

语法分析

(6. LR1 语法分析器)

魏恒峰

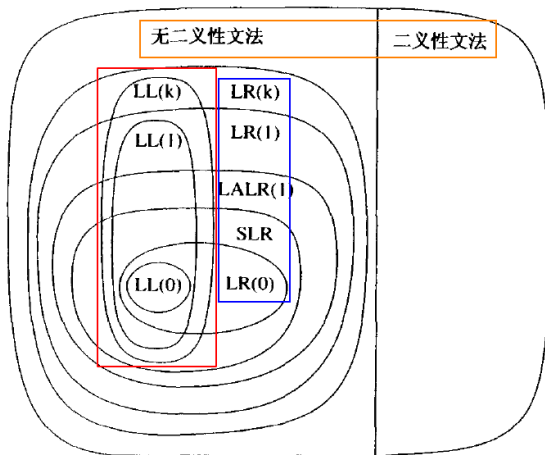
hfwei@nju.edu.cn

2021 年 12 月 03 日



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**

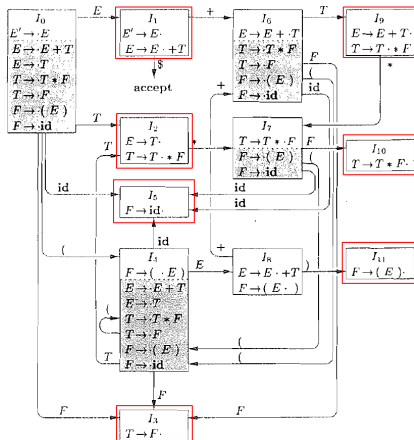
自底向上的、
不断归约的、
基于句柄识别自动机的、
适用于 LR 文法的、
 LR 语法分析器

$LR(0)$ 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



	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			

id + * () \$

$$(1) E \rightarrow E + T$$

$$(2) E \rightarrow T$$

$$(3) T \rightarrow T * F$$

$$(4) T \rightarrow F$$

$$(5) F \rightarrow (E)$$

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

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

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:

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

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

$$S \rightarrow L = R \mid R$$

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

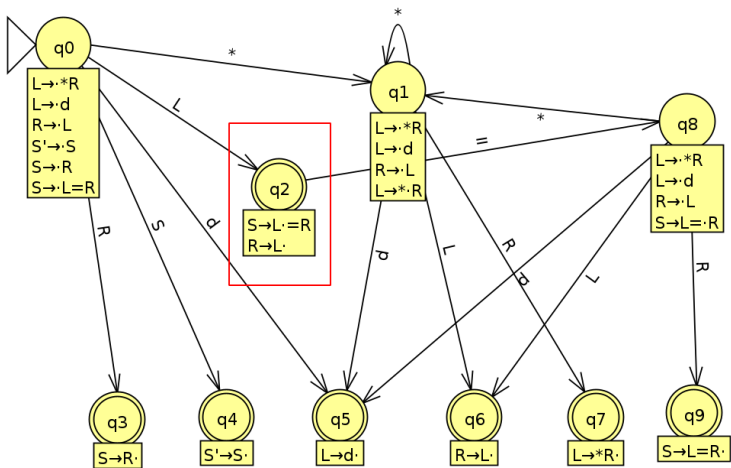
$$R \rightarrow L$$

$I_0:$ $S' \rightarrow \cdot S$ $S \rightarrow \cdot L = R$ $S \rightarrow \cdot R$ $L \rightarrow \cdot * R$ $L \rightarrow \cdot \mathbf{id}$ $R \rightarrow \cdot L$	$I_5:$ $L \rightarrow \mathbf{id} \cdot$ $I_6:$ $S \rightarrow L = \cdot R$ $R \rightarrow \cdot L$ $L \rightarrow \cdot * R$ $L \rightarrow \cdot \mathbf{id}$
$I_1:$ $S' \rightarrow S \cdot$	$I_7:$ $L \rightarrow * R \cdot$
$I_2:$ $S \rightarrow L \cdot = R$ $R \rightarrow L \cdot$	$I_8:$ $R \rightarrow L \cdot$
$I_3:$ $S \rightarrow R \cdot$	$I_9:$ $S \rightarrow L = R \cdot$
$I_4:$ $L \rightarrow * \cdot R$ $R \rightarrow \cdot L$ $L \rightarrow \cdot * R$ $L \rightarrow \cdot \mathbf{id}$	

$$[S \rightarrow L \cdot = R] \in I_2 \implies \text{ACTION}(I_2, =) \leftarrow s6$$

$$= \in \text{FOLLOW}(R) \implies \text{ACTION}(I_2, =) \leftarrow r5$$

即使考虑了 $\epsilon \in \text{FOLLOW}(A)$, 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



但该文法没有以 $R = \dots$ 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 $A \rightarrow \alpha$ 的后面

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

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

这表明只有 $a \in \text{FIRST}(\gamma)$ 时, 才可以进行 $A \rightarrow \alpha$ 归约

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

Definition ($LR(1)$ 项 (Item))

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

此处, a 是**向前看符号**, 数量为 **1**.

思想: α 在栈顶, 期望剩余输入中**开头**的是可以从 βa 推导出的符号串

$$[A \rightarrow \alpha \cdot, a]$$

只有下一个输入符号为 a 时, 才可以按照 $A \rightarrow \alpha$ 进行归约

LR(1)句柄识别自动机

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

```
SetOfItems CLOSURE(I) {  
    repeat  
        for ( I 中的每个项  $[A \rightarrow \alpha \cdot B\beta, a]$  )  
            for (  $G'$  中的每个产生式  $B \rightarrow \gamma$  )  
                for ( FIRST( $\beta a$ ) 中的每个终结符号  $b$  )  
                    将  $[B \rightarrow \cdot \gamma, b]$  加入到集合  $I$  中;  
    until 不能向  $I$  中加入更多的项;  
    return  $I$ ;  
}
```

$$\forall b \in \text{FIRST}(\beta a). [B \rightarrow \cdot \gamma, b] \in I$$

$LR(1)$ 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  $[A \rightarrow \alpha \cdot X \beta, a]$ )  
        将项  $[A \rightarrow \alpha X \cdot \beta, a]$  加入到集合  $J$  中;  
    return CLOSURE( $J$ );  
}
```

$$J = \text{GOTO}(I, X) = \text{CLOSURE}\left(\left\{[A \rightarrow \alpha X \cdot \beta, a] \mid [A \rightarrow \alpha \cdot X \beta, a] \in I\right\}\right)$$
$$(X \in N \cup T)$$

$LR(1)$ 句柄识别自动机

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

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

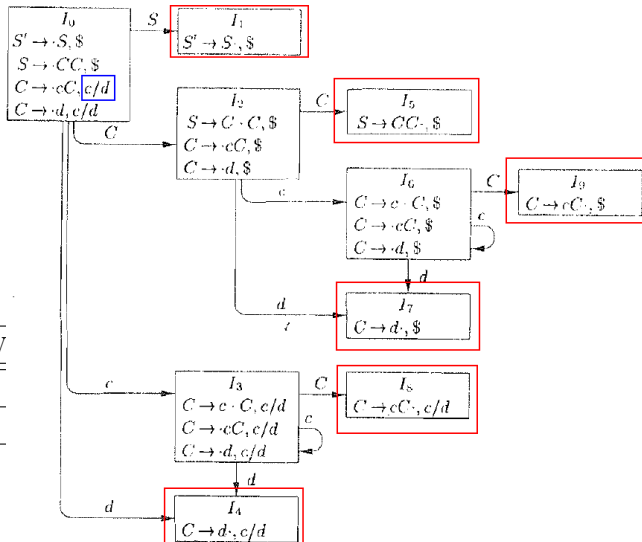
板书演示: LR(1) 自动机的构造过程

$$S' \rightarrow S$$

$$S \rightarrow C C$$

$$C \rightarrow c C \mid d$$

	FIRST	FOLLOW
S	$\{c, d\}$	$\$$
C	$\{c, d\}$	$\{c, d, \$\}$



LR(1) 分析表构造规则

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

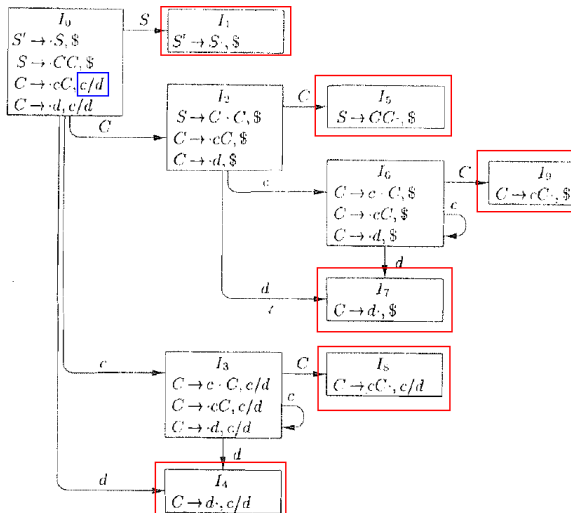
$$(2) \text{ GOTO}(I_i, A) = I_j \wedge A \in N \implies \text{GOTO}[i, A] \leftarrow gj$$

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

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

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)

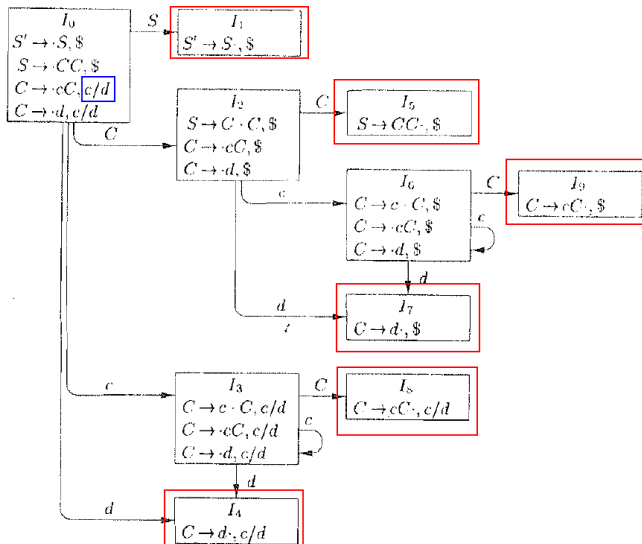
$w = ccdcd\$$

$S' \rightarrow S$

$S \rightarrow C C$

$C \rightarrow c C \mid d$

$L(G) = c^*dc^*d$



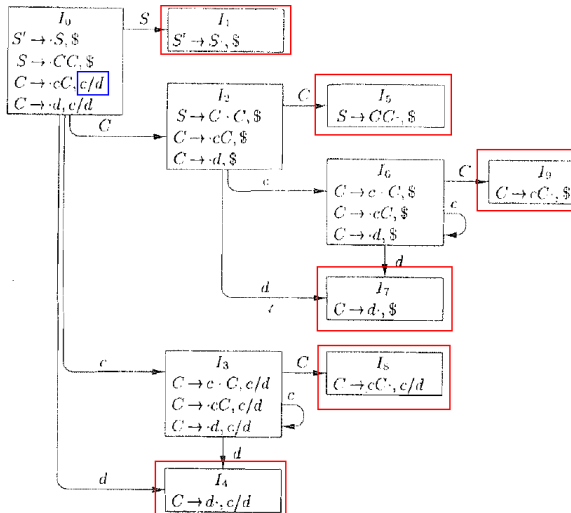
总结: $LR(0)$ 、 $SLR(1)$ 、 $LR(1)$ 的归约条件

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

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

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

$LR(1)$ 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		

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)

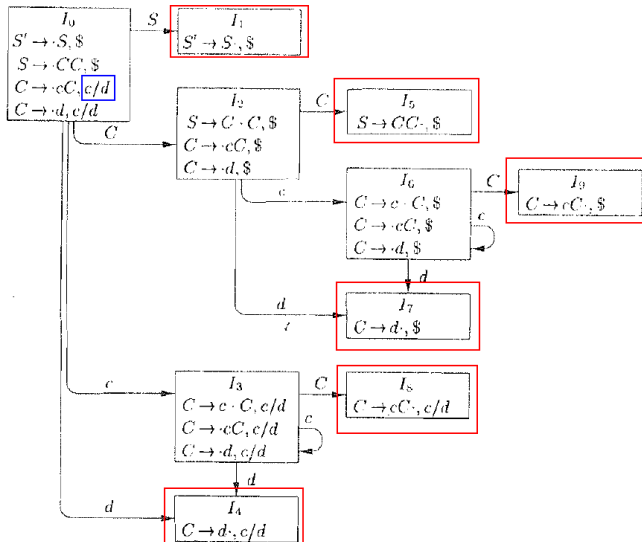
$w = ccdcd\$$

$S' \rightarrow S$

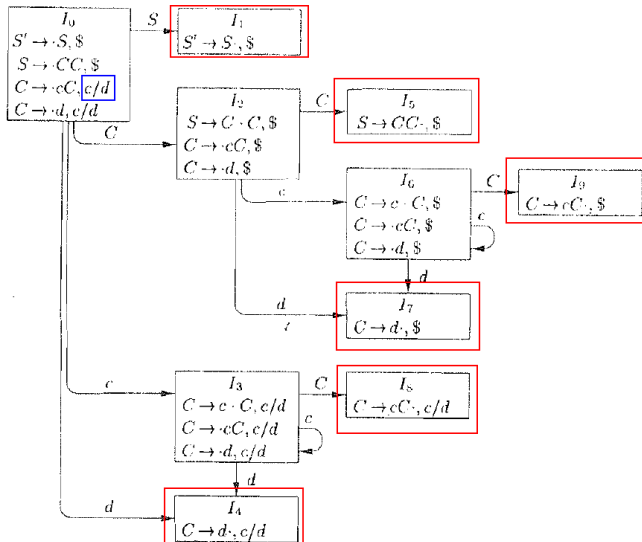
$S \rightarrow C C$

$C \rightarrow c C \mid d$

$L(G) = c^*dc^*d$



Q: 合并 I_4 与 I_7 为 I_{47} ($\{[C \rightarrow d \cdot, c/d/\$]\}$), 会怎样?

$w = ccd\$$ $w = cdcd\$$ $S' \rightarrow S$ $S \rightarrow C C$ $C \rightarrow c C \mid d$ $L(G) = c^*dc^*d$ 

继续合并 (I_8, I_9) 以及 (I_3, I_6)

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 I, J 的 GOTO 目标 (状态) 一定也是可以合并的

(I, J 出边集合相同, 且 $\text{GOTO}(I, X)$ 与 $\text{GOTO}(J, X)$ 具有相同的核项)

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 $[A \rightarrow \alpha \cdot, a]$ 与 $[B \rightarrow \beta \cdot a\gamma, b]$

则在 $LR(1)$ 自动机中,

存在某状态同时包含 $[A \rightarrow \alpha \cdot, a]$ 与 $[B \rightarrow \beta \cdot a\gamma, c]$ ($c \neq b$)

矛盾!

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。

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

$$S' \rightarrow S$$

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

$$A \rightarrow c$$

$$B \rightarrow c$$

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

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

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

Theorem

如果合并后的语法分析器**无冲突**, 则它的行为与原分析器本质上一**致**。

- (1) **接受**原分析器所接受的句子, 且状态转移相同
- (2) **拒绝**原分析器所拒绝的句子, 但可能多一些不必要的**归约**动作

(“实际上, 这个错误会在移入任何新的输入符号之前就被发现”)

(“两个分析器有相同的**移入**动作”)

$LALR(1)$ 语法分析器的优点

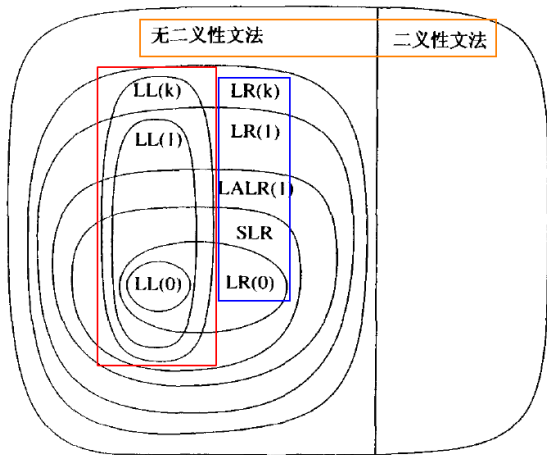
状态数量与 $SLR(1)$ 语法分析器的状态数量相同

($LALR(1)$ 与 $SLR(1)$ 都使用相同的 $LR(0)$ 核心项)

对于 $LR(1)$ 文法, 不会产生移入/归约冲突

Q: 但是你却通过 $LR(1)$ 自动机构造 $LALR(1)$ 项集族?

第 4.7.5 节 (本科教学版): 高效构造 $LALR(1)$ 语法分析表的方法



除 $LR(0)$ 外, 以上各种 LR 类文法对应的语言是等价的。

好消息: 善用 LR 语法分析器, 处理**二义性**文法

表达式文法

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

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

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

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

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

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

表达式文法: 使用 $SLR(1)$ 语法分析方法

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

$I_0: E' \rightarrow \cdot E$ $E \rightarrow \cdot E + E$ $E \rightarrow \cdot E * E$ $E \rightarrow \cdot (E)$ $E \rightarrow \cdot id$	$I_5: E \rightarrow E * \cdot E$ $E \rightarrow \cdot E + E$ $E \rightarrow \cdot E * E$ $E \rightarrow \cdot (E)$ $E \rightarrow \cdot id$
$I_1: E' \rightarrow E \cdot$ $E \rightarrow E \cdot + E$ $E \rightarrow E \cdot * E$	$I_6: E \rightarrow (E \cdot)$ $E \rightarrow E \cdot + E$ $E \rightarrow E \cdot * E$
$I_2: E \rightarrow (E \cdot)$ $E \rightarrow \cdot E + E$ $E \rightarrow \cdot E * E$ $E \rightarrow \cdot (E)$ $E \rightarrow \cdot id$	$I_7: E \rightarrow E + E \cdot$ $E \rightarrow E \cdot + E$ $E \rightarrow E \cdot * E$
$I_3: E \rightarrow id \cdot$	$I_8: E \rightarrow E * E \cdot$ $E \rightarrow E \cdot + E$ $E \rightarrow E \cdot * E$
$I_4: E \rightarrow E + \cdot E$ $E \rightarrow \cdot E + E$ $E \rightarrow \cdot E * E$ $E \rightarrow \cdot (E)$ $E \rightarrow \cdot id$	$I_9: E \rightarrow (E) \cdot$

$$\{+, *\} \subseteq FOLLOW(E)$$

考虑到**结合性与优先级**:

状态	ACTION						GOTO
	id	+	*	()	\$	
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	r5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

id + id * id

id + id + id

expr[**int** _p]

: (**INT**
| **ID**
)

({4 >= \$_p}? '*' **expr**[5]
| {3 >= \$_p}? '+' **expr**[4]
)*

;

条件语句文法

$stmt \rightarrow$ if $expr$ then $stmt$
 | if $expr$ then $stmt$ else $stmt$
 | other

$S' \rightarrow S$

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

条件语句文法: 使用 $SLR(1)$ 语法分析方法

$$S' \rightarrow S$$

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

$I_0:$	$S' \rightarrow \cdot S$ $S \rightarrow \cdot i S e S$ $S \rightarrow \cdot i S$ $S \rightarrow \cdot a$	$I_3:$	$S \rightarrow e \cdot$
$I_1:$	$S' \rightarrow S \cdot$	$I_4:$	$S \rightarrow i S \cdot e S$ $S \rightarrow i S \cdot$
$I_2:$	$S \rightarrow i \cdot S e S$ $S \rightarrow i \cdot S$ $S \rightarrow i S \cdot e S$ $S \rightarrow i S \cdot$ $S \rightarrow i a$	$I_5:$	$S \rightarrow i S e \cdot S$ $S \rightarrow i S e S \cdot$ $S \rightarrow i S \cdot$ $S \rightarrow \cdot a$
		$I_6:$	$S \rightarrow i S e S \cdot$

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	

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

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

$$S \rightarrow iEtSS' \mid a$$

$$S' \rightarrow eS \mid \epsilon$$

$$E \rightarrow b$$

非终结符号	输入符号					
	a	b	e	i	t	$\$$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S' \rightarrow \epsilon$ $S' \rightarrow eS$			$S' \rightarrow \epsilon$
E		$E \rightarrow b$				

if b then if b then a else a

解决二义性: 选择 $S' \rightarrow eS$, 将 **else** 与前面最近的 **then** 关联起来

```
prog : stat EOF;
```

```
stat : 'if' expr 'then' stat    // with higher priority  
    | 'if' expr 'then' stat 'else' stat  
    | expr  
    ;
```

Thank
You!



Office 926

hfwei@nju.edu.cn