

H3-1,H3-2

考虑文法： $S \rightarrow aSbS \mid bSaS \mid \epsilon$

- 为句子abab构造两个不同的最左推导，以此说明该文法是二义的

$S \Rightarrow_{lm} aSbS \Rightarrow_{lm} abS \Rightarrow_{lm} abaSbS \Rightarrow_{lm} ababS \Rightarrow_{lm} abab$

$S \Rightarrow_{lm} aSbS \Rightarrow_{lm} abSaSbS \Rightarrow_{lm} abaSbS \Rightarrow_{lm} ababS \Rightarrow_{lm} abab$

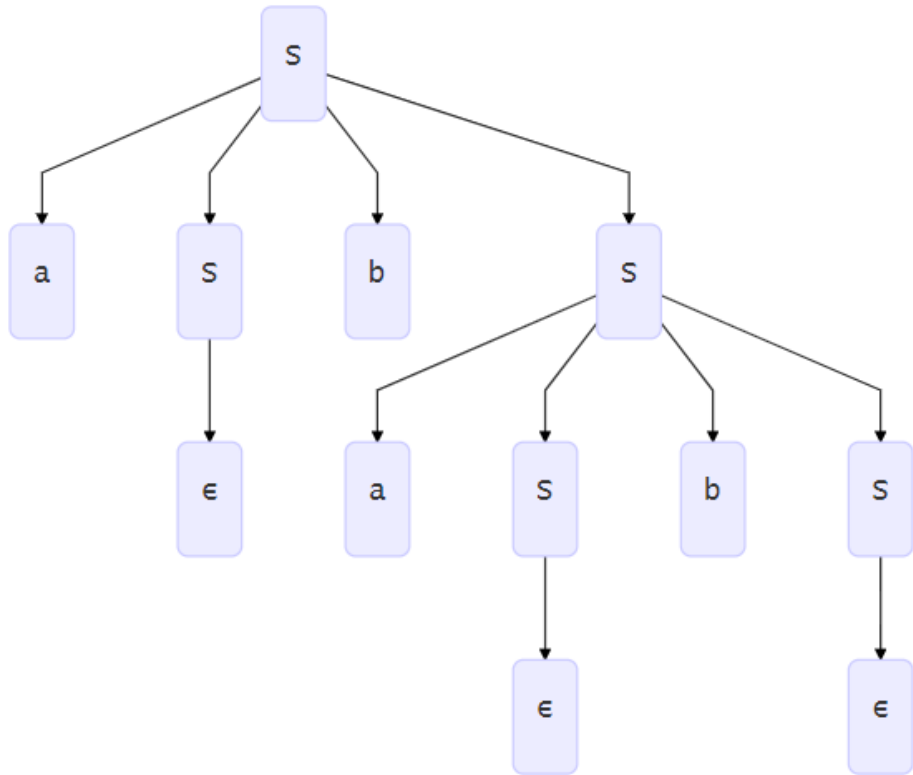
考虑文法： $S \rightarrow aSbS \mid bSaS \mid \epsilon$

- 为abab构造最右推导

$$S \Rightarrow_{rm} aSbS \Rightarrow_{lm} aSbaSbS \Rightarrow_{rm} aSbaSb \Rightarrow_{rm} aSbab \Rightarrow_{rm} abab$$

考虑文法： $S \rightarrow aSbS \mid bSaS \mid \epsilon$

- 为abab构造对应的分析树



考虑文法： $S \rightarrow aSbS \mid bSaS \mid \epsilon$

- 该文法产生的语言是什么

仅由a和b组成的，且两者数量相同的字符串

文法  $R \rightarrow R' \mid ' R \mid R R \mid R \mid (R) \mid a \mid b$

- 证明该文法产生字母表  $\{a, b\}$  上的所有正规式

先证该文法产生的串是字母表  $\{a, b\}$  上的产生式

$R \rightarrow a$  和  $R \rightarrow b$  产生  $a$  和  $b$ ，而  $a$  和  $b$  是字母表中的符号，因此是正规式

假设  $R_1$  和  $R_2$  分别产生正规式  $\alpha$  和  $\beta$ ，则

$R \rightarrow R_1 R_2$  产生正规式  $\alpha\beta$

$R \rightarrow R_1 \mid R_2$  产生正规式  $\alpha \mid \beta$

$R \rightarrow R_1^*$  产生正规式  $\alpha^*$

$R \rightarrow (R_1)$  产生正规式  $(\alpha)$

综上，该文法经多步推导产生的串均为字母表  $\{a, b\}$  上的产生式

文法  $R \rightarrow R' \mid ' R \mid R R \mid R \mid (R) \mid a \mid b$

- 证明该文法产生字母表  $\{a, b\}$  上的所有正规式

再证字母表  $\{a, b\}$  上的所有正规式都可以由此文法产生

归纳法: 通过正规式长度, 或者推导步数进行归纳.

归纳基础:  $R \Rightarrow a; R \Rightarrow b.$

归纳假设: 少于  $n$  步的推导产生字母表  $\{a, b\}$  上的正规式, 如  $R \Rightarrow^* x, R \Rightarrow^* y.$

归纳步骤:  $n$  步的最左推导:

$$1. R \Rightarrow R' \mid ' R \Rightarrow^* x' \mid ' R \Rightarrow^* x' \mid ' y$$

$$2. R \Rightarrow R R \Rightarrow^* x R \Rightarrow^* x y$$

$$3. R \Rightarrow R \Rightarrow^* x$$

$$4. R \Rightarrow (R) \Rightarrow^* (x)$$

综上, 字母表  $\{a, b\}$  上的所有正规式都可以由此文法产生

文法  $R \rightarrow R' \mid ' R \mid R R \mid R * \mid (R) \mid a \mid b$

为该文法写一个等价的非二义文法

$E \rightarrow E' \mid ' T \mid T$

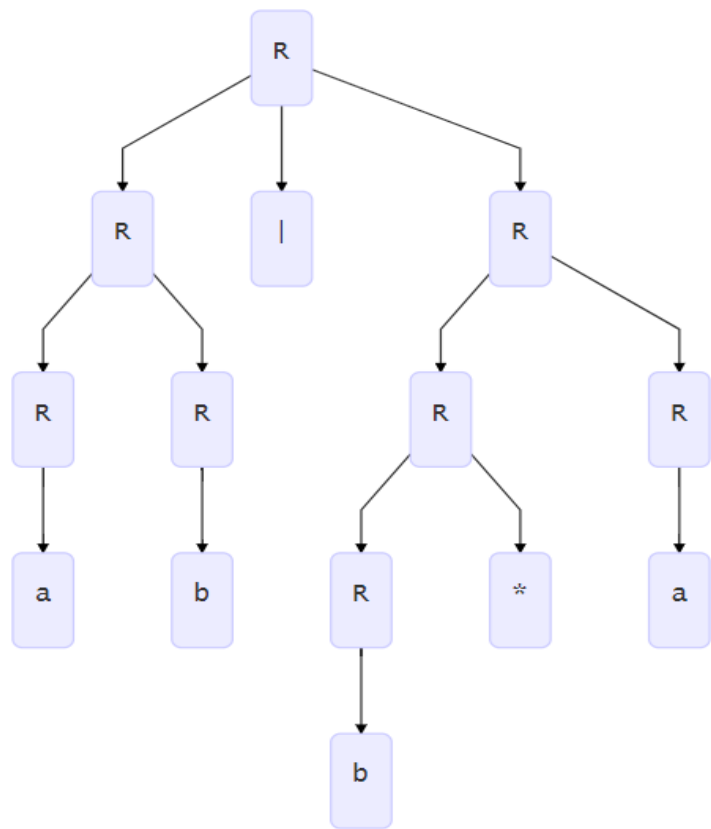
$T \rightarrow T F \mid F$

$F \rightarrow F * \mid (E) \mid a \mid b$

参考书上/ppt上对于加减乘除括号的非二义文法

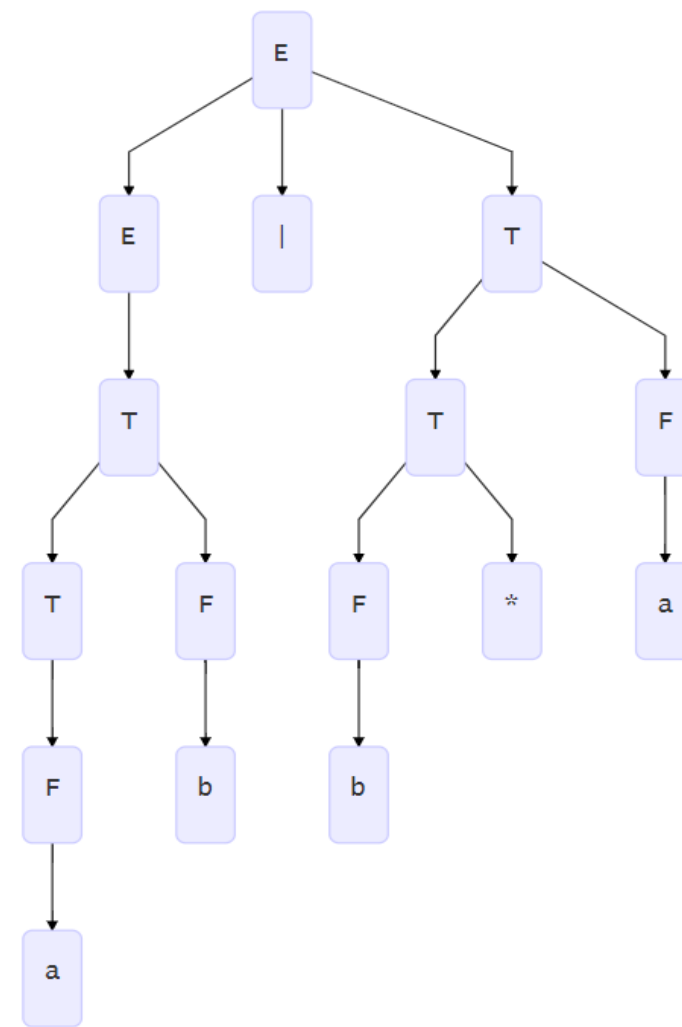
文法  $R \rightarrow R' \mid ' R \mid R R \mid R * \mid (R) \mid a \mid b$

为两个文法构造句子  $ab|b^*a$  的分析树



左为二义文法分析树

右为非二义文法分析树



为字母表  $\sigma = \{a, b\}$  上的下列语言设计文法

不含  $abb$  作为子串的所有串, 正规?

每个  $a$  之后最多只能出现一个  $b$   
要么前面没有  $a$ , 要么  $ab?$  的\*

$$S \rightarrow BA$$
$$A \rightarrow abA \mid aA \mid \epsilon$$
$$B \rightarrow bB \mid \epsilon$$
$$b^*(ab?)^*$$

正规!

为字母表  $\sigma = \{a, b\}$  上的下列语言设计文法

a的个数比b多的串,正规?

a的个数比b多, 可以拆解为多个a的个数比b多1的子串

$$S \rightarrow AS \mid A$$

$$A \rightarrow aE \mid bAA$$

$$B \rightarrow bE \mid aBB$$

$$E \rightarrow aB \mid bA \mid \epsilon$$

不正规

消除左递归： $S \rightarrow (L) \mid a$      $L \rightarrow L, S \mid S$

$S \rightarrow (L) \mid a$

$L \rightarrow S L'$

$L' \rightarrow , S L' \mid \epsilon$

注意:

$S \rightarrow (L) \mid a$

$L \rightarrow S$

没有左递归

H4-1,H4-2

下面的文法是否为LL(1)文法

$$S \rightarrow AB \mid PQx$$

$$A \rightarrow xy$$

$$B \rightarrow bc$$

$$P \rightarrow dP \mid \epsilon$$

$$Q \rightarrow aQ \mid \epsilon$$

$$First(AB) = \{x\}$$

$$First(PQx) = \{d, a, x\}$$

$$First(AB) \cap First(PQx) \neq \Phi$$

所以该文法不是LL(1)文法

# 构造下面文法的LL(1)分析表

$$S \rightarrow aBS \mid bAS \mid \epsilon$$

$$A \rightarrow bAA \mid a$$

$$B \rightarrow aBB \mid b$$

非终结符	输入符号		
	$a$	$b$	$\$$
$S$	$S \rightarrow aBS$	$S \rightarrow bAS$	$S \rightarrow \epsilon$
$A$	$A \rightarrow a$	$A \rightarrow bAA$	
$B$	$B \rightarrow aBB$	$B \rightarrow b$	

阅读并理解PL/O编译器，叙述PL/O编译器中test函数的作用以及他的三个参数的含义，并举例说明第二个参数带来的好处和引起的问题

- 三个参数的含义

- 可允许的下一个符号集合S1，如果当前符号不在此集合中，当即得到一个错误号
- 另加的停止符号集合S2，有些符号的出现，虽然无疑是错的，但他们绝对不应被忽略而跳过
- 整数n，表示有关错误的诊断号

- 第二个参数带来的好处与引发的问题

- 好处：分析出现错误后，不是短视地直接跳到下一个后继符号出现的位置，而是希望在停止符号处就能够停下，从而尝试发现更多的错误
- 问题：引入S2会让向后读的过程更早地结束,有时会结束在本应该跳过但是因为S2的引入而没有跳过的地方,使后续分析报更多的错误.

H5

# 为文法构造SLR和LALR分析表

- $E \rightarrow E + T | T$
- $T \rightarrow TF | F$
- $F \rightarrow F^* | a | b$

# SLR Items

$I_0:$   
 $E' \rightarrow .E$   
 $E \rightarrow .E + T$   
 $E \rightarrow .T$   
 $T \rightarrow .TF$   
 $T \rightarrow .F$   
 $F \rightarrow .F^*$   
 $F \rightarrow .a$   
 $F \rightarrow .b$

$I_1:$   
 $E' \rightarrow E.$   
 $E \rightarrow E. + T$

$I_2:$   
 $E \rightarrow T.$   
 $T \rightarrow T.F$   
 $F \rightarrow .F^*$   
 $F \rightarrow .a$   
 $F \rightarrow .b$

$I_3:$   
 $T \rightarrow F.$   
 $F \rightarrow F.*$

$I_4:$   
 $T \rightarrow a.$

$I_5:$   
 $T \rightarrow b.$

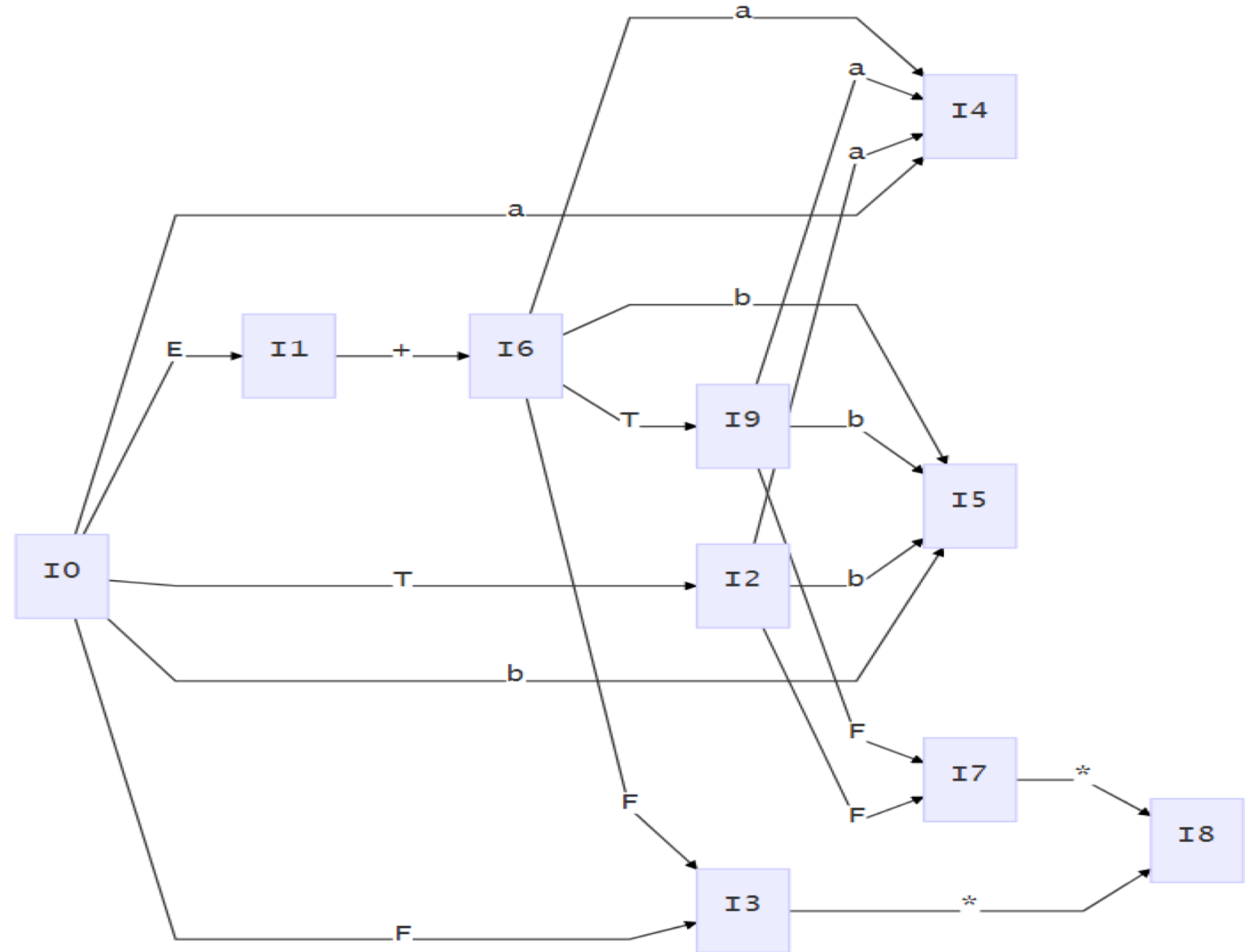
$I_6:$   
 $E \rightarrow E+.T$   
 $T \rightarrow .TF$   
 $T \rightarrow .F$   
 $F \rightarrow .F^*$   
 $F \rightarrow .a$   
 $F \rightarrow .b$

$I_7:$   
 $T \rightarrow TF.$   
 $F \rightarrow F.*$

$I_8:$   
 $F \rightarrow F*.$

$I_9:$   
 $E \rightarrow E + T.$   
 $T \rightarrow T.F$   
 $F \rightarrow .F^*$   
 $F \rightarrow .a$   
 $F \rightarrow .b$

# SLR DFA



# Production No.

1.  $E \rightarrow E + T$

2.  $E \rightarrow T$

3.  $T \rightarrow TF$

4.  $T \rightarrow F$

5.  $F \rightarrow F^*$

6.  $F \rightarrow a$

7.  $F \rightarrow b$

# SLR Table

state	action					goto		
	+	*	a	b	\$	e	t	f
0			s4	s5		1	2	3
1	s6				acc			
2	r2		s4	s5	r2			7
3	r4	s8	r4	r4	r4			
4	r6	r6	r6	r6	r6			
5	r7	r7	r7	r7	r7			
6			s4	s5			9	3
7	r3	s8	r3	r3	r3			
8	r5	r5	r5	r5	r5			
9	r1		s4	s5	r1			7

就是耐心地严格地按照算法3.5的closure进行迭代

# LR(1) 项目集规范族

$I_0:$

$E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_1:$

$E' \rightarrow E \cdot, \$$   
 $E \rightarrow E \cdot + T, +/\$$

$I_2:$

$E \rightarrow T \cdot, +/\$$   
 $T \rightarrow T \cdot F, +/a/b/\$$   
 $F \rightarrow F \cdot, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_3:$

$T \rightarrow F \cdot, +/a/b/\$$   
 $F \rightarrow F \cdot, */+/a/b/\$$

$I_4:$

$T \rightarrow a \cdot, */+/a/b/\$$

$I_5:$

$T \rightarrow b \cdot, */+/a/b/\$$

$I_6:$

$E \rightarrow E + \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_7:$

$T \rightarrow TF \cdot, +/a/b/\$$   
 $F \rightarrow F \cdot, */+/a/b/\$$

$I_8:$

$F \rightarrow F^* \cdot, */+/a/b/\$$

$I_9:$

$E \rightarrow E + T \cdot, +/\$$   
 $T \rightarrow T \cdot F, +/a/b/\$$   
 $F \rightarrow F \cdot, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

# LR(1) Item for $I_0$ (以 $I_0$ 为例)

$I_0:$ $E' \rightarrow \cdot E, \$$
--

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
→

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

$E + T, \$$   
→

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
→

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

$E + T, \$$   
→

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$

$T, +/\$$   
→

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/\$$   
 $T \rightarrow \cdot F, +/\$$

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

$E + T, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$

$T, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/\$$   
 $T \rightarrow \cdot F, +/\$$

$TF, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

$E + T, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$

$T, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/\$$   
 $T \rightarrow \cdot F, +/\$$

$TF, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$

$F, +/a/b/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, +/a/b/\$$   
 $F \rightarrow \cdot a, +/a/b/\$$   
 $F \rightarrow \cdot b, +/a/b/\$$

# LR(1) Item for $I_0$

$I_0:$   
 $E' \rightarrow \cdot E, \$$

$E, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, \$$   
 $E \rightarrow \cdot T, \$$

$E + T, \$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$

$T, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/\$$   
 $T \rightarrow \cdot F, +/\$$

$TF, +/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$

$F, +/a/b/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, +/a/b/\$$   
 $F \rightarrow \cdot a, +/a/b/\$$   
 $F \rightarrow \cdot b, +/a/b/\$$

$F^*, +/a/b/\$$   
 $\rightarrow$

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, +/a/b/\$/*$   
 $F \rightarrow \cdot a, +/a/b/\$/*$   
 $F \rightarrow \cdot b, +/a/b/\$/*$

就是耐心地严格地按照算法3.5的closure进行迭代

# LR(1) 项目集规范族

$I_0:$   
 $E' \rightarrow \cdot E, \$$   
 $E \rightarrow \cdot E + T, +/\$$   
 $E \rightarrow \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_1:$   
 $E' \rightarrow E \cdot, \$$   
 $E \rightarrow E \cdot + T, +/\$$

$I_2:$   
 $E \rightarrow T \cdot, +/\$$   
 $T \rightarrow T \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_3:$   
 $T \rightarrow F \cdot, +/a/b/\$$   
 $F \rightarrow F \cdot^*, */+/a/b/\$$

$I_4:$   
 $T \rightarrow a \cdot, */+/a/b/\$$

$I_5:$   
 $T \rightarrow b \cdot, */+/a/b/\$$

$I_6:$   
 $E \rightarrow E + \cdot T, +/\$$   
 $T \rightarrow \cdot TF, +/a/b/\$$   
 $T \rightarrow \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

$I_7:$   
 $T \rightarrow TF \cdot, +/a/b/\$$   
 $F \rightarrow F \cdot^*, */+/a/b/\$$

$I_8:$   
 $F \rightarrow F^* \cdot, */+/a/b/\$$

$I_9:$   
 $E \rightarrow E + T \cdot, +/\$$   
 $T \rightarrow T \cdot F, +/a/b/\$$   
 $F \rightarrow \cdot F^*, */+/a/b/\$$   
 $F \rightarrow \cdot a, */+/a/b/\$$   
 $F \rightarrow \cdot b, */+/a/b/\$$

最后,不需要合并,就是这0-9的十个项目集

细心!注意新产生的项目集是否已经存在  
复习的时候最好自己拿有同心项目集的题练练手

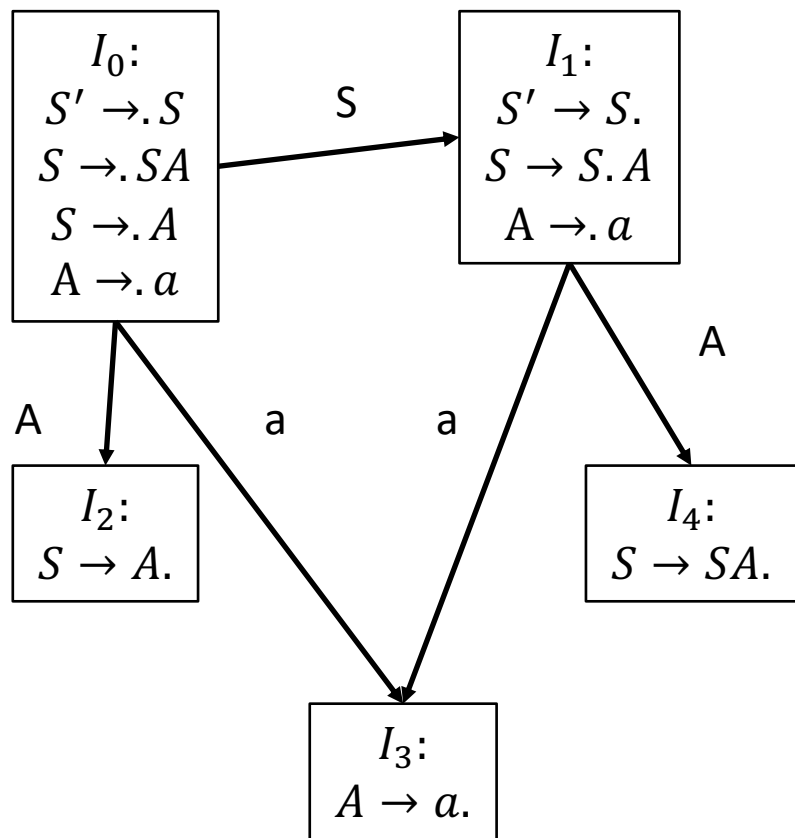
# LALR Table

state	action					goto		
	+	*	a	b	\$	e	t	f
0			s4	s5		1	2	3
1	s6				acc			
2	r2		s4	s5	r2			7
3	r4	s8	r4	r4	r4			
4	r6	r6	r6	r6	r6			
5	r7	r7	r7	r7	r7			
6			s4	s5			9	3
7	r3	s8	r3	r3	r3			
8	r5	r5	r5	r5	r5			
9	r1		s4	s5	r1			7

# 证明下面文法是SLR(1)文法，不是LL(1)文法

- $S \rightarrow SA|A$
- $A \rightarrow a$

- 证明不是LL(1)文法
  - 存在左递归，不是LL(1)文法
- 证明是SLR(1)文法
  - 产生式编号
  - 1.  $S \rightarrow SA$
  - 2.  $S \rightarrow A$
  - 3.  $A \rightarrow a$



$Follow(S) = \{a, \$\}$   
 $Follow(A) = \{a, \$\}$

status	action		goto	
	a	\$	s	a
0	s3		1	2
1	s3	acc		4
2	r2	r2		
3	r3	r3		
4	r1	r1		

证明是SLR(1),LALR(1),LR(1)文法,都需要完整的分析表  
 证明不是,可以只给出部分能够体现冲突的分析表,但是考试的时候建议都画完.

细心!注意新产生的项目集是否已经存在

H6-1,H6-2

考虑文法：

$$S \rightarrow I|R$$

$$I \rightarrow Fd|Id$$

$$R \rightarrow WpF$$

$$W \rightarrow Wd|\epsilon$$

$$F \rightarrow Fd|d$$

是否为LR(1)文法？

不是。

例如对于串ddpd和ddd,

对于第一个d,分析器无法确定应该按 $W \rightarrow \epsilon$ 规约,还是应该移进.

要想判断,需要知道后面是否有p

对于下面两个文法,哪个不是LR(1)文法?对该文法指出有移进-规约冲突的规范的LR(1)项目集.

$$S \rightarrow aAc$$
$$A \rightarrow Abb|b$$

是LR(1)文法

A看到第一个b时规约,  
如果后面是b,就移进两个终结符,  
如果后面是c就规约

$$S \rightarrow aAc$$
$$A \rightarrow bAb|b$$

不是LR(1)文法

看到b的时候不知道规约还是移进  
指出项目集,要把整个项目集规范族写出来,  
以免漏掉冲突

最后  $I_6$ :

$$A \rightarrow b.Ab, \quad b$$
$$A \rightarrow .bAb, \quad b$$
$$A \rightarrow b., \quad b$$
$$A \rightarrow .b, \quad b$$

有移进-规约冲突

为语言:

$$L = \{a^m b^n \mid 0 \leq m \leq 2n\}$$

(a的个数不超过b的个数的2倍)

写3个文法:LR(1)的, 二义的, 和非二义非LR(1)的

这类题是我认为整个编译原理里面最难搞的题.

我个人短时间内(比如考试的时候)想不出来,只能靠感觉猜测,靠尝试解答.

(去年考了这类题,我没有答出来)

LR(1):

最好是遇到a一直移进,等遇到了b,再规约下去.

也就是先匹配中间,再匹配两边

因为后面b可以很多,所以分为AB两部分,B由b组成

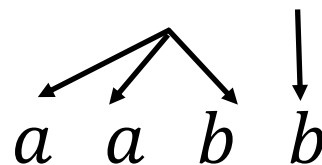
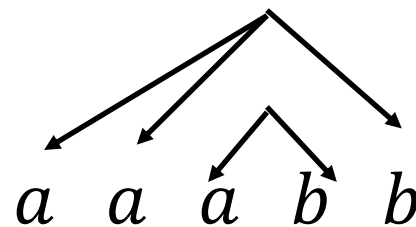
$$S \rightarrow AB$$

$$B \rightarrow Bb \mid \epsilon$$

对于A,按照右图,从中间开始规约,

然后按照m的奇偶,

$$A \rightarrow aaAb \mid ab \mid \epsilon$$



为语言:

$$L = \{a^m b^n \mid 0 \leq m \leq 2n\}$$

(a的个数不超过b的个数的2倍)

写3个文法:LR(1)的, 二义的, 和非二义非LR(1)的

二义的:

二义的最好写.

$$S \rightarrow aaSb \mid Sb \mid ab \mid \epsilon$$

*| b多 | m奇 |*

为语言:

$$L = \{a^m b^n \mid 0 \leq m \leq 2n\}$$

(a的个数不超过b的个数的2倍)

写3个文法:LR(1)的, 二义的, 和非二义非LR(1)的

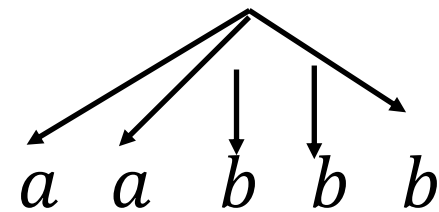
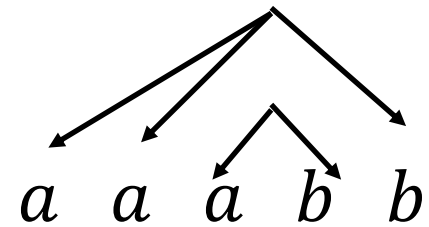
非二义且非LR(1):

这里为了非LR(1), 如右图, 让两边先匹配, 再匹配中间, 这样就能让中间b的数目影响LR(1)

$$S \rightarrow aaSb \mid A$$

$$B \rightarrow Bb \mid \epsilon$$

$$A \rightarrow aAb \mid B$$



为二义文法:

$$S \rightarrow A S \mid b$$

$$A \rightarrow S A \mid a$$

构造LR分析表时,分析动作冲突的条目?

冲突时如果不确定地选择动作,有哪些可能的动作序列?

没有完整的分析表,怎么确定有哪些冲突条目?

最后得到:

状态	action-a	action-b	action-\$	goto-A	goto-S
0	S4	S3		2	1
1	S4	S7	acc	5	6
2	S4	S3		2	8
3	r2	r2	r2		
4	r4	r4			
5	S4/r3	S7/r3		10	9
6	S4	S7		5	6
7	r2	r2			
8	S4/r1	S7/r1	r1	5	6
9	S4/r1	S7/r1		5	6
10	S4	S7		10	9

为二义文法:

$$S \rightarrow A S \mid b$$

$$A \rightarrow S A \mid a$$

构造LR分析表时,分析动作冲突的条目?

冲突时如果不确定地选择动作,有哪些可能的动作序列?

所以action[5,a], action[5,b], action[8,a],  
action[8,b], action[9,a], action[9,b]有冲突

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进? 规约?

1.  $S \rightarrow AS$
2.  $S \rightarrow b$
3.  $A \rightarrow SA$
4.  $A \rightarrow a$

### (1)移进

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进

1.  $S \rightarrow A S$
2.  $S \rightarrow b$
3.  $A \rightarrow S A$
4.  $A \rightarrow a$

### (1)移进

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进
0A2S8a4	b\$	r4:A->a规约
0A2S8A5	b\$	移进? 规约

1.  $S \rightarrow A S$
2.  $S \rightarrow b$
3.  $A \rightarrow S A$
4.  $A \rightarrow a$

(1)移进;(2)移进

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进
0A2S8a4	b\$	r4:A->a规约
0A2S8A5	b\$	移进

1.  $S \rightarrow A S$
2.  $S \rightarrow b$
3.  $A \rightarrow S A$
4.  $A \rightarrow a$

(1)移进;(2)移进

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进
0A2S8a4	b\$	r4:A->a规约
0A2S8A5	b\$	移进
0A2S8A5b7	\$	ERROR

1.  $S \rightarrow A S$
2.  $S \rightarrow b$
3.  $A \rightarrow S A$
4.  $A \rightarrow a$

(1)移进;(2)规约

栈	输入	动作
0	abab\$	移进
0a4	bab\$	r4:A->a规约
0A2	bab\$	移进
0A2b3	ab\$	r2:S->b规约
0A2S8	ab\$	移进
0A2S8a4	b\$	r4:A->a规约
0A2S8A5	b\$	r3:A->SA规约
0A2A2	b\$	移进
0A2A2b3	\$	r2:S->b规约
0A2A2S8	\$	r1:S->AS规约
0A2S8	\$	r1:S->AS规约
0S1	\$	acc

1.  $S \rightarrow AS$
2.  $S \rightarrow b$
3.  $A \rightarrow SA$
4.  $A \rightarrow a$

(1)规约;(2)移进  
和

(1)规约;(2)规约  
情况类似,不赘述

为文法:

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L, S \mid S$$

写语法制导定义,输出括号对数,嵌套括号最大深度

为了"输出",需要一个起始非终结符,  
是语法制导定义,不需要大括号

产生式	语义规则(括号对)	语义规则 (最大深度)
$S' \rightarrow S$	<code>print(S.pairs)</code>	<code>print(S.maxDepth)</code>
$S \rightarrow (L)$	<code>S.pairs=L.pairs+1</code>	<code>S.maxDepth=L.maxDepth+1</code>
$S \rightarrow a$	<code>S.pairs=0</code>	<code>S.maxDepth=0</code>
$L \rightarrow L1, S$	<code>L.pairs=L1.pairs+S.pairs</code>	<code>L.maxDepth=max(L1.maxDepth, S.maxDepth)</code>
$L \rightarrow S$	<code>L.pairs=S.pairs</code>	<code>L.maxDepth=S.maxDepth</code>

Thanks

祝大家期中复习愉快