



中山大學
SUN YAT-SEN UNIVERSITY



国家超级计算广州中心
NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

DCS290

Compilation Principle 编译原理

第四章 语法分析 (8)

郑馥丹

zhengfd5@mail.sysu.edu.cn

CONTENTS

目录

01

自顶向下分析
Top-Down Parsing

02

LL(1)分析
LL(1) Parsing

03

自底向上分析
Bottom-Up Parsing

04

LR分析
LR Parsing

6. 二义性文法的LR分析

- 对二义性文法的LR分析

- ① 消除二义性后分析

- ② 直接分析，但须添加额外的文法限制（如优先级或结合性等）

- 例：文法 $G[E]_1: E \rightarrow E + E | E * E | (E) | i$

- 消除二义性之后的等价文法为： $G[E]_2: E \rightarrow E + T | T, T \rightarrow T * F | F, F \rightarrow (E) | i$

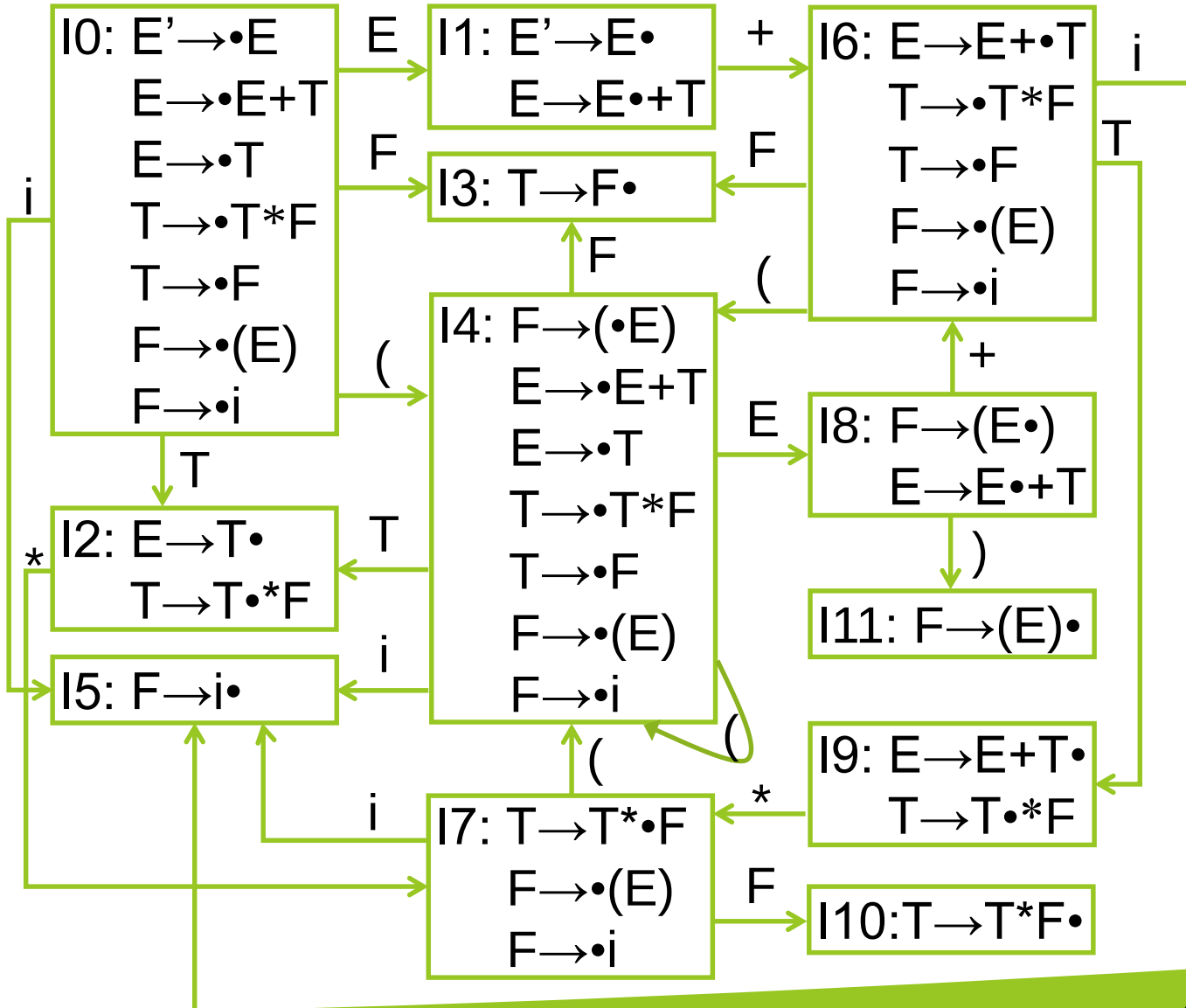
6. 二义性文法的LR分析

① 对无二义性的 $G[E]_2: E \rightarrow E+T|T, T \rightarrow T*F|F, F \rightarrow (E)|i$ 进行LR分析:

1. 拓广文法:

- (0) $E' \rightarrow E$ (1) $E \rightarrow E+T$ (2) $E \rightarrow T$
- (3) $T \rightarrow T*F$ (4) $T \rightarrow F$ (5) $F \rightarrow (E)$
- (6) $F \rightarrow i$

2. 构造LR(0)项目集及DFA



6. 二义性文法的LR分析

① 对无二义性的 $G[E]_2$: $E \rightarrow E+T|T$, $T \rightarrow T*F|F$, $F \rightarrow (E)|i$ 进行LR分析:

(0) $E' \rightarrow E$ (1) $E \rightarrow E+T$ (2) $E \rightarrow T$ (3) $T \rightarrow T*F$ (4) $T \rightarrow F$ (5) $F \rightarrow (E)$ (6) $F \rightarrow i$

3. 检查是否有冲突

I1:
 $E' \rightarrow E \cdot$
 $E \rightarrow E \cdot + T$

FOLLOW(E')={ $\$$ }与移进符号{+}无交集, 利用SLR(1)方法可解决冲突

I2:
 $E \rightarrow T \cdot$
 $T \rightarrow T \cdot * F$

FOLLOW(E)={+,), $\$$ }与移进符号{*}无交集, 利用SLR(1)方法可解决冲突

I9:
 $E \rightarrow E + T \cdot$
 $T \rightarrow T \cdot * F$

FOLLOW(E)={+,), $\$$ }与移进符号{*}无交集, 利用SLR(1)方法可解决冲突

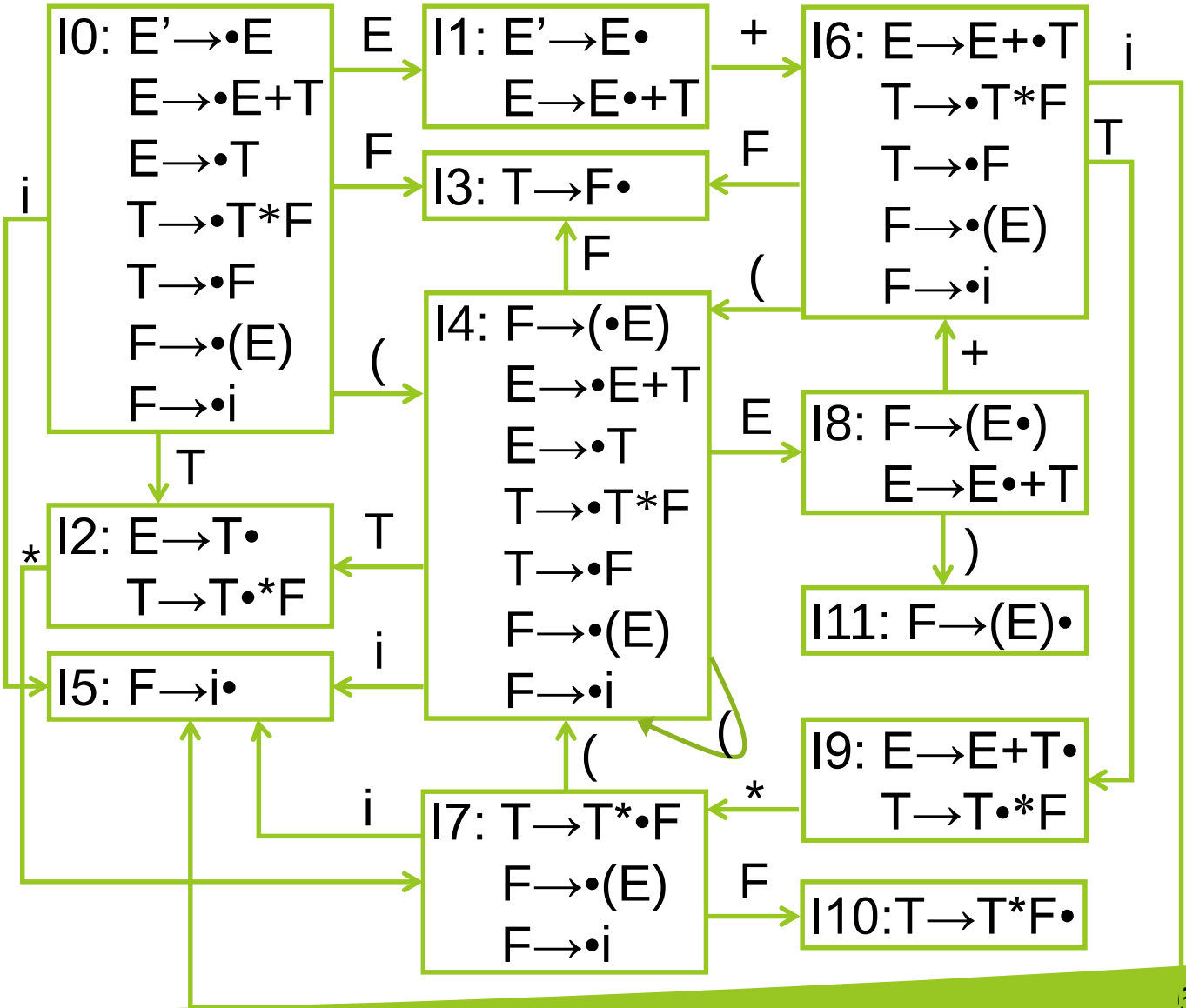
I1、I2、I9状态存在移进-归约冲突

6. 二义性文法的LR分析

(0) $E' \rightarrow E$ (1) $E \rightarrow E+T$ (2) $E \rightarrow T$ (3) $T \rightarrow T * F$ (4) $T \rightarrow F$ (5) $F \rightarrow (E)$ (6) $F \rightarrow i$

4. 进行SLR(1)分析

状态	ACTION						GOTO		
	i	+	*	(	)	\$	E	T	F
0	S ₅			S ₄			1	2	3
1		S ₆				acc			
2		r ₂	S ₇		r ₂	r ₂			
3		r ₄	r ₄		r ₄	r ₄			
4	S ₅			S ₄			8	2	3
5		r ₆	r ₆		r ₆	r ₆			
6	S ₅			S ₄				9	3
7	S ₅			S ₄					10
8		S ₆			S ₁₁				
9		r ₁	S ₇		r ₁	r ₁			
10		r ₃	r ₃		r ₃	r ₃			
11		r ₅	r ₅		r ₅	r ₅			



6. 二义性文法的LR分析

(0) $E' \rightarrow E$ (1) $E \rightarrow E+T$ (2) $E \rightarrow T$ (3) $T \rightarrow T*F$ (4) $T \rightarrow F$ (5) $F \rightarrow (E)$ (6) $F \rightarrow i$

4. 进行SLR(1)分析 **$i+i*i\$$**

状态	ACTION						GOTO		
	i	+	*	(	)	\$	E	T	F
0	S ₅			S ₄			1	2	3
1		S ₆				acc			
2		r ₂	S ₇		r ₂	r ₂			
3		r ₄	r ₄		r ₄	r ₄			
4	S ₅			S ₄			8	2	3
5		r ₆	r ₆		r ₆	r ₆			
6	S ₅			S ₄				9	3
7	S ₅			S ₄					10
8		S ₆			S ₁₁				
9		r ₁	S ₇		r ₁	r ₁			
10		r ₃	r ₃		r ₃	r ₃			
11		r ₅	r ₅		r ₅	r ₅			

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	$i+i*i\\$	S ₅	
2	05	\$i	$+i*i\$$	r ₆	3
3	03	\$F	$+i*i\$$	r ₄	2
4	02	\$T	$+i*i\$$	r ₂	1
5	01	\$E	$+i*i\$$	S ₆	
6	016	\$E+	$i*i\$$	S ₅	
7	0165	\$E+i	$*i\$$	r ₆	3
8	0163	\$E+F	$*i\$$	r ₄	9
9	0169	\$E+T	$*i\$$	S ₇	
10	01697	\$E+T*	$i\$$	S ₅	
11	016975	\$E+T*i	\$	r ₆	10
12	01697(10)	\$E+T*F	\$	r ₃	9
13	0169	\$E+T	\$	r ₁	
14	01	\$E	\$	acc	

6. 二义性文法的LR分析

② 直接分析：文法 $G[E]_1: E \rightarrow E+E | E^*E | (E) | i$

1. 拓广文法为：(0) $E' \rightarrow E$ (1) $E \rightarrow E+E$ (2) $E \rightarrow E^*E$ (3) $E \rightarrow (E)$ (4) $E \rightarrow i$

2. 可以分别验证该文法非LR(0)文法、非SLR(1)文法、亦非LR(1)文法

I7: $E \rightarrow E+E \cdot$
 $E \rightarrow E \cdot +E$
 $E \rightarrow E \cdot *E$

I8: $E \rightarrow E^*E \cdot$
 $E \rightarrow E \cdot +E$
 $E \rightarrow E \cdot *E$

FOLLOW(E)={+,*,),}\$与移进符号{+,*}有交集，利用SLR(1)方法无法解决冲突

非SLR(1)文法

I9: $E \rightarrow E+E \cdot, \$/+/*$
 $E \rightarrow E \cdot +E, \$/+/*$
 $E \rightarrow E \cdot *E, \$/+/*$

移进和归约的向前搜索符号集有交集，利用LR(1)方法无法解决冲突

非LR(1)文法

3. 但可以对其人为施加限制——利用优先关系和结合性：

- *的优先级高于+
- *和+都服从左结合

6. 二义性文法的LR分析

② 直接分析：文法 $G[E]_1: E \rightarrow E + E | E * E | (E) | i$

3. 但可以对其人为施加限制——利用优先关系和结合性：

- *的优先级高于+
- *和+都服从左结合

17: $E \rightarrow E + E \cdot$
 $E \rightarrow E \cdot + E$
 $E \rightarrow E \cdot * E$

***比+优先级高，所以遇到*则移进；** 符号栈 $E + E$ ，遇到*，应移进而非归约；
+服从左结合，所以遇到+则归约。 符号栈 $E + E$ ，遇到+，应归约。

18: $E \rightarrow E * E \cdot$
 $E \rightarrow E \cdot + E$
 $E \rightarrow E \cdot * E$

***比+优先级高，*服从左结合，** 符号栈 $E + E * E$ ，不管遇到+还是*都应归约。
所以不论遇到+或*都应归约。

6. 二义性文法的LR分析

② 直接分析: (0) $E' \rightarrow E$ (1) $E \rightarrow E + E$ (2) $E \rightarrow E * E$ (3) $E \rightarrow (E)$ (4) $E \rightarrow i$

4. 按上述优先关系和结合性构造SLR(1)分析表

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S ₃			S ₂			1
1		S ₄	S ₅			acc	
2	S ₃			S ₂			6
3		r ₄	r ₄		r ₄	r ₄	
4	S ₃			S ₂			7
5	S ₃			S ₂			8
6		S ₄	S ₅		S ₉		
7		r ₁	S ₅		r ₁	r ₁	
8		r ₂	r ₂		r ₂	r ₂	
9		r ₃	r ₃		r ₃	r ₃	

二义性文法SLR(1)分析表

VS.

状态	ACTION						GOTO		
	i	+	*	(	)	\$	E	T	F
0	S ₅			S ₄			1	2	3
1		S ₆				acc			
2		r ₂	S ₇		r ₂	r ₂			
3		r ₄	r ₄		r ₄	r ₄			
4	S ₅			S ₄			8	2	3
5		r ₆	r ₆		r ₆	r ₆			
6	S ₅			S ₄				9	3
7	S ₅			S ₄					10
8		S ₆			S ₁₁				
9		r ₁	S ₇		r ₁	r ₁			
10		r ₃	r ₃		r ₃	r ₃			
11		r ₅	r ₅		r ₅	r ₅			

无二义性文法SLR(1)分析表

6. 二义性文法的LR分析

② 直接分析: (0) $E' \rightarrow E$ (1) $E \rightarrow E + E$ (2) $E \rightarrow E * E$ (3) $E \rightarrow (E)$ (4) $E \rightarrow i$

5. 进行SLR(1)分析 **$i+i*i\$$**

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S ₃			S ₂			1
1		S ₄	S ₅			acc	
2	S ₃			S ₂			6
3		r ₄	r ₄		r ₄	r ₄	
4	S ₃			S ₂			7
5	S ₃			S ₂			8
6		S ₄	S ₅		S ₉		
7		r ₁	S ₅		r ₁	r ₁	
8		r ₂	r ₂		r ₂	r ₂	
9		r ₃	r ₃		r ₃	r ₃	

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	i+i*i\$	S ₃	
2	03	\$i	+i*i\$	r ₄	1
3	01	\$E	+i*i\$	S ₄	
4	014	\$E+	i*i\$	S ₃	
5	0143	\$E+i	*i\$	r ₄	7
6	0147	\$E+E	*i\$	S ₅	
7	01475	\$E+E*	i\$	S ₃	
8	014753	\$E+E*i	\$	r ₄	8
9	014758	\$E+E*E	\$	r ₂	7
10	0147	\$E+E	\$	r ₁	1
11	01	\$E	\$	acc	

6. 二义性文法的LR分析

② 直接分析：(0) $E' \rightarrow E$ (1) $E \rightarrow E + E$ (2) $E \rightarrow E * E$ (3) $E \rightarrow (E)$ (4) $E \rightarrow i$

5. 进行SLR(1)分析 i+i*i\$

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	i+i*i\$	S ₃	
2	03	\$i	+i*i\$	r ₄	1
3	01	\$E	+i*i\$	S ₄	
4	014	\$E+	i*i\$	S ₃	
5	0143	\$E+i	*i\$	r ₄	7
6	0147	\$E+E	*i\$	S₅	
7	01475	\$E+E*	i\$	S ₃	
8	014753	\$E+E*i	\$	r ₄	8
9	014758	\$E+E*E	\$	r₂	7
10	0147	\$E+E	\$	r₁	1
11	01	\$E	\$	acc	

二义性文法SLR(1)分析

VS.

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	i+i*i\$	S ₅	
2	05	\$i	+i*i\$	r ₆	3
3	03	\$F	+i*i\$	r ₄	2
4	02	\$T	+i*i\$	r ₂	1
5	01	\$E	+i*i\$	S ₆	
6	016	\$E+	i*i\$	S ₅	
7	0165	\$E+i	*i\$	r ₆	3
8	0163	\$E+F	*i\$	r ₄	9
9	0169	\$E+T	*i\$	S ₇	
10	01697	\$E+T*	i\$	S ₅	
11	016975	\$E+T*i	\$	r ₆	10
12	01697(10)	\$E+T*F	\$	r ₃	9
13	0169	\$E+T	\$	r ₁	
14	01	\$E	\$	acc	

无二义性文法SLR(1)分析

6. 二义性文法的LR分析

- 二义性文法——悬空-else[Dangling-else]:

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

if E1 then if E2 then S1 else S2

- 其对应的无二义性的文法为:

$$stmt \rightarrow matched_stmt \mid open_stmt$$

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

$$open_stmt \rightarrow \text{if } expr \text{ then } stmt$$

if E1 then if E2 then S1 else S2

$$| \text{ if } expr \text{ then } matched_stmt \text{ else } open_stmt$$

6. 二义性文法的LR分析

- 二义性文法——悬空-else[Dangling-else]:

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

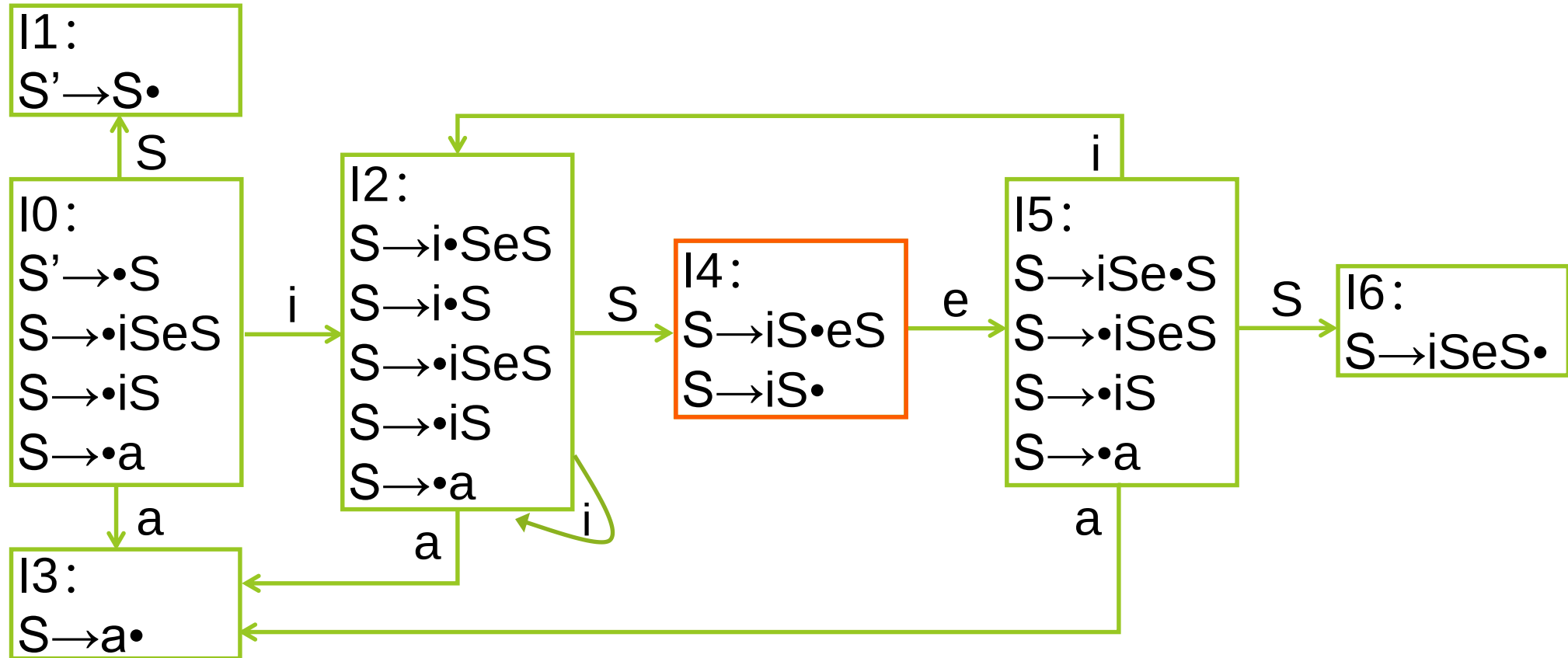
- 假设用**S**表示 $stmt$, 用**i**表示 $\text{if } expr \text{ then}$, **e**表示 else , **a**表示 other :

$$S \rightarrow iS \mid iSeS \mid a$$

- 拓广为: (0) $S' \rightarrow S$ (1) $S \rightarrow iSeS$ (2) $S \rightarrow iS$ (3) $S \rightarrow a$

6. 二义性文法的LR分析

- 对该文法进行LR分析：(0) $S' \rightarrow S$ (1) $S \rightarrow iSeS$ (2) $S \rightarrow iS$ (3) $S \rightarrow a$



存在移进-归约冲突；

但 $FOLLOW(S) = \{e, \$\}$ ，因此用SLR(1)无法解决冲突；

可尝试添加人为限制：else总是与离它最近的then配对。

6. 二义性文法的LR分析

- 对该文法进行LR分析：(0) $S' \rightarrow S$ (1) $S \rightarrow iSeS$ (2) $S \rightarrow iS$ (3) $S \rightarrow a$

I4:

$S \rightarrow iS \cdot eS$

$S \rightarrow iS \cdot$

存在移进-归约冲突；

但 $FOLLOW(S) = \{e, \$\}$ ，因此用SLR(1)无法解决冲突；

可尝试添加人为限制：else总是与离它最近的then配对。

假设符号栈中为*\$iS*，面临输入符号*e*，恢复成为Dangling-else文法的表达即为：

符号栈中为*\$if expr then stmt*，面临输入符号*else*，

此时，按照“**else总是与离它最近的then配对**”的限制条件，应考虑**移进**，**冲突得以解决**。

6. 二义性文法的LR分析

- 对该文法进行LR分析：(0) $S' \rightarrow S$ (1) $S \rightarrow iSeS$ (2) $S \rightarrow iS$ (3) $S \rightarrow a$
 - 对输入 **if** *expr* **then if** *expr* **then other else other**\$ **ii***aea*\$

状态	ACTION				GOTO
	i	e	a	\$	
0	S ₂		S ₃		1
1				acc	
2	S ₂		S ₃		4
3		r ₃		r ₃	
4		S ₅		r ₂	
5	S ₂		S ₃		6
6		r ₁		r ₁	

Dangling-else文法的SLR(1)分析表

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	ii <i>aea</i> \$	S ₂	
2	02	\$i	<i>iaea</i> \$	S ₂	
3	022	\$ii	<i>aea</i> \$	S ₃	
4	0223	\$iia	<i>ea</i> \$	r ₃	4
5	0224	\$iiS	<i>ea</i> \$	S₅	
6	02245	\$iiSe	<i>a</i> \$	S ₃	
7	022453	\$iiSea	\$	r ₃	6
8	022456	\$iiSeS	\$	r ₁	4
9	024	\$iS	\$	r ₂	1
10	01	\$S	\$	acc	

Dangling-else文法的SLR(1)分析过程

7. LR分析中的错误恢复[Error Recovery]

- 若能构造出**没有冲突**的LR分析表，则输入的句子中的**所有错误都能被发现**
(LR分析表中的空白位置)
- 各种LR分析发现错误的位置可能不一样
 - 在发现错误前需要执行的归约次数： $LR(0) \geq SLR(1) \geq LALR(1) \geq LR(1)$
- 回顾：错误恢复
 - Parser在Error情况下应该做什么？
 - ✓ **给出Error信息**：Error信息越详尽越好
 - ✓ **从Error中恢复**：能够继续对其余输入进行解析（不要因发现错误而停止语法分析）

7. LR分析中的错误恢复[Error Recovery]

- 回顾：恐慌模式[Panic Mode]

- 是一种错误恢复策略
- 恐慌：遇到危险先跑再说，不尝试精确修复错误，直接跳过
- 目标：让parser尽快回到可以继续分析的状态，而不是纠结于错误的细节
- 基本思想：当parser遇到一个无法处理的错误时，采取“紧急避险”的方式，简单粗暴地忽略/跳过输入中的一些符号，直到输入中出现一些特定的符号

7. LR分析中的错误恢复[Error Recovery]

• 恐慌模式[Panic Mode]

(0) $E' \rightarrow E$ (1) $E \rightarrow E + E$ (2) $E \rightarrow E * E$ (3) $E \rightarrow (E)$ (4) $E \rightarrow i$

- 发现错误：当前状态面临当前输入符没有合法动作（分析表对应位置空白）
- 解决方式：
 - ✓ 从栈顶往下扫描
 - ✓ 直到发现某个状态 S ，它有一个对应于某个非终结符 A 的GOTO目标
 - ✓ 丢弃零个或多个输入符号，直到发现一个可能合法地跟在 A 之后的符号 a 为止
 - ✓ 将GOTO(S , A)压入栈中
 - ✓ 继续进行正常的语法分析

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S_3			S_2			1
1		S_4	S_5			acc	
2	S_3			S_2			6
3		r_4	r_4		r_4	r_4	
4	S_3			S_2			7
5	S_3			S_2			8
6		S_4	S_5		S_9		
7		r_1	S_5		r_1	r_1	
8		r_2	r_2		r_2	r_2	
9		r_3	r_3		r_3	r_3	

假设当前状态栈为01478，输入字符串为 $i*i+i\$$

7. LR分析中的错误恢复[Error Recovery]

- **短语级别的错误恢复**

- 检查LR分析表中的**每个报错条目**
- 根据语言的使用方法来决定程序员所犯的何种错误最有可能引起这个语法错误
- 构造出适当的恢复过程：给出具体的错误处理例程
- 在分析表的ACTION表中**每个空条目中填写一个指向错误处理例程的指针**

7. LR分析中的错误恢复[Error Recovery]

• 短语级别的错误恢复

– 例：文法 $G[E]: E \rightarrow E + E | E * E | (E) | i$

状态	ACTION						GOTO
	i	+	*	(	)	\$	
0	S ₃			S ₂			1
1		S ₄	S ₅			acc	
2	S ₃			S ₂			6
3		r ₄	r ₄		r ₄	r ₄	
4	S ₃			S ₂			7
5	S ₃			S ₂			8
6		S ₄	S ₅		S ₉		
7		r ₁	S ₅		r ₁	r ₁	
8		r ₂	r ₂		r ₂	r ₂	
9		r ₃	r ₃		r ₃	r ₃	

SLR(1)分析表

状态	ACTION						GOTO
	i	+	*	(	)	\$	
0	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	1
1	e ₃	S ₄	S ₅	e ₃	e ₂	acc	
2	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	6
3	r ₄	r ₄	r ₄	r ₄	r ₄	r ₄	
4	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	7
5	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	8
6	e ₃	S ₄	S ₅	e ₃	S ₉	e ₄	
7	r ₁	r ₁	S ₅	r ₁	r ₁	r ₁	
8	r ₂	r ₂	r ₂	r ₂	r ₂	r ₂	
9	r ₃	r ₃	r ₃	r ₃	r ₃	r ₃	

带错误恢复例程的SLR(1)分析表

7. LR分析中的错误恢复[Error Recovery]

• 短语级别的错误恢复

– 例：文法 $G[E]: E \rightarrow E + E | E * E | (E) | i$

- ✓ r_j : 对那些在某些输入上执行特定归约动作的状态，将报错条目替换为归约动作，延后报错的时间；
- ✓ e_1 : 在状态0,2,4,5上，这些状态原本期望读入一个运算分量的第一个符号(i或左括号)，但实际却读入+或*或\$，此时：将状态0,2,4,5在i上的移进目标，即**状态3压入状态栈，符号i压入符号栈**，并报错：**缺少运算分量**；

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S_3	e_1	e_1	S_2	e_2	e_1	1
1	e_3	S_4	S_5	e_3	e_2	acc	
2	S_3	e_1	e_1	S_2	e_2	e_1	6
3	r_4	r_4	r_4	r_4	r_4	r_4	
4	S_3	e_1	e_1	S_2	e_2	e_1	7
5	S_3	e_1	e_1	S_2	e_2	e_1	8
6	e_3	S_4	S_5	e_3	S_9	e_4	
7	r_1	r_1	S_5	r_1	r_1	r_1	
8	r_2	r_2	r_2	r_2	r_2	r_2	
9	r_3	r_3	r_3	r_3	r_3	r_3	

带错误恢复例程的SLR(1)分析表

7. LR分析中的错误恢复[Error Recovery]

• 短语级别的错误恢复

– 例：文法 $G[E]: E \rightarrow E + E | E * E | (E) | i$

✓ e_2 ：在状态0,1,2,4,5上，遇到右括号，此时：
从输入中删除右括号，并报错：**不匹配的右括号**；

✓ e_3 ：在状态1,6上，期待读入一个运算符却发现了一个i或左括号，此时：将状态1,6在+上的移进目标，即**状态4压入状态栈，符号+移进符号栈**，并报错：**缺少运算符**；

✓ e_4 ：在状态6上发现输入是结束符\$，此时：将状态6在右括号上的移进目标，即**状态9压入状态栈中，右括号压入符号栈**，并报错：**缺少右括号**。

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S_3	e_1	e_1	S_2	e_2	e_1	1
1	e_3	S_4	S_5	e_3	e_2	acc	
2	S_3	e_1	e_1	S_2	e_2	e_1	6
3	r_4	r_4	r_4	r_4	r_4	r_4	
4	S_3	e_1	e_1	S_2	e_2	e_1	7
5	S_3	e_1	e_1	S_2	e_2	e_1	8
6	e_3	S_4	S_5	e_3	S_9	e_4	
7	r_1	r_1	S_5	r_1	r_1	r_1	
8	r_2	r_2	r_2	r_2	r_2	r_2	
9	r_3	r_3	r_3	r_3	r_3	r_3	

带错误恢复例程的SLR(1)分析表

7. LR分析中的错误恢复[Error Recovery]

• 短语级别的错误恢复

– 例：文法G[E]: $E \rightarrow E + E | E * E | (E) | i$

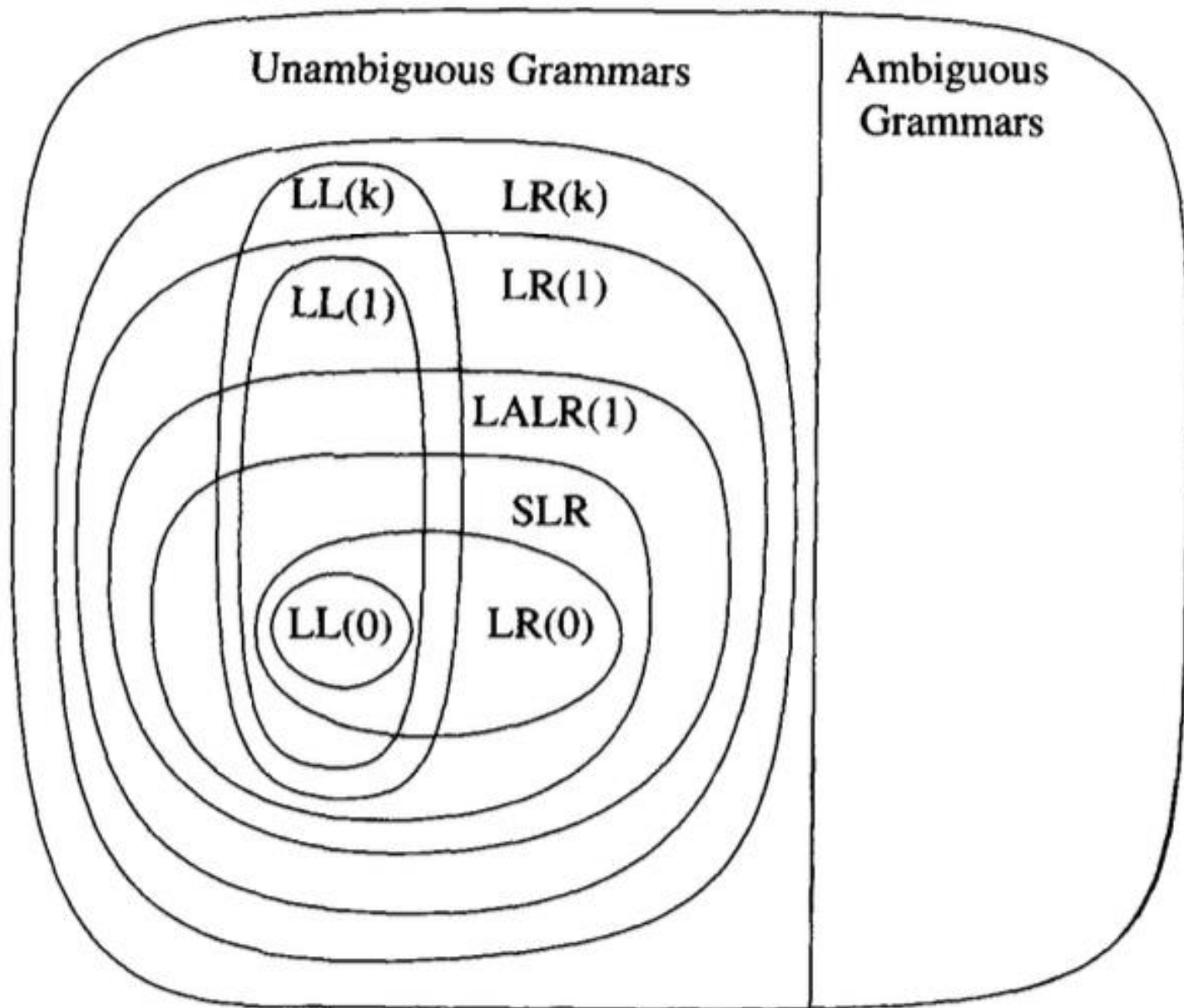
步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	\$	i+)\$	S ₃	
2	03	\$i	+) \$	r ₄	1
3	01	\$E	+) \$	S ₄	
4	014	\$E+	) \$	e ₂ : 不匹配的右括号, 从输入中删除右括号	
5	014	\$E+	\$	e ₁ : 缺少运算分量, 将状态3压入状态栈, 符号i压入符号栈	
6	0143	\$E+i	\$	r ₄	7
7	0147	\$E+E	\$	r ₁	1
8	01	\$E	\$	end	

状态	ACTION						GOTO
	i	+	*	(	)	\$	E
0	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	1
1	e ₃	S ₄	S ₅	e ₃	e ₂	acc	
2	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	6
3	r ₄	r ₄	r ₄	r ₄	r ₄	r ₄	
4	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	7
5	S ₃	e ₁	e ₁	S ₂	e ₂	e ₁	8
6	e ₃	S ₄	S ₅	e ₃	S ₉	e ₄	
7	r ₁	r ₁	S ₅	r ₁	r ₁	r ₁	
8	r ₂	r ₂	r ₂	r ₂	r ₂	r ₂	
9	r ₃	r ₃	r ₃	r ₃	r ₃	r ₃	

带错误恢复例程的SLR(1)分析表

总结

- 自顶向下语法分析
 - 递归下降预测分析
 - LL(1)分析
- 自底向上语法分析[移进-归约]
 - 算法优先分析
 - LR分析
 - ✓ LR(0)
 - ✓ SLR(1)
 - ✓ LR(1)
 - ✓ LALR(1)



总结

- 重点掌握

- 进行LL(1)分析

- ✓ 消除左公因子、左递归、二义性等情况
 - ✓ 求FIRST集、FOLLOW集
 - ✓ 构造LL(1)分析表
 - ✓ 进行LL(1)分析

- 进行LR分析——LR(0)、SLR(1)、LR(1)、LALR(1)

- ✓ 构建项目集、CLOSURE函数、GOTO函数、DFA
 - ✓ 构建LR分析表
 - ✓ 进行LR分析
 - ✓ 能分辨出一个文法是属于哪种LR文法

第四章课后作业（2）

- 证明文法 $G[S]: S \rightarrow Ca|bCe|De|bDa, C \rightarrow d, D \rightarrow d$ 是LR(1)而不是LALR(1)的，构造LR(1)分析表，并对输入字符串bda\$进行LR(1)分析，写出分析的全过程（表格形式）。
- 提交要求：
 - 文件命名：学号-姓名-第四章作业(2)；
 - 文件格式：.pdf文件；
 - 手写版、电子版均可；若为手写版，则拍照后转成pdf提交，但**须注意将照片旋转为正常角度，且去除照片中的多余信息**；电子版如word等转成pdf提交；
 - 提交到超算习堂（第四章作业(2)）处；
 - 提交ddl：**4月22日晚上12:00**；
 - **重要提示：不得抄袭！**