

## 题目 1

考虑文法

$$S \rightarrow AS|b$$

$$A \rightarrow SA|a$$

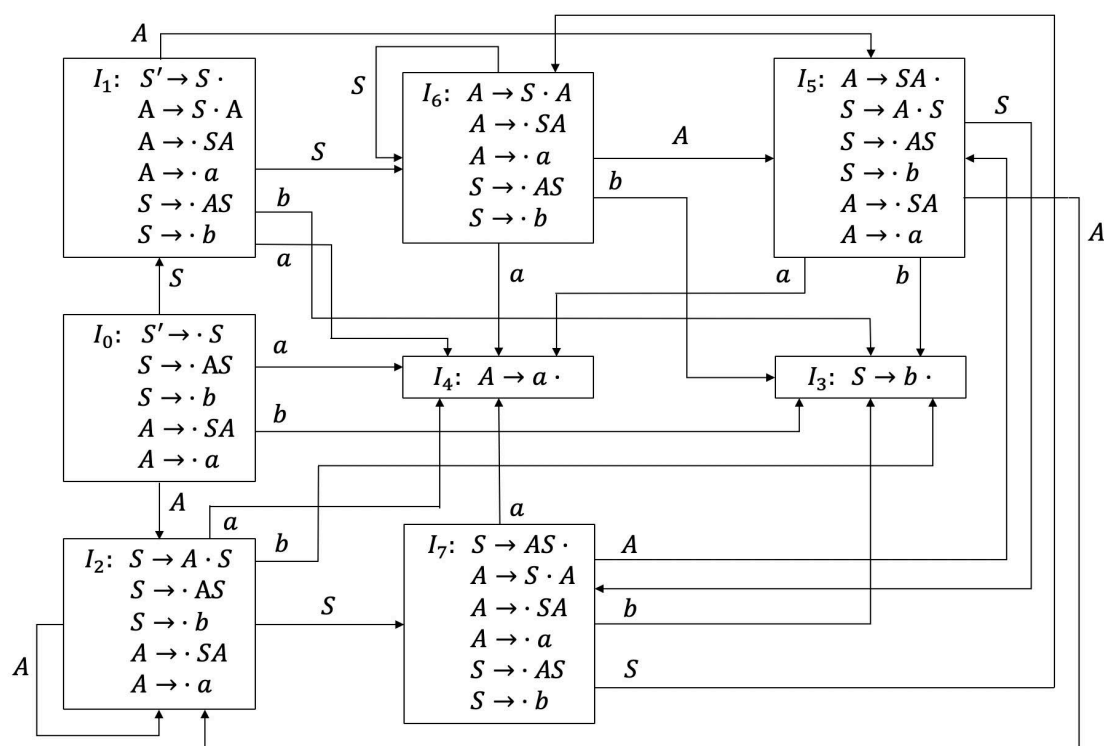
- (1) 列出这个文法的所有 LR(0) 项目。
- (2) 构造这个文法的 LR(0) 项目集规范族及识别活前缀的 DFA。
- (3) 这个文法是 SLR 的吗？若是，构造出它的 SLR 分析表。
- (4) 这个文法是 LALR 或 LR(1) 的吗？

解答：

- (1) 该文法的所有 LR(0) 项目如下所示：

编号	项目	编号	项目	编号	项目
1	$S' \rightarrow \cdot S$	2	$S' \rightarrow S \cdot$	3	$S \rightarrow \cdot AS$
4	$S \rightarrow A \cdot S$	5	$S \rightarrow AS \cdot$	6	$S \rightarrow \cdot b$
7	$S \rightarrow b \cdot$	8	$A \rightarrow \cdot SA$	9	$A \rightarrow S \cdot A$
10	$A \rightarrow SA \cdot$	11	$A \rightarrow \cdot a$	12	$A \rightarrow a \cdot$

- (2) 根据上述的 LR(0) 项目，我们可以直接构造出如下包含 LR(0) 项目集规范族的 DFA。



(3) 观察上述 LR(0) 项目集规范族，我们可以发现状态  $I_1, I_5, I_7$  存在移进归约冲突，因此我们首先计算各非终结符的 FOLLOW 集如下。

- $\text{FOLLOW}(S') = \{\#, a, b\}$
- $\text{FOLLOW}(S) = \{\#, a, b\}$
- $\text{FOLLOW}(A) = \{a, b\}$

接下来我们再判断上述三个状态的移进归约冲突是否可以消解。

- $I_1$  :  $\text{FOLLOW}(S')$  不包含  $a$ 、 $b$ ，冲突可消解。
- $I_5$  :  $\text{FOLLOW}(A)$  中包含  $a$ 、 $b$ ，冲突不可消解。
- $I_7$  :  $\text{FOLLOW}(S)$  中包含  $a$ 、 $b$ ，冲突不可消解。

因此我们可以判断该文法不是 SLR 文法。

(4) 首先我们列出 LR(1) 项目集规范族如下：

$I_0: S' \rightarrow \cdot S, \#$ $S \rightarrow \cdot AS, \#/a/b$ $S \rightarrow \cdot b, \#/a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$	$I_1: S' \rightarrow S \cdot, \#$ $A \rightarrow S \cdot A, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$	$I_2: S \rightarrow A \cdot S, \#/a/b$ $S \rightarrow \cdot AS, \#/a/b$ $S \rightarrow \cdot b, \#/a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$	$I_3: A \rightarrow a \cdot, a/b$  $I_4: S \rightarrow b \cdot, \#/a/b$
$I_5: A \rightarrow SA \cdot, a/b$ $S \rightarrow A \cdot S, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$	$I_6: A \rightarrow S \cdot A, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$	$I_7: S \rightarrow b \cdot, a/b$	$I_8: S \rightarrow AS \cdot, \#/a/b$ $A \rightarrow S \cdot A, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$
$I_9: S \rightarrow AS \cdot, a/b$ $A \rightarrow S \cdot A, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$	$I_{10}: S \rightarrow A \cdot S, a/b$ $S \rightarrow \cdot AS, a/b$ $S \rightarrow \cdot b, a/b$ $A \rightarrow \cdot SA, a/b$ $A \rightarrow \cdot a, a/b$		

我们可以发现在  $I_9$  状态中，同时包含项目  $S \rightarrow AS \cdot, a/b$  与  $A \rightarrow \cdot a, a/b$ ，因此遇到搜索符号  $a$  时，将难以判断该移进还是归约，存在“移进-归约”冲突，即该文法不是 LR(1) 文法，更不是 LALR 文法。

## 题目 2

证明下面的文法是 LALR(1) 的但不是 SLR(1) 的。

$$S \rightarrow Aa|bAc|dc|bda$$

$$A \rightarrow d$$

解答：我们直接给出 LR(1) 项目集规范族，如下所示。

$I_0: S' \rightarrow \cdot S, \#$ $S \rightarrow \cdot Aa, \#$ $S \rightarrow \cdot bAc, \#$ $S \rightarrow \cdot dc, \#$ $S \rightarrow \cdot bda, \#$ $A \rightarrow \cdot d, a$	$I_1: S' \rightarrow S \cdot, \#$	$I_3: S \rightarrow b \cdot Ac, \#$ $S \rightarrow b \cdot da, \#$ $A \rightarrow \cdot d, c$	$I_5: S \rightarrow Aa \cdot, \#$	$I_8: S \rightarrow dc \cdot, \#$
	$I_2: S \rightarrow A \cdot a, \#$	$I_4: S \rightarrow d \cdot c, \#$ $A \rightarrow d \cdot, a$	$I_6: S \rightarrow bA \cdot c, \#$	$I_9: S \rightarrow bAc \cdot, \#$
		$I_7: S \rightarrow bd \cdot a, \#$ $A \rightarrow d \cdot, c$	$I_{10}: S \rightarrow bda \cdot, \#$	

不难发现上述各项目集中并未出现无法消解的冲突，因此该文法是 LR(1) 的。进一步，我们可以发现上述项目集中不存在同心集，因此不需要合并，即该文法是 LALR(1) 的。

继续观察，我们可以发现状态  $I_4: \{[S \rightarrow d \cdot c, \#], [A \rightarrow d \cdot, a]\}$ ，对应于 LR(0) 项目中的状态  $\{[S \rightarrow d \cdot c], [A \rightarrow d \cdot]\}$ ，即出现了“移进-归约”冲突。

进一步，我们可以求出  $FOLLOW(A) = \{a, c\}$ ，包含  $c$ ，因此冲突不可消解，该文法不是 SLR(1) 的。

## 题目 3

证明下面的文法是 LR(1) 的但不是 LALR(1) 的。

$$S \rightarrow Aa|bAc|Bc|bBa$$

$$A \rightarrow d$$

$$B \rightarrow d$$

解答：我们直接给出 LR(1) 项目集规范族，如下所示。

$I_0: S' \rightarrow \cdot S, \#$ $S \rightarrow \cdot Aa, \#$ $S \rightarrow \cdot bAc, \#$ $S \rightarrow \cdot Bc, \#$ $S \rightarrow \cdot bBa, \#$ $A \rightarrow \cdot d, a$ $B \rightarrow \cdot d, c$	$I_1: S' \rightarrow S \cdot, \#$	$I_3: S \rightarrow b \cdot Ac, \#$ $S \rightarrow b \cdot Ba, \#$ $A \rightarrow \cdot d, c$ $B \rightarrow \cdot d, a$	$I_6: S \rightarrow Aa \cdot, \#$
	$I_2: S \rightarrow A \cdot a, \#$	$I_4: S \rightarrow B \cdot c, \#$	$I_7: S \rightarrow bA \cdot c, \#$
	$I_5: A \rightarrow d \cdot, a$ $B \rightarrow d \cdot, c$		$I_8: S \rightarrow bB \cdot a, \#$
$I_9: A \rightarrow d \cdot, c$ $B \rightarrow d \cdot, a$	$I_{10}: S \rightarrow Bc \cdot, \#$	$I_{11}: S \rightarrow bAc \cdot, \#$	$I_{12}: S \rightarrow bBa \cdot, \#$

不难发现上述各项目集中并未出现无法消解的冲突，因此该文法是 LR(1) 的。然后我们再验证是否是 LALR(1)，因此将同心集  $I_5$  与  $I_9$  进行合并，得到  $I_{59} : \{[A \rightarrow d \cdot, a/c], [B \rightarrow d \cdot, a/c]\}$ 。很明显  $I_{59}$  是一个含有“归约-归约”冲突的集合，因此上述文法不是 LALR(1) 的。

#### 题目 4

给出下面表达式的逆波兰表示（后缀式）

- (1)  $a * (-b + c)$
- (2)  $\text{not } A \text{ or not } (C \text{ or not } D)$
- (3)  $a + b * (c + d / e)$
- (4)  $(A \text{ and } B) \text{ or } (\text{not } C \text{ or } D)$
- (5)  $-a + b * (-c + d)$
- (6)  $(A \text{ or } B) \text{ and } (C \text{ or not } D \text{ and } E)$

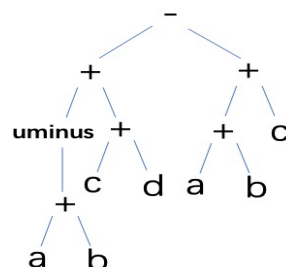
解答：逻辑运算符优先级：not > and > or

- (1)  $a \ b \ - \ c \ + \ *$
- (2)  $A \ \text{not} \ C \ D \ \text{not} \ \text{or} \ \text{not} \ \text{or}$
- (3)  $a \ b \ c \ d \ e \ / \ + \ * \ +$
- (4)  $A \ B \ \text{and} \ C \ \text{not} \ D \ \text{or} \ \text{or}$
- (5)  $a \ - \ b \ c \ - \ d \ + \ * \ +$
- (6)  $A \ B \ \text{or} \ C \ D \ \text{not} \ E \ \text{and} \ \text{or} \ \text{and}$

#### 题目 5

请将表达式  $-(a + b) * (c + d) - (a + b + c)$  分别表示成三元式、间接三元式和四元式序列。

解答：上述表达式对应的语法树如下所示：



$$T_0 := a + b$$

$$T_1 := -T_0$$

$$T_2 := c + d$$

$$T_3 := T_1 * T_2$$

$$T_4 := a + b$$

$$T_5 := T_4 + c$$

$$T_6 := T_3 - T_5$$

· 四元式：

op	$arg_1$	$arg_2$	result
(0) +	a	b	$T_0$
(1) uminus	$T_0$		$T_1$
(2) +	c	d	$T_2$
(3) *	$T_1$	$T_2$	$T_3$
(4) +	a	b	$T_4$
(5) +	$T_4$	c	$T_5$
(6) -	$T_3$	$T_5$	$T_6$

· 三元式：

op	$arg_1$	$arg_2$
(0) +	a	b
(1) uminus	(0)	
(2) +	c	d
(3) *	(1)	(2)
(4) +	a	b
(5) +	(4)	c
(6) -	(3)	(5)

· 间接三元式：

间接代码
(0)
(1)
(2)
(3)
(0)
(4)
(5)

op	$arg_1$	$arg_2$
(0) +	a	b
(1) uminus	(0)	
(2) +	c	d
(3) *	(1)	(2)
(4) +	(0)	c
(5) -	(3)	(4)