightarrow + TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow * FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid \text{id}$$

注明：红色部分代表了
 α ，蓝色部分代表了 β



□处理任意数量的A产生式

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

其中 β_i 都不以A开头

改为:

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \varepsilon$$



□非直接左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Sd \mid \varepsilon$$

□先变换成直接左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Aad \mid bd \mid \varepsilon$$

□再消除左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow bd A' \mid A'$$

$$A' \rightarrow adA' \mid \varepsilon$$



□有左因子的(left -factored)文法:

$$\diamond A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$$

□提左因子(left factoring)

❖ 推后选择产生式的时机，以便获取更多信息

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \text{ 等价于}$$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 \mid \beta_2$$



□例 悬空 $else$ 的文法

$$\begin{aligned} stmt &\rightarrow \text{if } expr \text{ then } stmt \text{ else } stmt \\ &\quad | \text{if } expr \text{ then } stmt \\ &\quad | \text{other} \end{aligned}$$

提左因子

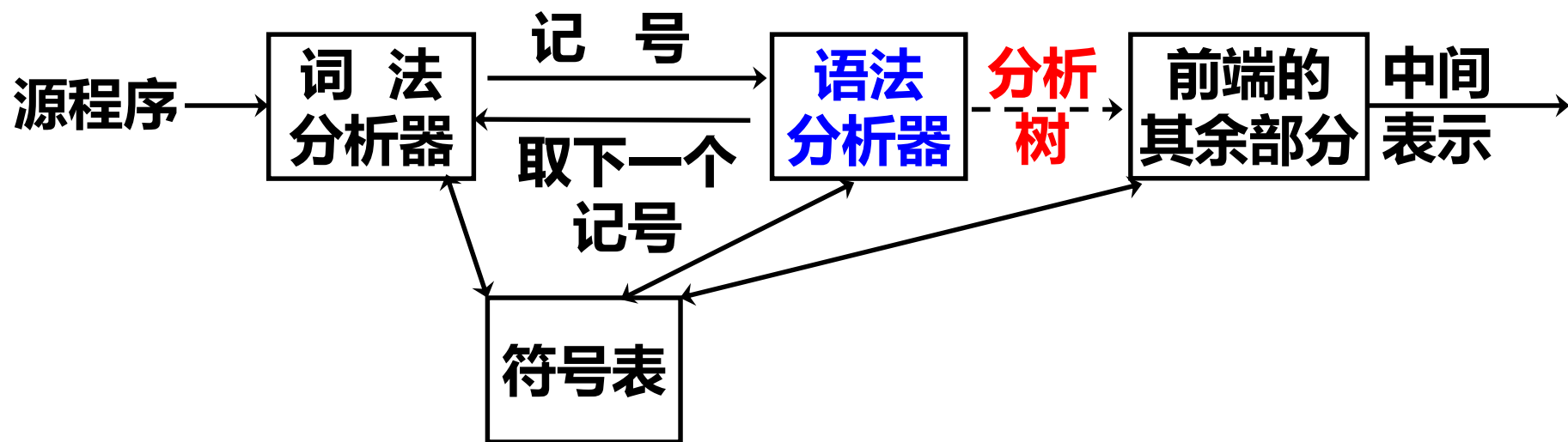
$$\begin{aligned} stmt &\rightarrow \text{if } expr \text{ then } stmt \text{ optional_else_part} \\ &\quad | \text{other} \\ \text{optional_else_part} &\rightarrow \text{else } stmt \\ &\quad | \epsilon \end{aligned}$$

算法仍然二义!!!



□复杂的回溯→代价太高

- ❖非终结符有可能有多个产生式
- ❖由于信息缺失，无法准确预测选择哪一个
- ❖考虑到往往需要对多个非终结符进行推导展开，因此尝试的路径可能呈指数级爆炸



□ 语法分析方法概述

❖ 自顶向下与自底向上方法的区别

❖ 自顶向下分析方法

➢ 递归下降分析方法

➢ 消除左递归、提取左公因子

➢ LL(1)文法及非递归预测分析方法



□ Predictive parsing

□ 与递归下降法相似，但

- ❖ 不会对若干产生式进行尝试
- ❖ 没有回溯
- ❖ 通过向前看一些记号来预测需要用到的产生式

□ 此方法接受 $LL(k)$ 文法

- ❖ L -means “left-to-right” scan of input
- ❖ L -means “leftmost derivation”
- ❖ k -means “predict based on k tokens of lookahead”
- ❖ In practice, $LL(1)$ is used



□对文法加什么样的限制可以保证没有回溯？

□先定义两个和文法有关的函数

❖ $\text{FIRST}(\alpha) = \{a \mid \alpha \Rightarrow^* a..., a \in V_T\}$

意义：可从 α 推导得到的串的首符号的集合

❖ $\text{FOLLOW}(A) = \{a \mid S \Rightarrow^* ...A\textcolor{red}{a}..., a \in V_T\}$

意义：可能在推导过程中紧跟在A右边的终结符号的集合



□ 计算FIRST(X), $X \in V_T \cup V_N$

❖ $X \in V_T$, $\text{FIRST}(X) = \{X\}$

❖ $X \in V_N$ 且 $X \rightarrow \epsilon$

则将 ϵ 加入到 $\text{FIRST}(X)$

❖ $X \in V_N$ 且 $X \rightarrow Y_1 Y_2 \dots Y_k$

➤ 如果 $a \in \text{FIRST}(Y_i)$ 且 ϵ 在 $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$ 中, 则将 a 加入到 $\text{FIRST}(X)$

➤ 如果 ϵ 在 $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_k)$ 中, 则将 ϵ 加入到 $\text{FIRST}(X)$

FIRST集合只包括终结符和 ϵ



□例 $E \rightarrow TE'$
 $E' \rightarrow + TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow * FT' \mid \varepsilon$
 $F \rightarrow (E) \mid \text{id}$

□ $X \in V_T, \text{FIRST}(X) = \{X\}$

□ $X \in V_N$ 且 $X \rightarrow \varepsilon, \varepsilon \in \text{FIRST}(X)$

□ $X \in V_N$ 且 $X \rightarrow Y_1 Y_2 \dots Y_k$

❖ 如果 $a \in \text{FIRST}(Y_i)$ 且 $\varepsilon \in \text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$ 中, 则 $a \in \text{FIRST}(X)$

❖ 如果 $\varepsilon \in \text{FIRST}(Y_1), \dots, \text{FIRST}(Y_k)$ 中, 则 $\varepsilon \in \text{FIRST}(X)$

$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{ (, \text{id} \}$

$\text{FIRST}(E') = \{ +, \varepsilon \}$

$\text{FIRST}(T') = \{ *, \varepsilon \}$



□ 计算FOLLOW(A), $A \in V_N$

- ❖ $\$$ 加入到 FOLLOW(A), 当 A 是开始符号, $\$$ 是输入串的结束符号
- ❖ 如果 $A \rightarrow \alpha B\beta$, 则 $\text{FIRST}(\beta) - \{\epsilon\}$ 加入到 FOLLOW(B)
- ❖ 如果 $A \rightarrow \alpha B$ 或 $A \rightarrow \alpha B\beta$ 且 $\epsilon \in \text{FIRST}(\beta)$, 则 FOLLOW(A) 加入到 FOLLOW(B)



□例 $E \rightarrow TE'$
 $E' \rightarrow + TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow * FT' \mid \varepsilon$
 $F \rightarrow (E) \mid \text{id}$

- 当A是开始符号, $\$ \in \text{FOLLOW}(A)$
- $A \rightarrow \alpha B\beta$, $\text{FIRST}(\beta) - \{\varepsilon\} \subseteq \text{FOLLOW}(B)$
- $A \rightarrow \alpha B$ 或 $A \rightarrow \alpha B\beta$ 且 $\varepsilon \in \text{FIRST}(\beta)$,
 $\text{FOLLOW}(A) \subseteq \text{FOLLOW}(B)$

$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{ (, \text{id} \}$

$\text{FIRST}(E') = \{ +, \varepsilon \}$

$\text{FRIST}(T') = \{ *, \varepsilon \}$

$\text{FOLLOW}(E) = \{), \$ \} = \text{FOLLOW}(E')$

$\text{FOLLOW}(T) = \{ +,), \$ \} = \text{FOLLOW}(T')$

$\text{FOLLOW}(F) = \{ *, +,), \$ \}$



□LL(1)文法的定义

任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下列条件:

- ❖ $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$
- ❖ 若 $\beta \Rightarrow^* \varepsilon$, 那么 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$



□LL(1)文法的定义

任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下列条件:

❖ **$\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$**

❖ 若 $\beta \Rightarrow^* \varepsilon$, 那么 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$

□该条件存在的必要性

❖ 容易理解

❖ 每次通过输入词法单元记号和**FIRST**集合匹配产生式的时候, 需要有唯一的选择



□LL(1)文法的定义

任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下列条件:

❖ $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$

❖ 若 $\beta \Rightarrow^* \varepsilon$, 那么 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$

□假设 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \{a\}$

$a \in \text{FIRST}(\alpha): A \Rightarrow^* a\alpha'$

$a \in \text{FOLLOW}(A): B \Rightarrow^* \dots A a \dots$



□LL(1)文法的定义

任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下列条件:

❖ $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$

❖ 若 $\beta \Rightarrow^* \varepsilon$, 那么 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$

□假设 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \{a\}$

$a \in \text{FIRST}(\alpha): A \Rightarrow^* a\alpha'$

$a \in \text{FOLLOW}(A): B \Rightarrow^* \dots A a \dots$

由于 $\beta \Rightarrow^* \varepsilon$, 所以遇到 a 时, 无法判断用哪一个产生式

❖ 可以用 $A \rightarrow \alpha$ 来对 A 进行展开

❖ 亦可以用 $A \rightarrow \beta$ 和 $\beta \Rightarrow^* \varepsilon$ 最后把 A 消掉



□LL(1)文法的定义

任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下列条件:

❖ $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$

❖ 若 $\beta \Rightarrow^* \varepsilon$, 那么 $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$

□LL(1)文法有一些明显的性质

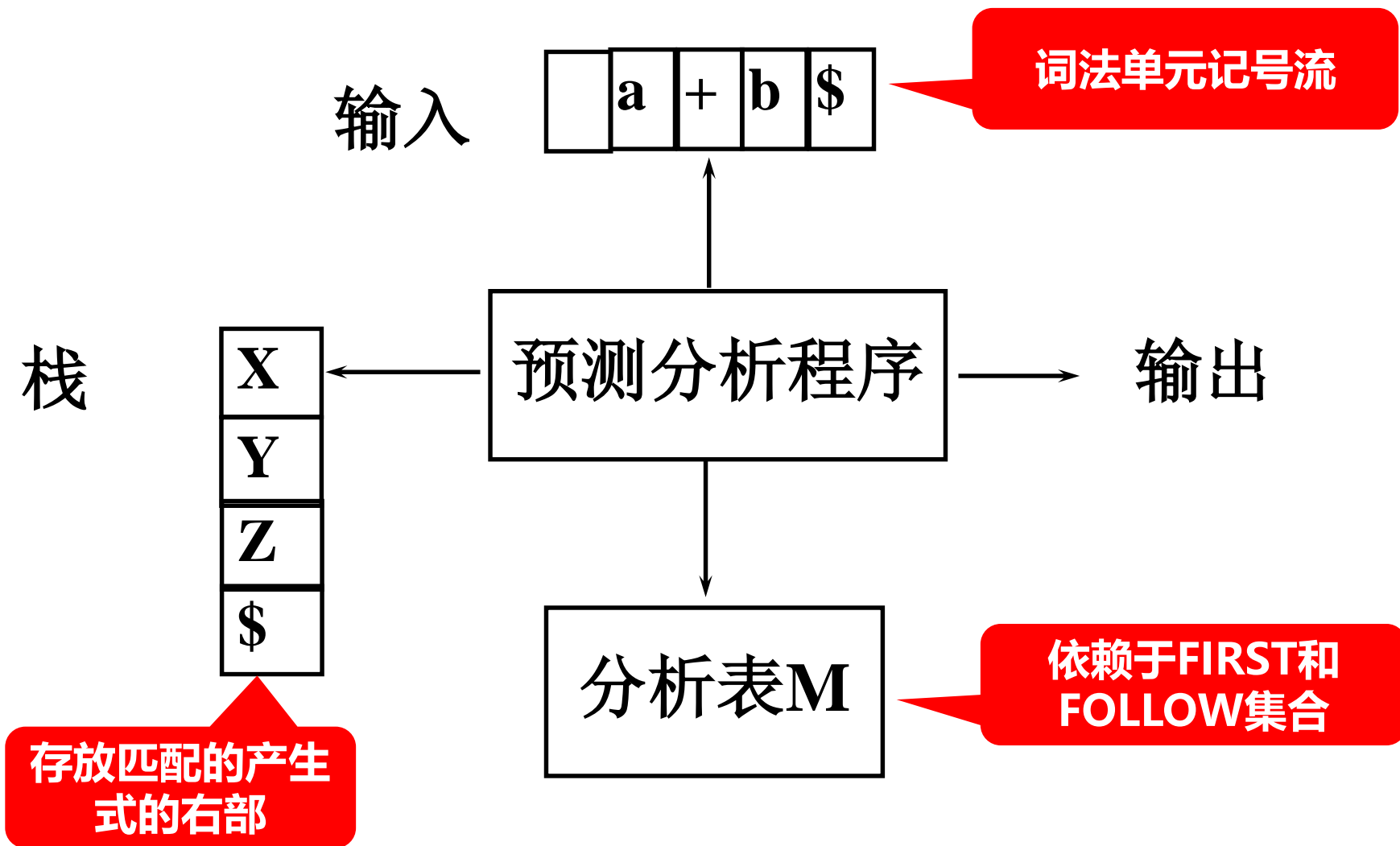
❖ 没有公共左因子

❖ 不是二义的

❖ 不含左递归



非递归的预测分析





□ 对文法的每个产生式 $A \rightarrow \alpha$, 执行(1)和(2)

- ❖ (1) 对 $\text{FIRST}(\alpha)$ 的每个终结符 a , 把 $A \rightarrow \alpha$ 加入 $M[A, a]$
- ❖ (2) 如果 ϵ 在 $\text{FIRST}(\alpha)$ 中, 对 $\text{FOLLOW}(A)$ 的每个终结符 b (包括 $\$$) , 把 $A \rightarrow \alpha$ 加入 $M[A, b]$

M 中其它没有定义的条目都是error



预测分析表M的构造



□行：非终结符；列：终结符 或\$；单元：产生式

| 非终结符 | 输入符号 | | | | | |
|------|---------------------------|------------------------------|-----------------------|---------------------|------------------------------|------------------------------|
| | id | + | * | (|) | \$ |
| E | $E \rightarrow TE'$ | | | $E \rightarrow TE'$ | | |
| E' | | $E' \rightarrow +TE'$ | | | $E' \rightarrow \varepsilon$ | $E' \rightarrow \varepsilon$ |
| T | $T \rightarrow FT'$ | | | $T \rightarrow FT'$ | | |
| T' | | $T' \rightarrow \varepsilon$ | $T' \rightarrow *FT'$ | | $T' \rightarrow \varepsilon$ | $T' \rightarrow \varepsilon$ |
| F | $F \rightarrow \text{id}$ | | | $F \rightarrow (E)$ | | |



预测分析器接受输入 $id * id + id$ 的前一部分动作

| 栈 | 输 入 | 输 出 |
|--------------|-------------|------------------------------|
| $\$E 'T' id$ | $id + id\$$ | $F \rightarrow id$ |
| $\$E 'T'$ | $+ id\$$ | 匹配id |
| $\$E'$ | $+ id\$$ | $T' \rightarrow \varepsilon$ |
| $\$E 'T+$ | $+ id\$$ | $E' \rightarrow +TE'$ |
| $\$E 'T+$ | $id\$$ | 匹配+ |
| $\$E 'T'F$ | $id\$$ | $T \rightarrow FT'$ |
| $\$E 'T'id$ | $id\$$ | $F \rightarrow id$ |
| $\$E 'T'$ | $\$$ | 匹配id |



预测分析器接受输入 $id * id + id$ 的所有动作

| 栈 | 输 入 | 输 出 |
|----------|-----|------------------------------|
| $\$E'T'$ | \$ | $T' \rightarrow \varepsilon$ |
| $\$E'$ | \$ | $E' \rightarrow \varepsilon$ |
| \$ | \$ | Finished |
| | | |
| | | |
| | | |
| | | |

例: $stmt \rightarrow \text{if } expr \text{ then } stmt \text{ } e_part \mid \text{other}$
 $e_part \rightarrow \text{else } stmt \mid \varepsilon$ $expr \rightarrow b$

| 非终结符 | 输入符号 | | | |
|-----------|---------------------------------|----------------------|---|-----|
| | other | b | else | ... |
| $stmt$ | $stmt \rightarrow \text{other}$ | | | |
| e_part | | | $e_part \rightarrow$ $\text{else } stmt$ $e_part \rightarrow \varepsilon$ | |
| $expr$ | | $expr \rightarrow b$ | | |

多重定义条目意味着文法左递归或者是二义的



例：删去 $e_part \rightarrow \varepsilon$ ，这正好满足else和近的then配对
LL(1)文法：预测分析表无多重定义的条目

| 非终结符 | 输入符号 | | | |
|-----------|--------------------------|----------------------|---|-----|
| | other | b | else | ... |
| $stmt$ | $stmt \rightarrow other$ | | | |
| e_part | | | $e_part \rightarrow$ else $stmt$ $e_part \rightarrow \varepsilon$ | |
| $expr$ | | $expr \rightarrow b$ | | |