

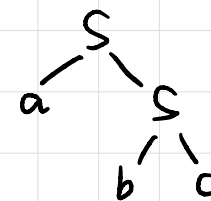
如何区别 LR(0), SLR(1), LR(1), LALR(1)

为什么要拓广文法?

① 归约项目 ( $A \rightarrow \alpha \cdot$ ) · 在最后

eg:  $k[s]: S \rightarrow aS | bc$  句子 abc

② 接受项目 ( $S \rightarrow \alpha \cdot$ ) 开始文法对应的 (永远中有 1 个)  
all



③ 移进项目 ( $A \rightarrow \alpha \cdot x \beta$ ) · 后面是终结符

自底向上

④ 待约项目 ( $A \rightarrow \alpha \cdot x \beta$ ) · 后面是非终结符

LR(0)文法: 不存在冲突项目 (移进-归约, 归约-归约)

构表步骤: 拓广文法 —— 项目集规范族 —— LR(0)分析表

1k: (是否存在冲突项目) { 不存在  $\uparrow$   
存在 (不是 LR(0)文法) 无法构表

LR(0)  $\subset$  SLR(1)

SLR(1)文法: 存在冲突项目 (移进-归约) 目的: 减少冲突的发生

构表步骤: 拓广文法 —— 项目集规范族 —— SLR(1)分析表

(存在冲突项目)

$\rightarrow$  根据 FOLLOW 集来判断 {  $= \emptyset$ , 是 SLR(1)文法  
 $\neq \emptyset$ , 无法构表

LR(1)文法: 当存在冲突项目 (移进-归约), 无 (归约-归约) 冲突

构表步骤: 拓广文法 —— (带向前搜索符) 项目集规范族 —— LR(1)分析表

(存在冲突项目)

$\rightarrow$  根据向前搜索符来判断 {  $= \emptyset$ , 是 LR(1)文法  
 $\neq \emptyset$ , 无法构表

LR(1)  $\subset$  LALR(1)

任何一个 SLR(1) 文法一定是一个 LALR(1) 文法。

LALR(1)文法: 合并同心集后无 (归约-归约) 冲突

(表达式相同, 搜索符不同)

构表步骤: 拓广文法 —— (带向前搜索符) 项目集规范族 —— LALR(1)分析表

LR(1) 可以认为是, 当 SLR(1) 解决不了时, 出现的一种方法

则为 SLR(1) 文法

$I_k$ :  
 $S \rightarrow \alpha \cdot aB$   
 $A \rightarrow \alpha \cdot$   
 $B \rightarrow \alpha \cdot$

$\{a\} \cap \text{Follow}(A) = \emptyset$  无(移进-归约)冲突  
 $\{a\} \cap \text{Follow}(B) = \emptyset$

$\text{Follow}(A) \cap \text{Follow}(B) \neq \emptyset$ , 有(归约-归约)冲突

无法用 SLR(1) 解决

LALR(1) 与 LR(1) 的不同之处是当输入串有误时, LR(1) 能够及时发现错误, 而 LALR(1) 则可能还继续执行一些多余的归约动作, 但决不会执行新的移进, 即 LALR(1) 能够像 LR(1) 一样准确地指出出错的地点。就文法的描述能力来说, 有下面的结论:

$\text{LR}(0) \subset \text{SLR}(1) \subset \text{LR}(1) \subset \text{无二义文法}$

例 3.23 判断下述文法 G[S] 是哪类 LR 文法。

G[S]: (1)  $S \rightarrow L=R$   
(2)  $S \rightarrow R$   
(3)  $L \rightarrow *R$   
(4)  $L \rightarrow i$   
(5)  $R \rightarrow L$

[解答] 首先将文法 G[S] 拓广为 G'[S']:

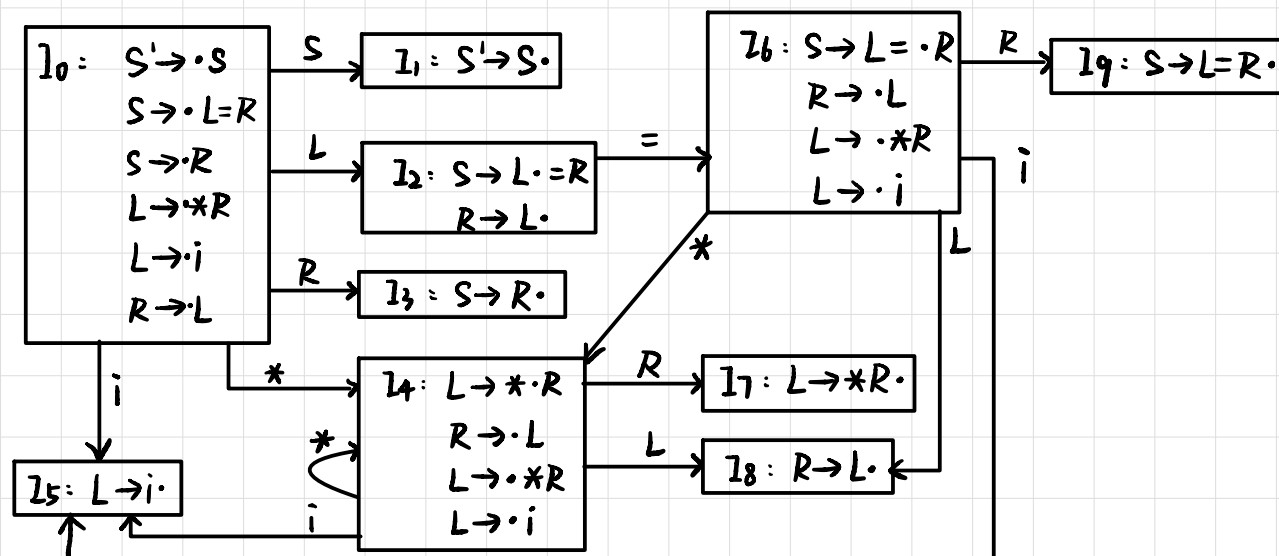
G'[S']: (0)  $S' \rightarrow S$   
(1)  $S \rightarrow L=R$   
(2)  $S \rightarrow R$   
(3)  $L \rightarrow *R$   
(4)  $L \rightarrow i$   
(5)  $R \rightarrow L$

① LR(0) X

② SLR(1) X

③ LR(1) ✓

④ LALR(1)



不是 LR(0) 文法



FOLLOW集

$FOLLOW(S') = \{ \# \}$

$FOLLOW(S) = \{ \# \}$

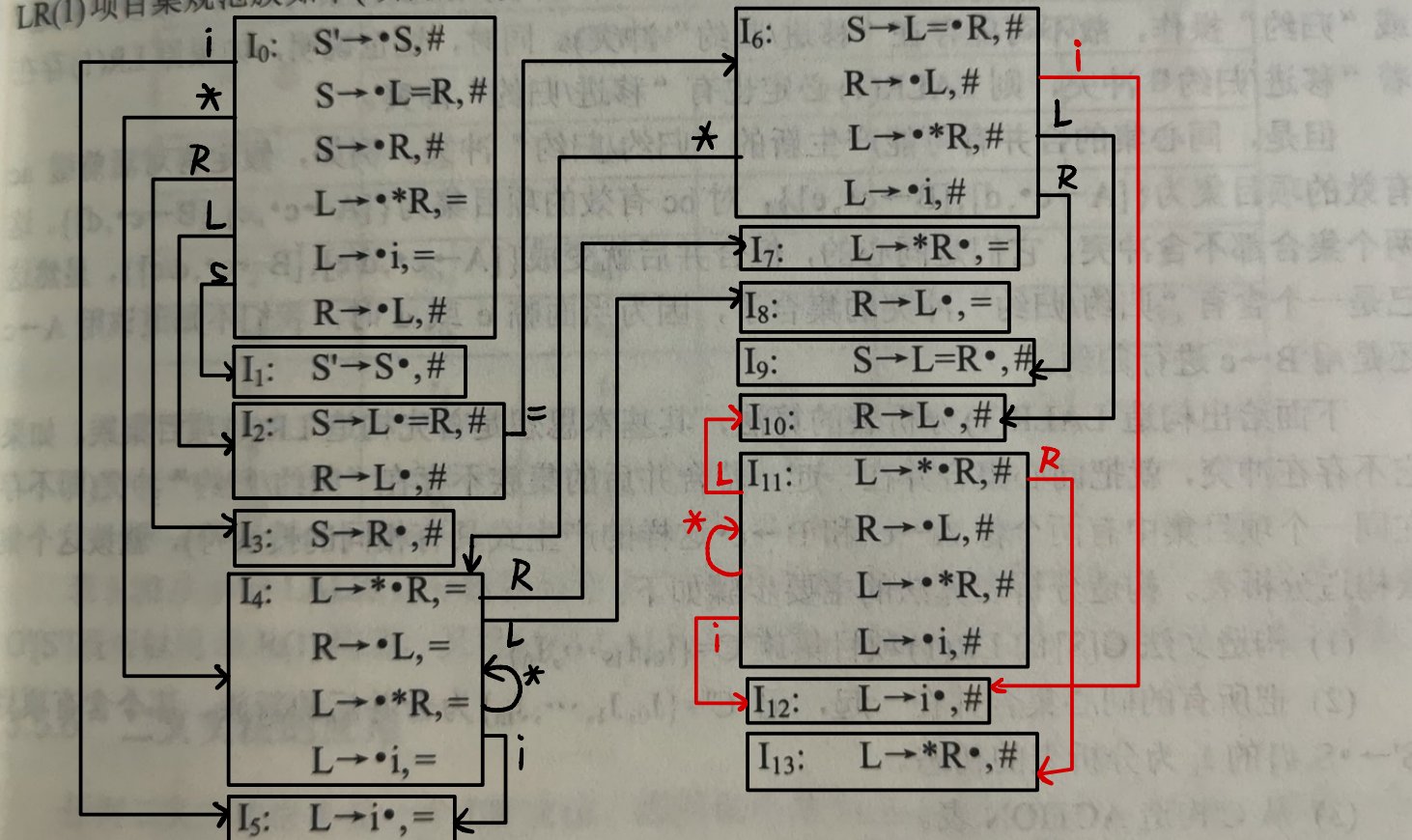
$FOLLOW(L) = \{ =, \# \}$

$FOLLOW(R) = \{ \#, = \}$

$G[S']$ : (0)  $S' \rightarrow S$   
 (1)  $S \rightarrow L=R$   
 (2)  $S \rightarrow R$   
 (3)  $L \rightarrow *R$   
 (4)  $L \rightarrow i$   
 (5)  $R \rightarrow L$

$\{=\} \cap FOLLOW(R) = \{=\} \cap \{\#,=\} \neq \emptyset$  不是SLR(1)文法

所以文法  $G[S']$  不是 LR(1) 文法。判断是否是 LR(1) 文法则首先要构造 LR(1) 项目集规范族。因此，构造文法  $G[S']$  的 LR(1) 项目集规范族如下(项目集  $I_0$  由  $S' \rightarrow \cdot S, \#$  开始):



此时,  $I_2$  的移进项目  $[S \rightarrow L \cdot =R, \#]$  和归约项目  $[R \rightarrow L \cdot, \#]$  有

$\{=\} \cap \{\#\} = \emptyset$

LR(1) 文法



3.23 请指出图 3-46 中的 LR 分析表 (a)、(b)、(c) 分属 LR(0)、SLR(1) 和 LR(1) 中的哪一种，并说明理由。

| 状态 | ACTION         |                | GOTO |   |
|----|----------------|----------------|------|---|
|    | b              | #              | S    | B |
| 0  | s <sub>3</sub> |                | 1    | 2 |
| 1  |                | acc            |      |   |
| 2  | s <sub>4</sub> |                |      | 5 |
| 3  | r <sub>2</sub> |                |      |   |
| 4  |                | r <sub>2</sub> |      |   |
| 5  |                | r <sub>1</sub> |      |   |

(a) LR(1)

| 状态 | ACTION         |                |                | GOTO |
|----|----------------|----------------|----------------|------|
|    | a              | b              | #              | T    |
| 0  | s <sub>2</sub> | s <sub>3</sub> |                | 1    |
| 1  |                |                | acc            |      |
| 2  | s <sub>2</sub> | s <sub>3</sub> |                | 4    |
| 3  | r <sub>2</sub> | r <sub>2</sub> | r <sub>2</sub> |      |
| 4  | r <sub>1</sub> | r <sub>1</sub> | r <sub>1</sub> |      |

LR(0)

(b)

| 状态 | ACTION         |                |                | GOTO |
|----|----------------|----------------|----------------|------|
|    | i              | k              | #              | P    |
| 0  | s <sub>1</sub> | s <sub>3</sub> |                | 2    |
| 1  | s <sub>1</sub> | s <sub>3</sub> |                | 4    |
| 2  |                |                | acc            |      |
| 3  |                |                | r <sub>2</sub> |      |
| 4  |                |                | r <sub>1</sub> |      |

SLR(1)

(c)

3.27 一个非 LR(1) 的文法如下：

$G[L]: L \rightarrow MLb \mid a$

$M \rightarrow \varepsilon$

请给出所有“移进/归约”冲突的 LR(1) 项目集，以说明该文法确实不是 LR(1) 的。

【解答】 先将文法  $G[L]$  拓广为  $G[L']$

$G[L']:$  (0)  $L' \rightarrow L$

(1)  $L \rightarrow MLb$

(2)  $L \rightarrow a$

(3)  $M \rightarrow \varepsilon$

如果按 LR(1) 方法构造分析表时出现“移进/归约”冲突，则项目集规范族中一定如下形式的项目：

$[A \rightarrow \alpha \cdot b\beta, a]$  和  $[A \rightarrow \alpha \cdot, b]$

即移进符号与向前搜索符号相同。

$I_0: L' \rightarrow \cdot L, \#$

$L \rightarrow \cdot MLb, \#$

$L \rightarrow \cdot a, \#$

$M \rightarrow \cdot, a$

$I_1: L' \rightarrow L \cdot, \#$

$I_2: L \rightarrow M \cdot Lb, \#$

$L \rightarrow \cdot MLb, b$

$L \rightarrow \cdot a, b$

$M \rightarrow \cdot, a$

$I_3: L \rightarrow ML \cdot b, \#$

$I_4: L \rightarrow M \cdot Lb, b$

$L \rightarrow \cdot MLb, b$

$L \rightarrow \cdot a, b$

$M \rightarrow \cdot, a$

$I_5: L \rightarrow MLb \cdot, \#$

由  $I_0$  中  $L \rightarrow \cdot a, \#$  和  $M \rightarrow \cdot, a$  可知  $\{a\} \cap \{a\} \neq \emptyset$

由  $I_2$  中  $L \rightarrow \cdot a, b$  和  $M \rightarrow \cdot, a$  可知  $\{a\} \cap \{a\} \neq \emptyset$

由  $I_4$  中  $L \rightarrow \cdot a, b$  和  $M \rightarrow \cdot, a$  可知  $\{a\} \cap \{a\} \neq \emptyset$

不是 LR(1) 文法



3.29 已知文法  $G[S]: S \rightarrow aAd \mid ;Bd \mid aB \uparrow \mid ;A \uparrow$   
 $A \rightarrow a$

$B \rightarrow a$

(1) 试判断  $G[S]$  是否为 LALR(1) 文法。

【解答】 (1) 将文法  $G[S]$  拓广为文法  $G[S']$ :

$G[S']$ : (0)  $S' \rightarrow S$

(1)  $S \rightarrow aAd$

(2)  $S \rightarrow ;Bd$

(3)  $S \rightarrow aB \uparrow$

(4)  $S \rightarrow ;A \uparrow$

(5)  $A \rightarrow a$

(6)  $B \rightarrow a$

判断  $G[S]$  是否为 LALR(1) 文法的方法是: 首先构造 LR(1) 项目集族, 如果它不存在冲突, 就把同心集合并在一起; 若合并后的集族不存在“归约/归约”冲突(即不存在同一个项目集中有两个像  $A \rightarrow c \cdot$  和  $B \rightarrow c \cdot$  这样具有相同搜索符的产生式), 则表明  $G[S]$  是 LALR(1) 文法。

$I_0: S' \rightarrow \cdot S, \#$

$S \rightarrow \cdot aAd, \#$

$S \rightarrow \cdot ;Bd, \#$

$S \rightarrow \cdot aB \uparrow, \#$

$S \rightarrow \cdot ;A \uparrow, \#$

$I_1: S' \rightarrow S \cdot, \#$

$I_2: S \rightarrow a \cdot Ad, \#$

$S \rightarrow a \cdot B \uparrow, \#$

$A \rightarrow \cdot a, d$

$B \rightarrow \cdot a, \uparrow$

$I_3: S \rightarrow ; \cdot Bd, \#$

$S \rightarrow ; \cdot A \uparrow, \#$

$A \rightarrow \cdot a, \uparrow$

$B \rightarrow \cdot a, d$

$I_4: S \rightarrow aA \cdot d, \#$

$I_5: S \rightarrow aB \cdot \uparrow, \#$

$I_6: A \rightarrow a \cdot, d$

$B \rightarrow a \cdot, \uparrow$

$I_7: S \rightarrow ;B \cdot d, \#$

$I_8: S \rightarrow ;A \cdot \uparrow, \#$

$I_9: A \rightarrow a \cdot, \uparrow$

$B \rightarrow a \cdot, d$

$I_{10}: S \rightarrow aAd \cdot, \#$

$I_{11}: S \rightarrow aB \uparrow \cdot, \#$

$I_{12}: S \rightarrow ;Bd \cdot, \#$

$I_{13}: S \rightarrow ;A \uparrow \cdot, \#$

根据 LR(1) 项目集族, 将同心集合并(即去掉向前搜索符后两个项目的产生式相同)。经检查, 只有  $I_6$  与  $I_9$  同心, 即将  $I_6$  和  $I_9$  合并为  $I_{69}$ :

$I_{69}: A \rightarrow a \cdot, \uparrow / d$

$B \rightarrow a \cdot, \uparrow / d$

此时出现了“归约/归约”冲突, 即对“ $\uparrow$ ”或“ $d$ ”不知是用  $A \rightarrow a \cdot$  归约, 还是用  $B \rightarrow a \cdot$  归约, 故  $G[S]$  不是 LALR 文法。