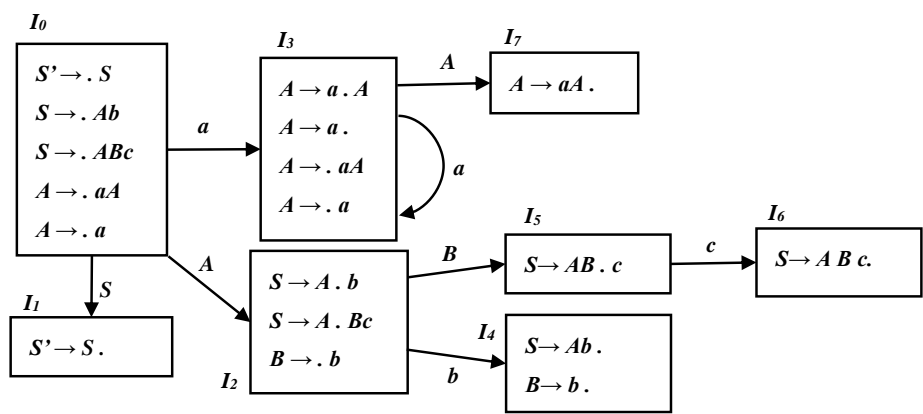


(若发现问题，请及时告知)

A1. 给定文法 $G[S]$:

- (1) $S \rightarrow Ab$
- (2) $S \rightarrow ABc$
- (3) $A \rightarrow aA$
- (4) $A \rightarrow a$
- (5) $B \rightarrow b$

(1) 下图是相应于 $G[S]$ 的增广文法 $G[S']$ 的 LR(0) 自动机:



文法 $G[S]$ 不是LR(0) 文法。试指出 $G[S']$ 的 LR(0) 自动机中存在哪些冲突的状态？并指出这些状态的冲突类别，即是移进-归约冲突还是归约-归约冲突？

(2) 完善 $G[S]$ 的 SLR (1) 分析表，并说明 $G[S]$ 是 SLR (1) 文法:

状态	ACTION				GOTO		
	a	b	c	#	S	A	B
0	s3				1		
1				acc			
2		s4					5
3							
4							
5							
6							
7		r3					

(3) 基于上面的 SLR(1) 分析表进行 LR 分析，若分析过程中设有符号栈，则栈

中存放的整个符号串（初始栈顶符号# 除外）对应于文法 $G[S]$ 的 _____。

- A. 某个活前缀 B. 某个可归约串 C. 某个句柄 D. 某个LR(0)项目

(4) 基于上面的 SLR(1) 分析表进行 LR 分析，若处于某个正常状态（未出错状态）时在栈顶已形成句柄，且当前所面临的输入符号是 b ，那么这样的状态可能有 _____ 个。

- A. 1个 B. 2个 C. 3个 D. 4个

(5) 基于上面的 SLR(1) 分析表进行 LR 分析，若处于某个出错状态时所面临的输入符号是 b ，那么这样的状态可能有 _____ 个。

- A. 1个 B. 2个 C. 3个 D. 4个

参考解答：

(1) I3为移进-归约冲突 和 I4为归约-归约冲突。

(2)

状态	ACTION				GOTO		
	a	b	c	#	S	A	B
0	s3				1	2	
1				acc			
2		s4					5
3	s3	r4				7	
4			r5	r1			
5			s6				
6				r2			
7		r3					

是SLR(1)文法的原因：上表中无多重表项（每个表项最多一个动作），或者说出每个冲突状态的解决方法。

(3) A

(4) B

(5) C

A2. 给定下列文法 $G[S]$ ：

(1) $S \rightarrow BA$

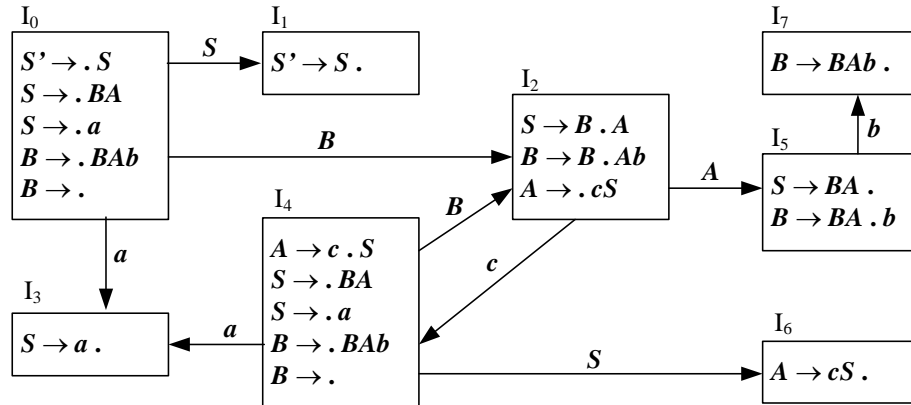
(2) $S \rightarrow a$

(3) $B \rightarrow BA b$

(4) $B \rightarrow \varepsilon$

(5) $A \rightarrow c S$

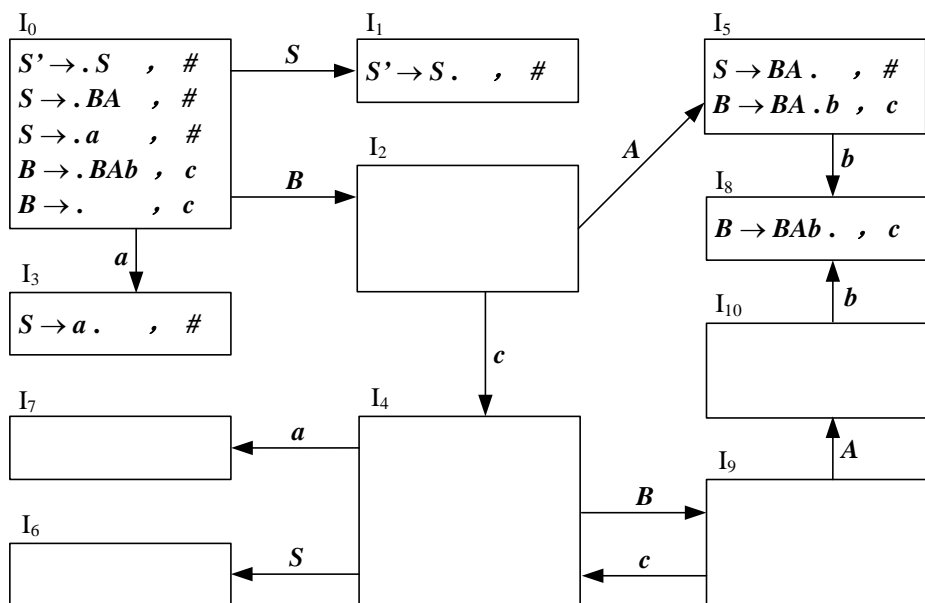
(1) 下图是相应于 $G[S]$ 的增广文法 $G[S']$ 的 LR(0) 自动机：



文法 $G[S]$ 不是LR(0) 文法。试指出 $G[S']$ 的 LR(0) 自动机中存在哪些冲突的状态？并指出这些状态的冲突类别，即是移进-归约冲突还是归约-归约冲突？

(2) 文法 $G[S]$ 也不是 SLR(1) 文法。试解释为什么？

(3) 下图是相应于 $G[S]$ 的增广文法 $G[S']$ 的 LR(1) 自动机，但部分状态所对应的项目集未给出，试补齐之（即分别给出状态 I_2, I_4, I_6, I_7, I_9 和 I_{10} 对应的项目集）。



(4) $G[S]$ 是否LR(1)文法? 若不是, 则指出 $G[S]$ 的 LR(1) 自动机中有冲突的状态。

(5) 下图表示 $G[S]$ 的 LR(1) 分析表和 SLR(1) 分析表中有冲突状态的行所对应的内容, 上半部分是 LR(1) 分析表, 下半部分是 SLR(1) 分析表, 但表中的状态号和表项中的内容没有给出, 试补齐之。(注意: 仅考虑LR(1) 和 SLR(1) 分析表, 不考虑 LR(0) 分析表; 仅需列出分析表中有冲突的状态所对应的行)

	冲突的状态	ACTION				GOTO		
		a	b	c	$\#$	S	A	B
LR(1) 分析表								
SLR(1) 分析表								

(6) 可以通过限定措施来解决 LR(1) 分析表和 SLR(1) 分析表中冲突的状态。请

给出你解决冲突的方案，并根据所给方案，分别修改题5中有冲突的表项。

(7) 根据题6中修改后新的 LR(1) 分析表或 SLR(1) 分析表, 分别进行 LR 分析。

对于一个有语法错误的输入符号串, 一般哪一种分析方法更快一些? 所谓“更快”是指发现错误时经过的步数较少, 而每一“步”是指进行一次“移进”或者完成一次“归约”的动作。请给出一个会发生语法错误的输入符号串 (长度不超过5), 用以支持你的结论。

参考解答:

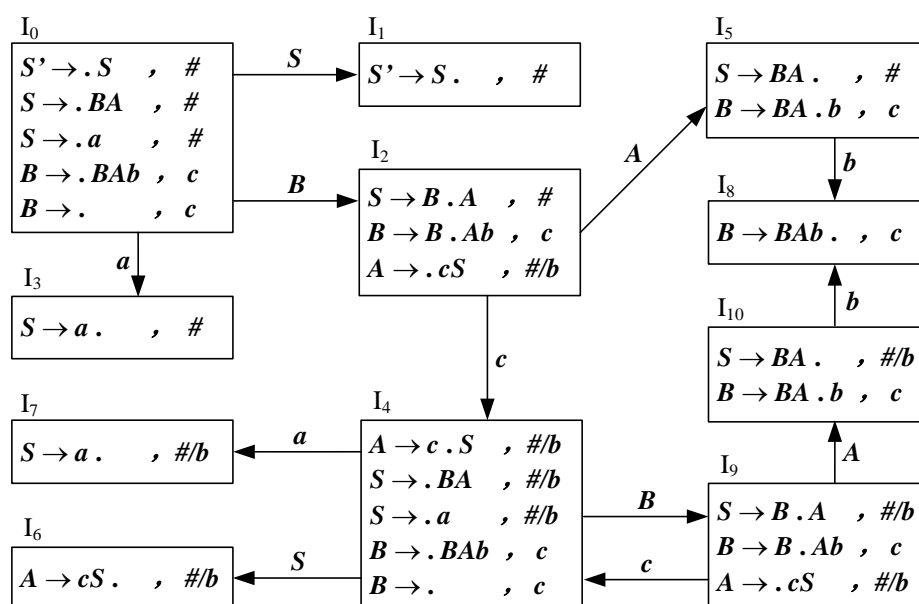
(1) $G[S]$ 的 LR(0) 自动机中存在3个冲突的状态: I_0 、 I_4 和 I_5 ; 均为移进-归约冲。

(2) $FOLLOW(S) = \{b, \#\}$ 。 (含 $FOLLOW(B) = \{c\}$ 也不多余)

因为 $FOLLOW(S)$ 中含 b , 所以 I_5 的移进-归约冲突无法用 SLR(1) 方法解决;

(含“ I_0 和 I_4 的移进-归约冲突可以用 SLR(1) 方法解决”, 也不多余)。

(3)



(4)

不是，冲突的状态仅一个： I_{10} 。

(5)

	冲突的状态	ACTION				GOTO		
		a	b	c	$\#$	S	A	B
LR(1) 分析表	10		s8, r1		r1			
SLR(1) 分析表	5		s7, r1		r1			

(6) 对于上述LR(1)分析和SLR(1)分析中的冲突，若限定“冲突发生时若输入为 b ，则优先进行移进”，则二者的冲突状态均可以化解。

根据这一解决方案，我们修改题5中的表项，使得LR(1)分析表中

$ACTION[10, b] = \{s8\}$ ，SLR(1)分析表中 $ACTION[5, b] = \{s7\}$ 。

(7) LR(1)分析发现错误更快}。比如，输入符号串 ab ，LR(1)分析在第1步“移进”后发现错误，而SLR(1)分析在第1步“移进”，第2步“归约”后发现错误。

.....

以下是 Lecture04 文档中的题目

1. 下面的文法 $G[S]$ 描述由命题变量 p 、 q ，联结词 \wedge （合取）、 \vee （析取）、 \neg （否定）构成的命题公式集合：

$$\begin{aligned} S &\rightarrow S \vee T \mid T \\ T &\rightarrow T \wedge F \mid F \\ F &\rightarrow \neg F \mid p \mid q \end{aligned}$$

试分别指出句型 $\neg F \vee \neg q \wedge p$ 和 $\neg F \vee p \wedge \neg F \vee T$ 的所有短语，直接短语。如果这些句型同时也是右句型，那么还要给出其句柄。请将结果填入下表中：

句型	短语	直接短语	句柄
$\neg F \vee \neg q \wedge p$			
$\neg F \vee p \wedge \neg F \vee T$			

参考解答：

句型	短语	直接短语	句柄
$\neg F \vee \neg q \wedge p$	$\neg F \vee \neg q \wedge p$ $\neg F$ $\neg q \wedge p$ $\neg q$ q p	$\neg F$ q p	$\neg F$
$\neg F \vee p \wedge \neg F \vee T$	$\neg F \vee p \wedge \neg F \vee T$ $\neg F \vee p \wedge \neg F$ $\neg F$ $p \wedge \neg F$ p $\neg F$	$\neg F$ p $\neg F$	无

2. (1) 给定文法 $G[S]$ ：

$$\begin{aligned} S &\rightarrow A B \\ A &\rightarrow a A \mid \varepsilon \\ B &\rightarrow B b \mid \varepsilon \end{aligned}$$

- (a) 构造该文法 $G[S]$ 的 LR(0) 有限状态机。
- (b) 说明该文法不是 LR(0) 文法。
- (c) 该文法是否 SLR(1) 文法？为什么？

(2) 给定文法 $G[S]$ ：

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

完成同（1）一样三个问题（a），（b），（c）。

参考解答：

（2）首先变换文法为增广文法。增加如下产生式

$$S' \rightarrow S$$

得到增广文法如下

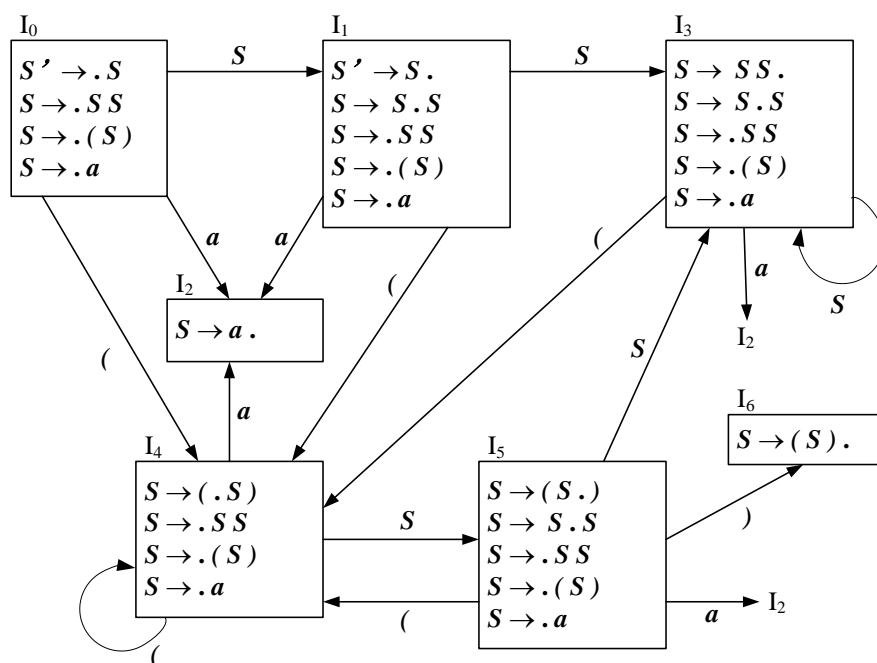
$$S' \rightarrow S$$

$$S \rightarrow S S$$

$$S \rightarrow (S)$$

$$S \rightarrow a$$

（a）该文法的LR（0）有限状态机状态转换图如下：



（b）状态I3中包含移进-归约冲突，所以G(S)不是LR(0)文法。

（c）对于状态I3，由于FOLLOW(S)={a, (,), #}，在面临第一个符号是a时，不能选择是归约成S还是移进a，所以，不是SLR(1)文法。

4. 给定 SLR(1) 文法 G[S]:

$$(1) S \rightarrow A b$$

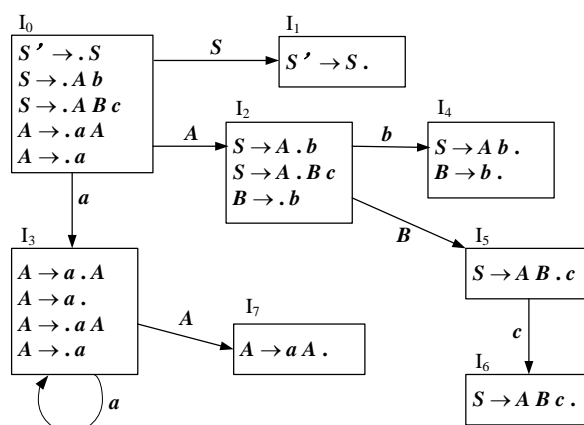
$$(2) S \rightarrow A B c$$

$$(3) A \rightarrow a A$$

$$(4) A \rightarrow a$$

$$(5) B \rightarrow b$$

其 LR(0) 有限状态机如下图所示：



(a) 构造该文法的 SLR(1) 分析表。

(b) 若采用 SLR(1) 方法对于 $L(G)$ 中的某一输入串进行分析，当栈顶出现句柄 a 时，余留输入符号串中的第一个符号是什么？

参考解答：

(a) SLR(1) 分析表：

状态	ACTION				GOTO		
	a	b	c	#	S	A	B
0	s ₃				1	2	
1				acc			
2		s ₄					5
3	s ₃	r ₄				7	
4			r ₅	r ₁			
5			s ₆				
6				r ₂			
7		r ₃					

(b) b

5. 给定 SLR(1) 文法 $G[S]$ ：

(1) $S \rightarrow a S a$

(2) $S \rightarrow b S b$

(3) $S \rightarrow c$

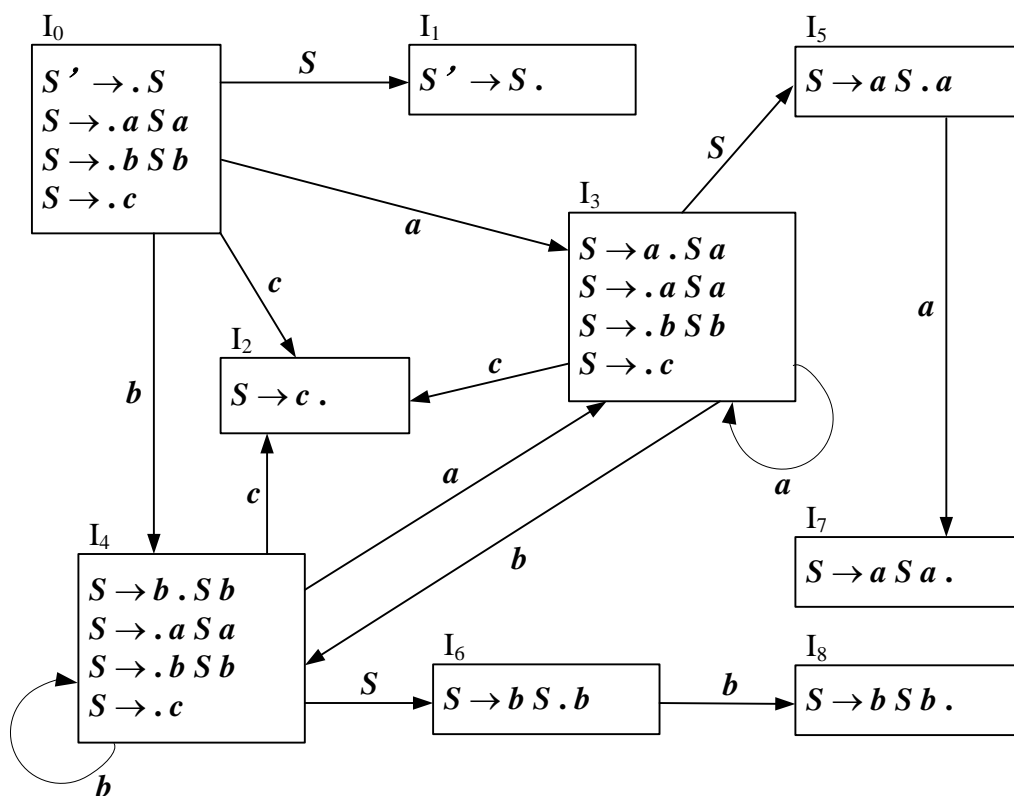
(a) 构造该文法的 LR(0) 有限状态机

(b) 构造该文法的 SLR(1) 分析表。

(c) 若根据以上 SLR(1)分析表对于 $L(G)$ 中的某一输入串执行 SLR(1) 分析过程，初始时符号栈存放符号 $\#$ 。当扫描过串 $abbcb$ 后，分析栈中的符号串是什么（以进栈先后次序给出）？当前可归约的句柄是什么？

参考解答：

(a)



(b)

状态	ACTION				GOTO
	a	b	c	#	S
0	s ₃	s ₄	s ₂		1
1				acc	
2	r ₃	r ₃		r ₃	
3	s ₃	s ₄	s ₂		5
4	s ₃	s ₄	s ₂		6
5	s ₇				
6		s ₈			
7	r ₁	r ₁		r ₁	
8	r ₂	r ₂		r ₂	

(c) 当输入扫描过串 *abbc* 后，分析栈中的符号串是什么 #abbSb（或 abbSb）。当前可规约的句柄是 bSb。

8. (1) 构造下列增广文法 $G[P']$ 的 LR(1) FSM，验证原文法是 LR(1) 文法：

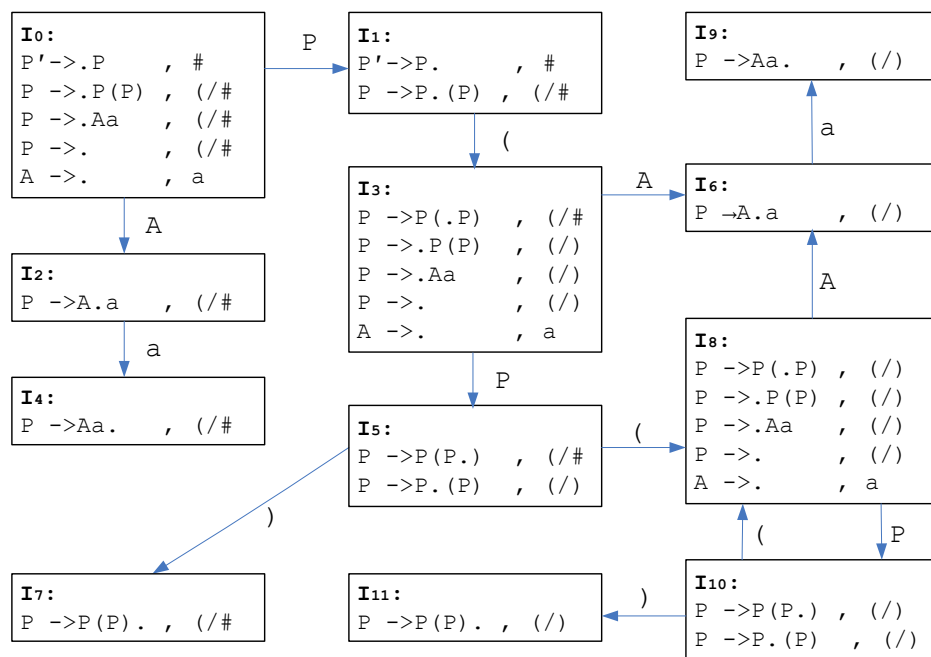
- (0) $P' \rightarrow P$
- (1) $P \rightarrow P(P)$
- (2) $P \rightarrow Aa$
- (3) $P \rightarrow \varepsilon$
- (4) $A \rightarrow \varepsilon$

其中 P' , P , A 为非终结符

(2) 通过合并同芯集（状态）的方法构造相应于上述 LR(1) FSM 的 LALR(1) FSM，并判断原文法是否 LALR(1) 文法？

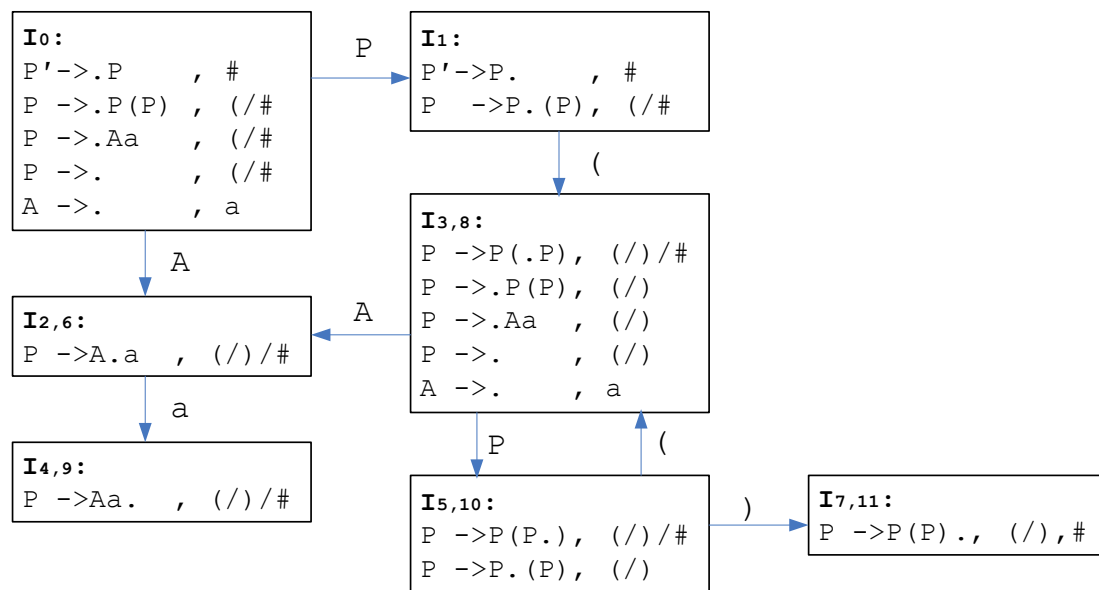
参考解答：

构造增广文法 $G(P')$ 的 LR(1) 项目集族和转换函数如下：



其中的每个状态均无冲突，所以原文法是 LR(1) 文法。

可以看出：I2 和 I6，I3 和 I8，I4 和 I9，I5 和 I10，I7 和 I11 是同芯状态。通过合并同芯状态的方法构造相应于上述 LR(1) 转换图的 LALR(1) 转换图 如下：



可以看出：所有状态都不存在移进-归约和归约-归约冲突。所以，原文法是 LALR(1) 文法。

12. 已知某文法 $G[S]$ 的 LALR(1) 分析表如下：

状态	ACTION	GOTO
----	--------	------

	<i>a</i>	<i>t</i>	<i>g</i>	<i>c</i>	#	<i>S</i>
0	s11	s8		s4		1
1				s2	acc	
2			s3			
3	s11	s8		s4		16
4	s5					
5	s6					
6				s7		
7			r1	r1	R1	
8			s9			
9				s10		
10	s11	s8		s4		14
11	s11	s8		s4		12
12			s13	s2		
13	s11	s8		s4		15
14			r4	s2	R4	
15			r2	s2	R2	
16			r3	s2	R3	

并且已知各规则右边语法符号的个数以及左边的非终结符如下：

规则编号	1	2	3	4
右部长度	4	4	4	4
左部符号	S	S	S	S

(a) 写出使用上述 LALR(1)分析器分析下面串的过程（只需写出前 10 步，列出所有可能的 ri ， sj 序列，注意先后次序）：

acaaccgtgccaacgatgccaa ...

(b) 试指出该串相对于上述文法的句柄。

参考解答：

(a) s11, s4, s5, s6, s7, r1, s2, s3, s8, s9, ...

(b) 原始文法（不在题目中公开）是

$S \rightarrow caac \mid aSgS \mid ScgS \mid tgcS$

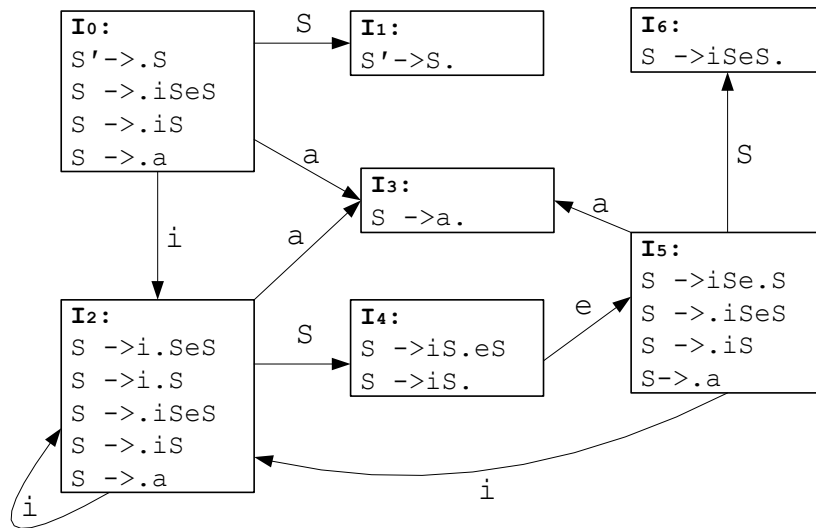
该串相对于上述文法的句柄是：caac

13. 给定如下文法 $G[S]$:

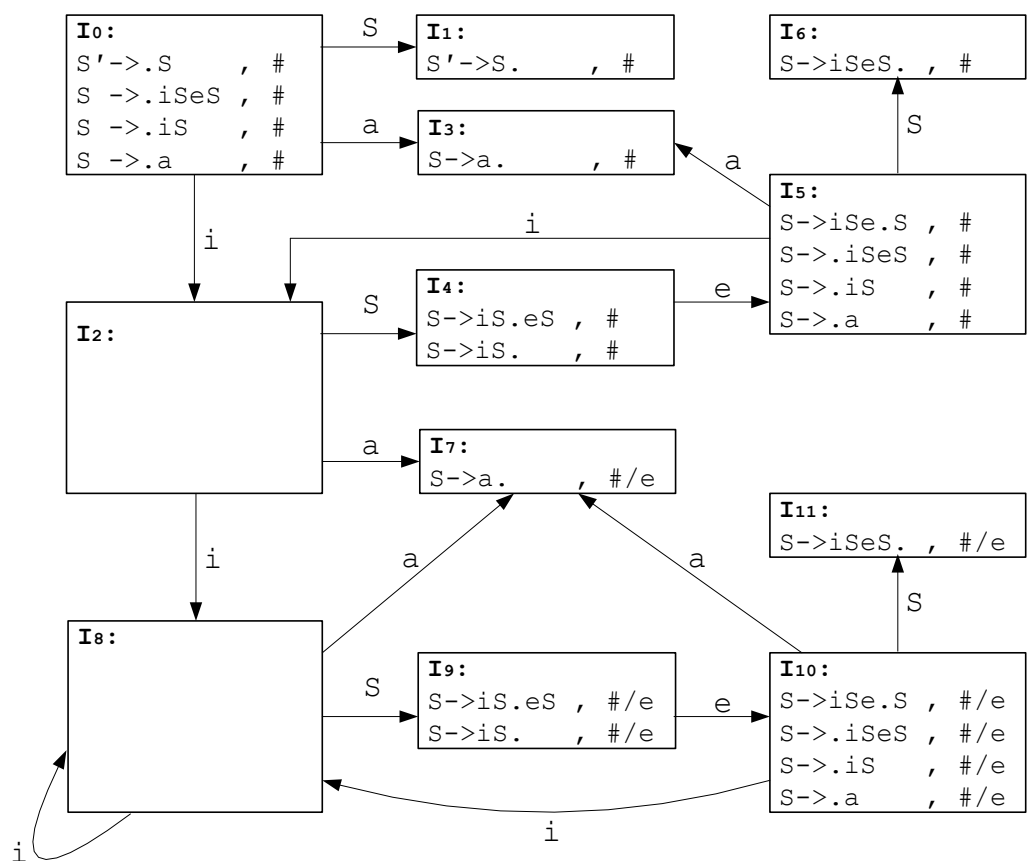
(1) $S \rightarrow \underline{\text{if}} \ S \ \underline{\text{else}} \ S$

- (2) $S \rightarrow \underline{\text{if}}\ S$
 (3) $S \rightarrow a$

为文法 $G[S]$ 增加产生式 $S' \rightarrow S$, 得到增广文法 $G'[S']$, 下图是相应的LR(0)自动机 (i 表示 if, e 表示 else):



- (1) 指出LR(0)自动机中的全部冲突状态及其冲突类型, 以说明文法 $G[S]$ 不是LR(0)文法。
- (2) 文法 $G[S]$ 也不是SLR(1)文法。为什么?
- (3) 下图表示文法 $G[S]$ 的LR(1)自动机, 部分状态所对应的项目集未给出, 试补齐之 (即分别给出状态 I_2 , I_8 , 和 I_{10} 对应的的项目集。



- (4) 指出LR(1)自动机中的全部冲突状态，这说明文法 $G[S]$ 也不是 LR(1) 文法。
- (5) 若规定最近匹配原则，即 `else` 优先匹配左边靠近它的未匹配的if，则可以解决上述2个自动机中的状态冲突。下图表示文法 $G[S]$ 在规规定这一规则情况下的SLR(1)分析表，状态 4~6 对应的行未给出，试补齐之。

状态	ACTION				GOTO
	i	e	a	#	S
0	s2		s3		1
1				acc	
2	s2		s3		4
3		r3		r3	
4					
5					
6					

下图表示文法 $G[S]$ 在规规定这一规则情况下的LR(1)分析表，状态 4, 7 和 9 对

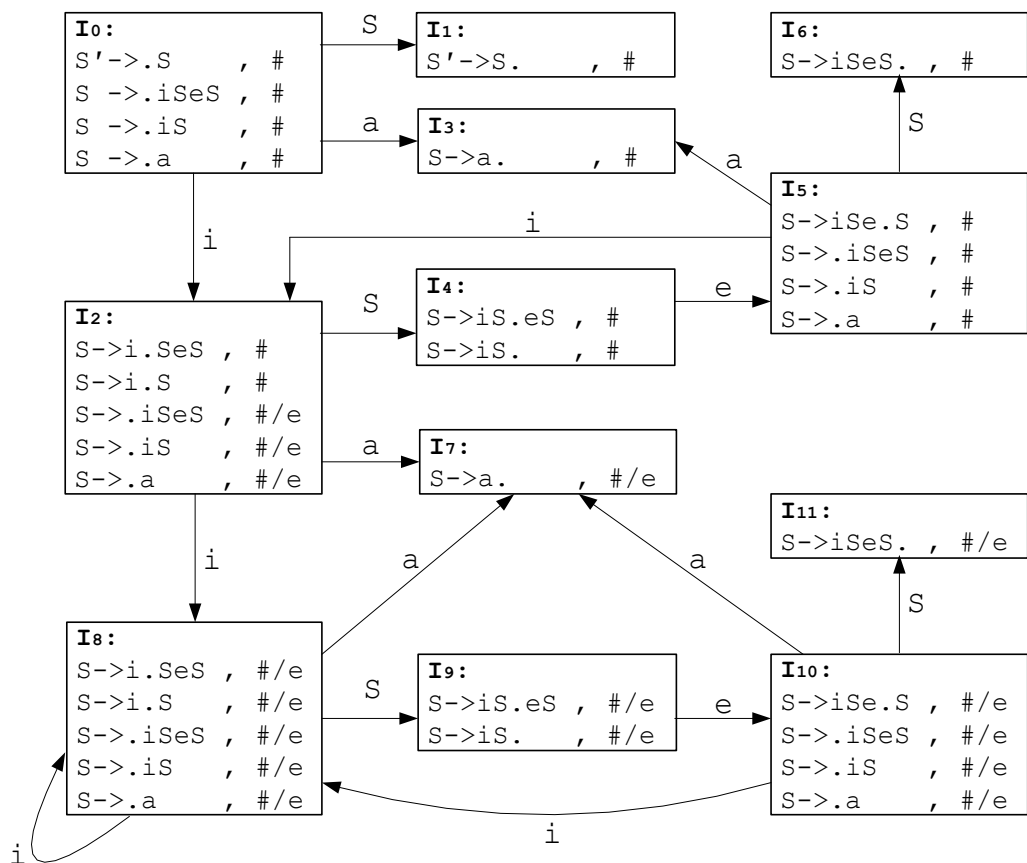
应的行未给出，试补齐之。

状态	ACTION				GOTO
	i	e	a	#	S
0	s2		s3		1
1				acc	
2	s8		s7		4
3				r3	
4					
5	s2		s3		6
6				r1	
7					
8	s8		s7		9
9					
10	s8		s7		11
11		r1		r1	

- (6) 对于文法G[S]中正确的句子，基于上述两个分析表均可以成功进行LR分析。然而，对于不属于文法G[S]中的句子，两种分析过程发现错误的速度不同，即发现错误时所经过的移进/归约总步数有差异。试给出一个长度不超过10的句子（即所包含的终结符个数不超过10），使得两种分析过程发现错误的速度不同。哪一个更快？对于你给的例子，两种分析过程分别到达哪个状态会发现错误？

参考解答：

- (1) 状态 I_4 有冲突，为移进-归约冲突。
- (2) 因 $\text{Follow}(S) = \{ e, \# \}$ ， $e \in \text{Follow}(S)$ ，所以状态 I_4 的移进-归约冲突不可解决，所以该文法不是 SLR(1) 文法。
- (3) 完整的 LR(1) 自动机如下：



(4) 状态 I9 有冲突，同样为移进-归约冲突。

(5) 完整的 SLR(1)分析表

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	

完整的 LR(1)分析表

状态	ACTION				GOTO
	i	e	a	#	S
0	s2		s3		1
1				acc	
2	s8		s7		4
3				r3	
4		s5		r2	
5	s2		s3		6
6				r1	
7		r3		r3	
8	s8		s7		9
9		s10		r2	
10	s8		s7		11
11		r1		r1	

（6）如对于句子 a else，用LR(1)分析1步后到达状态3发现错误，用SLR(1)分析2步后到达状态1发现错误，所以LR(1) 分析更快。

另解：如对于句子 if a else a else，用LR(1)分析5步后到达状态3发现错误，用SLR(1)分析8步后到达状态1发现错误，所以LR(1) 分析更快。