

编译原理

第四章 自上而下语法分析

方徽星

扬州大学 信息工程学院(505)

fanghuixing@yzu.edu.cn

2018年春季学期

本章主要内容

一. 下推自动机

二. 自上而下分析法存在的问题

三. 自上而下分析法的一般方法

四. LL(1)文法

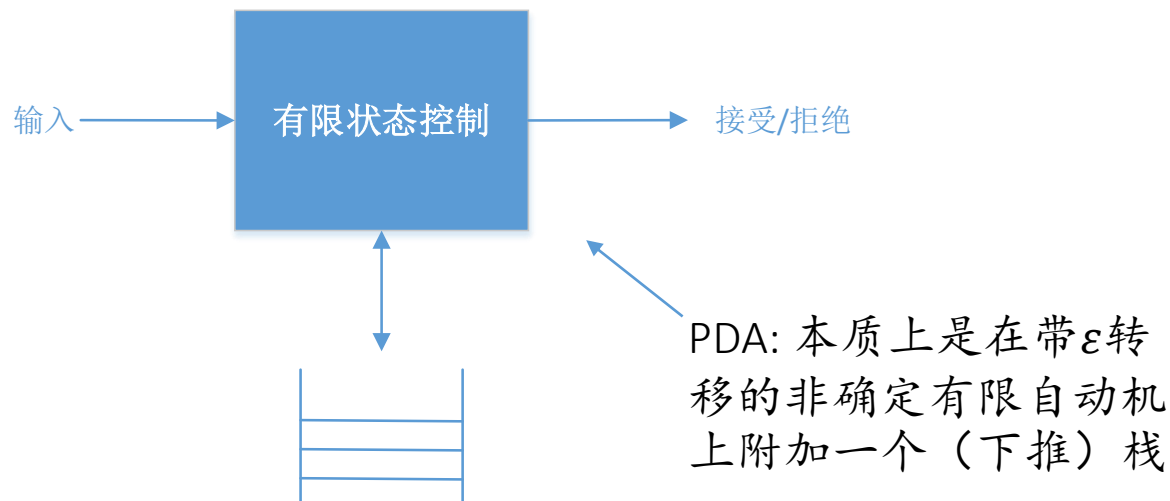
五. 递归下降和非递归的预测分析

六. 预测分析表的构造

第一节 下推自动机

1.1 PDA的组成

- 大部分高级程序设计语言的语句可以用上下文无关文法描述
- 下推自动机(PDA: Pushdown Automata)能够识别上下文无关文法所描述的语言



1.1 PDA的组成

- PDA是一个7元组 $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$:
 - Q : 有限的状态集合
 - Σ : 输入符号字母表 (有限集合)
 - Γ : 有限的栈字母表, 是能够被推入栈的符号集合
 - $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$, 转移函数, δ 的输出是序对的有限集合
 - q_0 : 初始状态 $q_0 \in Q$
 - Z_0 : 初始符号 $Z_0 \in \Gamma$, 开始时栈中只有这个符号
 - F : 接受状态集合, $F \subseteq Q$

1.2 PDA的工作原理

- 设上下文无关文法 $G = (V_N, \Sigma, P, S)$, 构造下推自动机 $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$:
 - $Q = F = \{q_0\}$, 控制器只有一个状态
 - $\Gamma = V_N \cup \Sigma$, 非终结符集合与终结符集合的并
 - $Z_0 = S$, 栈初始符号与文法初始符号相同
 - 转移函数:
 - 推导: 对于形如: $A \rightarrow r$ 的产生式, 有 $\delta(q, \varepsilon, A) = (q, r)$
 - 匹配: 对于输入符号 $a \in \Sigma$, 有 $\delta(q, a, a) = (q, \varepsilon)$

1.2 PDA的工作原理

- 例：构造PDA接受语言 $L = \{a^n cb^n \mid n \geq 0\}$

解：生成 L 的文法产生式可以表示如下：

$$S \rightarrow aSb \mid c$$

相应的PDA $M = (\{q\}, \{a, b, c\}, \{S, a, b, c\}, \delta, q, S, \{q\})$ 的转移函数：

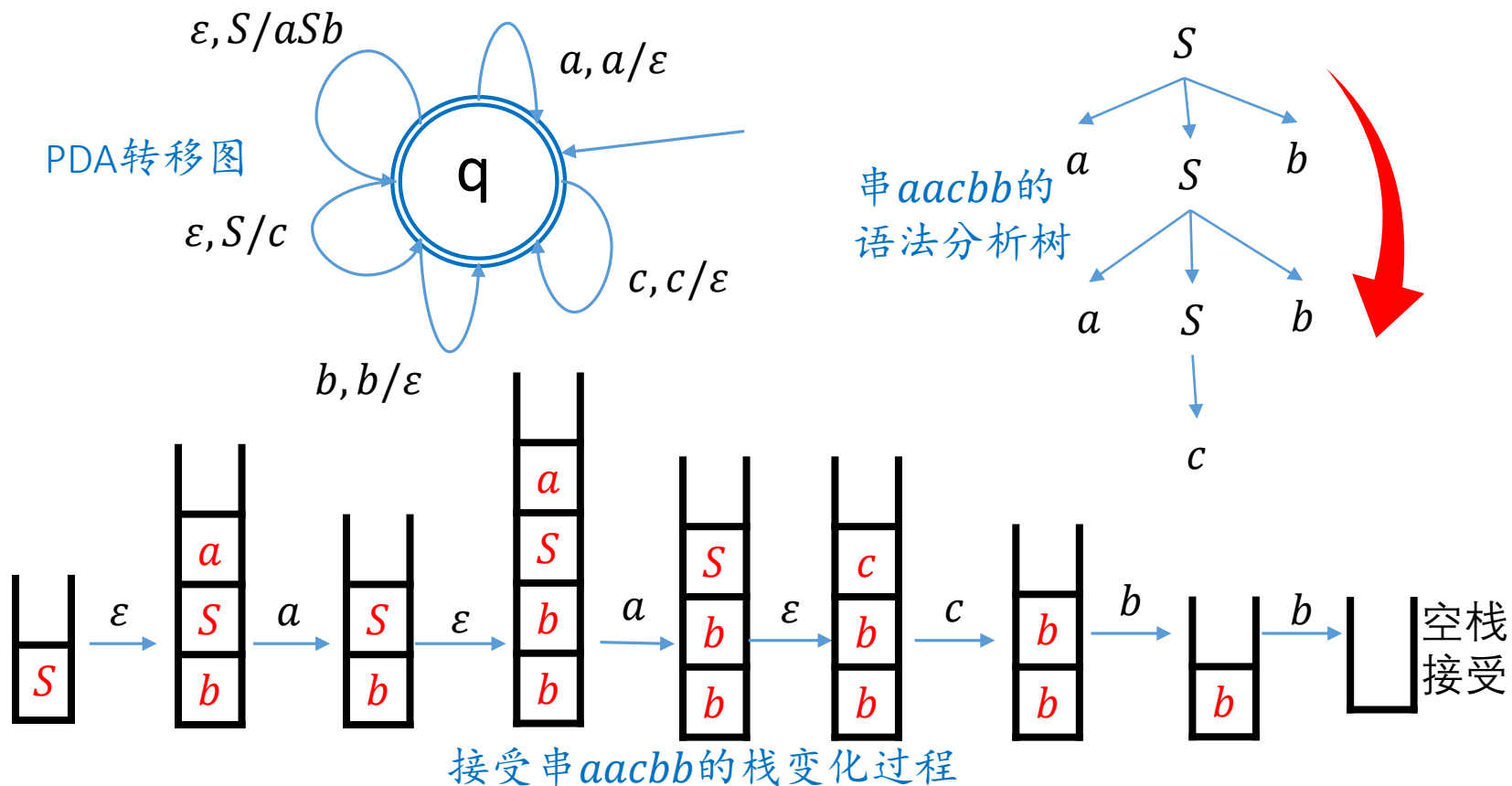
- $\delta(q, \varepsilon, S) = \{(q, aSb), (q, c)\}$
- $\delta(q, a, a) = (q, \varepsilon)$
- $\delta(q, b, b) = (q, \varepsilon)$
- $\delta(q, c, c) = (q, \varepsilon)$

对输入串 $aacbb$ 的分析过程为：

$$\begin{aligned} & (q, aacbb, S) \vdash (q, aacbb, aSb) \vdash (q, acbb, Sb) \\ & \vdash (q, acbb, aSbb) \vdash (q, cbb, Sbb) \vdash (q, cbb, cbb) \\ & \vdash (q, bb, bb) \vdash (q, b, b) \vdash (q, \varepsilon, \varepsilon) \text{(栈空, 接受)} \end{aligned}$$

1.2 PDA的工作原理

- 例：构造PDA接受语言 $L = \{a^n cb^n \mid n \geq 0\}$ (续)



第二节 自上而下分析法存在的问题

2.1 左递归的消除

- 左递归：如果一个文法中有一个非终结符号 A 使得对某个串 α 存在一个推导 $A \xRightarrow{+} A\alpha$ ，则该文法是左递归的(Left Recursive)
- 自上而下语法分析方法不能处理左递归文法，需要通过转换消除左递归
- 直接左递归

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

其中 β_i 都不以 A 开头，将 A 的产生式替换为：

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

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

2.1 左递归的消除

- 例：考虑文法： $S \rightarrow Aa \mid b$, $A \rightarrow Sd \mid \varepsilon$

其中 S 是左递归的： $S \Rightarrow Aa \Rightarrow Sda$ ，但不是直接左递归的，使用 S 的产生式对 $A \rightarrow Sd$ 中的 S 进行代换，可以得到下面的文法：

- $S \rightarrow Aa \mid b$
- $A \rightarrow Aad \mid bd \mid \varepsilon$

然后删除其中的直接左递归，可得

- $S \rightarrow Aa \mid b$
- $A \rightarrow bdA' \mid A'$
- $A' \rightarrow adA' \mid \varepsilon$

2.2 回溯的消除

- **回溯**：进行推导时，如果有多个产生式可选，则选择具有不确定性，若选错则需回到最近选择处进行重选
- 使用预测：设 $A \in V_N$ ，且 $A \rightarrow \alpha \mid \beta$ ，当 A 为栈顶符号，而输入符号为 a 时：
 - 若 $a \in First(\alpha)$ ， $a \notin First(\beta)$ ，则选 $A \rightarrow \alpha$
 - 若 $a \notin First(\alpha)$ ， $a \in First(\beta)$ ，则选 $A \rightarrow \beta$
 - 若 $a \notin First(\alpha)$ ， $a \notin First(\beta)$ ，则表示输入有错
 - 若 $a \in First(\alpha)$ ， $a \in First(\beta)$ ，提取左公因子
- $First(\alpha) = \{a \mid \overset{*}{\alpha} \Rightarrow a \dots, a \in V_T\}$ ，是从文法符号串 α 推导得到的串首符号集合

2.2 回溯的消除

- 提取左公因子：当无法决定在两个产生式中如何选择时，可以通过改写产生式来**推后选择**，等**读入了足够多的输入**再做出正确的选择
- 例如：文法中有产生式 $A \rightarrow \alpha \beta_1 \mid \alpha \beta_2$ ，如果输入的开头是从 α 推导得到的非空串，则可以改造产生式如下：
 - $A \rightarrow \alpha A'$
 - $A' \rightarrow \beta_1 \mid \beta_2$其中 α 为左公因子

2.2 回溯的消除

- 提取左公因子算法

- 输入：文法G

- 输出：等价的提取了左公因子的文法

- 方法：对于每个非终结符 A ，找出它的两个或多个选项之间的最长公共前缀 α ，如果 $\alpha \neq \varepsilon$ 则将所有 A 的产生式 $A \rightarrow \alpha \beta_1 \mid \alpha \beta_2 \mid \cdots \mid \alpha \beta_n \mid \gamma$ 改写为

- $A \rightarrow \alpha A' \mid \gamma$

- $A' \rightarrow \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$

- 其中 γ 表示所有不以 α 开头的产生式体； A' 是非终结符号

- 不断应用此改写方法，直到每个非终结符号的任意两个产生式体都没有公共前缀为止

2.2 回溯的消除

- 例：考虑如下文法

- $S \rightarrow iEtS \mid iEtSeS \mid a$

- $E \rightarrow b$

提取左公因子后，新文法为：

- $S \rightarrow iEtSS' \mid a$

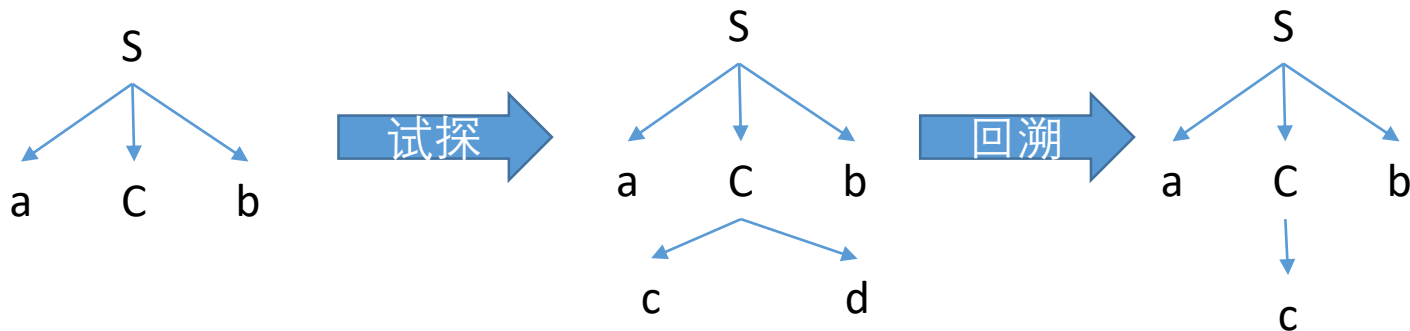
- $S' \rightarrow eS \mid \varepsilon$

- $E \rightarrow b$

第三节 自上而下分析法的一般方法

3. 自上而下分析法的一般方法

- 自上而下分析：对任何输入串，从文法开始符号出发，自上而下，从左到右地构建输入串的语法分析树
 - 本质上是为输入串寻找最左推导
 - 一种试探过程：反复使用不同的产生式以匹配输入
 - 文法应该没有左递归（避免陷入无限循环）
 - 试探与回溯效率低、代价高，在实践中价值不大



第四节 LL(1)文法

4. LL(1)文法

- 回溯消除，对文法符号串 α ，定义：

$$First(\alpha) = \{a \mid \alpha \xRightarrow{*} a \dots, a \in V_T\}$$

[注意：当 $\alpha \xRightarrow{*} \epsilon$ 时， $\epsilon \in First(\alpha)$]

如果对非终结符A的任意两个不同选择 α 和 β ：

$$First(\alpha) \cap First(\beta) = \emptyset$$

当要求A匹配输入串时，就可以准确指派某个选择

- 用提取公因子方法可以使A的所有选择的开始符号集合两两不相交

4. LL(1)文法

例：考虑如下文法产生式

- $E \rightarrow TE'$
 - $E' \rightarrow +TE' \mid \varepsilon$
 - $T \rightarrow FT'$
 - $T' \rightarrow * FT' \mid \varepsilon$
 - $F \rightarrow (E) \mid i$
- 当处理非终结符 E' 时, $First(E') = \{+, \varepsilon\}$, 遇到 ‘+’ 时, 选择 $E' \rightarrow +TE'$; **当遇到 ‘)’ 的时候, 如何选择?**

为了处理非终结符 A 可推出 ε 的情况, 需要知道 **A 后面紧跟的是什么终结符**

4. LL(1)文法

- 如果 ε 属于非终结符 A 的某个选择的开始符号集合，则需要定义 A 的后继符号集合

$$Follow(A) = \{a \mid S \xRightarrow{*} \dots Aa \dots, a \in V_T\}$$

如果 A 是某个句型的最右符号，则令特殊符号

$$\$ \in Follow(A)$$

- 约定 $\$$ 是一个特殊的“结束标记”符号，且不是任何文法符号

4. LL(1)文法

- 计算文法符号 X 的 $First(X)$ ，不断应用下列规则直到没有新的终结符号或 ε 可以被加入到任何 $First$ 集合中
 - 如果 X 是一个终结符号，则 $First(X) = \{X\}$
 - 如果 X 是一个非终结符号，且有产生式
$$X \rightarrow Y_1 Y_2 \dots Y_k (k \geq 1)$$
如果 $[\exists i. a \in First(Y_i)] \wedge [\forall j < i. \varepsilon \in First(Y_j)]$ 则把 a 加入到 $First(X)$ 中
 - 如果有产生式 $X \rightarrow \varepsilon$ ，则 $\varepsilon \in First(X)$

4. LL(1)文法

- 计算文法符号串 $X_1X_2 \dots X_n$ 的 $First$ 集合
 - $\forall a \in First(X_1). a \neq \varepsilon \Rightarrow a \in First(X_1X_2 \dots X_n)$
 - $[\forall i < j, \varepsilon \in First(X_i)] \Rightarrow [\forall a \in First(X_j). a \neq \varepsilon \Rightarrow a \in First(X_1X_2 \dots X_n)]$
 - $\forall i. \varepsilon \in First(X_i) \Rightarrow \varepsilon \in First(X_1X_2 \dots X_n)$
- 例：计算文法符号串 XY 的 $First$ 集合，其中
 - $X \rightarrow \varepsilon \mid a$
 - $Y \rightarrow b$

$$First(XY) = ?$$

4. LL(1)文法

- 计算非终结符号 A 的 $Follow$ 集合，应用如下规则直到没有新的终结符可以加入到任意的 $Follow$ 集合
 - 将 $\$$ 放入 $Follow(S)$ ，其中 S 是开始符号， $\$$ 是输入右端的结束符
 - 若有产生式 $B \rightarrow \alpha A \beta$ ，则 $First(\beta)$ 中除了 ε 之外的符号均在 $Follow(A)$ 中
 - 若有
 - 产生式 $B \rightarrow \alpha A$ ，或
 - 产生式 $B \rightarrow \alpha A \beta$ 且 $\varepsilon \in First(\beta)$则把 $Follow(B)$ 中全部符号放入 $Follow(A)$

4. LL(1)文法

- 例：考虑如下文法产生式

- $E \rightarrow TE'$

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

- $T \rightarrow FT'$

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

- $F \rightarrow (E)$

$$\textit{Follow}(T) = ?$$

4. LL(1)文法

- 例：考虑如下文法产生式

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E)$

$Follow(E) = \{\$,)\}$

- 因 E 是开始符号，所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$ ，则 $First()$ 中除 ε 外的符号均在 $Follow(E)$ 中，所以 $) \in Follow(E)$

4. LL(1)文法

- 例：考虑如下文法产生式

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E)$

$Follow(E) = \{\$,)\}$

- 因 E 是开始符号，所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$ ，则 $First()$ 中除 ε 外的符号均在 $Follow(E)$ 中，所以 $) \in Follow(E)$

$Follow(E') = \{\$,)\}$

- 因 $E \rightarrow TE'$ ，所以 $Follow(E)$ 中所有符号均在 $Follow(E')$

4. LL(1)文法

- 例：考虑如下文法产生式

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E)$

$Follow(T) = \{+, \$,)\}$

- 因 $E \rightarrow TE'$ ，所以 $First(E')$ 中除 ε 外的符号均在 $Follow(T)$ 中，则 $+\in Follow(T)$
- $E' \rightarrow +TE'$ (同上)
- 因 $E' \rightarrow +TE'$ ，且 $\varepsilon \in First(E')$ 则 $Follow(E')$ 中所有符号均在 $Follow(T)$ 中

$Follow(E) = \{ \$,) \}$

- 因 E 是开始符号，所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$ ，则 $First(E)$ 中除 ε 外的符号均在 $Follow(E)$ 中，所以 $) \in Follow(E)$

$Follow(E') = \{ \$,) \}$

- 因 $E \rightarrow TE'$ ，所以 $Follow(E)$ 中所有符号均在 $Follow(E')$

4. LL(1)文法

- 例：考虑如下文法产生式

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E)$

$Follow(T) = \{+, \$,)\}$

- 因 $E \rightarrow TE'$ ，所以 $First(E')$ 中除 ε 外的符号均在 $Follow(T)$ 中，则 $+\in Follow(T)$
- $E' \rightarrow +TE'$ (同上)
- 因 $E' \rightarrow +TE'$ ，且 $\varepsilon \in First(E')$ 则 $Follow(E')$ 中所有符号均在 $Follow(T)$ 中

$Follow(E) = \{ \$,) \}$

- 因 E 是开始符号，所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$ ，则 $First(E)$ 中除 ε 外的符号均在 $Follow(E)$ 中，所以 $) \in Follow(E)$

$Follow(T') = \{+, \$,)\}$

- 因 $T \rightarrow FT'$ ，所以 $Follow(T)$ 中所有符号均在 $Follow(T')$

$Follow(E') = \{ \$,) \}$

- 因 $E \rightarrow TE'$ ，所以 $Follow(E)$ 中所有符号均在 $Follow(E')$

4. LL(1)文法

• 例：考虑如下文法产生式

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E)$

$Follow(E) = \{ \$,) \}$

- 因 E 是开始符号, 所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$, 则 $First()$ 中除 ε 外的符号均在 $Follow(E)$ 中, 所以 $) \in Follow(E)$

$Follow(E') = \{ \$,) \}$

- 因 $E \rightarrow TE'$, 所以 $Follow(E)$ 中所有符号均在 $Follow(E')$

$Follow(T) = \{ +, \$,) \}$

- 因 $E \rightarrow TE'$, 所以 $First(E')$ 中除 ε 外的符号均在 $Follow(T)$ 中, 则 $+ \in Follow(T)$
- $E' \rightarrow +TE'$ (同上)
- 因 $E' \rightarrow +TE'$, 且 $\varepsilon \in First(E')$ 则 $Follow(E')$ 中所有符号均在 $Follow(T)$ 中

$Follow(T') = \{ +, \$,) \}$

- 因 $T \rightarrow FT'$, 所以 $Follow(T)$ 中所有符号均在 $Follow(T')$

$Follow(F) = \{ *, +, \$,) \}$

- 因 $T \rightarrow FT'$, 所以 $First(T')$ 中除 ε 外的符号均在 $Follow(F)$ 中, 则 $* \in Follow(F)$
- 因 $T' \rightarrow * FT'$, 且 $\varepsilon \in First(T')$ 则 $Follow(T')$ 中所有符号均在 $Follow(F)$ 中

4. LL(1)文法

- 如果文法G中所有产生式 $A \rightarrow \alpha \mid \beta$ 满足如下条件
 1. $First(\alpha) \cap First(\beta) = \emptyset$
 2. 若 $\beta \xRightarrow{*} \varepsilon$, 则 $First(\alpha) \cap Follow(A) = \emptyset$ ($\alpha \xRightarrow{*} \varepsilon$ 也类似)

则G是LL(1)文法

- ✓ 第一个L代表从左(**L**eft)向右地扫描输入
 - ✓ 第二个L代表产生最左推导(**L**eftmost derivation)
 - ✓ 1表示在决定语法分析器的每步动作时需要向前看下一个输入符号
- LL(1)文法: 不是二义的, 不含左递归, 没有公共左因子

第五节 递归下降和非递归的预测分析

5.1 递归下降的预测分析

- **预测分析：**根据当前输入符号为非终结符确定采用哪种选择(LL(1)文法满足要求)
- **递归下降(Recursive-Descent)的预测分析：**
 - 为每一个非终结符写一个分析过程
 - 文法的定义是递归的，因此过程也是递归的
 - 处理输入串时，首先执行开始符号所对应的过程
 - 然后根据产生式右部出现的非终结符，依次调用相应的过程
 - 逐步下降的过程调用序列隐含地建立了输入的分析树

5.1 递归下降的预测分析

- 一个非终结符A对应的典型过程

```
A() {  
    选择一个A的产生式,  $A \rightarrow X_1X_2 \dots X_k$ ;  
    for ( $i = 1$  to  $k$ ) {  
        if ( $X_i$ 是一个非终结符)  
            调用过程 $X_i()$ ;  
        else if ( $X_i$ 等于当前输入符)  
            读取下一个输入  
        else /*发生错误*/;  
    }  
}
```

5.1 递归下降的预测分析

- 例：考虑文法

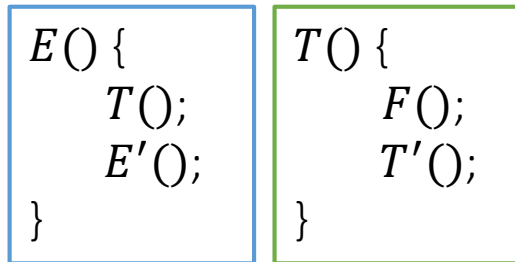
- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E) \mid i$

$E() \{$
 $T();$
 $E'();$
 $\}$

5.1 递归下降的预测分析

- 例：考虑文法

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow *FT' \mid \varepsilon$
- $F \rightarrow (E) \mid i$



5.1 递归下降的预测分析

- 例：考虑文法

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E) \mid i$

$E() \{$ $T();$ $E'();$ $\}$	$T() \{$ $F();$ $T'();$ $\}$
---------------------------------------	---------------------------------------

```
E'() {  
    if (token == '+') {  
        nextToken();  
        T();  
        E'();  
    }  
}
```

5.1 递归下降的预测分析

- 例：考虑文法

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow *FT' \mid \varepsilon$
- $F \rightarrow (E) \mid i$

<pre>E() { T(); E'(); }</pre>	<pre>T() { F(); T'(); }</pre>
---	---

```
E'() {  
    if (token == '+') {  
        nextToken();  
        T();  
        E'();  
    }  
}
```

```
T'() {  
    if (token == '*') {  
        nextToken();  
        F();  
        T'();  
    }  
}
```

5.1 递归下降的预测分析

- 例：考虑文法

- $E \rightarrow TE'$
- $E' \rightarrow +TE' \mid \varepsilon$
- $T \rightarrow FT'$
- $T' \rightarrow * FT' \mid \varepsilon$
- $F \rightarrow (E) \mid i$

```
E() {  
    T();  
    E'();  
}
```

```
T() {  
    F();  
    T'();  
}
```

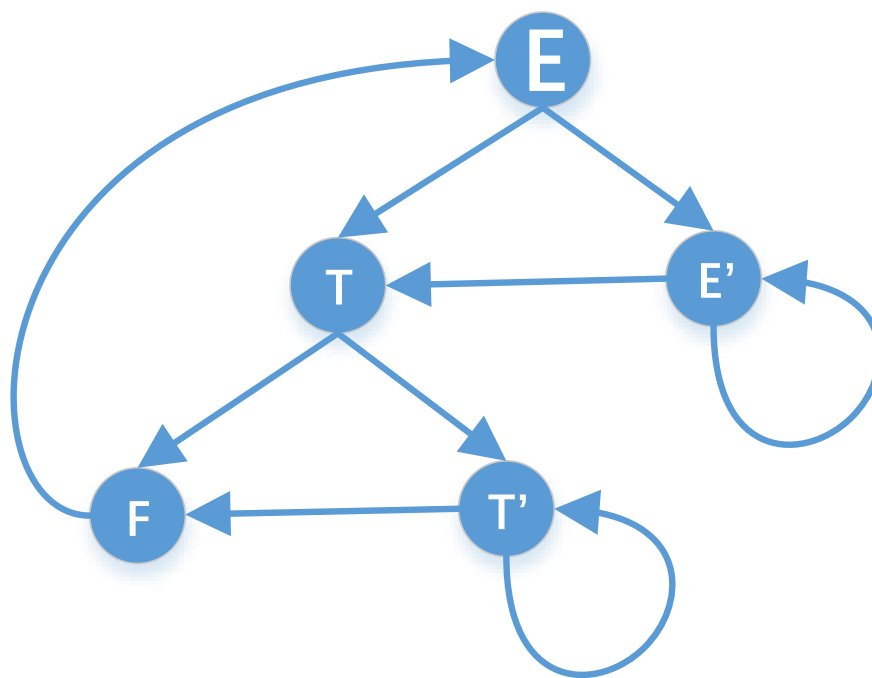
```
E'() {  
    if (token == '+') {  
        nextToken();  
        T();  
        E'();  
    }  
}
```

```
T'() {  
    if (token == '*') {  
        nextToken();  
        F();  
        T'();  
    }  
}
```

```
F() {  
    if (token == 'i') {  
        nextToken();  
    }  
    else if (token == '(') {  
        nextToken();  
        E();  
        if (token == ')') {  
            nextToken();  
        } else ERROR;  
    } else ERROR;  
}
```

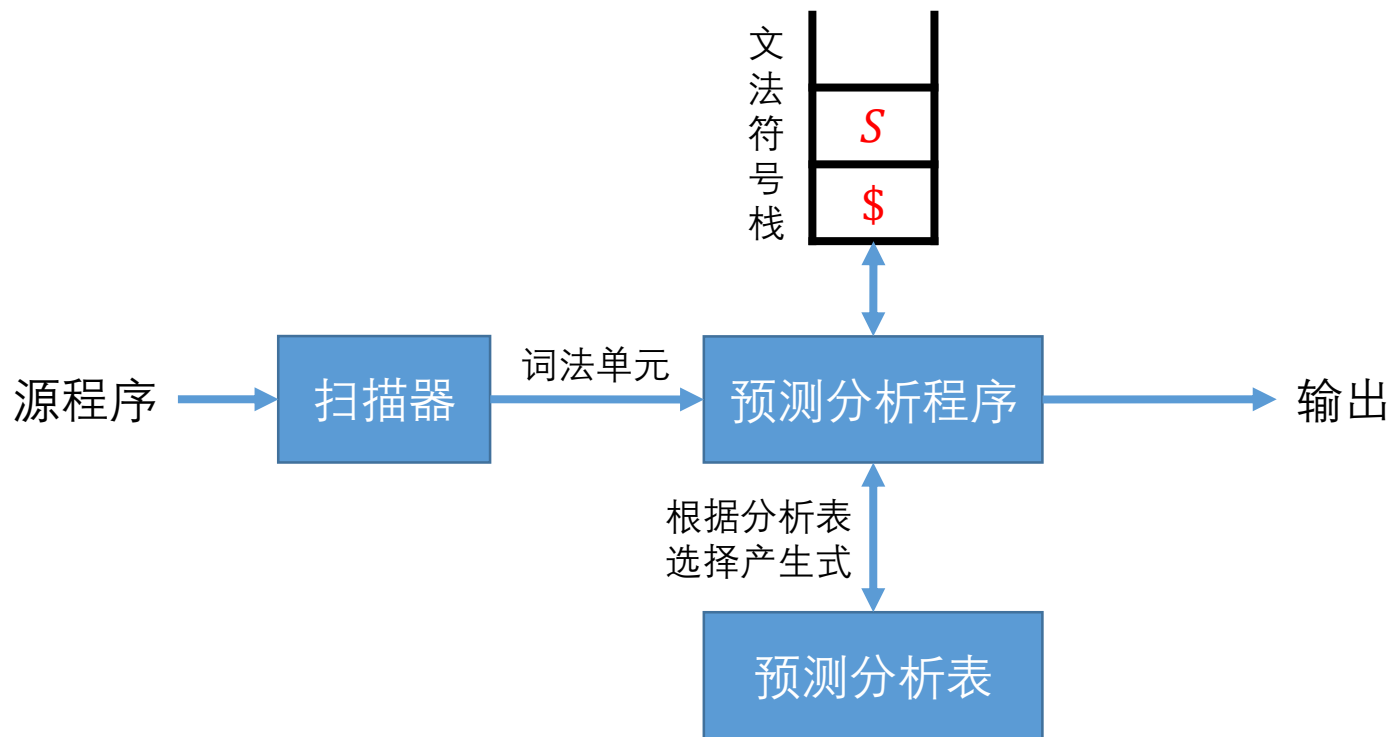
5.1 递归下降的预测分析

- 例：考虑文法
 - $E \rightarrow TE'$
 - $E' \rightarrow +TE' \mid \varepsilon$
 - $T \rightarrow FT'$
 - $T' \rightarrow * FT' \mid \varepsilon$
 - $F \rightarrow (E) \mid i$



5.2 非递归的预测分析

- 分析表驱动的预测分析器模型



5.2 非递归的预测分析

- 分析表驱动的预测分析算法

输入:

串 w 和 G 的预测分析表 M

输出:

若 w 在 $L(G)$ 中则输出 w 的一个最左推导; 否则给出错误指示

方法:

开始时, 输入缓冲区中是 $w\$$;
 G 的开始符号 S 位于栈顶,
 S 的下面是符号 $\$$

```
设置  $ip$  指向  $w$  的第一个符号  
 $X :=$  栈顶符号;  
while( $X \neq \$$ ){  
    if( $X$  等于  $ip$  所指向的符号){  
        弹出栈顶符号;  
         $ip := \text{nextToken}()$ ;  
    }  
    else if( $X$  是终结符号) error();  
    else if( $M[X, a]$  是报错条目) error();  
    else if( $M[X, a] = X \rightarrow Y_1 Y_2 \dots Y_k$ ){  
        输出产生式  $X \rightarrow Y_1 Y_2 \dots Y_k$ ;  
        弹出栈顶符号;  
        将  $Y_k, Y_{k-1}, \dots, Y_1$  压栈,  $Y_1$  位于栈顶;  
    }  
     $X :=$  栈顶符号;  
}
```

5.2 非递归的预测分析

- 例：考虑文法

$$\begin{aligned} E &\rightarrow TE' & E' &\rightarrow +TE' \mid \varepsilon & T &\rightarrow FT' \\ F &\rightarrow (E) \mid \mathbf{id} & T' &\rightarrow * FT' \mid \varepsilon \end{aligned}$$

非终结 符号	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	$F \rightarrow \mathbf{id}$	—	—	$F \rightarrow (E)$	—	—

预测分析表M

5.2 非递归的预测分析

- 处理输入 **id + id * id\$** 推导过程

$E \xrightarrow{\text{id}} TE'$	$E \xrightarrow{(} TE'$
$E' \xrightarrow{+} +TE'$	$E' \xrightarrow{)} \varepsilon$
$E' \xrightarrow{\$} \varepsilon$	$T \xrightarrow{\text{id}} FT'$
$T \xrightarrow{(} FT'$	$T' \xrightarrow{+} \varepsilon$
$T' \xrightarrow{*} *FT'$	$T' \xrightarrow{)} \varepsilon$
$T' \xrightarrow{\$} \varepsilon$	$F \xrightarrow{\text{id}} \text{id}$
$F \xrightarrow{(} (E)$	

迁移关系表示
预测分析表M

已匹配	栈	输入	动作
	$E\$$	id + id * id\$	
	$TE'\$$	id + id * id\$	输出 $E \rightarrow TE'$
	$FT'E'\$$	id + id * id\$	输出 $T \rightarrow FT'$
	id $T'E'\$$	id + id * id\$	输出 $F \rightarrow \text{id}$
id	$T'E'\$$	+id * id\$	匹配 id
id	$E'\$$	+id * id\$	输出 $T' \rightarrow \varepsilon$
id	$+TE'\$$	+id * id\$	输出 $E' \rightarrow +TE'$
id +	$TE'\$$	id * id\$	匹配 +
id +	$FT'E'\$$	id * id\$	输出 $T \rightarrow FT'$
?	?	?	?

第六节 预测分析表的构造

6. 预测分析表的构造

- 预测分析表构造算法

- 输入：文法G

- 输出：预测分析表M

- 方法：对于G的每个产生式 $A \rightarrow \alpha$ ：

- 对于 $First(\alpha)$ 中的每个终结符 c ，将 $A \rightarrow \alpha$ 加入到 $M[A, c]$ 中

- 若 ε 在 $First(\alpha)$ 中，那么对于 $Follow(A)$ 中的每个终结符 d （包括特殊符号 $\$$ ），将 $A \rightarrow \alpha$ 加入到 $M[A, d]$ 中

- 完成上述操作之后，若 $M[A, c]$ 中没有产生式，则设置为‘—’，代表错误

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First
E	$\{ (, \mathbf{id} \}$
E'	$\{ +, \varepsilon \}$
T	$\{ (, \mathbf{id} \}$
F	$\{ (, \mathbf{id} \}$
T'	$\{ *, \varepsilon \}$

$$Follow(E) = \{ \$,) \}$$

- 因 E 是开始符号，所以 $\$ \in Follow(E)$
- 因 $F \rightarrow (E)$ ，则 $First()$ 中除 ε 外的符号均在 $Follow(E)$ 中，所以 $) \in Follow(E)$

$$Follow(E') = \{ \$,) \}$$

- 因 $E \rightarrow TE'$ ，所以 $Follow(E)$ 中所有符号均在 $Follow(E')$

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First
E	$\{ (, \mathbf{id} \}$
E'	$\{ +, \varepsilon \}$
T	$\{ (, \mathbf{id} \}$
F	$\{ (, \mathbf{id} \}$
T'	$\{ *, \varepsilon \}$

$$Follow(T) = \{ +, \$,) \}$$

- 因 $E \rightarrow TE'$ ，所以 $First(E')$ 中除 ε 外的符号均在 $Follow(T)$ 中，则 $+ \in Follow(T)$
- $E' \rightarrow +TE'$ (同上)
- 因 $E' \rightarrow +TE'$ ，且 $\varepsilon \in First(E')$ 则 $Follow(E')$ 中所有符号均在 $Follow(T)$ 中

$$Follow(T') = \{ +, \$,) \}$$

- 因 $T \rightarrow FT'$ ，所以 $Follow(T)$ 中所有符号均在 $Follow(T')$

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First
E	$\{ (, \mathbf{id} \}$
E'	$\{ +, \varepsilon \}$
T	$\{ (, \mathbf{id} \}$
F	$\{ (, \mathbf{id} \}$
T'	$\{ *, \varepsilon \}$

$$Follow(F) = \{ *, +, \$,) \}$$

- 因 $T \rightarrow FT'$ ，所以 $First(T')$ 中除 ε 外的符号均在 $Follow(F)$ 中，则 $* \in Follow(F)$
- 因 $T' \rightarrow * FT'$ ，且 $\varepsilon \in First(T')$ 则 $Follow(T')$ 中所有符号均在 $Follow(F)$ 中

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First	Follow
E	{(, id}	{\$,)}
E'	{+, ε }	{\$,)}
T	{(, id}	{+, \$,)}
F	{(, id}	{*, +, \$,)}
T'	{*, ε }	{+, \$,)}

对于 $E \rightarrow TE'$, 有 $First(TE') = \{(, \mathbf{id}\}$

- 所以将 $E \rightarrow TE'$ 加入 $M[E, (]$ 和 $M[E, \mathbf{id}]$ 中
- $\varepsilon \notin First(TE')$, 没有操作

非终结符号	输入符号					
	id	+	*	()	\$
E	$E \rightarrow TE'$	—	—	$E \rightarrow TE'$	—	—

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First	Follow
E	$\{(\mathbf{id})\}$	$\{\$,)\}$
E'	$\{+, \varepsilon\}$	$\{\$,)\}$
T	$\{(\mathbf{id})\}$	$\{+, \$,)\}$
F	$\{(\mathbf{id})\}$	$\{*, +, \$,)\}$
T'	$\{*, \varepsilon\}$	$\{+, \$,)\}$

对于 $E' \rightarrow \varepsilon$ ，有 $First(\varepsilon) = \{\varepsilon\}$ 对于 $Follow(E')$ 中的每个终结符

- $\$$ ，将 $E' \rightarrow \varepsilon$ 加入 $M[E', \$]$
- $)$ ，将 $E' \rightarrow \varepsilon$ 加入 $M[E',)]$

对于 $E' \rightarrow +TE'$ ，有 $First(+TE') = \{+\}$

- 将 $E' \rightarrow +TE'$ 加入 $M[E', +]$ 中
- $\varepsilon \notin First(+TE')$ ，没有操作

非终结符号	输入符号					
	\mathbf{id}	$+$	$*$	$($	$)$	$\$$
E	$E \rightarrow TE'$	—	—	$E \rightarrow TE'$	—	—
E'	—	$E' \rightarrow +TE'$	—	—	$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First	Follow
E	{(, id }	{\$,)}
E'	{+, ε }	{\$,)}
T	{(, id }	{+, \$,)}
F	{(, id }	{*, +, \$,)}
T'	{*, ε }	{+, \$,)}

对于 $T \rightarrow FT'$, 有 $First(FT') = \{(, \mathbf{id}\}$

- 将 $T \rightarrow FT'$ 加入 $M[T, (]$ 和 $M[T, \mathbf{id}]$ 中
- $\varepsilon \notin First(FT')$, 没有操作

非终结符号	输入符号					
	id	+	*	()	\$
E	$E \rightarrow TE'$	—	—	$E \rightarrow TE'$	—	—
E'	—	$E' \rightarrow +TE'$	—	—	$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$	—	—	$T \rightarrow FT'$	—	—

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

NT	First	Follow	非终结符号	输入符号					
				id	+	*	()	\$
E	{(, id}	{\$,)}	E	$E \rightarrow TE'$	—	—	$E \rightarrow TE'$	—	—
E'	{+, ε }	{\$,)}	E'	—	$E' \rightarrow +TE'$	—	—	$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	{(, id}	{+, \$,)}	T	$T \rightarrow FT'$	—	—	$T \rightarrow FT'$	—	—
F	{(, id}	{*, +, \$,)}	T'	—	$T' \rightarrow \varepsilon$	$T' \rightarrow * FT'$	—	$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
T'	{*, ε }	{+, \$,)}							

对于 $T' \rightarrow * FT'$, 有 $First(* FT') = \{*\}$

- 将 $T' \rightarrow * FT'$ 加入 $M[T', *]$ 中
- $\varepsilon \notin First(FT')$, 没有操作

对于 $T' \rightarrow \varepsilon$, 有 $First(\varepsilon) = \{\varepsilon\}$ 对于 $Follow(T')$ 中的每个终结符

- +, 将 $T' \rightarrow \varepsilon$ 加入 $M[T', +]$
- \$, 将 $T' \rightarrow \varepsilon$ 加入 $M[T', \$]$
-), 将 $T' \rightarrow \varepsilon$ 加入 $M[T',)]$

6. 预测分析表的构造

- 例：考虑文法

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

非终结 符号	输入符号					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$	—	—	$E \rightarrow TE'$	—	—
<i>E'</i>	—	$E' \rightarrow +TE'$	—	—	$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
<i>T</i>	$T \rightarrow FT'$	—	—	$T \rightarrow FT'$	—	—
<i>T'</i>	—	$T' \rightarrow \varepsilon$	$T' \rightarrow * FT'$	—	$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
<i>F</i>	$F \rightarrow \mathbf{id}$	—	—	$F \rightarrow (E)$	—	—

对于 $F \rightarrow (E)$, 有 $First((E)) = \{(\}$
 • 将 $F \rightarrow (E)$ 加入 $M[F, (]$ 中

对于 $F \rightarrow \mathbf{id}$, 有 $First(\mathbf{id}) = \{\mathbf{id}\}$
 • 将 $F \rightarrow \mathbf{id}$ 加入 $M[F, \mathbf{id}]$ 中

小结

一. 下推自动机概念，形式定义，用途

二. 左递归，回溯等问题

四. LL(1)文法定义，First、Follow集合

五. 递归下降分析、非递归的表驱动分析

六. 预测分析表的构造算法