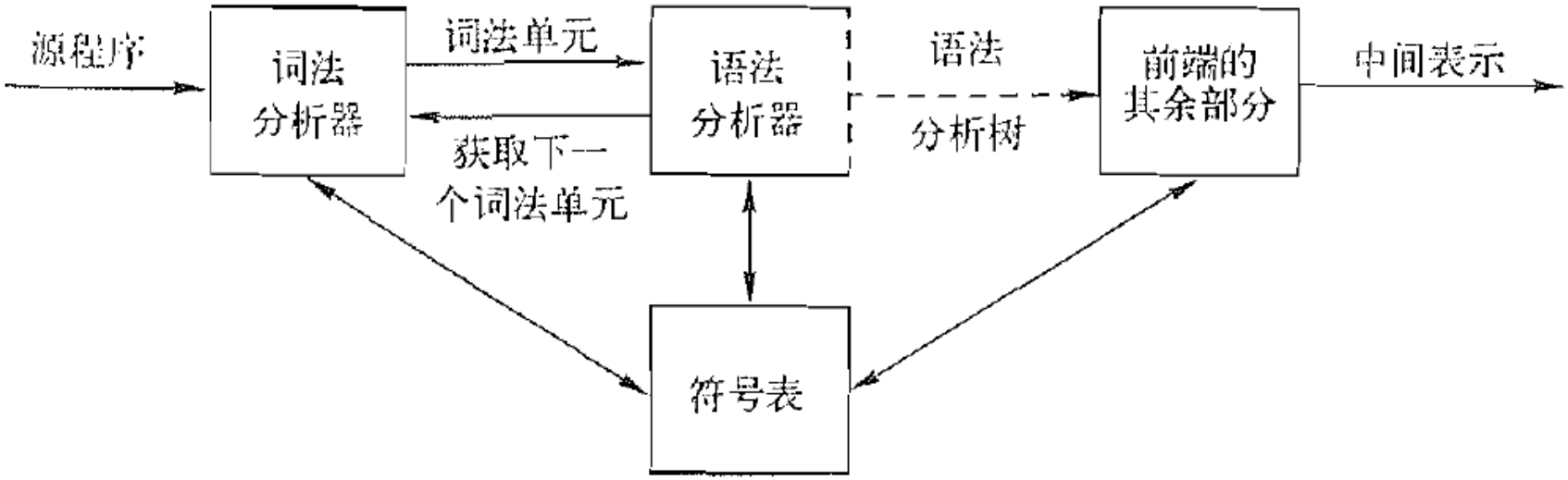
语法分析

魏恒峰

hfwei@nju.edu.cn

2020 年 12 月 2 日

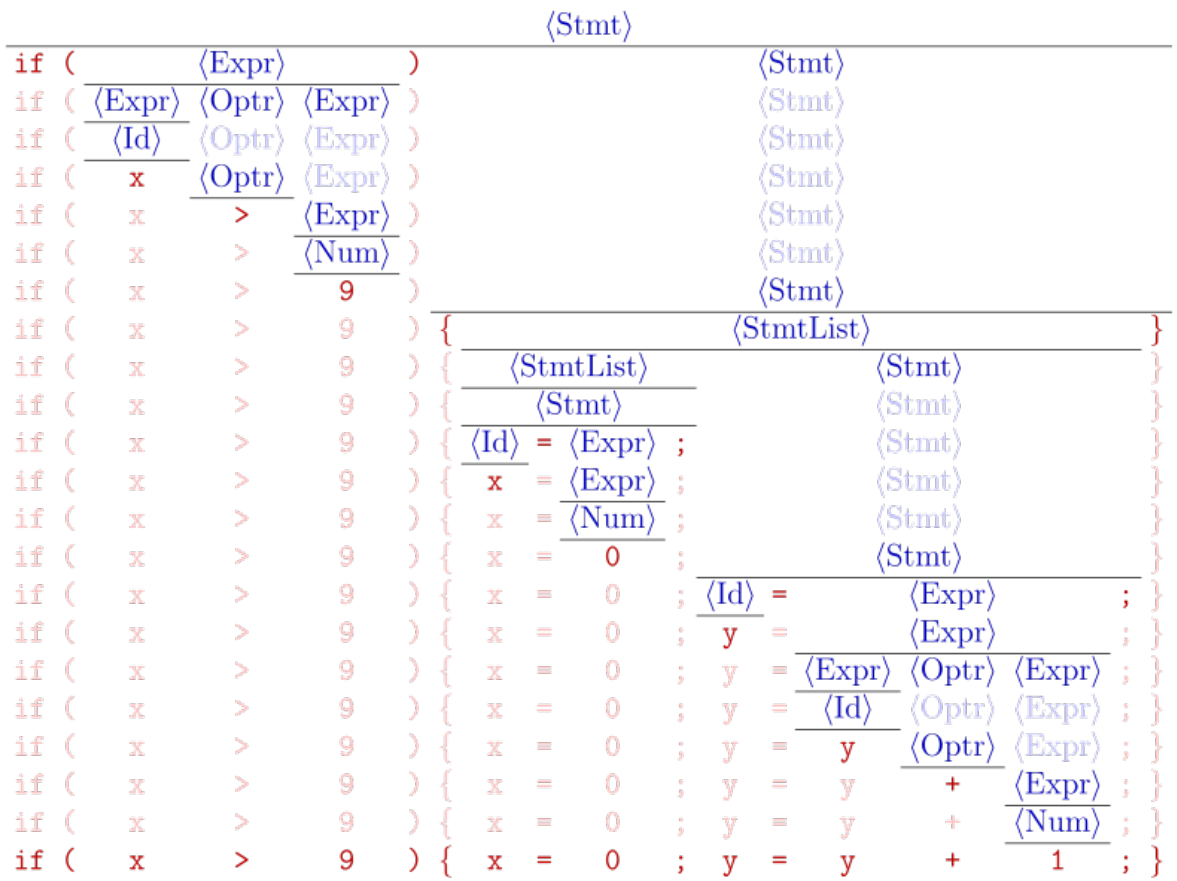
输入: 词法单元流 & **语言的语法规则**



输出: 语法分析树 (Parse Tree)

语法分析举例

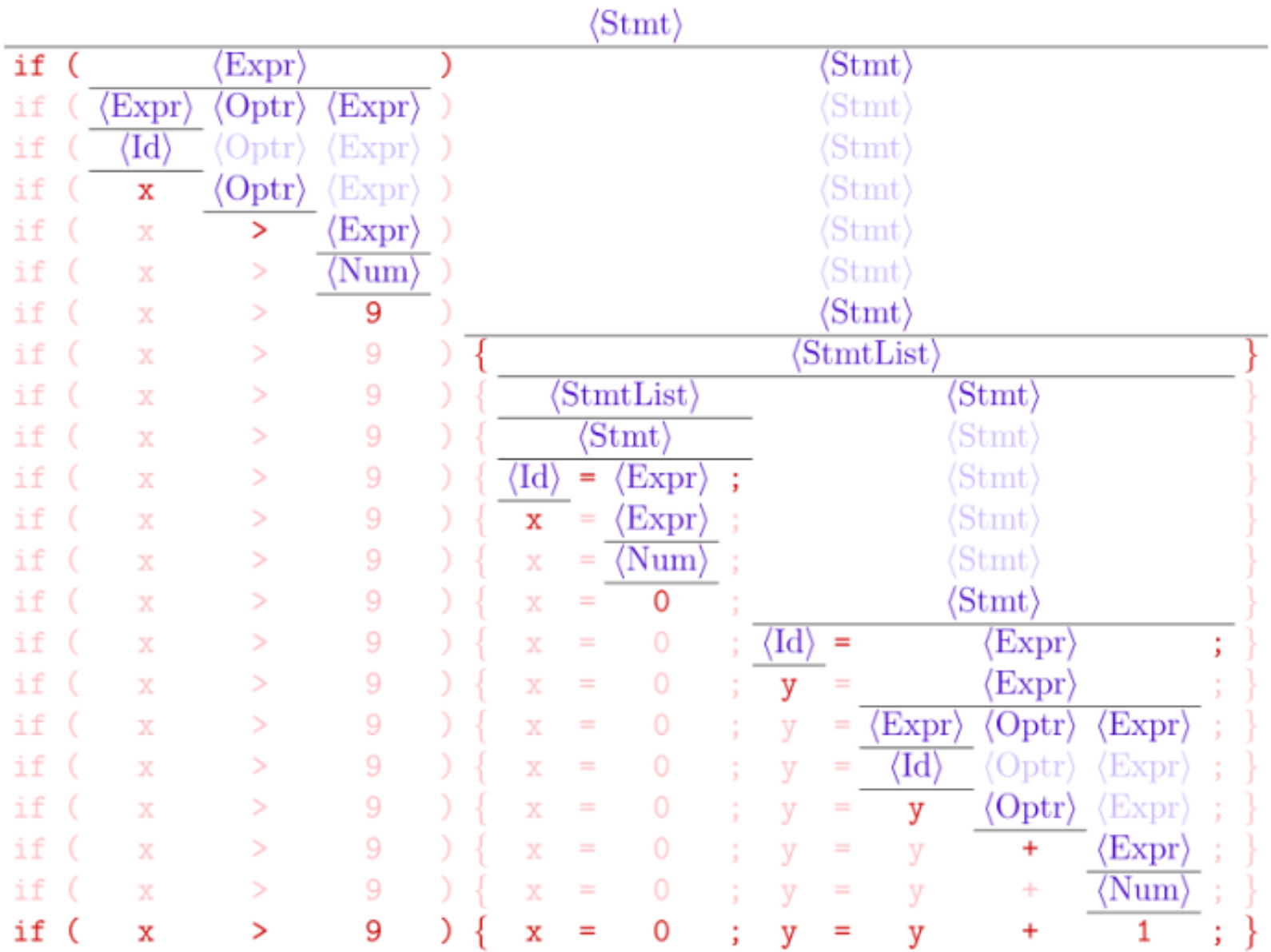
⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



语法分析阶段的主题之一：**上下文无关文法**

⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +

语法分析阶段的主题之二: **构建语法分析树**

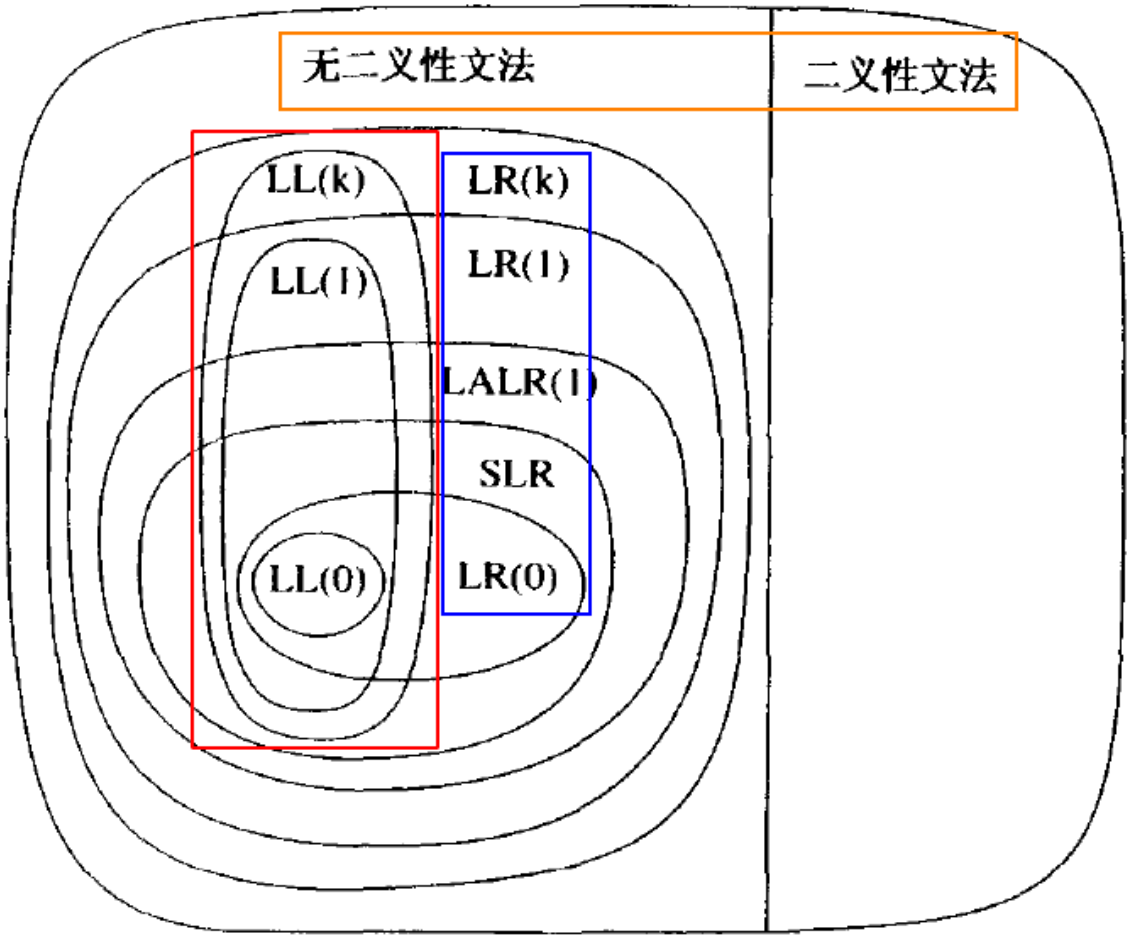


语法分析阶段的主题之三: **错误恢复**

报错、**恢复**、继续分析

只考虑**无二义性**的文法

这意味着, 每个句子对应唯一的一棵语法分析树



今日份主题: LR **语法分析器**

自底向上的、

不断归约的、

基于句柄识别自动机的、

适用于LR **文法**的、

LR 语法分析器

自底向上构建语法分析树

根节点是文法的起始符号 S

叶节点是词法单元流 $w\$$

仅包含终结符号与特殊的**文件结束符** $\$$

自底向上构建语法分析树

根节点是文法的起始符号 S

每个**中间非终结符节点**表示**使用它的某条产生式进行归约**

叶节点是词法单元流 $w\$$

仅包含终结符号与特殊的**文件结束符** $\$$

自顶向下的“推导”与自底向上的“归约”

$$E \underset{\text{rm}}{\Longrightarrow} T \underset{\text{rm}}{\Longrightarrow} T * F \underset{\text{rm}}{\Longrightarrow} T * \mathbf{id} \underset{\text{rm}}{\Longrightarrow} F * \mathbf{id} \underset{\text{rm}}{\Longrightarrow} \mathbf{id} * \mathbf{id}$$

(1) $E \to E + T$

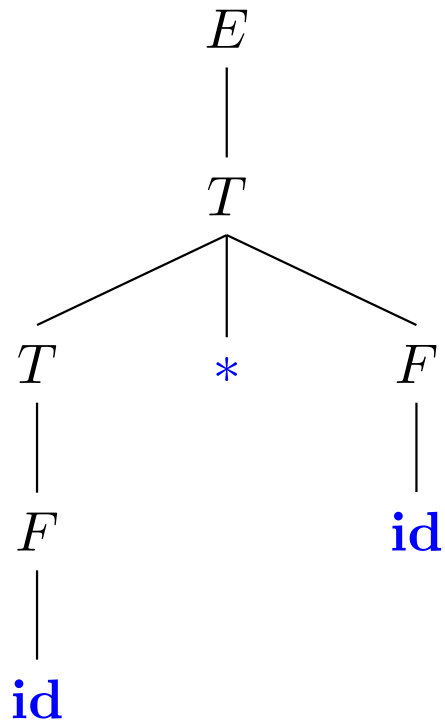
(2) $E \to T$

(3) $T \to T * F$

(4) $T \to F$

(5) $F \to (E)$

(6) $F \to \mathbf{id}$



$w = \mathbf{id} * \mathbf{id}$

$$E \Longleftarrow T \Longleftarrow T * F \Longleftarrow T * \mathbf{id} \Longleftarrow F * \mathbf{id} \Longleftarrow \mathbf{id} * \mathbf{id}$$

“推导” $(A \to \alpha)$ 与 “归约” $(A \leftarrow \alpha)$

$$S \triangleq \gamma_0 \Longrightarrow \ldots \gamma_{i-1} \Longrightarrow \gamma_i \Longrightarrow \gamma_{r+1} \Longrightarrow \cdots \Longrightarrow r_n = w$$

$$S \triangleq \gamma_0 \Longleftarrow \ldots \gamma_{i-1} \Longleftarrow \gamma_i \Longleftarrow \gamma_{r+1} \Longleftarrow \cdots \Longleftarrow r_n = w$$

自底向上语法分析器为输入构造**反向推导**

LR 语法分析器

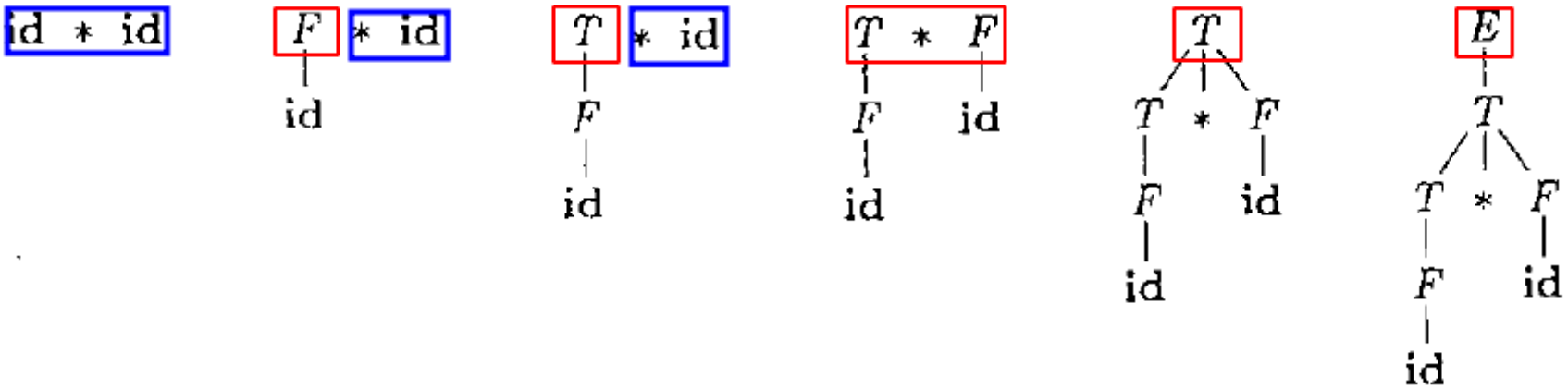
L：**从左向右** (Left-to-right) 扫描输入

R：构建**反向** (Reverse) **最右推导**

“反向最右推导”与“从左到右扫描”相一致

LR 语法分析器的状态

在任意时刻, 语法分析树的**上边缘**与**剩余的输入**构成当前句型



$$E \Longleftarrow T \Longleftarrow T * F \Longleftarrow T * \mathbf{id} \Longleftarrow F * \mathbf{id} \Longleftarrow \mathbf{id} * \mathbf{id}$$

LR 语法分析器使用**栈**存储语法分析树的**上边缘**

它包含了语法分析器目前所知的所有信息

板书演示 “栈” 上操作

(1) $E \to E + T$

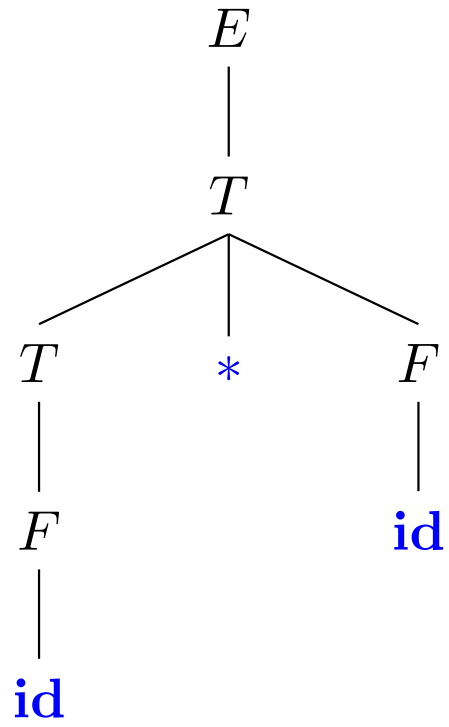
(2) $E \to T$

(3) $T \to T * F$

(4) $T \to F$

(5) $F \to (E)$

(6) $F \to \mathbf{id}$



$w = \mathbf{id} * \mathbf{id}$

两大操作: **移入输入符号** 与 **按产生式归约**

直到栈中仅剩开始符号 S, 且输入已结束, 则成功停止

基于栈的 LR 语法分析器

Q_1：何时归约? (何时移入?)

Q_2：按哪条产生式进行归约?

基于栈的 LR 语法分析器

(1) $E \to E + T$

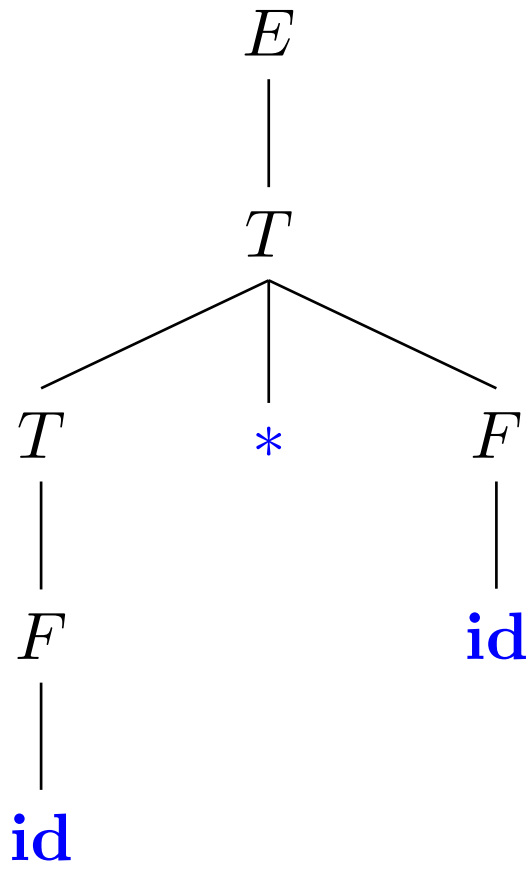
(2) $E \to T$

(3) $T \to T * F$

(4) $T \to F$

(5) $F \to (E)$

(6) $F \to \mathbf{id}$



为什么第二个 F 以 $T * F$ 整体被归约为 T?

这与**栈**的当前状态 "$T * F$" 相关

LR 分析表指导 LR 语法分析器

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

在**当前状态 (编号)**下, 面对**当前文法符号**时, 该采取什么**动作**

ACTION 表指明动作, GOTO 表仅用于归约时的状态转换

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

sn	移入输入符号, 并进入**状态** n
rk	使用k **号产生式**进行归约
gn	转换到**状态** n
acc	成功接受, 结束
空白	错误

再次板书演示 “栈” 上操作: 移入与归约

(1) $E \to E + T$

(2) $E \to T$

(3) $T \to T * F$

(4) $T \to F$

(5) $F \to (E)$

(6) $F \to \mathbf{id}$

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

$w = \mathbf{id} * \mathbf{id\$}$

栈中存储语法分析器的**状态 (编号)**, “编码” 了语法分析树的上边缘

```
1:  procedure LR()
2:      PUSH(S, s_0)                                  ▷ 或 PUSH(S, $_{s_0})
3:      token ← NEXT-TOKEN()
4:      while (1) do
5:          s ← TOP(S)
6:          if ACTION[s, token] = s_i then            ▷ 移入
7:              PUSH(S, i)                            ▷ 或 PUSH(S, token_{s_i})
8:              token ← NEXT-TOKEN()
9:          else if ACTION[s, token] = r_j then       ▷ 归约; j : A → α
10:             |α| 次 POP(S)
11:             s ← TOP(S)
12:             PUSH(S, GOTO[s, A])   ▷ 转换状态; 或 PUSH(S, A_{GOTO[s,A]})
13:         else if ACTION[s, token] = acc then       ▷ 接受
14:             break
15:         else
16:             ERROR(...)
```

行号	栈	符号	输入	动作
(1)	0	\$	id * id \$	移入到 5
(2)	0 5	\$ id	* id \$	按照 $F \to$ id 归约
(3)	0 3	\$ F	* id \$	按照 $T \to F$ 归约
(4)	0 2	\$ T	* id \$	移入到 7
(5)	0 2 7	\$ T *	id \$	移入到 5
(6)	0 2 7 5	\$ T * id	\$	按照 $F \to$ id 归约
(7)	0 2 7 10	\$ T * F	\$	按照 $T \to T * F$ 归约
(8)	0 2	\$ T	\$	按照 $E \to T$ 归约
(9)	0 1	\$ E	\$	接受

句型

$w =$ **id** $*$ **id**\$ 的分析过程

如何构造 LR 分析表?

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

在**当前状态 (编号)**下, 面对**当前文法符号**时, 该采取什么**动作**

状态是什么？如何跟踪状态？

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

状态是语法分析树的上边缘，存储在栈中

可以用**自动机**跟踪状态变化（自动机中的路径 $\Leftrightarrow$ 栈中符号/状态编号）

何时归约？使用哪条产生式进行归约？

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

必要条件: 当前状态中, 已观察到某个产生式的完整右部

对于 LR 文法, 这是当前**唯一**的选择

何时归约？使用哪条产生式进行归约？

Definition (句柄 (Handle))

在输入串的 (唯一) 反向最右推导中, **如果**下一步是逆用产生式 $A \to \alpha$ 将 α 归约为 A, 则称 α 是当前句型的**句柄**。

最右句型	句柄	归约用的产生式
$\mathbf{id}_1 * \mathbf{id}_2$	$\mathbf{id}_1$	$F \to \mathbf{id}$
$F * \mathbf{id}_2$	F	$T \to F$
$T * \mathbf{id}_2$	$\mathbf{id}_2$	$F \to \mathbf{id}$
$T * F$	$T * F$	$T \to T * F$
T	T	$E \to T$

LR 语法分析器的关键就是高效**寻找每个归约步骤所使用的句柄**。

句柄可能在哪里？

Theorem

存在一种 LR 语法分析方法，保证**句柄总是出现在栈顶**。

句柄可能在哪里？

Theorem

存在一种 LR 语法分析方法，保证**句柄总是出现在栈顶**。

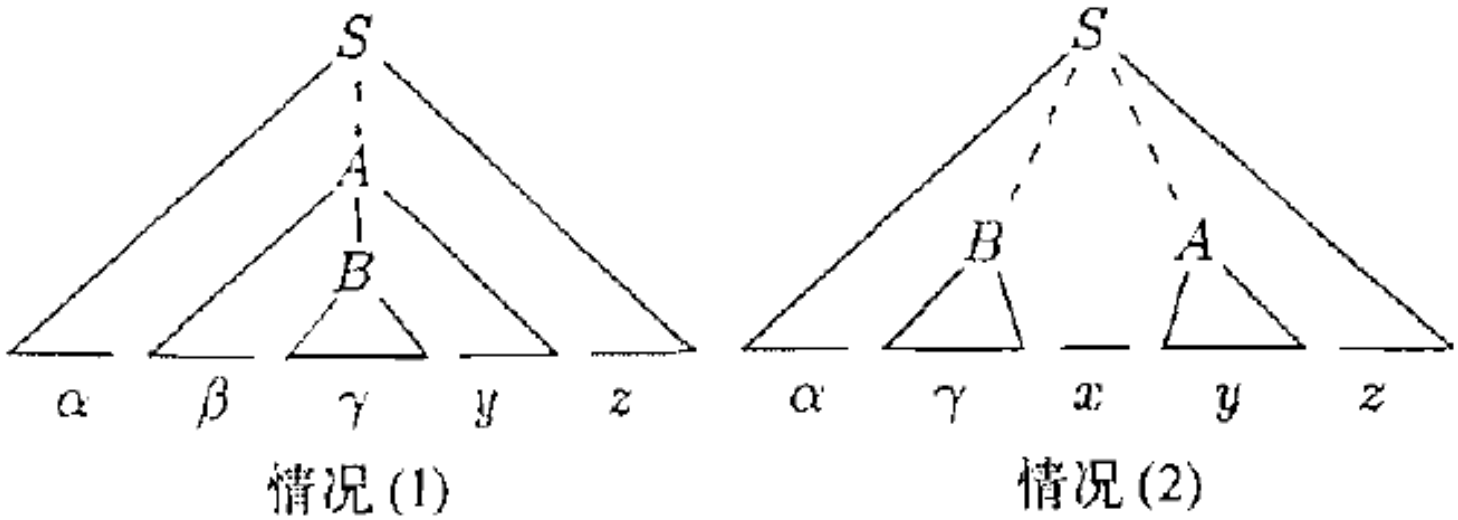


图 4-29　一个最右推导中两个连续步骤的两种情况

$$S \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha Az \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha\beta Byz \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha\beta\gamma yz \quad S \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha BxAz \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha Bxyz \overset{*}{\underset{\text{rm}}{\Longrightarrow}} \alpha\gamma xyz$$

可以用**自动机**跟踪状态变化

(自动机中的路径 $\Leftrightarrow$ 栈中符号/状态编号)

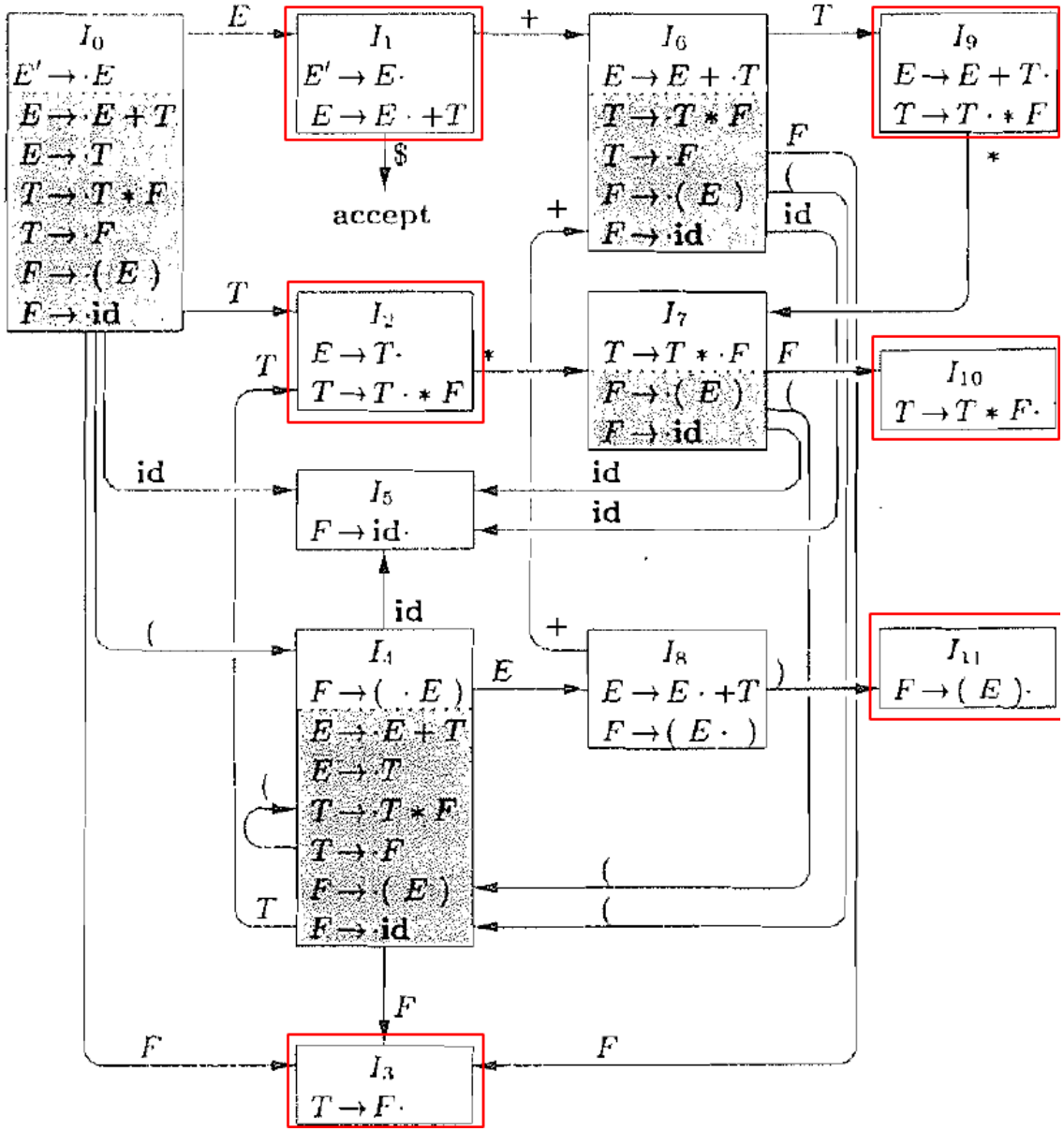
Theorem

存在一种 LR 语法分析方法，保证**句柄总是出现在栈顶**。

在自动机的当前状态识别可能的句柄 (观察到的完整右部)

(自动机的当前状态 $\Leftrightarrow$ 栈顶)

$LR(0)$ 句柄识别有穷状态自动机 (Handle-Finding Automaton)

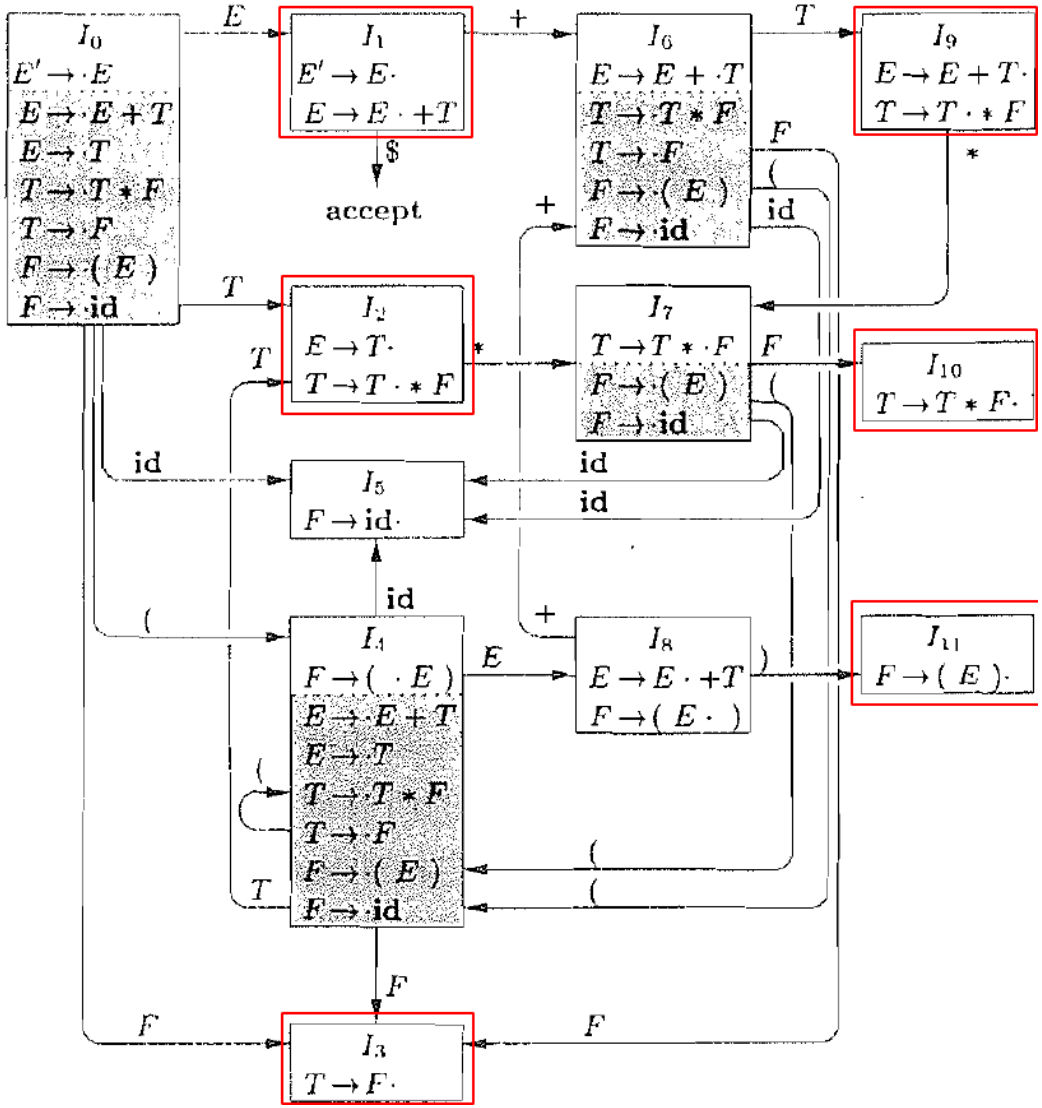


$LR(0)$ 句柄识别自动机

为给定的**文法** G构造相应的句柄识别自动机

该自动机用于识别该**文法** G **所允许的所有可能的句柄**

$LR(0)$ 句柄识别自动机



状态是什么?

状态刻画了“当前观察到的**针对所有产生式的右部的前缀**”

Definition ($LR(0)$ 项 (Item))

文法 G 的一个 $LR(0)$ **项**是 G 的某个产生式加上一个位于体部的**点**。

$$A \to XYZ$$

$$A \to \cdot XYZ$$

$$A \to X \cdot YZ$$

$$A \to XY \cdot Z$$

$$A \to XYZ \cdot$$

(产生式 $A\epsilon$ 只有一个项 $A \to \cdot$)

状态刻画了“当前观察到的**针对所有产生式的右部的前缀**”

Definition ($LR(0)$ 项 (Item))

文法 G 的一个 $LR(0)$ **项**是 G 的某个产生式加上一个位于体部的**点**。

$$A \to XYZ$$

$$A \to \cdot XYZ$$

$$A \to X \cdot YZ$$

$$A \to XY \cdot Z$$

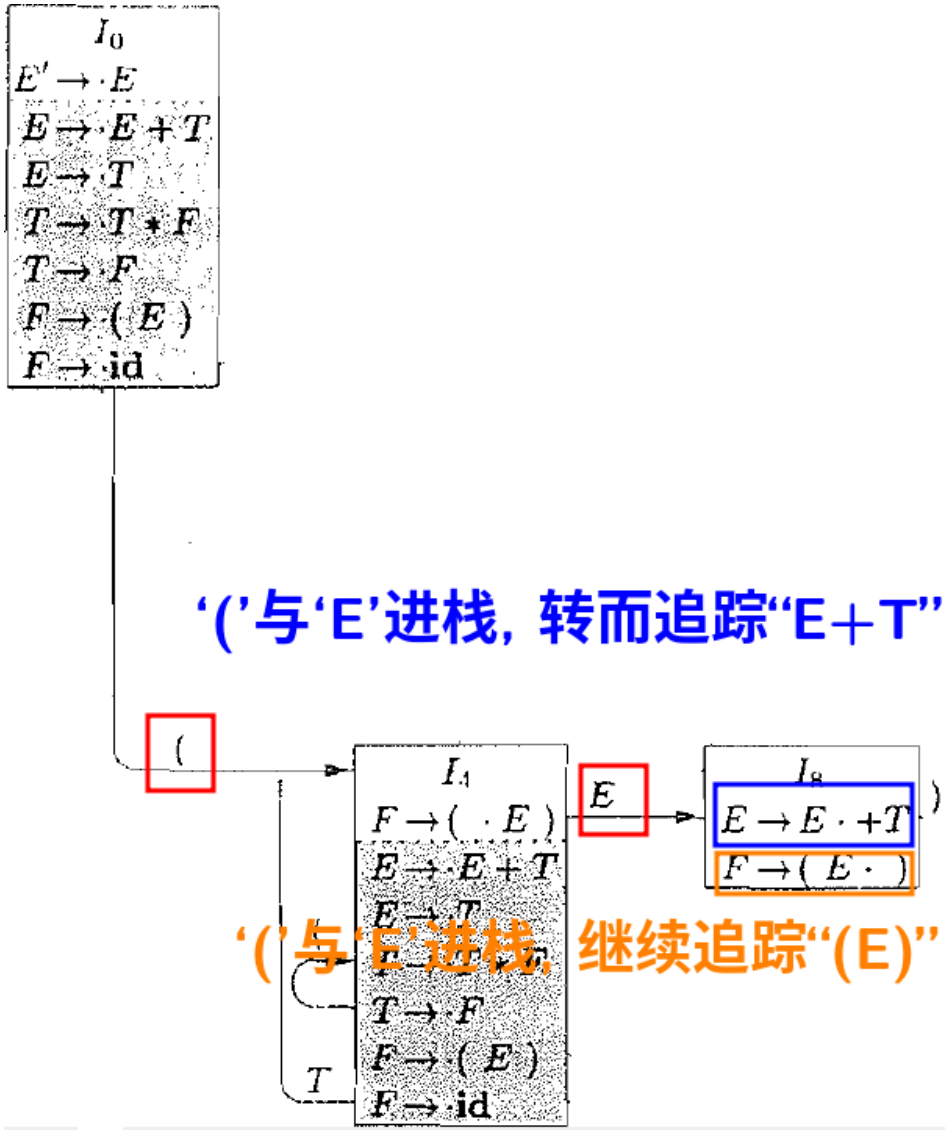
$$A \to XYZ \cdot$$

(产生式 $A\epsilon$ 只有一个项 $A \to \cdot$)

项指明了语法分析器已经观察到了某个产生式的哪些部分

点指示了**栈顶**, 左边 (与路径) 是栈中内容, 右边是期望看到的文法符号串

状态刻画了“当前观察到的**针对所有产生式的右部的前缀**”



点指示了**栈顶**，左边 (与路径) 是栈中内容，右边是期望看到的文法符号串

状态刻画了“当前观察到的**针对所有产生式的右部的前缀**”

Definition (项集)

项集就是若干**项**构成的集合。

因此, 句柄识别自动机的一个**状态**可以表示为一个**项集**

状态刻画了“当前观察到的**针对所有产生式的右部的前缀**”

Definition (项集)

项集就是若干**项**构成的集合。

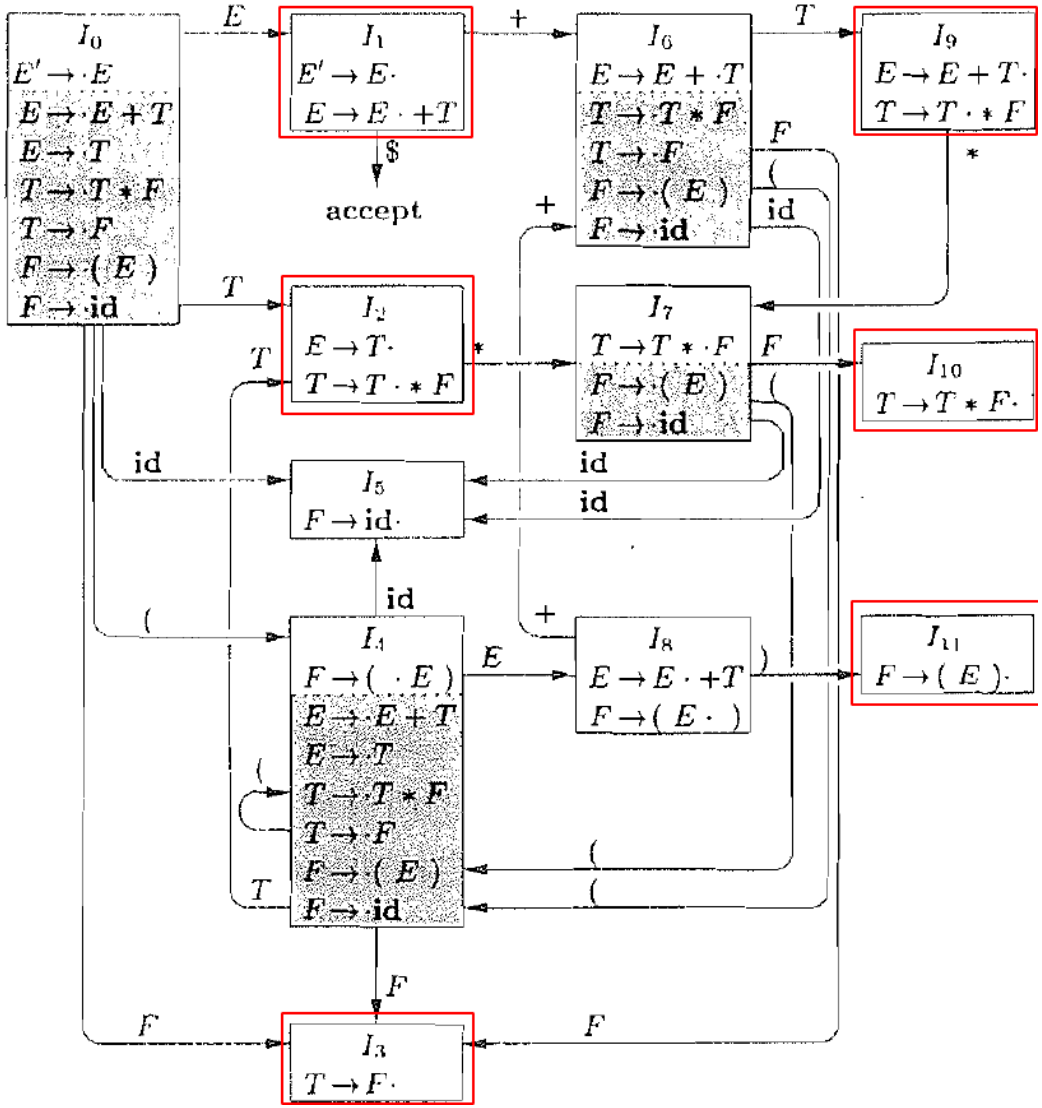
因此, 句柄识别自动机的一个**状态**可以表示为一个**项集**

Definition (项集族)

项集族就是若干**项集**构成的集合。

因此, 句柄识别自动机的**状态集**可以表示为一个**项集族**

$LR(0)$ 句柄识别自动机



项、项集、项集族

Definition (增广文法 (Augmented Grammar))

文法 G 的**增广文法**是在 G 中加入产生式 $S' \to S$ 得到的文法。

目的: 告诉语法分析器何时停止分析并接受输入符号串

当语法分析器**面对** \$且**要使用** $S' \to S$ **进行归约**时, 输入符号串被接受

Definition (增广文法 (Augmented Grammar))

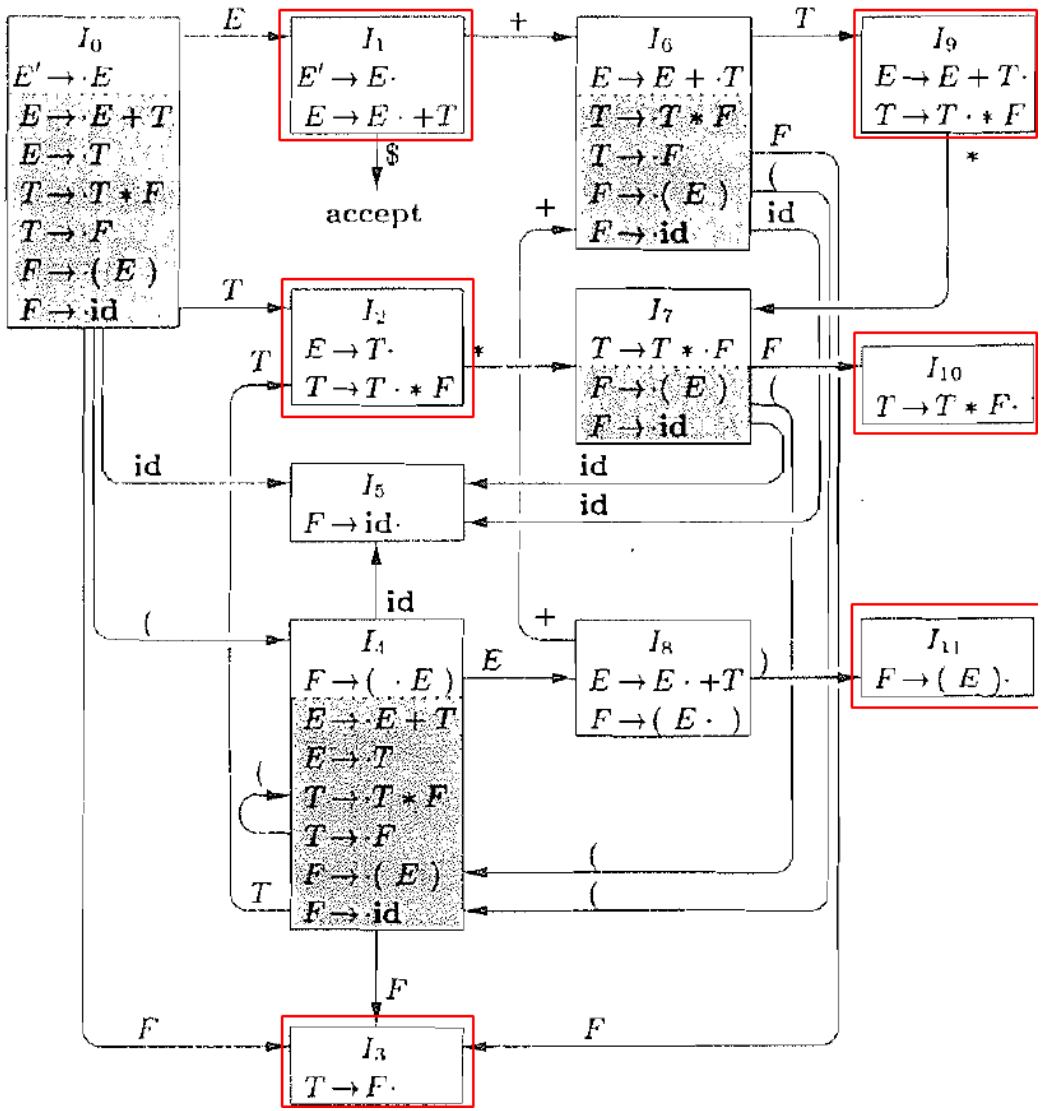
文法 G 的**增广文法**是在 G 中加入产生式 $S' \to S$ 得到的文法。

目的: 告诉语法分析器何时停止分析并接受输入符号串

当语法分析器**面对** \$且**要使用** $S' \to S$ **进行归约**时, 输入符号串被接受

注: 此 “接受” (输入串) 非彼 “接受” (句柄识别自动机)

$LR(0)$ 句柄识别自动机



注: 此 “接受” (输入串) 非彼 “接受” (句柄识别自动机)

$LR(0)$ 句柄识别自动机

初始状态是什么?

点指示了**栈顶**, 左边 (与路径) 是栈中内容, 右边是期望看到的文法符号串

(0) $E' \to E$

(1) $E \to E + T$

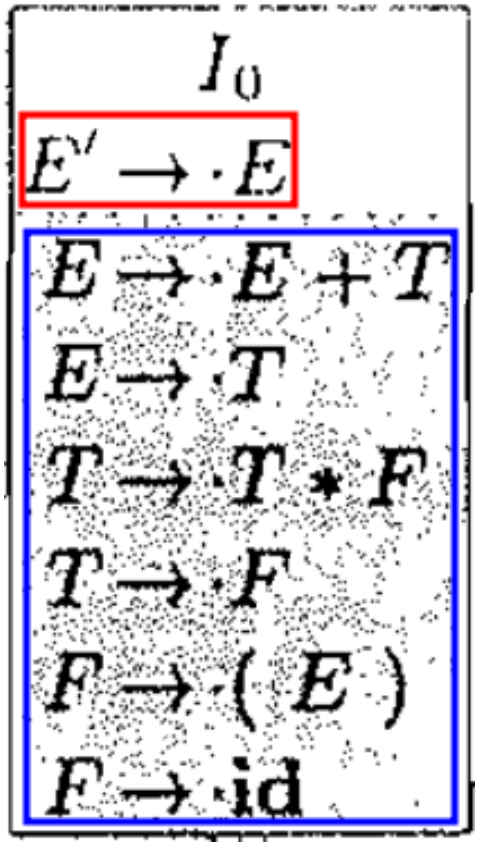
(2) $E \to T$

(3) $T \to T * F$

(4) $T \to F$

(5) $F \to (E)$

(6) $F \to \mathbf{id}$

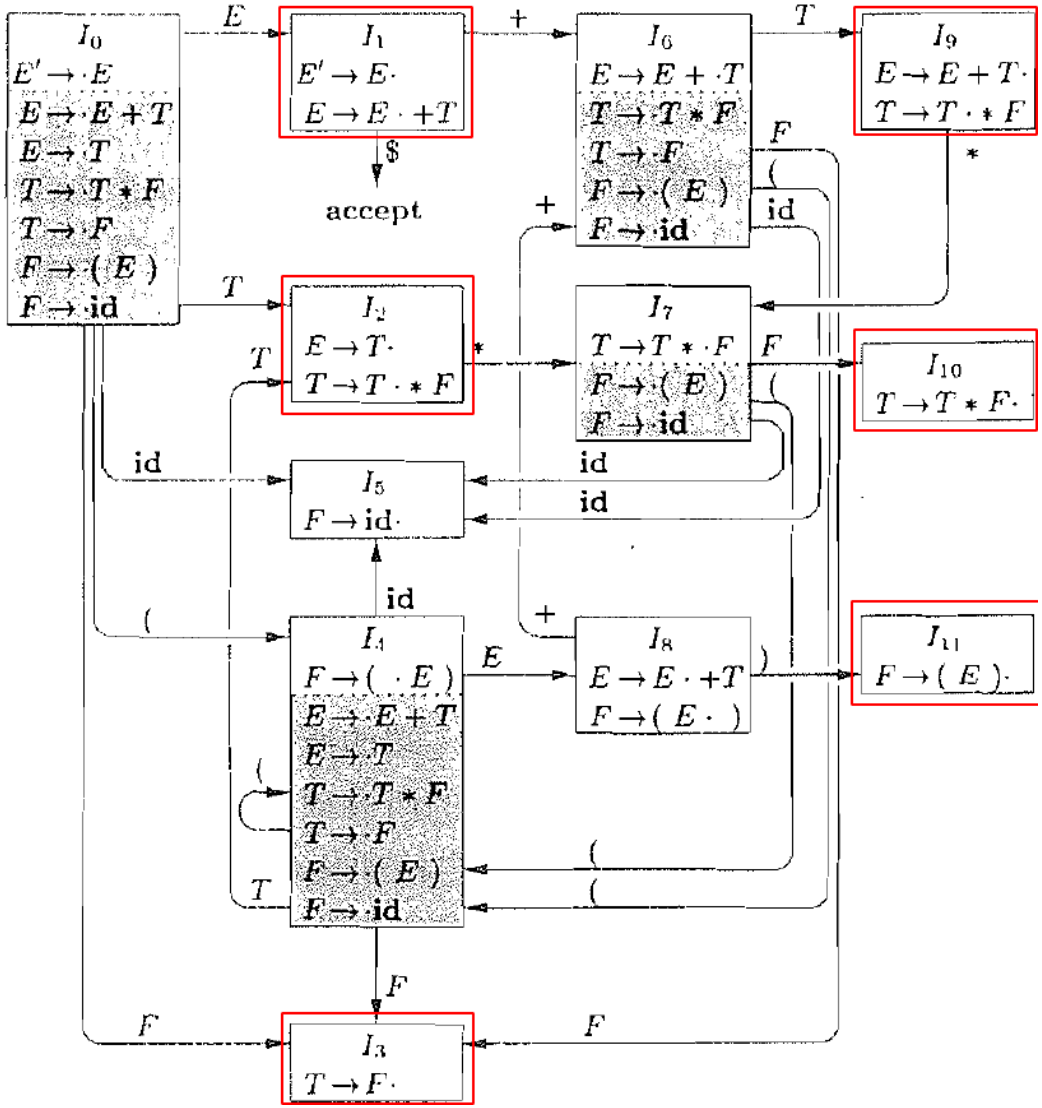


CLOSURE($\{[E' \to \cdot E]\}$)

$LR(0)$ 句柄识别自动机

状态之间如何转移?

板书演示 $LR(0)$ **句柄识别自动机**的构造过程



状态编号约定

```
SetOfItems CLOSURE(I) {
        J = I;
        repeat
                for (J中的每个项 A → α·Bβ)
                        for (G的每个产生式 B → γ)
                                if (项 B → ·γ 不在J中)
                                        将 B → ·γ 加入J中;
        until 在某一轮中没有新的项被加入到J中;
        return J;
}
```

$$J = \text{GOTO}(I, X) = \text{CLOSURE}\left(\left\{[A \to \alpha X \cdot \beta] \middle| [A \to \alpha \cdot X\beta] \in I\right\}\right)$$

$$(X \in N \cup T \cup \{\$\})$$

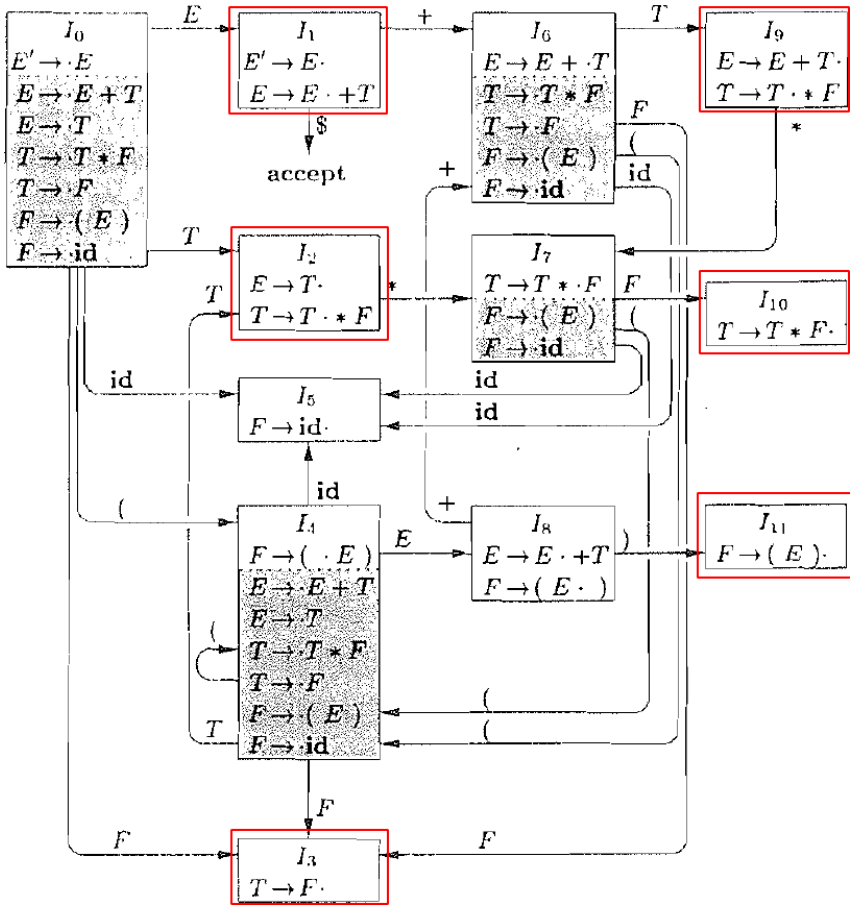
```
void items(G') {
    C = {CLOSURE({[S' → ·S]})};
    repeat
        for (C中的每个项集I)
            for (每个文法符号X)
                if (GOTO(I, X)非空且不在C中)
                    将GOTO(I, X)加入C中;
    until 在某一轮中没有新的项集被加入到C中;
}
```

初始状态

下一个状态

图 4-33 规范 LR(0) 项集族的计算

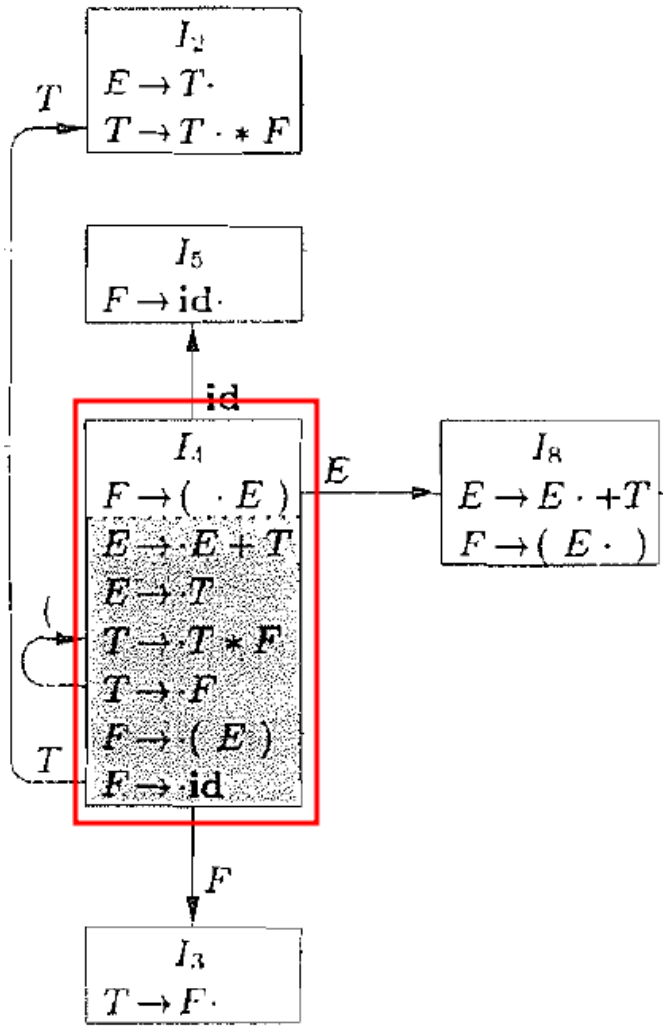
$LR(0)$ 分析表



	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

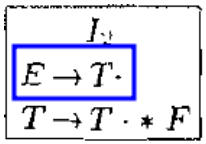
GOTO 函数被拆分成 ACTION 表 (针对终结符) 与 GOTO 表 (针对非终结符)

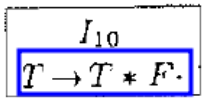
(1) $[A \to \alpha \cdot a\beta] \in I_i \wedge a \in T \wedge \text{GOTO}(I_i, a) = I_j \implies \text{ACTION}[i, a] \leftarrow sj$



	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

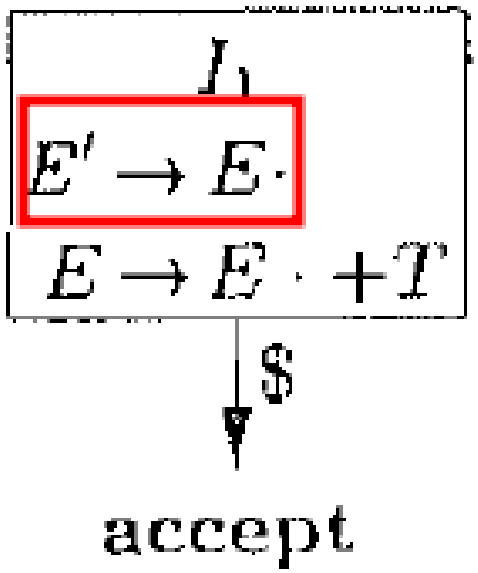
(2) $\text{GOTO}(I_i, A) = I_j \implies \text{ACTION}[i, A] \leftarrow gj$





	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

(3) $[k : A \to \alpha\cdot] \in I_i \wedge A \neq S' \implies \forall t \in T \cup \{\$\}.\ \text{ACTION}[i, t] = rk$



	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

(4) $[S' \to S\cdot] \in I_i \implies \text{ACTION}[i, \$] \leftarrow acc$

$LR(0)$ 分析表

(1) $[A \to \alpha \cdot a\beta] \in I_i \wedge a \in T \wedge \text{GOTO}(I_i, a) = I_j \implies \text{ACTION}[i, a] \leftarrow sj$

(2) $\text{GOTO}(I_i, A) = I_j \implies \text{ACTION}[i, A] \leftarrow gj$

(3) $[k : A \to \alpha\cdot] \in I_i \wedge A \neq S' \implies \forall t \in T \cup \{\$\}.\ \text{ACTION}[i, t] = rk$

(4) $[S' \to S\cdot] \in I_i \implies \text{ACTION}[i, \$] \leftarrow acc$

Definition ($LR(0)$ 文法)

如果文法 G 的$LR(0)$ **分析表**是**无冲突**的, 则 G 是 $LR(0)$ 文法。

	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

非 $LR(0)$ 分析表/文法

$LR(0)$ 分析表每一行 (状态) **所选用的归约产生式是相同的**

	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	$s5$			$s4$			$g1$	$g2$	$g3$
1		$s6$				acc			
2	$r2$	$r2$	$s7, r2$	$r2$	$r2$	$r2$			
3	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$			
4	$s5$			$s4$			$g8$	$g2$	$g3$
5	$r6$	$r6$	$r6$	$r6$	$r6$	$r6$			
6	$s5$			$s4$				$g9$	$g3$
7	$s5$			$s4$					$g10$
8		$s6$			$s11$				
9	$r1$	$r1$	$s7, r1$	$r1$	$r1$	$r1$			
10	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$			
11	$r5$	$r5$	$r5$	$r5$	$r5$	$r5$			

归约时不需要向前看, 这就是 "0" 的含义

$LR(0)$ 语法分析器

L：**从左向右** (Left-to-right) 扫描输入

R：构建**反向** (Reverse) **最右推导**

0：**归约**时无需向前看

$LR(0)$ 自动机与栈之间的互动关系

向前走 $\Leftrightarrow$ 移入

回溯 $\Leftrightarrow$ 归约

自动机才是本质，栈是实现方式

(用栈记住 “来时的路”, 以便回溯)

$SLR(1)$ 分析表

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

归约:

(3) $[k : A \to \alpha\cdot] \in I_i \wedge A \neq S' \implies \forall t \in \textsc{Follow}(A).\ \textsc{action}[i, t] = rk$

Definition ($SLR(1)$ 文法)

如果文法 G 的$SLR(1)$ **分析表**是**无冲突**的, 则 G 是 $SLR(1)$ 文法。

无冲突: ACTION 表中每个单元格最多只有一种动作

状态	ACTION						GOTO		
	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

两类可能的冲突: “移入/归约” 冲突、“归约/归约” 冲突

非 $SLR(1)$ 文法举例

$$S \longrightarrow L = R \mid R$$

$$L \longrightarrow * R \mid \mathbf{id}$$

$$R \longrightarrow L$$

I_0: $S' \to \cdot S$
$S \to \cdot L = R$
$S \to \cdot R$
$L \to \cdot * R$
$L \to \cdot \mathbf{id}$
$R \to \cdot L$

I_1: $S' \to S\cdot$

I_2: $S \to L\cdot = R$
$R \to L\cdot$

I_3: $S \to R\cdot$

I_4: $L \to *\cdot R$
$R \to \cdot L$
$L \to \cdot * R$
$L \to \cdot \mathbf{id}$

I_5: $L \to \mathbf{id}\cdot$

I_6: $S \to L = \cdot R$
$R \to \cdot L$
$L \to \cdot * R$
$L \to \cdot \mathbf{id}$

I_7: $L \to *R\cdot$

I_8: $R \to L\cdot$

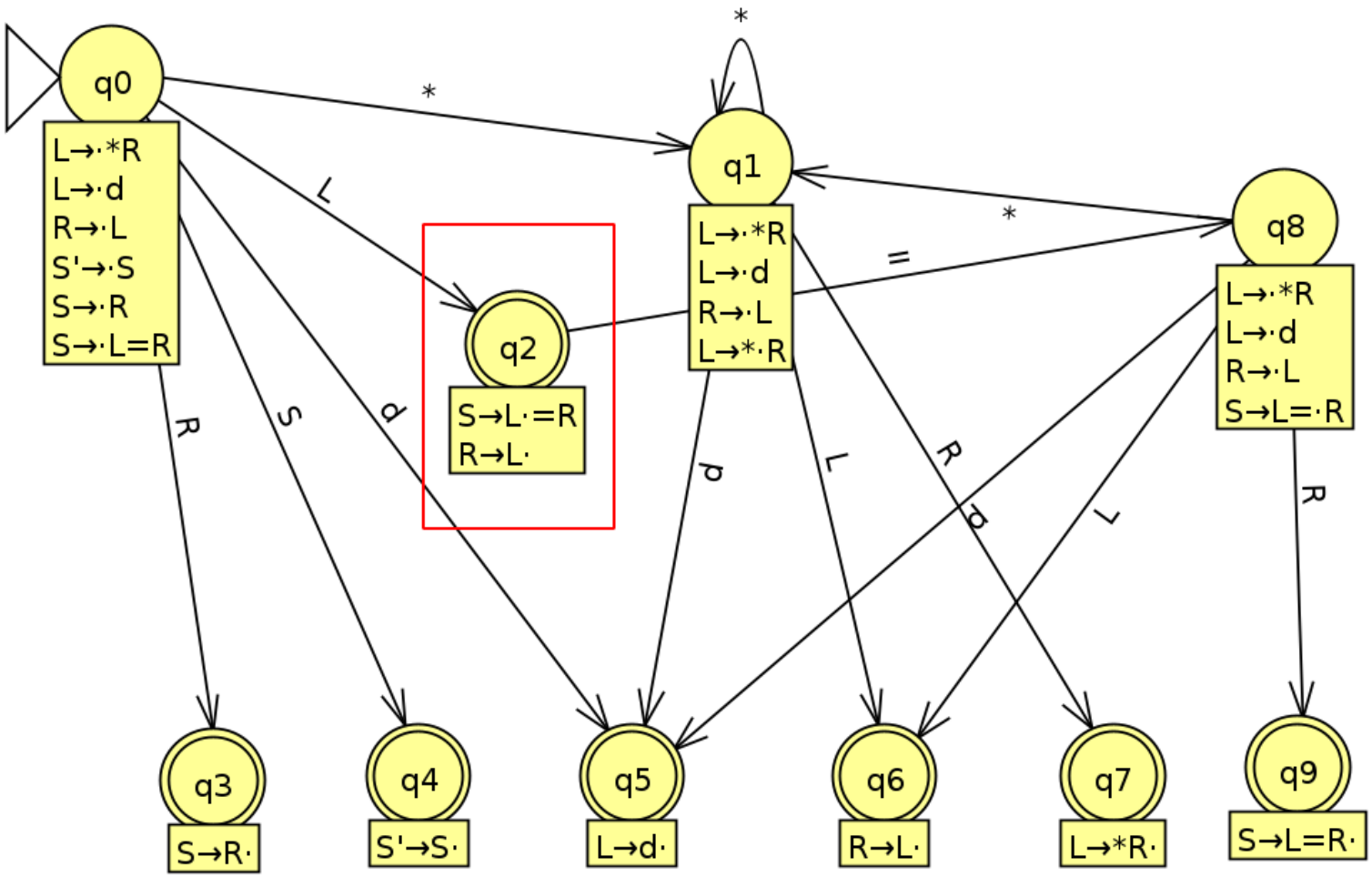
I_9: $S \to L = R\cdot$

$$[S \to L\cdot = R] \in I_2 \implies \textsc{action}(I_2, =) \leftarrow s6$$

$$= \in \textsc{Follow}(R) \implies \textsc{action}(I_2, =) \leftarrow r5$$

即使考虑了$= \in \textsc{Follow}(A)$, 对该文法来说仍然不够

因为, 这仅仅说明在**某个**句型中, a 可以跟在 A 后面



该文法没有**以** $R = \cdots$ **开头**的最右句型

希望 LR 语法分析器的每个状态能**尽可能精确**地

指明**哪些输入符号可以跟在句柄 $A \to \alpha$ 的后面**

希望 LR 语法分析器的每个状态能**尽可能精确**地

指明**哪些输入符号可以跟在句柄 $A \to \alpha$ 的后面**

在 $LR(0)$ 自动机中, 某个项集 I_j 中包含 $[A \to \alpha\cdot]$

则在之前的某个项集 I_i 中包含 $[B \to \beta \cdot A\gamma]$

这表明只有 $a \in \textsc{First}(\gamma)$ 时, 才可以进行 $A \to \alpha$ 归约

希望 LR 语法分析器的每个状态能**尽可能精确**地

指明**哪些输入符号可以跟在句柄 $A \to \alpha$ 的后面**

在 $LR(0)$ 自动机中, 某个项集 I_j 中包含 $[A \to \alpha\cdot]$

则在之前的某个项集 I_i 中包含 $[B \to \beta \cdot A\gamma]$

这表明只有 $a \in \textsc{First}(\gamma)$ 时, 才可以进行 $A \to \alpha$ 归约

但是, 对 I_i 求闭包时, 仅得到 $[A \to \cdot\alpha]$, 丢失了 $\textsc{First}(\gamma)$ 信息

Definition ($LR(1)$ 项 (Item))

$$[A \to \alpha \cdot \beta, a] \qquad (a \in T \cup \{\$\})$$

此处, a 是**向前看符号**, 数量为 **1**.

Definition ($LR(1)$ 项 (Item))

$$[A \to \alpha \cdot \beta, a] \qquad (a \in T \cup \{\$\})$$

此处, a 是**向前看符号**, 数量为 **1**.

思想: α 在栈顶, 且输入中开头的是可以从 βa 推导出的符号串

$LR(1)$句柄识别自动机

$$[A \to \alpha \cdot B\beta, a] \in I \qquad (a \in T \cup \{\$\})$$

```
SetOfIteins CLOSURE(I) {
        repeat
                for ( I中的每个项 [A → α·Bβ, a] )
                        for ( G'中的每个产生式 B → γ )
                                for ( FIRST(βa)中的每个终结符号 b )
                                        将 [B → ·γ, b] 加入到集合 I中;
        until 不能向 I 中加入更多的项;
        return I;
}
```

$$\forall b \in \textsc{First}(\beta a).\ [B \to \cdot\gamma, b] \in I$$

$LR(1)$句柄识别自动机

SetOfItems GOTO(I, X) {
    将 J 初始化为空集；
    for (I 中的每个项 $[A \to \alpha \cdot X\beta, a]$)
        将项 $[A \to \alpha X \cdot \beta, a]$ 加入到集合 J 中；
    return CLOSURE(J);
}

$$J = \text{GOTO}(I, X) = \text{CLOSURE}\left(\left\{[A \to \alpha X \cdot \beta] \middle| [A \to \alpha \cdot X\beta] \in I\right\}\right)$$

$LR(1)$句柄识别自动机

```
void items(G') {
    将 C 初始化为{CLOSURE ({[S' → ·S, $]})}
    repeat
        for ( C 中的每个项集 I )
            for ( 每个文法符号 X )
                if ( GOTO(I, X) 非空且不在 C 中 )
                    将 GOTO(I, X) 加入 C 中;
    until 不再有新的项集加入到 C 中;
}
```

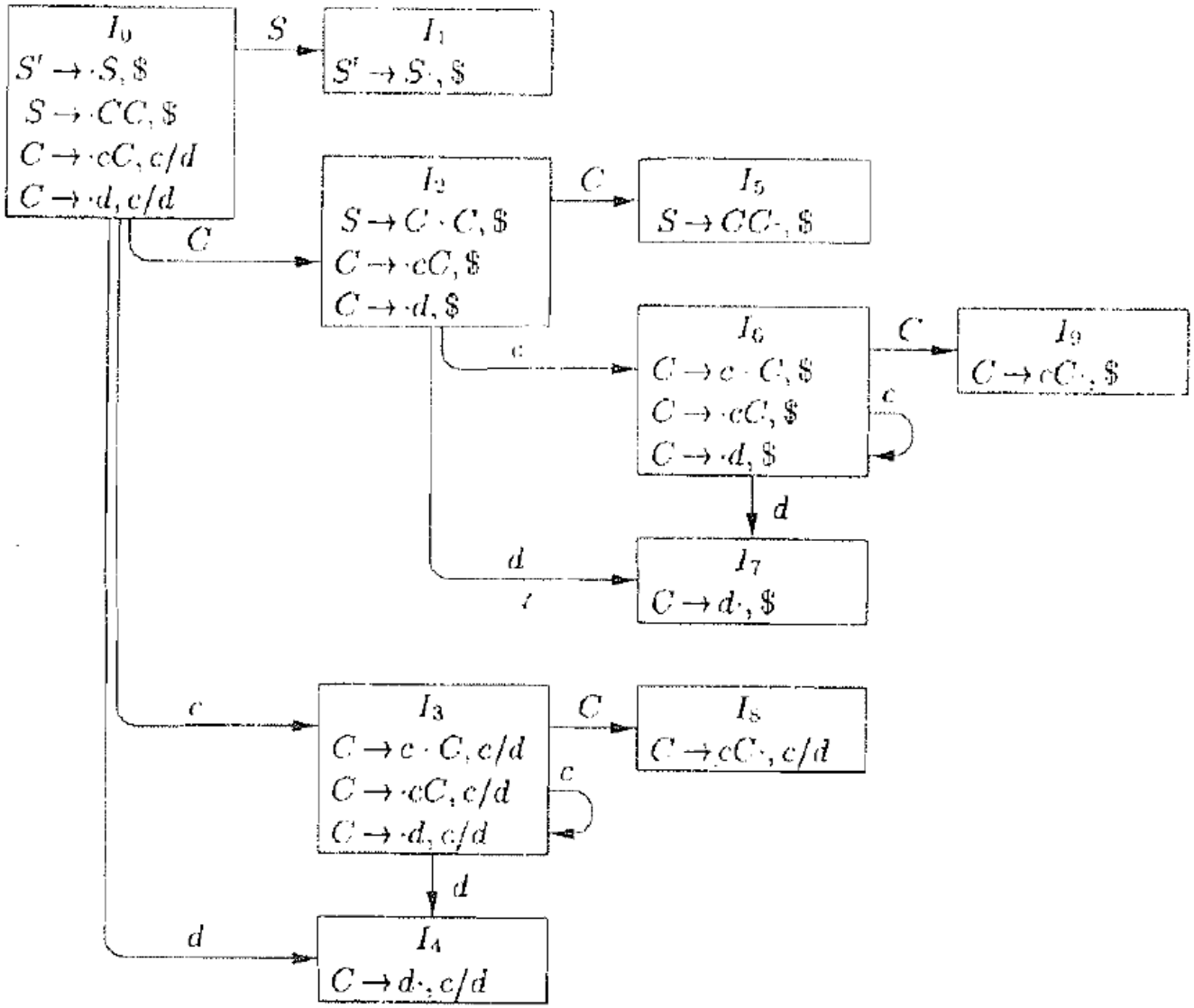
初始状态: $\text{CLOSURE}([S' \to \cdot S, \$])$

板书演示: $LR(1)$ 自动机的构造过程

$$S' \longrightarrow S$$

$$S \longrightarrow C\ C$$

$$C \longrightarrow c\ C \mid d$$



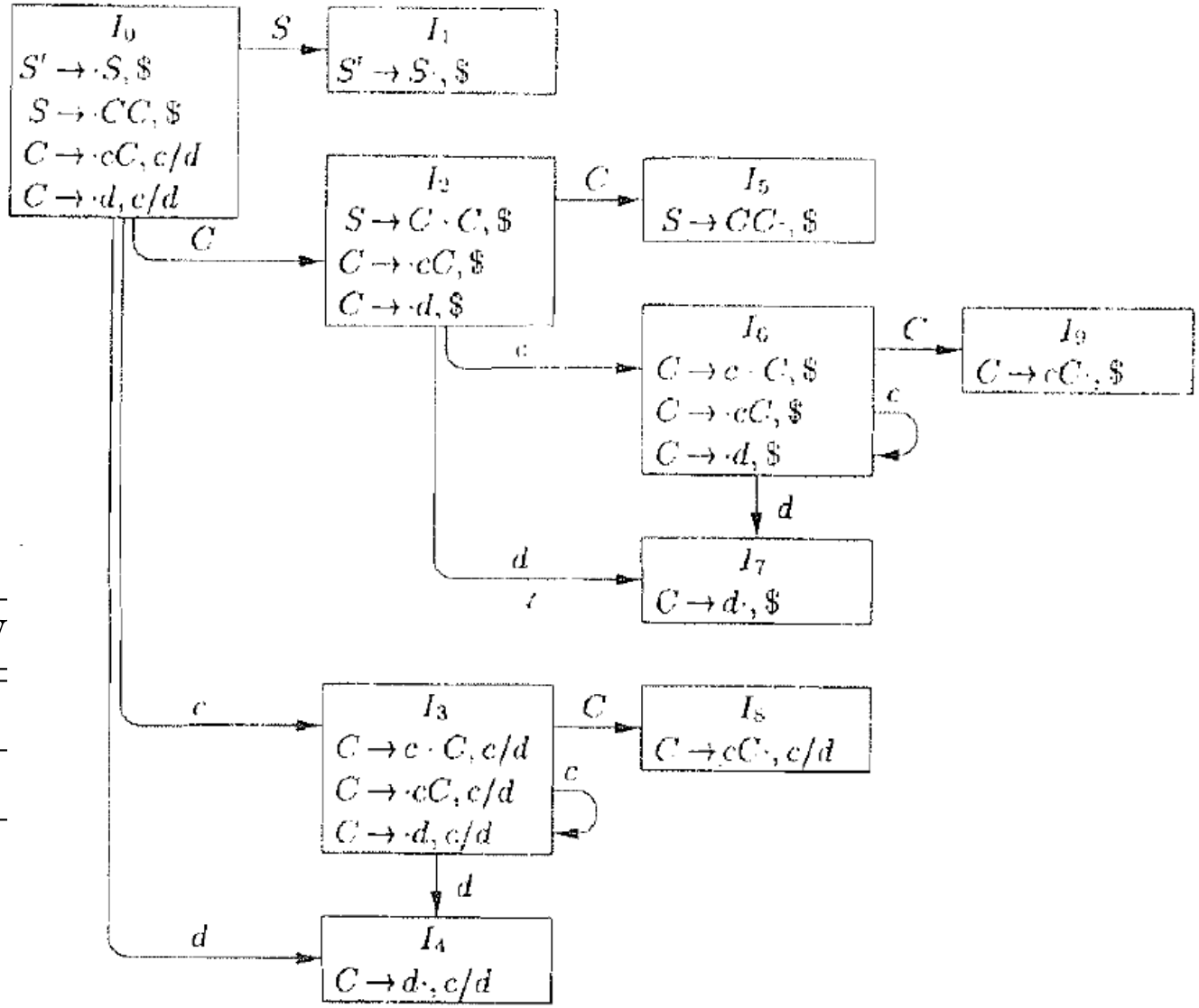
板书演示: $LR(1)$ 自动机的构造过程

$$S' \longrightarrow S$$

$$S \longrightarrow C\ C$$

$$C \longrightarrow c\ C \mid d$$

	First	Follow
S	$\{c, d\}$	$\$$
C	$\{c, d\}$	$\{c, d, \$\}$



$LR(1)$ 自动机构建 $LR(1)$ 分析表

(1) $[A \to \alpha \cdot a\beta, b] \in I_i \wedge a \in T \wedge \text{GOTO}(I_i, a) = I_j \implies \text{ACTION}[i, a] \leftarrow sj$

(2) $\text{GOTO}(I_i, A) = I_j \implies \text{ACTION}[i, A] \leftarrow gj$

(3) $[k : A \to \alpha \cdot, a] \in I_i \wedge A \neq S' \implies \text{ACTION}[i, a] = rk$

(4) $[S' \to S \cdot, \$] \in I_i \implies \text{ACTION}[i, \$] \leftarrow acc$

$LR(1)$ 自动机构建 $LR(1)$ 分析表

(1) $[A \to \alpha \cdot a\beta, b] \in I_i \wedge a \in T \wedge \text{GOTO}(I_i, a) = I_j \implies \text{ACTION}[i, a] \leftarrow sj$

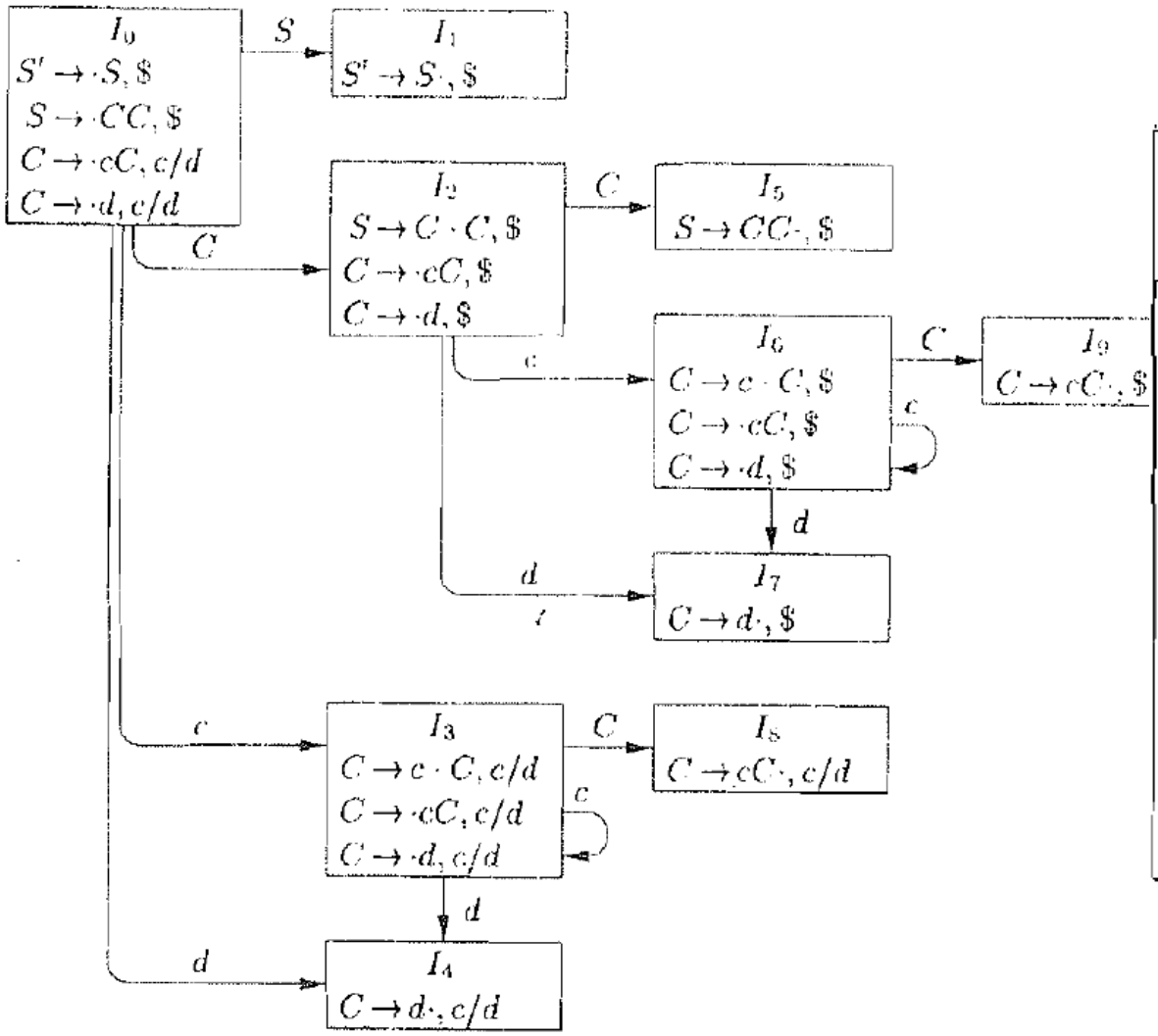
(2) $\text{GOTO}(I_i, A) = I_j \implies \text{ACTION}[i, A] \leftarrow gj$

(3) $[k : A \to \alpha \cdot, a] \in I_i \wedge A \neq S' \implies \text{ACTION}[i, a] = rk$

(4) $[S' \to S \cdot, \$] \in I_i \implies \text{ACTION}[i, \$] \leftarrow acc$

Definition ($LR(1)$ 文法)

如果文法 G 的$LR(1)$ **分析表**是**无冲突**的, 则 G 是 $LR(1)$ 文法。



状态	ACTION			GOTO	
	c	d	$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

$LR(1)$ 通过**不同的向前看符号**, 区分了状态对 $(3,6)$, $(4,7)$ 与 $(8,9)$

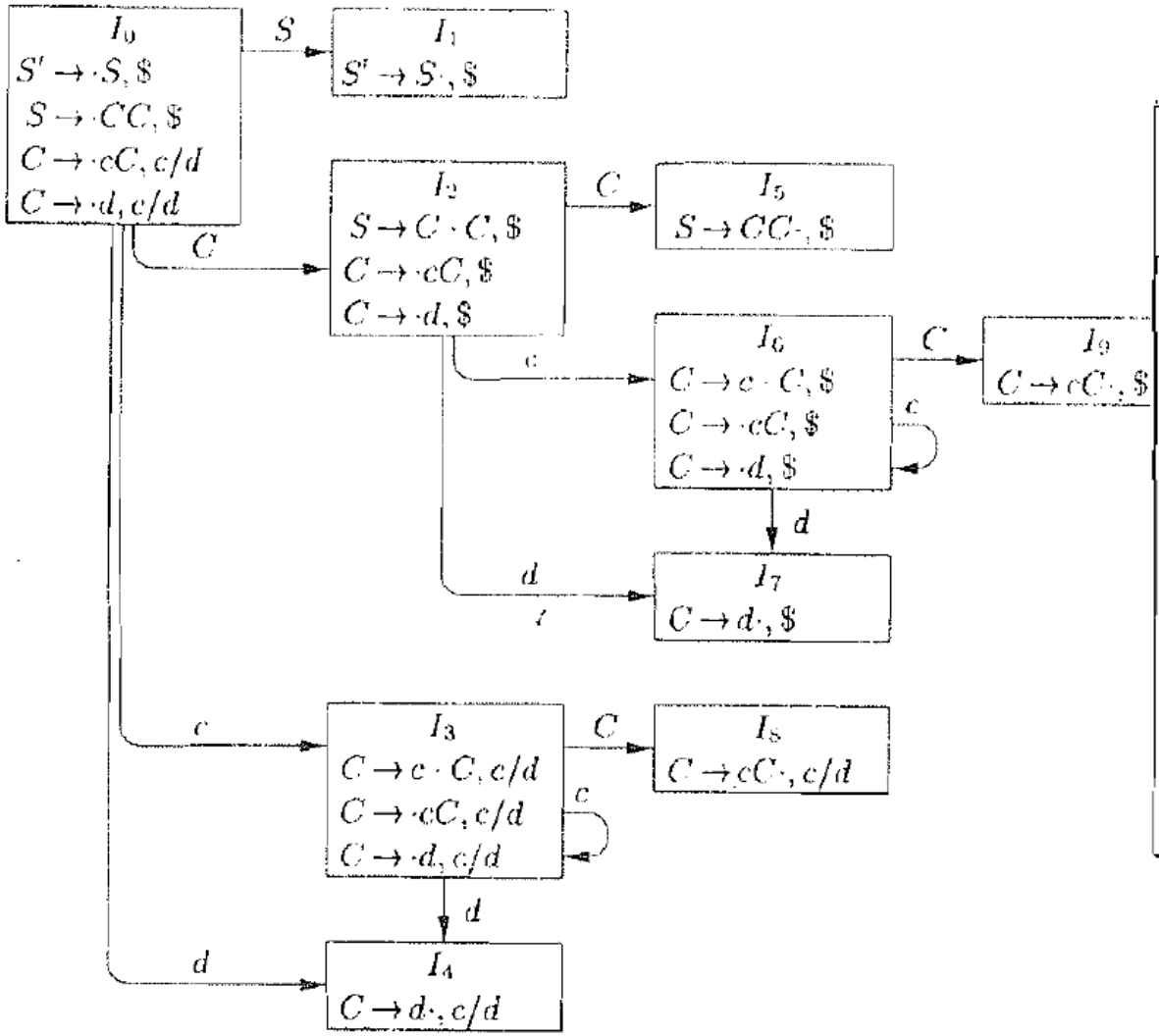
$LR(0)$、$SLR(1)$、$LR(1)$ 的**归约**条件

$$[k : A \to \alpha\cdot] \in I_i \wedge A \neq S' \implies \forall t \in T \cup \{\$\}.\ \textsc{action}[i, t] = rk$$

$$[k : A \to \alpha\cdot] \in I_i \wedge A \neq S' \implies \forall t \in \textsc{Follow}(A).\ \textsc{action}[i, t] = rk$$

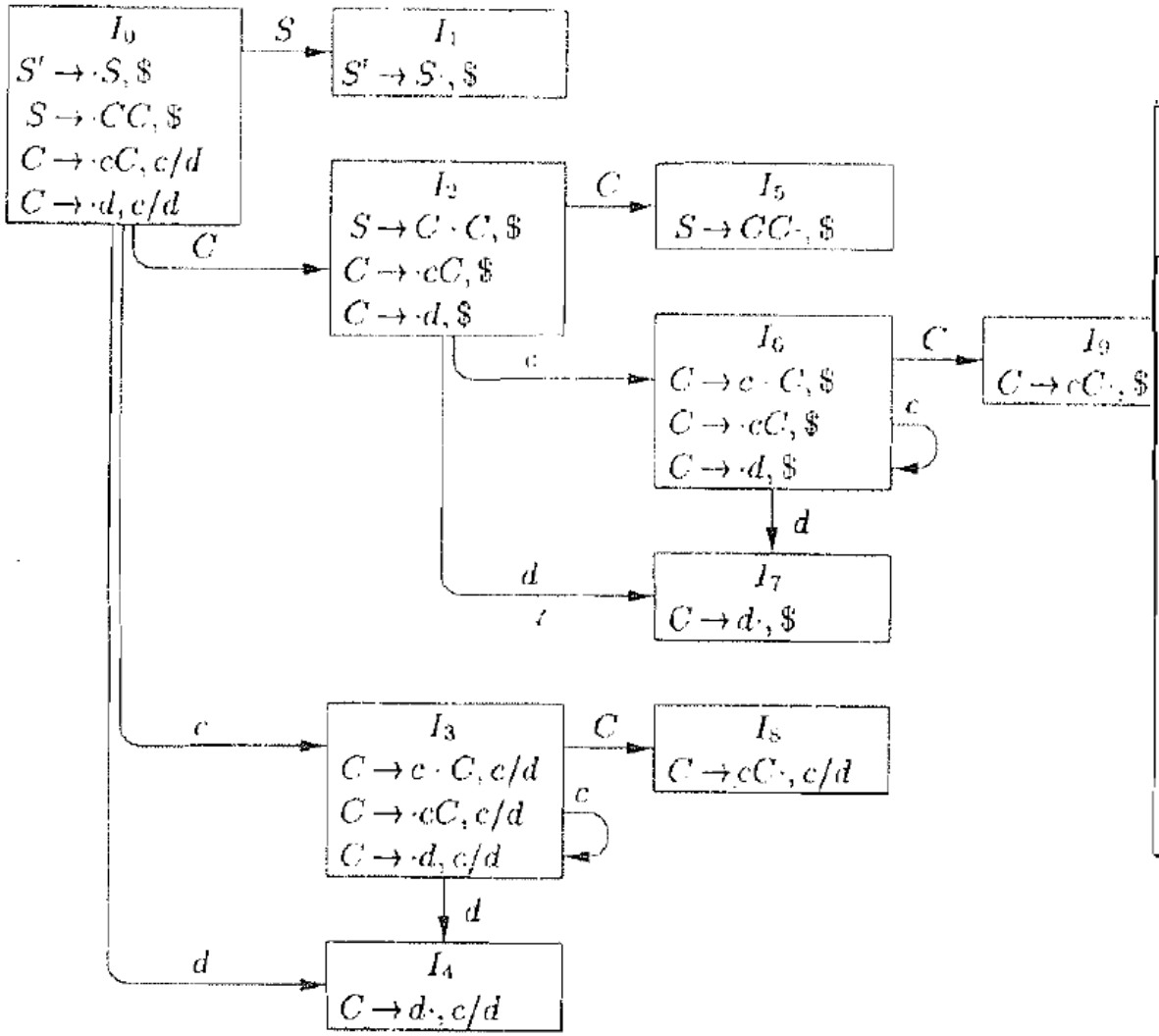
$$[k : A \to \alpha\cdot, a] \in I_i \wedge A \neq S' \implies \textsc{action}[i, a] = rk$$

$LR(1)$ 虽然强大, 但是生成的 $LR(1)$ 分析表可能过大, 状态过多



状态	ACTION			GOTO	
	c	d	$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

$LR(1)$ 虽然强大, 但是生成的 $LR(1)$ 分析表可能过大, 状态过多



状态	ACTION			GOTO	
	c	d	$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

$LALR(1)$: 合并具有相同**核心** $LR(0)$项的状态 (忽略不同的向前看符号)

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

Q：GOTO 函数怎么办?

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

状态	ACTION			GOTO	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

Q：GOTO 函数怎么办?

A：可以合并的状态的 GOTO 目标 (状态) 一定也是可以合并的

Q：对于 $LR(1)$ 文法，合并得到的 $LALR(1)$ 分析表是否会引入冲突？

Theorem

$LALR(1)$ 分析表**不会**引入移入/归约冲突。

Q：对于 $LR(1)$ 文法，合并得到的 $LALR(1)$ 分析表是否会引入冲突？

Theorem

$LALR(1)$ 分析表**不会**引入移入/归约冲突。

反证法

假设合并后出现 $[A \to \alpha\cdot, a]$ 与 $[B \to \beta \cdot a\gamma, b]$

则在 $LR(1)$ 自动机中，

存在某状态同时包含 $[A \to \alpha\cdot, a]$ 与 $[B \to \beta \cdot a\gamma, c]$

Q : 对于 $LR(1)$ 文法, 合并得到的 $LALR(1)$ 分析表是否会引入冲突?

Theorem

$LALR(1)$ 分析表**可能会**引入归约/归约冲突。

$$L(G) = \{acd, ace, bcd, bce\}$$

$S' \longrightarrow S$

$S \longrightarrow a\ A\ d \mid b\ B\ d \mid a\ B\ e \mid b\ A\ e$

$A \longrightarrow c$

$B \longrightarrow c$

Q : 对于 $LR(1)$ 文法, 合并得到的 $LALR(1)$ 分析表是否会引入冲突?

Theorem

$LALR(1)$ 分析表**可能会**引入归约/归约冲突。

$$L(G) = \{acd, ace, bcd, bce\}$$

$S' \longrightarrow S$

$S \longrightarrow a\ A\ d \mid b\ B\ d \mid a\ B\ e \mid b\ A\ e$

$A \longrightarrow c$

$B \longrightarrow c$

$$\{[A \to c\cdot, d], [B \to c\cdot, e]\}$$
$$\{[A \to c\cdot, e], [B \to c\cdot, d]\}$$

$$\{[A \to c\cdot, d/e], [B \to c\cdot, d/e]\}$$

好消息: 善用 LR 语法分析器, 处理**二义性**文法

表达式文法

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$$

$$E \to E + T \mid T$$

$$T \to T * F \mid F$$

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

$$E \to TE'$$

$$E' \to +\, TE' \mid \epsilon$$

$$T \to FT'$$

$$T' \to *\, FT' \mid \epsilon$$

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

表达式文法: 使用 $SLR(1)$ 语法分析方法

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$$

I_0:
$E' \to \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot (E)$
$E \to \cdot \mathbf{id}$

I_1:
$E' \to E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_2:
$E \to (\cdot E)$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot (E)$
$E \to \cdot \mathbf{id}$

I_3:
$E \to \mathbf{id}\cdot$

I_4:
$E \to E + \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot (E)$
$E \to \cdot \mathbf{id}$

I_5:
$E \to E * \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot (E)$
$E \to \cdot \mathbf{id}$

I_6:
$E \to (E\cdot)$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_7:
$E \to E + E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_8:
$E \to E * E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_9:
$E \to (E)\cdot$

$$\{+, *\} \subseteq \textsc{Follow}(E)$$

表达式文法: 使用 $SLR(1)$ 语法分析方法

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$$

I_0: $E' \to \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot(E)$
$E \to \cdot\mathbf{id}$

I_1: $E' \to E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_2: $E \to (\cdot E)$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot(E)$
$E \to \cdot\mathbf{id}$

I_3: $E \to \mathbf{id}\cdot$

I_4: $E \to E + \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot(E)$
$E \to \cdot\mathbf{id}$

I_5: $E \to E * \cdot E$
$E \to \cdot E + E$
$E \to \cdot E * E$
$E \to \cdot(E)$
$E \to \cdot\mathbf{id}$

I_6: $E \to (E\cdot)$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_7: $E \to E + E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_8: $E \to E * E\cdot$
$E \to E\cdot + E$
$E \to E\cdot * E$

I_9: $E \to (E)\cdot$

$$\{+, *\} \subseteq \textsc{Follow}(E)$$

考虑到**结合性与优先级**:

状态	ACTION						GOTO
	id	+	*	(	)	$	E
0	s3			s2			1
1		s4	s5			acc	
2	s3			s2			6
3		r4	r4		r4	r4	
4	s3			s2			7
5	s3			s2			8
6		s4	s5		s9		
7		r1	s5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

条件语句文法

$$
\begin{array}{lcl}
\textit{stmt} & \longrightarrow & \textbf{if } \textit{expr} \textbf{ then } \textit{stmt} \\
 & | & \textbf{if } \textit{expr} \textbf{ then } \textit{stmt} \textbf{ else } \textit{stmt} \\
 & | & \textbf{other}
\end{array}
$$

$$S' \longrightarrow S$$

$$S \longrightarrow i\ S\ e\ S \mid i\ S \mid a$$

条件语句文法: 使用 $SLR(1)$ 语法分析方法

$$S' \rightarrow S$$

$$S \rightarrow i\ S\ e\ S \mid i\ S \mid a$$

I_0: $S' \rightarrow \cdot S$
$S \rightarrow \cdot iSeS$
$S \rightarrow \cdot iS$
$S \rightarrow \cdot a$

I_1: $S' \rightarrow S\cdot$

I_2: $S \rightarrow i\cdot SeS$
$S \rightarrow i\cdot S$
$S \rightarrow \cdot iSeS$
$S \rightarrow \cdot iS$
$S \rightarrow \cdot a$

I_3: $S \rightarrow a\cdot$

I_4: $S \rightarrow iS\cdot eS$
$S \rightarrow iS$

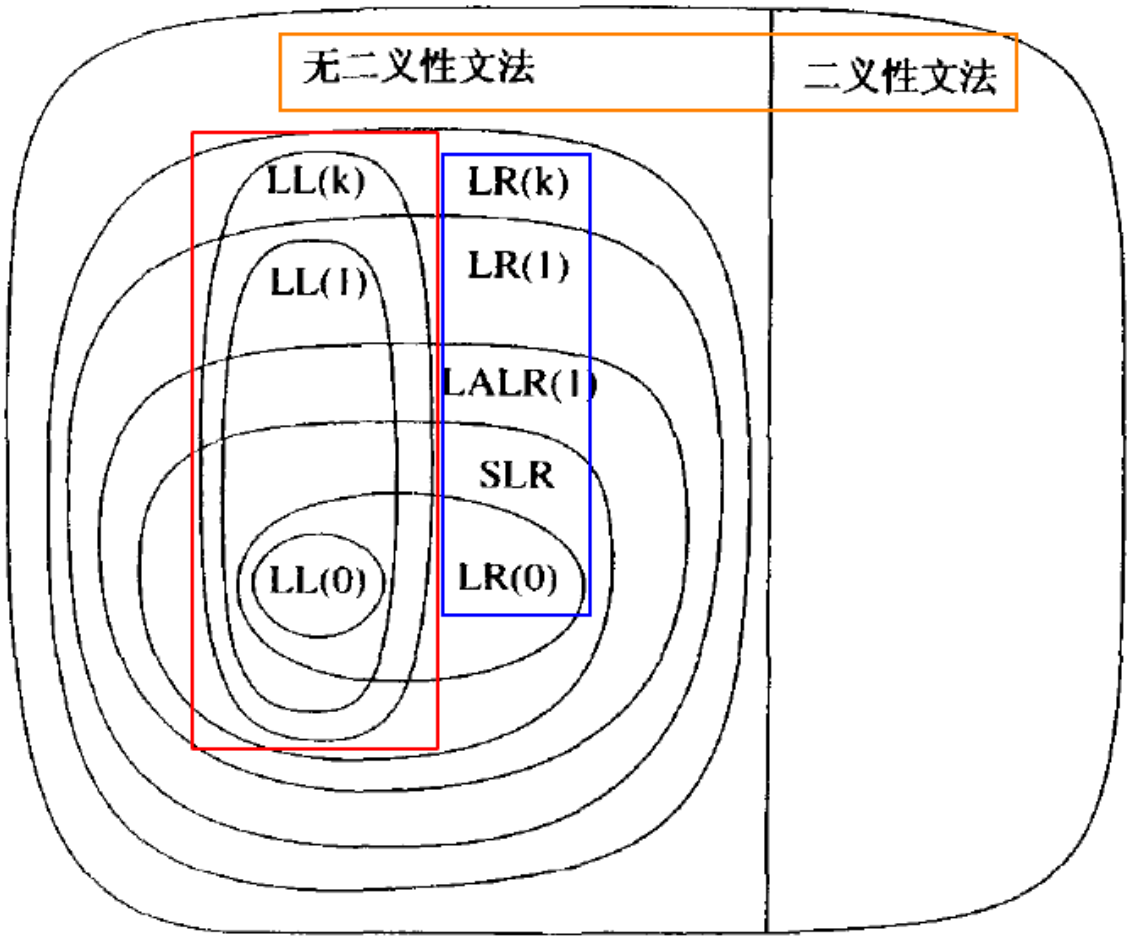
I_5: $S \rightarrow iSe\cdot S$
$S \rightarrow \cdot iSeS$
$S \rightarrow \cdot iS$
$S \rightarrow \cdot a$

I_6: $S \rightarrow iSeS\cdot$

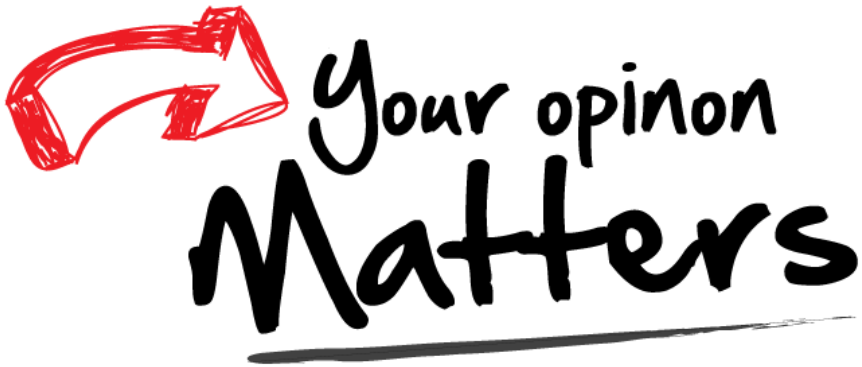
状态	ACTION				GOTO
	i	e	a	\$	S
0	s2		s3		1
1				acc	
2	s2		s3		4
3		r3		r3	
4		s5		r2	
5	s2		s3		6
6		r1		r1	

$e \in \text{FOLLOW}(S)$

$\text{ACTION}[4, e] = s5$


无二义性文法
二义性文法
LL(k)
LR(k)
LL(1)
LR(1)
LALR(1)
SLR
LL(0)
LR(0)


Thank
You!



Office 926

hfwei@nju.edu.cn