



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

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

2. LR(0)分析

- 小结——LR(0)分析过程

- ① 对文法进行拓广：对文法 $G[S]$ ，增加一条产生式 $S' \rightarrow S$ ，拓广为文法 $G'[S']$
- ② 根据产生式构造LR(0)项目集：CLOSURE函数和GOTO函数
- ③ 根据项目集构造LR(0)DFA
- ④ 根据LR(0)DFA构造LR(0)分析表
- ⑤ 根据LR(0)分析表进行LR(0)分析

2. LR(0)分析

- 项目集中的冲突：一个项目集中存在下列情况称为项目冲突：

- 移进-归约冲突

- ✓ 移进项目 $A \rightarrow \alpha \cdot a \beta$ 和归约项目 $B \rightarrow r \cdot$ 同在一个项目集中，当面临输入符 a 时，不能确定移进 a 还是把 r 归约为 B ；

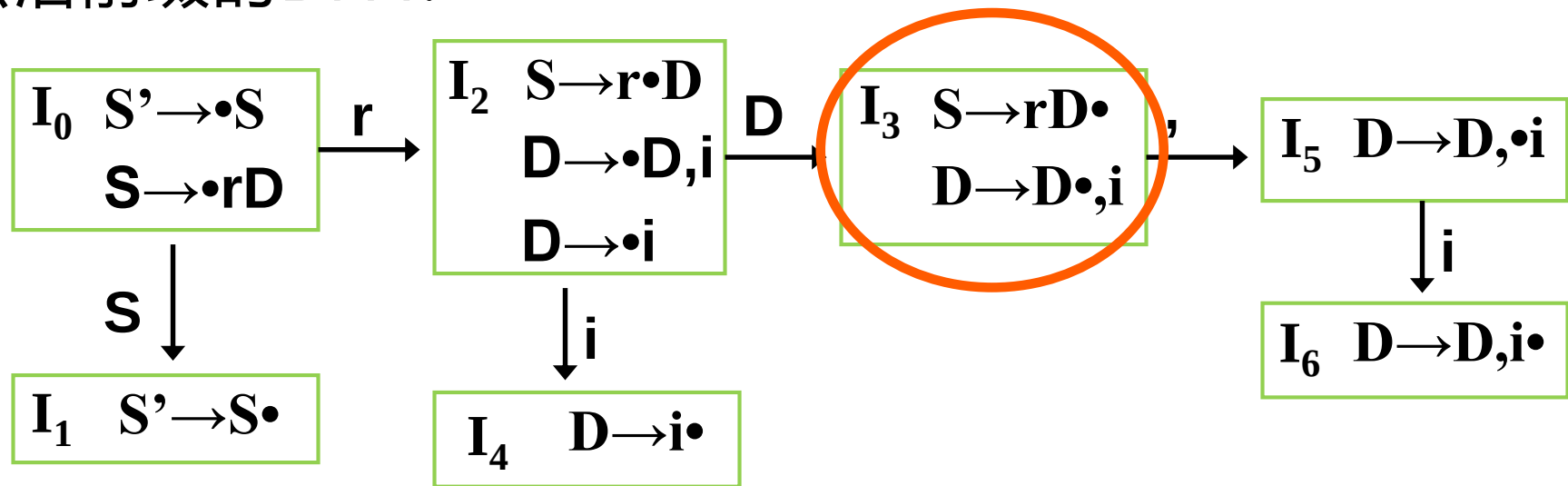
- 归约-归约冲突

- ✓ 归约项目 $A \rightarrow \beta \cdot$ 和归约项目 $B \rightarrow r \cdot$ 同在一个项目集中，不管面临什么输入符都不能确定把 β 归约为 A 还是把 r 归约为 B 。

一个文法的LR(0)项目集规范族中的项目集不存在移进-归约冲突，也不存在归约-归约冲突时，称该文法为LR(0)文法。

2. LR(0)分析

- 例：文法 $G'[S']$: (0) $S' \rightarrow S$ (1) $S \rightarrow rD$ (2) $D \rightarrow D,i$ (3) $D \rightarrow i$
- 识别文法活前缀的DFA:



不管面临什么输入符号，都归约

当面临输入为“,”时，移进

在项目集 I_3 中 $S \rightarrow rD \bullet$ 为**归约项目**， $D \rightarrow D \bullet,i$ 为**移进项目**，存在**移进-归约冲突**，该文法不是LR(0)文法。

3. SLR(1)分析

- SLR(1)分析方法可解决这种冲突

- 基本思想:

- ✓ 对于LR(0)有**冲突**的**项目集**用**向前查看输入符号串的1个符号**的办法加以解决

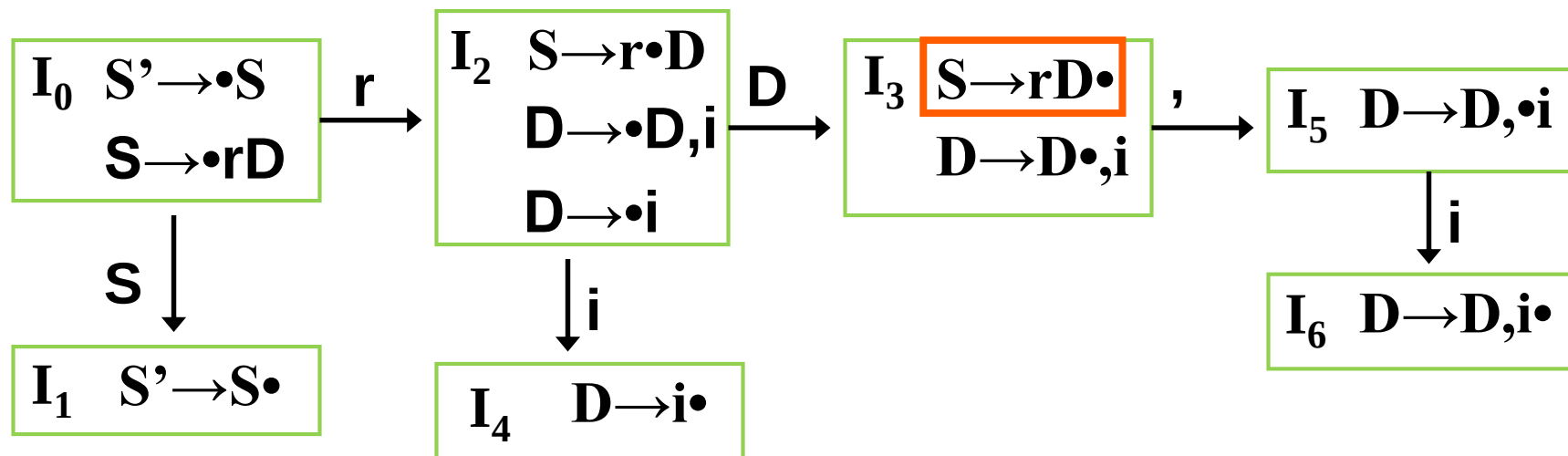
- 解决方法:

- ✓ 对归约项目 $A \rightarrow r\bullet$, **只有当输入符号 $a \in FOLLOW(A)$ 才进行归约**, 缩小归约范围, 有可能解决冲突。

而不是不管面临什么
输入符号都归约

3. SLR(1)分析

- 例：文法 $G'[S']$: (0) $S' \rightarrow S$ (1) $S \rightarrow rD$ (2) $D \rightarrow D,i$ (3) $D \rightarrow i$
- 识别文法活前缀的DFA:



	Follow集
S'	\$
S	\$
D	\$,

对项目集 I_3 :
 当前输入符为\$时, 进行归约;
 当前输入符为,时, 进行移进操作;
 冲突得以解决。

3. SLR(1)分析

- SLR(1)的处理方法:

- 一个LR(0)项目集I含有冲突项目 $I=\{X\rightarrow\alpha\cdot b\beta, A\rightarrow r\cdot, B\rightarrow\delta\cdot\}$, 其中 $b\in V_T$, 如果 $FOLLOW(A)$ 和 $FOLLOW(B)$ 互不相交, 且不含 b , 则当状态I面临某输入符 a 时, 可采用下列动作:
 - ✓ 若 $a=b$, 则移进;
 - ✓ 若 $a\in FOLLOW(A)$, 则用产生式 $A\rightarrow r$ 归约;
 - ✓ 若 $a\in FOLLOW(B)$, 则用产生式 $B\rightarrow\delta$ 归约;
 - ✓ 此外, 报错。

上述解决项目集(状态)中冲突的方法称为SLR(1)方法 (Simple因为只对有冲突的状态才向前查看一个符号, 以确定动作)

3. SLR(1)分析

- 如果一个文法的LR(0)项目集中某些项目集所含有的**动作冲突都能用SLR(1)方法解决**，则称这个文法为**SLR(1)文法**
- SLR(1)分析表的构造算法
 - 与LR(0)分析表的构造类似
 - **仅对表格中有冲突的行采用SLR(1)方法区分移进或归约对应的列**
- 改进的SLR(1)分析表的构造算法
 - 对所有的归约项目仅对**当前输入符号**包含在该**归约项目左部非终结符的FOLLOW集**中，才采取归约动作。

3. SLR(1)分析

- SLR(1)分析表的构造方法:

- 含 $S' \rightarrow \bullet S$ 项目的项目集为初态

- 分析表的ACTION表和GOTO表构造步骤为:

- ✓ 若项目 $A \rightarrow \alpha \bullet a \beta \in k$, $a \in V_T$, 且 $GOTO(k, a) = j$, 则置 $ACTION[k, a] = 'S_j'$, 移进;

- ✓ 若项目 $A \rightarrow \alpha \bullet B \beta \in k$, $B \in V_N$, 且 $GOTO(k, B) = j$, 则置 $GOTO[k, B] = 'j'$;

- ✓ 若项目 $A \rightarrow \alpha \bullet \in k$, 且产生式 $A \rightarrow \alpha$ 的编号为 j , 则对任何 a (终结符和'\$'), 且 $a \in FOLLOW(A)$, 置 $ACTION[k, a] = 'r_j'$, 归约;

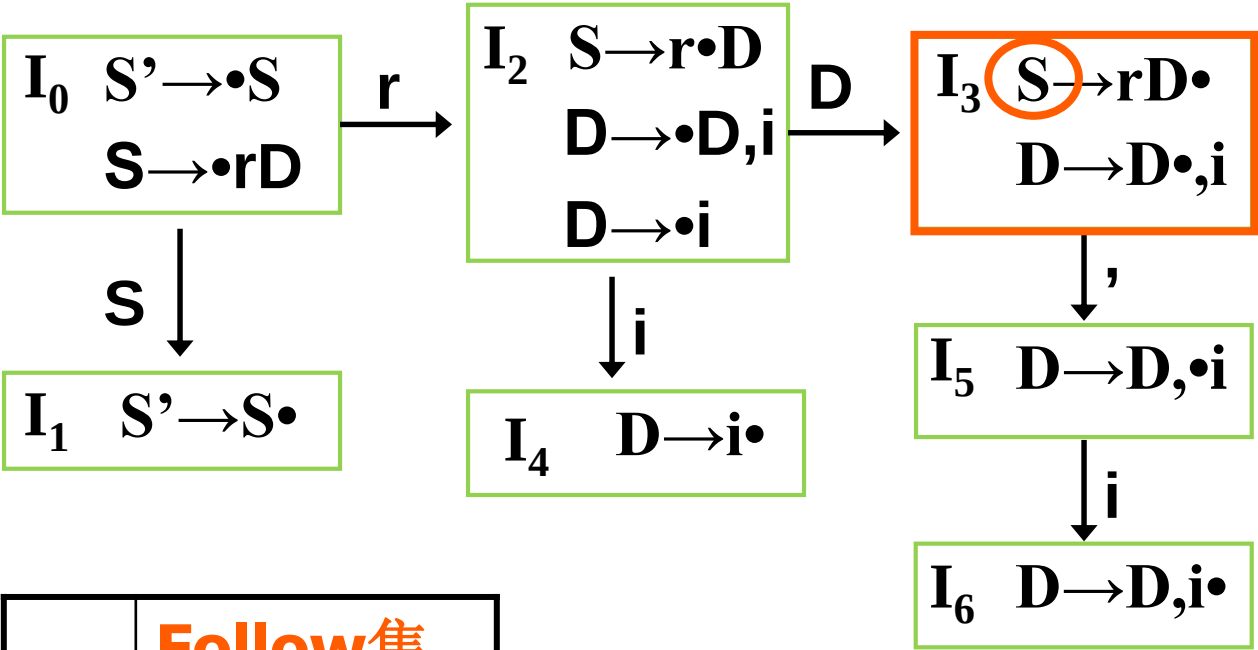
与LR(0)的唯一区别,
SLR(1)的1之所在

- ✓ 若项目 $S' \rightarrow S \bullet \in k$, 则置 $ACTION[k, \$] = 'acc'$, ACCEPT;

- ✓ 不能用上述方法填入的分析表的元素可空着, 表示ERROR。

3. SLR(1)分析

• 例：文法G'[S']： (0) $S' \rightarrow S$ (1) $S \rightarrow rD$ (2) $D \rightarrow D,i$ (3) $D \rightarrow i$



	Follow集
S'	\$
S	\$
D	\$,

状态	ACTION				GOTO	
	r	,	i	\$	S	D
0	S ₂				A	
1				acc		
2			S ₄			3
3		S ₅		r ₁		
4		r ₃		r ₃		
5			S ₆			
6		r ₂		r ₂		

3. SLR(1)分析

- 仍有许多文法构造的LR(0)项目集规范族存在的动作冲突不能用SLR(1)方法解决

文法G[S']:

(0) $S' \rightarrow S$

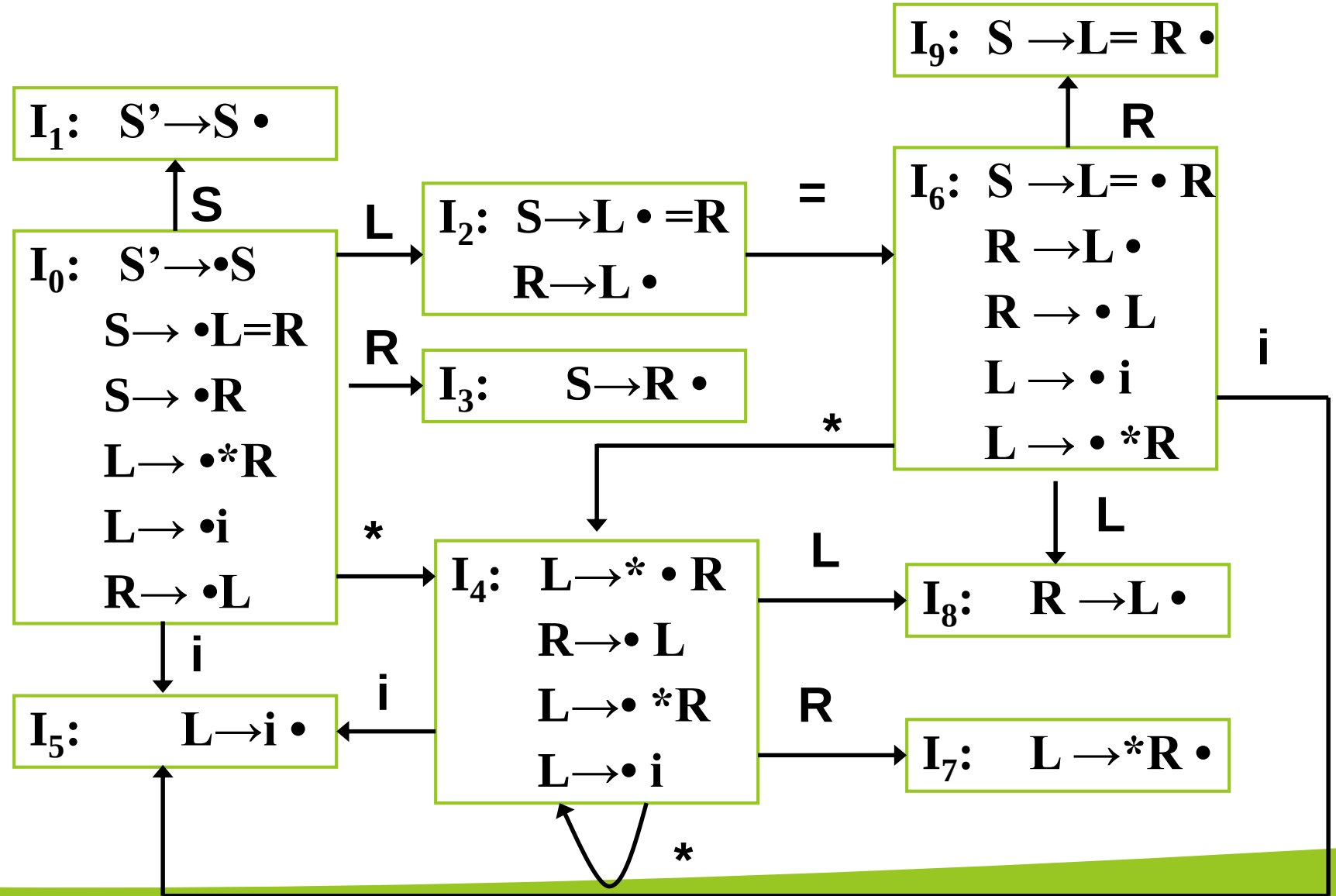
(1) $S \rightarrow L=R$

(2) $S \rightarrow R$

(3) $L \rightarrow *R$

(4) $L \rightarrow i$

(5) $R \rightarrow L$



3. SLR(1)分析

$$S' \rightarrow S$$

$$S \rightarrow L=R \mid R$$

$$L \rightarrow *R \mid i$$

$$R \rightarrow L$$

	Follow集
S'	\$
S	\$
L	= \$
R	\$ =

$$I_6: S \rightarrow L= \bullet R$$

$$R \rightarrow L \bullet$$

$$R \rightarrow \bullet L$$

$$L \rightarrow \bullet i$$

$$L \rightarrow \bullet * R$$

Follow(R)={\$,=}与移进符号集{*,i}交集为空, 可用SLR(1)方法解决冲突

$$I_2: S \rightarrow L \bullet = R$$

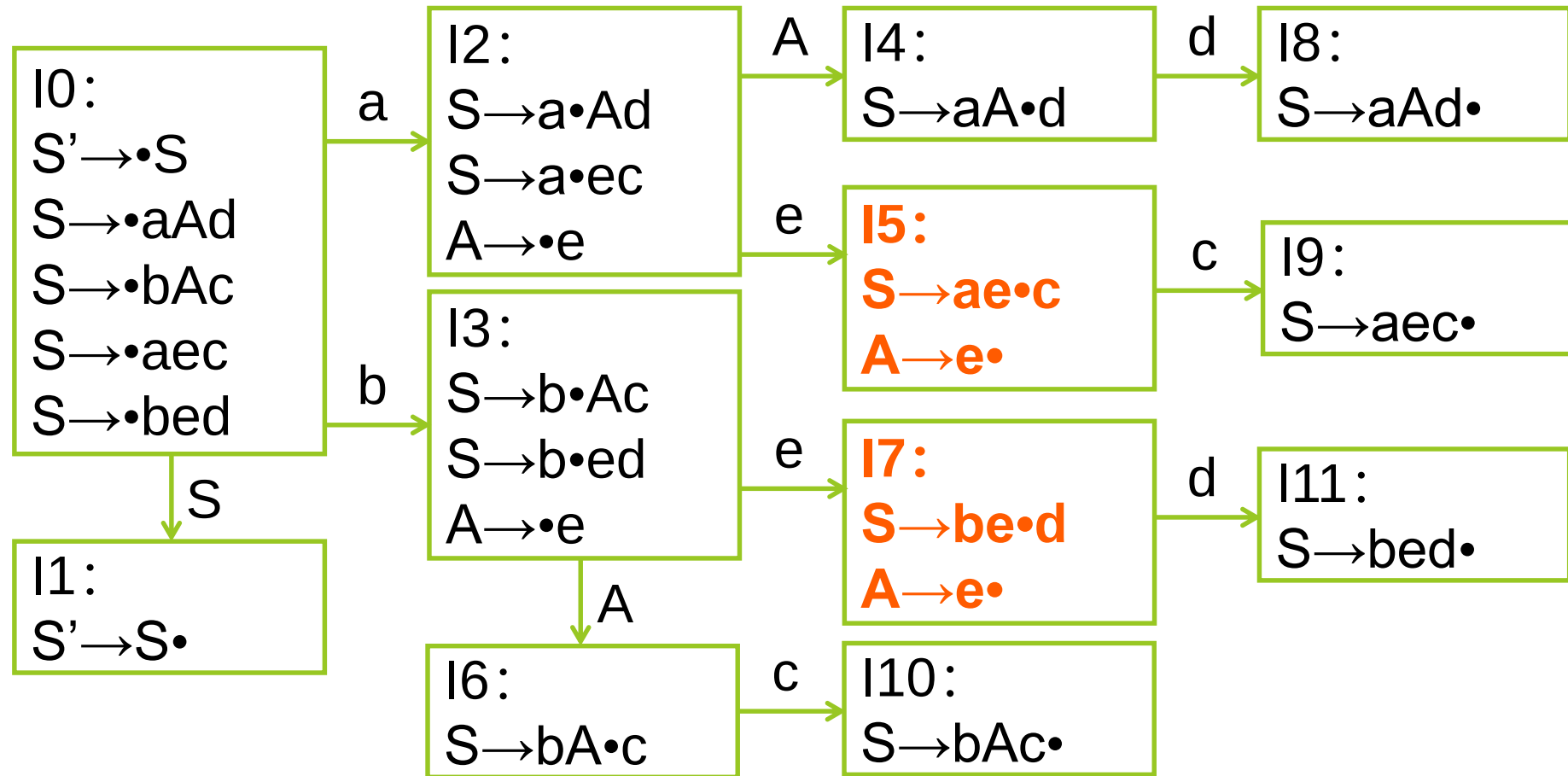
$$R \rightarrow L \bullet$$

Follow(R)={\$,=}与移进符号集{=}交集不为空, SLR(1)方法不能解决冲突, 该文法不是SLR(1)文法。

3. SLR(1)分析

• 例：文法 $G'[S']$:

(0) $S' \rightarrow S$ (1) $S \rightarrow aAd$ (2) $S \rightarrow bAc$ (3) $S \rightarrow aec$ (4) $S \rightarrow bed$ (5) $A \rightarrow e$



3. SLR(1)分析

• 例：文法 $G'[S']$:

(0) $S' \rightarrow S$ (1) $S \rightarrow aAd$ (2) $S \rightarrow bAc$ (3) $S \rightarrow aec$ (4) $S \rightarrow bed$ (5) $A \rightarrow e$

I5:

$S \rightarrow ae \bullet c$

$A \rightarrow e \bullet$

FOLLOW(A)={c,d}与移进符号集{c}交集不为空,
SLR(1)仍无法解决冲突。

I7:

$S \rightarrow be \bullet d$

$A \rightarrow e \bullet$

FOLLOW(A)={c,d}与移进符号集{d}交集不为空,
SLR(1)仍无法解决冲突。

3. SLR(1)分析

- SLR(1)用FOLLOW信息作为展望信息（只对属于FOLLOW集的输入符号归约），缩小了归约范围，消除了一些无效归约，解决了项目集中的一些简单的冲突。
- 尽管FOLLOW(A)中包含了所有含A的句型中A后的可能终结符，但Follow(A)可能与移进符号有交集，且并不是每个含有A的句型中，A的后面都可以出现Follow(A)中的每一个符号，所以SLR(1)未能从根本上消除所有无效归约。

4. LR(1)分析

- LR(1)分析的基本思想

- LR(1)方法按每个具体的句型设置展望信息。
- 例：如果存在如下句型 $\dots\alpha A a\dots$, $\dots\beta A b\dots$, $\dots\gamma A c\dots$, 则

$\text{FOLLOW}(A)=\{a,b,c\}$

- ✓ 处理到句型 $\dots\alpha A$, 只当输入符号为a时归约;
- ✓ 处理到句型 $\dots\beta A$, 只当输入符号为b时归约;
- ✓ 处理到句型 $\dots\gamma A$, 只当输入符号为c时归约。

4. LR(1)分析

- LR(1)项目

- 在 LR(0)项目基础上增加一个终结符，所增加的终结符称为向前搜索符 (lookahead)，表示产生式右端完整匹配后所允许在剩余符号串中的下一个终结符
- LR(1)项目形如：[$A \rightarrow \alpha \bullet \beta$, **a**]，其中， $A \rightarrow \alpha \bullet \beta$ 同LR(0)项目，**a为向前搜索符，或为终结符，或为输入结束标志符\$**
- 对于形如：[$A \rightarrow \alpha \bullet$, a]的LR(1)项目，对应LR(0)的归约项目，但只有当下一个输入符是a时才能进行归约
- 对于其它形式的LR(1)项目，a只起到信息传承的作用

4. LR(1)分析

- 初始项目集

- $[S' \rightarrow \bullet S, \$]$ 属于初始项目集，此时，把\$作为向前搜索符，表示要把S产生式的右部归约成S时，必须面临输入符为\$才行

- CLOSURE函数

- ① 如果I是一个项目集，I的任何项目都属于CLOSURE(I)
 - ② 若有项目 $[A \rightarrow \alpha \bullet B \beta, a]$ 属于CLOSURE(I)， $B \rightarrow \gamma$ 是文法的产生式， $\beta \in V^*$ ，
 $b \in \text{FIRST}(\beta a)$ ，则 $[B \rightarrow \bullet \gamma, b]$ 也属于CLOSURE(I)
 - ③ 重复②直到CLOSURE(I)不再扩大为止。

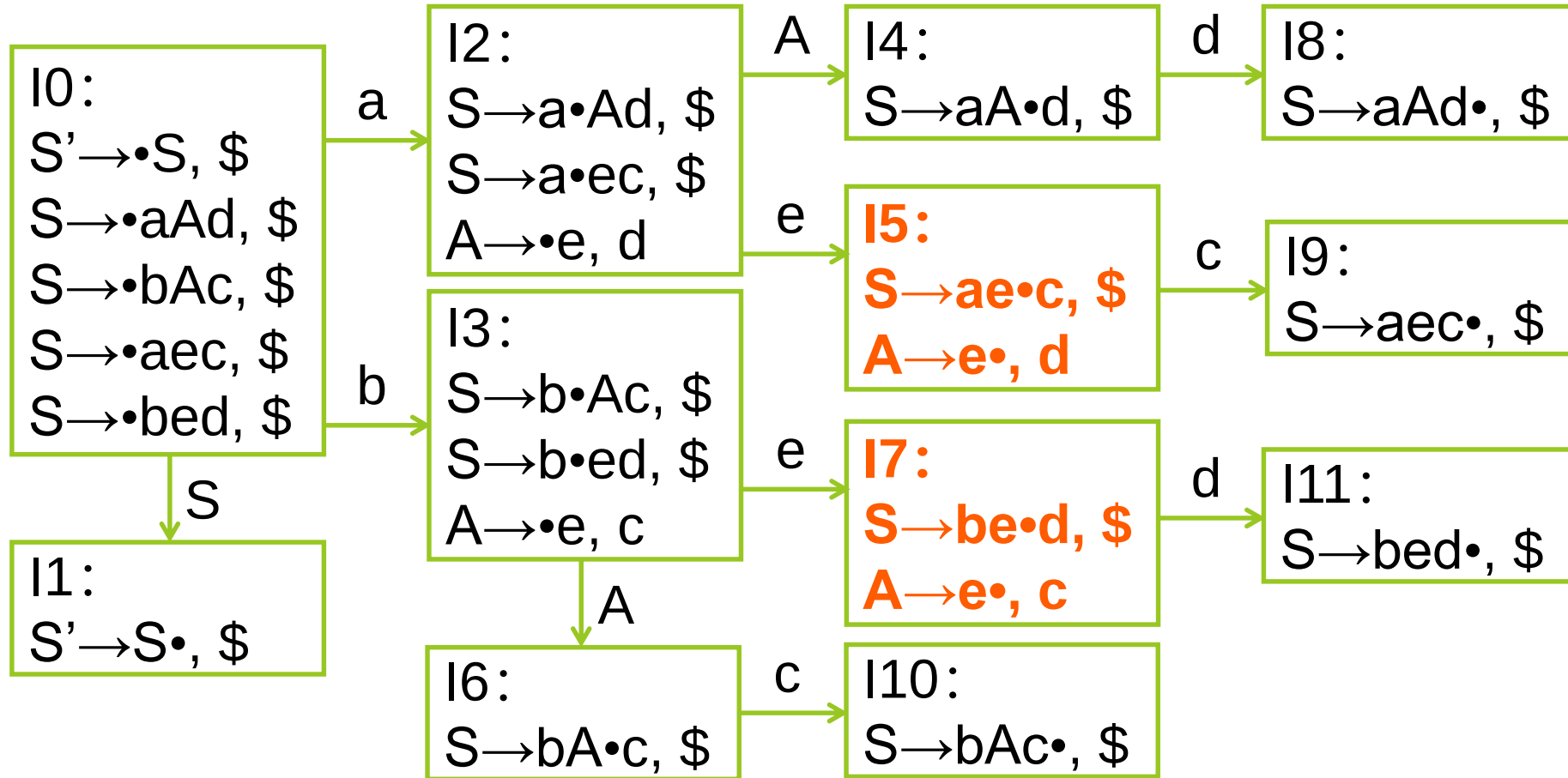
- GOTO函数

- 与LR(0)相似
 - $\text{GOTO}(I, X) = \text{CLOSURE}(J)$ ， $J = \{\text{任何形如}[A \rightarrow \alpha X \bullet \beta, a] \text{的项目} \mid [A \rightarrow \alpha \bullet X \beta, a] \in I\}$

4. LR(1)分析

• 例：文法 $G'[S']$:

(0) $S' \rightarrow S$ (1) $S \rightarrow aAd$ (2) $S \rightarrow bAc$ (3) $S \rightarrow aec$ (4) $S \rightarrow bed$ (5) $A \rightarrow e$



4. LR(1)分析

• 例：文法 $G'[S']$:

(0) $S' \rightarrow S$ (1) $S \rightarrow aAd$ (2) $S \rightarrow bAc$ (3) $S \rightarrow aec$ (4) $S \rightarrow bed$ (5) $A \rightarrow e$

I5:

$S \rightarrow ae \bullet c, \$$

$A \rightarrow e \bullet, d$

遇到输入字符c时, 移进;
遇到输入字符d时, 用产生式 $A \rightarrow e$ 归约;
冲突得以解决。

I7:

$S \rightarrow be \bullet d, \$$

$A \rightarrow e \bullet, c$

遇到输入字符d时, 移进;
遇到输入字符c时, 用产生式 $A \rightarrow e$ 归约;
冲突得以解决。

4. LR(1)分析

- 如果一个文法的LR(1)项目集中没有移进-归约或归约-归约冲突，则称该文法为LR(1)文法。

随堂练习 (9)

- 若文法 G' 为：(0) $S' \rightarrow S$ (1) $S \rightarrow BB$ (2) $B \rightarrow aB$ (3) $B \rightarrow b$ ，请画出其对应的LR(1)DFA。

