



语法分析 规范的LR方法

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2026年09月21日



SLR(1)文法的描述能力有限



$S \rightarrow V = E$
 $S \rightarrow E$
 $V \rightarrow * E$
 $V \rightarrow \text{id}$
 $E \rightarrow V$

I_0 :
 $S' \rightarrow \cdot S$
 $S \rightarrow \cdot V = E$
 $S \rightarrow \cdot E$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$
 $E \rightarrow \cdot V$

V

I_2 :
 $S \rightarrow V \cdot = E$
 $E \rightarrow V \cdot$

$=$

I_6 :
 $S \rightarrow V = \cdot E$
 $E \rightarrow \cdot V$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$



SLR(1)文法的描述能力有限



$S \rightarrow V = E$
 $S \rightarrow E$
 $V \rightarrow * E$
 $V \rightarrow \text{id}$
 $E \rightarrow V$

I_0 :
 $S' \rightarrow \cdot S$
 $S \rightarrow \cdot V = E$
 $S \rightarrow \cdot E$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$
 $E \rightarrow \cdot V$

V

I_2 :
 $S \rightarrow V \cdot = E$
 $E \rightarrow V \cdot$

$=$

I_6 :
 $S \rightarrow V = \cdot E$
 $E \rightarrow \cdot V$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$

项目 $S \rightarrow V \cdot = E$ 使得
 $\text{action}[2, =] = \text{s6}$

项目 $E \rightarrow V \cdot$ 使得
 $\text{action}[2, =] = \text{r5}$
因为 $\text{Follow}(E) = \{=, \$\}$



SLR(1)文法的描述能力有限



$S \rightarrow V = E$
 $S \rightarrow E$
 $V \rightarrow * E$
 $V \rightarrow \text{id}$
 $E \rightarrow V$

I_0 :
 $S' \rightarrow \cdot S$
 $S \rightarrow \cdot V = E$
 $S \rightarrow \cdot E$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$
 $E \rightarrow \cdot V$

V

I_2 :
 $S \rightarrow V \cdot = E$
 $E \rightarrow V \cdot$

$=$

I_6 :
 $S \rightarrow V = \cdot E$
 $E \rightarrow \cdot V$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$

产生移进-归约冲突,
但该文法不是二义的。

项目 $S \rightarrow V \cdot = E$ 使得
 $\text{action}[2, =] = \text{s6}$

项目 $E \rightarrow V \cdot$ 使得
 $\text{action}[2, =] = \text{r5}$
因为 $\text{Follow}(E) = \{=, \$\}$



- 目标：在识别活前缀DFA的**状态**中，**增加信息**，**排除一些不正确的归约操作**
- 方法：添加了前向搜索符
 - 一个项目 $A \rightarrow \alpha \cdot \beta$ ，如果最终用这个产生式进行归约之后，期望看见的符号是 a ，则这个加点项的前向搜索符是 a 。
 - 上述项目可以写成： $A \rightarrow \alpha \cdot \beta, a$
- 与SLR(1)分析的区别
 - 项目集的定义发生了改变： $LR(0) \Rightarrow LR(1)$
 - $\text{closure}(I)$ 和GOTO函数需要修改



- 目标：在识别活前缀DFA的**状态**中，**增加信息**，**排除一些不正确的归约操作**
- 方法：添加了前向搜索符
 - 一个项目 $A \rightarrow a \cdot \beta$ ，如果最终用这个产生式进行归约之后，期望看见的符号是 a ，则这个加点项的前向搜索符是 a 。
 - 上述项目可以写成： $A \rightarrow a \cdot \beta, a$
- 与SLR(1)分析的区别
 - 项目集的定义发生了改变： $LR(0) \Rightarrow LR(1)$
 - $\text{closure}(I)$ 和GOTO函数需要修改



- 目标：在识别活前缀DFA的**状态**中，**增加信息**，**排除一些不正确的归约操作**
- 方法：添加了前向搜索符
 - 一个项目 $A \rightarrow a \cdot \beta$ ，如果最终用这个产生式进行归约之后，期望看见的符号是 a ，则这个加点项的前向搜索符是 a 。
 - 上述项目可以写成： $A \rightarrow a \cdot \beta, a$
- 与SLR(1)分析的区别
 - **项目集**的定义发生了改变： $LR(0) \Rightarrow LR(1)$
 - **closure(I)** 和 **GOTO** 函数需要修改



- LR(1)项目:

$$[A \rightarrow a \cdot \beta, a]$$

- 当项目由两个分量组成，第一分量为SLR中的项，第二分量为搜索符(向前看符号)
- LR(1)中的1代表了搜索符 a 的长度



- **LR(1)项目:**

$$[A \rightarrow a \cdot \beta, a]$$

- 当项目由两个分量组成，第一分量为SLR中的项，第二分量为搜索符(向前看符号)
- LR(1)中的1代表了搜索符 a 的长度

- **使用注意事项:**

- 当 β 不为空时， a 不起作用
- 当 β 为空时，如果下一个输入符号是 a ，将按照 $A \rightarrow a$ 进行归约
 - a 的集合是FOLLOW(A)的子集



- LR(1)项目:

$$[A \rightarrow a \cdot \beta, a]$$

- 当项目由两个分量组成, 第一分量为SLR中的项, 第二分量为搜索符(向前看符号)

- LR(1)中的1代表了搜索符 a 的长度

- LR(1)项目 $[A \rightarrow a \cdot \beta, a]$ 对活前缀 γ 有效:

- 如果存在着推导 $S \Rightarrow_{rm}^* \delta A w \Rightarrow_{rm} \delta a \beta w$, 其中:

- $\gamma = \delta a$;

- a 是 w 的第一个符号, 或者 w 是 ϵ 且 a 是 $\$$



• 例 $S \rightarrow BB$

$B \rightarrow bB \mid a$

LR(1)项目 $[A \rightarrow a \cdot \beta, a]$ 对活前缀 γ 有效：
存在着推导 $S \Rightarrow_{rm}^* \delta A w \Rightarrow_{rm} \delta a \beta w$ ，其中：
 $\gamma = \delta a$ ；
 a 是 w 的第一个符号，或者 w 是 ϵ 且 a 是 $\$$

考虑 $[B \rightarrow b \cdot B, b]$ 项，

从最右推导 $S \Rightarrow_{rm}^* bbBba \Rightarrow_{rm} bbbBba$ 看出：

令 $A = B, a = b, \beta = B, \delta = bb, \gamma = \delta a = bbb, w = ba$

$[B \rightarrow b \cdot B, b]$ 对活前缀 $\gamma = bbb$ 是有效的



• 例 $S \rightarrow BB$

$B \rightarrow bB \mid a$

LR(1)项目 $[A \rightarrow a \cdot \beta, a]$ 对活前缀 γ 有效：
存在着推导 $S \Rightarrow_{rm}^* \delta A w \Rightarrow_{rm} \delta a \beta w$ ，其中：
 $\gamma = \delta a$ ；
 a 是 w 的第一个符号，或者 w 是 ϵ 且 a 是 $\$$

考虑 $[B \rightarrow b \cdot B, b]$ 项，

从最右推导 $S \Rightarrow_{rm}^* bbBba \Rightarrow_{rm} bbbBba$ 看出：

令 $A = B$, $a = b$, $\beta = B$, $\delta = bb$, $\gamma = \delta a = bbb$, $w = ba$

$[B \rightarrow b \cdot B, b]$ 对活前缀 $\gamma = bbb$ 是有效的



规范的LR分析—步骤



- **构造LR(1)项目集规范族**
 - 也就是构造识别活前缀的DFA
- **构造规范的LR分析表**
 - 状态之间的转换关系



- 基础运算1：计算闭包CLOSURE(I)

- I中的任何项目都属于CLOSURE(I)
- 若有项目 $[A \rightarrow \alpha \cdot B \beta, a]$ 在CLOSURE(I)中，而 $B \rightarrow \gamma$ 是文法中的产生式， b 是FIRST(βa)中的元素，则 $[B \rightarrow \cdot \gamma, b]$ 也属于CLOSURE(I)

保证在用 $B \rightarrow \gamma$ 进行归约后，

- 出现的输入字符 b 是句柄 $\alpha B \beta$ 中 B 的后继符号
- 或者是 $\alpha B \beta$ 归约为 A 后可能出现的终结符。



- **基础运算2：通过GOTO(I,X)算CLOSURE(J)**

- 将J置为空集
- 若有项目 $[A \rightarrow a \cdot X \beta, a]$ 在I中，那么将项目 $[A \rightarrow a X \cdot \beta, a]$ 放入J中
- 计算并返回CLOSURE(J)

注意： GOTO(I,X)中的X可以是终结符或非终结符



• 具体算法

- 初始项目集 I_0 :

$I_0 = \text{CLOSURE}([S' \rightarrow \cdot S, \$])$ 将\$作为向前的搜索符

- 设C为最终返回的项目集族, 初始为 $C = \{I_0\}$

- 重复以下步骤

- 对C中的任意项目集I, 重复

- 对每一个文法符号X(终结符或非终结符)

- 如果 $\text{GOTO}(I, X) \neq \emptyset$ 且 $\text{GOTO}(I, X) \notin C$, 那么将 $\text{GOTO}(I, X)$ 放入C

- 注: 上述 $\text{GOTO}(I, X)$ 是上一页ppt中计算闭包的GOTO

- 当C中项目集不再增加为止



构造LR(1)项目集族：举例



$$S' \rightarrow \cdot S, \$ \quad I_0$$

步骤一：从初始项开始



构造LR(1)项目集族：举例



$$\begin{array}{l} S' \rightarrow \cdot S, \$ \\ S \rightarrow \cdot BB \end{array} I_0$$

步骤二：计算非核心项目的第一个分量



构造LR(1)项目集族：举例


$$\begin{array}{l} S' \rightarrow \cdot S, \$ \\ S \rightarrow \cdot BB, \$ \end{array} I_0$$

步骤三：通过FIRST(ϵ \$) 计算非核心项目的第二个分量



构造LR(1)项目集族：举例



$S' \rightarrow \cdot S, \$$ $S \rightarrow \cdot BB, \$$ $B \rightarrow \cdot bB$ $B \rightarrow \cdot a$	I_0
---	-------

步骤二：计算非核心项目的
第一个分量



构造LR(1)项目集族：举例



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

$S \rightarrow \cdot BB, \$$

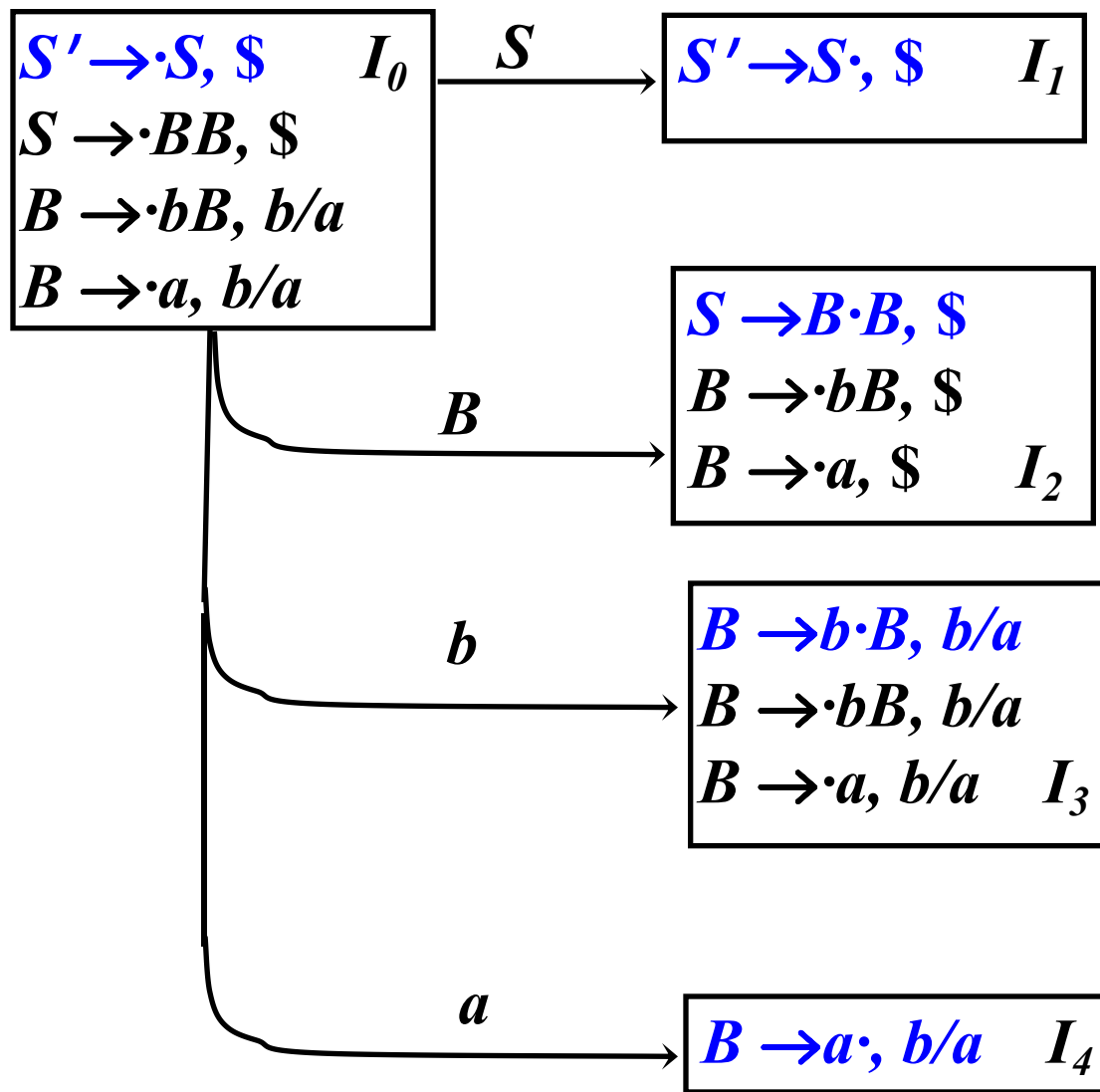
$B \rightarrow \cdot bB, b/a$

$B \rightarrow \cdot a, b/a$

步骤三：通过FIRST(B\$) 计算非核心项目的第二个分量

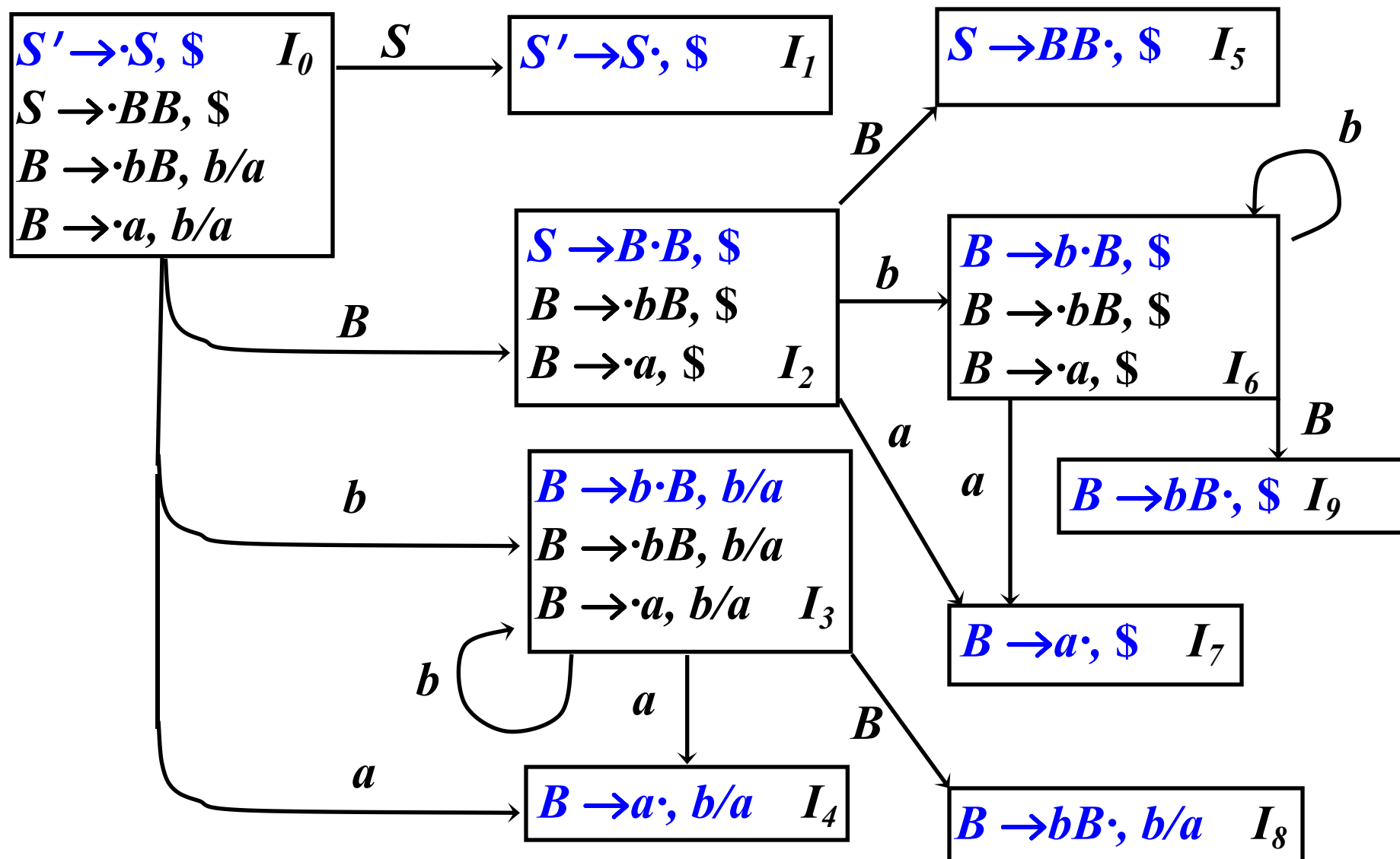


构造LR(1)项目集族：举例





构造LR(1)项目集族：举例





- 构造识别拓广文法 G' 活前缀的DFA

- 基于LR(1)项目族来构造

- 状态 i 的 $action$ 函数如下确定:

- 如果 $[A \rightarrow a \cdot a\beta, b]$ 在 I_i 中, 且 $goto(I_i, a) = I_j$, 那么置 $action[i, a]$ 为 sj (此时, 不看 b)
- 如果 $[A \rightarrow a \cdot, a]$ 在 I_i 中, 且 $A \neq S'$, 那么置 $action[i, a]$ 为 rj (此时, 不再看 FOLLOW(A))
- 如果 $[S' \rightarrow S \cdot, \$]$ 在 I_i 中, 那么置 $action[i, \$] = acc$

如果上述构造出现了冲突, 那么文法就不是LR(1)的



构造规范的LR分析表



- 构造识别拓广文法 G' 活前缀的DFA
- 状态 i 的 $action$ 函数如下确定：
 - 参见上页ppt
- 状态 i 的 $goto$ 函数如下确定：
 - 如果 $goto(I_i, A) = I_j$, 那么 $goto[i, A] = j$



- 构造识别拓广文法 G' 活前缀的DFA
- 状态 i 的 $action$ 函数如下确定：
 - 参见上页ppt
- 状态 i 的 $goto$ 函数如下确定：
 - 如果 $goto(I_i, A) = I_j$, 那么 $goto[i, A] = j$
- 分析器的初始状态是包含 $[S' \rightarrow \cdot S, \$]$ 的项目集对应的状态

用上面规则未能定义的所有条目都置为 **error**



非SLR(1)文法-revisit



$S \rightarrow V = E$
 $S \rightarrow E$
 $V \rightarrow * E$
 $V \rightarrow \text{id}$
 $E \rightarrow V$

$I_0:$
 $S' \rightarrow \cdot S$
 $S \rightarrow \cdot V = E$
 $S \rightarrow \cdot E$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$
 $E \rightarrow \cdot V$

V

$I_2:$
 $S \rightarrow V \cdot = E$
 $E \rightarrow V \cdot$

$=$

$I_6:$
 $S \rightarrow V = \cdot E$
 $E \rightarrow \cdot V$
 $V \rightarrow \cdot * E$
 $V \rightarrow \cdot \text{id}$

产生移进-归约冲突,
但该文法不是二义的。

项目 $S \rightarrow V \cdot = E$ 使得
 $\text{action}[2, =] = \text{s6}$

项目 $E \rightarrow V \cdot$ 使得
 $\text{action}[2, =] = \text{r5}$
 因为 $\text{Follow}(E) = \{=, \$\}$



非SLR(1)但是LR(1)文法


$$S \rightarrow V = E$$
$$S \rightarrow E$$
$$V \rightarrow * E$$
$$V \rightarrow \text{id}$$
$$E \rightarrow V$$
$$I_0:$$
$$S' \rightarrow \cdot S, \$$$
$$S \rightarrow \cdot V = E, \$$$
$$S \rightarrow \cdot E, \$$$
$$V \rightarrow \cdot * E, =$$
$$V \rightarrow \cdot \text{id}, =$$
$$E \rightarrow \cdot V, \$$$
$$V \rightarrow \cdot * E, \$$$
$$V \rightarrow \cdot \text{id}, \$$$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

$$I_0:$$

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow \alpha \cdot B \beta, \alpha]$

这里: $[S' \rightarrow \epsilon \cdot S \epsilon, \$]$

$\text{FIRST}(\beta \alpha)$



$\text{FIRST}(\epsilon \$) = \{\$ \}$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

$$I_0:$$

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow a \cdot B \beta, a]$

这里: $[S' \rightarrow \epsilon \cdot S \epsilon, \$]$

$\text{FIRST}(\beta a)$



$\text{FIRST}(\epsilon \$) = \{\$ \}$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

$$I_0:$$

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow a \cdot B \beta, a]$

这里: $[S \rightarrow \varepsilon \cdot V = E, \$]$

$\text{FIRST}(\beta a)$



$\text{FIRST}(= E\$) = \{=\}$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

I_0 :

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow a \cdot B \beta, a]$

这里: $[S \rightarrow \varepsilon \cdot V = E, \$]$

$\text{FIRST}(\beta a)$



$\text{FIRST}(= E\$) = \{=\}$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

$$I_0:$$

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow a \cdot B \beta, a]$

这里: $[S \rightarrow \varepsilon \cdot V = E, \$]$

$\text{FIRST}(\beta a)$



$\text{FIRST}(= E\$) = \{=\}$



非SLR(1)但是LR(1)文法



$$S \rightarrow V = E$$

$$S \rightarrow E$$

$$V \rightarrow * E$$

$$V \rightarrow \text{id}$$

$$E \rightarrow V$$

$$I_0:$$

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot V = E, \$$$

$$S \rightarrow \cdot E, \$$$

$$V \rightarrow \cdot * E, =$$

$$V \rightarrow \cdot \text{id}, =$$

$$E \rightarrow \cdot V, \$$$

$$V \rightarrow \cdot * E, \$$$

$$V \rightarrow \cdot \text{id}, \$$$

计算闭包:

定义里: $[A \rightarrow a \cdot B \beta, a]$

这里: $[S \rightarrow \varepsilon \cdot E \varepsilon, \$]$

FIRST(βa)



FIRST($\varepsilon \$$) = { $\$$ }



非SLR(1)但是LR(1)文法


$$S \rightarrow V = E$$
$$S \rightarrow E$$
$$V \rightarrow * E$$
$$V \rightarrow \text{id}$$
$$E \rightarrow V$$
$$I_0:$$
$$S' \rightarrow \cdot S, \$$$
$$S \rightarrow \cdot V = E, \$$$
$$S \rightarrow \cdot E, \$$$
$$V \rightarrow \cdot * E, =$$
$$V \rightarrow \cdot \text{id}, =$$
$$E \rightarrow \cdot V, \$$$
$$V \rightarrow \cdot * E, \$$$
$$V \rightarrow \cdot \text{id}, \$$$



非SLR(1)但是LR(1)文法


$$S \rightarrow V = E$$
$$S \rightarrow E$$
$$V \rightarrow * E$$
$$V \rightarrow \text{id}$$
$$E \rightarrow V$$
$$I_0:$$
$$S' \rightarrow \cdot S, \$$$
$$S \rightarrow \cdot V = E, \$$$
$$S \rightarrow \cdot E, \$$$
$$V \rightarrow \cdot * E, =/\$$$
$$V \rightarrow \cdot \text{id}, =/\$$$
$$E \rightarrow \cdot V, \$$$

可通过合并搜索符简化



非SLR(1)但是LR(1)文法



$S \rightarrow V = E$
 $S \rightarrow E$
 $V \rightarrow * E$
 $V \rightarrow \text{id}$
 $E \rightarrow V$

I_0 :
 $S' \rightarrow \cdot S, \$$
 $S \rightarrow \cdot V = E, \$$
 $S \rightarrow \cdot E, \$$
 $V \rightarrow \cdot * E, =/\$$
 $V \rightarrow \cdot \text{id}, =/\$$
 $E \rightarrow \cdot V, \$$

$V \longrightarrow$

I_2 :
 $S \rightarrow V \cdot = E, \$$
 $E \rightarrow V \cdot, \$$

项目 $[S \rightarrow V \cdot = E, \$]$ 使得
 $\text{action}[2, =] = \text{s6}$

项目 $[E \rightarrow V \cdot, \$]$ 使得
 $\text{action}[2, \$] = \text{r5}$
因为 $\{\$ \}$ 是 $\text{Follow}(E) = \{=, \$\}$ 的真子集

每一个SLR(1)文法都是LR(1)的



LR分析法总结



		SLR	LR(1)
初始状态		$[S' \rightarrow \cdot S]$	$[S' \rightarrow \cdot S, \$]$
项目集		LR(0) CLOSURE(I)	LR(1), CLOSURE(I) 搜索符考虑 FIRST (βa)
动作	移进	$[A \rightarrow \alpha \cdot a \beta] \in I_i$ $GOTO(I_i, a) = I_j$ ACTION [i, a] = sj	$[A \rightarrow \alpha \cdot a \beta, b] \in I_i$ $GOTO(I_i, a) = I_j$ ACTION [i, a] = sj
	归约	$[A \rightarrow \alpha \cdot] \in I_i, A \neq S'$ $a \in FOLLOW(A)$ ACTION [i, a] = rj	$[A \rightarrow \alpha \cdot, a] \in I_i$ $A \neq S'$ ACTION [i, a] = rj
	接受	$[S' \rightarrow S \cdot] \in I_i$ ACTION [$i, \$$] = acc	$[S' \rightarrow S \cdot, \$] \in I_i$ ACTION [$i, \$$] = acc
	出错	空白条目	空白条目
GOTO		$GOTO(I_i, A) = I_j$ GOTO [i, A] = j	$GOTO(I_i, A) = I_j$ GOTO [i, A] = j
状态量		少(几百)	多(几千)



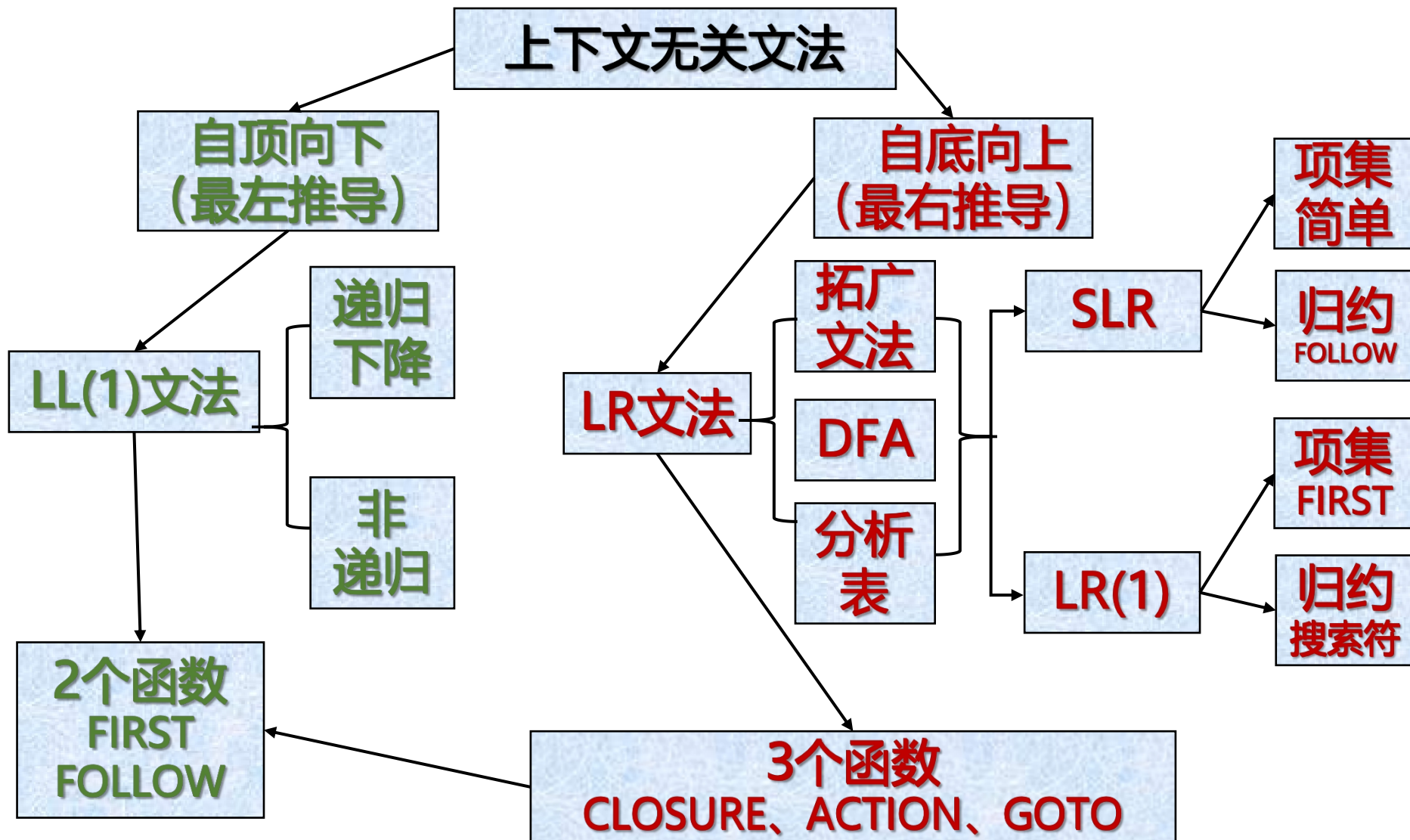
LR和LL分析方法的比较



	LR(1)方 法	LL(1)方 法
建立分析树	自底而上	自顶而下
归约or推导	规范归约	最左推导
决定使用产生式的时机	看见产生式整个右部推出的串后(句柄)	看见产生式推出的第一个终结符后
对文法的限制	无	无左递归、无公共左因子
分析表	状态 $\times$ 文法符号, 大	非终结符 $\times$ 终结符, 小
分析栈	状态栈, 信息更多	文法符号栈
确定句柄	根据栈顶状态和下一个符号便可以确定句柄和归约所用产生式	无句柄概念
语法错误	决不会将出错点后的符号移入分析栈	和LR一样, 决不会读过出错点而不报错



语法分析技术总结





教会AI写代码， 但由你定义灵魂！

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2026年09月21日