

# Canonical LR Parsing Table

<<CanonicalLRParsingTable.ppt>>

# Canonical LR Parsing Table的建構

1. 在建構狀態時,如能提供更多的資訊,則可避免不必要的**Reduction**的動作.其方法便是明白的指出哪些輸入符號(緊接在文法規則 $A \rightarrow \alpha$ 之後),需要執行 “**Reduce  $A \rightarrow \alpha$** ” 的動作.

# Canonical LR Parsing Table的建構

2. LR(1)項目中的1表示事先向前看一個輸入符號
  - I. 在 $[A \rightarrow \alpha \bullet \beta, a]$ 時,  $\beta \neq \epsilon$ , 則前瞻符號(Lookahead Symbol)  $a$ 在此並沒有發生作用, 僅會去執行Shift 的動作
  - II. 在 $[A \rightarrow \alpha \bullet, a]$ 時, 若  $a$ 為下一個輸入的Terminal Symbol, 則執行Reduce  $A \rightarrow \alpha$ 的動作

# Canonical LR Parsing Table的建構

3. 建構LR(1) item與LR(0)item集合的方式非常類似,僅需要將CLOSURE與GOTO的兩個函式做些許的修改即可
  - I. CLOSURE(I)的運算
    - a) 每個於項目集合I中的LR(1)項目皆存在於CLOSURE(I)中
    - b) 若 $[A \rightarrow \alpha \bullet B \beta, a]$ 存在於項目集合I中,且存在一文法規則 $B \rightarrow r$ ,並且b是FIRST( $\beta a$ )中的元素,則將 $[B \rightarrow \bullet r, b]$ 加於CLOSURE(I)中

# Function CLOSURE(I)

Begin

Repeat

for (每一個在項目集合I中的項目  $[A \rightarrow \alpha \bullet B \beta, a]$  且每一個文法G中的文法規則  $B \rightarrow r$ , 以及每一個終端符號  $b \in \text{FIRST}(\beta a)$ , 會使得  $[B \rightarrow \bullet r, b]$  不在目前的項目集合I中) Do

將  $[B \rightarrow \bullet r, b]$  加於項目集合I中;

Until 無其他向住可在加入項目集合I中);

Return I;

End;

# Canonical LR Parsing Table的建構

## II. GOTO(I,X)的運算

- a) 若 $[A \rightarrow \alpha \bullet X \beta, a]$ 於項目集合I中,則 $[A \rightarrow \alpha X \bullet \beta, a]$ 與其所產生之CLOSURE皆於GOTO(I,X)中

Algorithm Function GOTO(I,X)

Begin

令Y為所有會使得 $[A \rightarrow \alpha X \bullet \beta, a]$ 在項目集合I中之 $[A \rightarrow \alpha \bullet X \beta, a]$ 項目所構成的集合;

Return CLOSURE(Y)之結果;

End;

### III. 項目集合的建構

Algorithm Function ITEM( $G'$ )

Begin

$D := \{\text{CLOSURE}(\{S' \rightarrow S\})\};$

Repeat

For each 項目集合  $I \in D$ , 且對於文法符號  $X$  不會使得

$\text{GOTO}(I, X)$  為空集合 或不會使得  $\text{GOTO}(I, X)$  在  $D$  中) Do

將  $\text{GOTO}(I, X)$  加於  $D$  中;

Until (無其他項目集合可在加入  $D$  中);

End;

# Example: Grammar $G'$ 之LR(1)項目集合的建構

$G'$ :

$S' \rightarrow S$

$S \rightarrow EE$

$E \rightarrow cE | b$

1.  $I_0: \{ S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot EE, \$ \quad (\text{FIRST}(\varepsilon \$) = \{ \$ \})$

$E \rightarrow \cdot cE, c/b \quad (\text{FIRST}(E \$) = \{ c, b \})$

$E \rightarrow \cdot b, c/b \}$

a)  $\text{GOTO}(I_0, S)$

$I_1: \{ S' \rightarrow S \cdot, \$ \}$



# Example: Grammar $G'$ 之LR(1)項目集合的建構

1.  $I_0$ :

b) GOTO( $I_0, E$ )

$I_2: \{ S \rightarrow E \cdot E, \$$

$E \rightarrow \cdot cE\$$  (FIRST( $\epsilon \$$ ) = { \$ })

$E \rightarrow \cdot b, \$ \}$

c) GOTO( $I_0, c$ )

$I_3: \{ E \rightarrow c \cdot E, c/b$

$E \rightarrow \cdot cE, c/b$  (FIRST( $\epsilon c / \epsilon b$ ) = { c, b })

$E \rightarrow \cdot b, c/b \}$

d) GOTO( $I_0, b$ )

$I_4: \{ E \rightarrow b \cdot, c/b \}$

# Example: Grammar $G'$ 之LR(1)項目集合的建構

2.  $I_2$

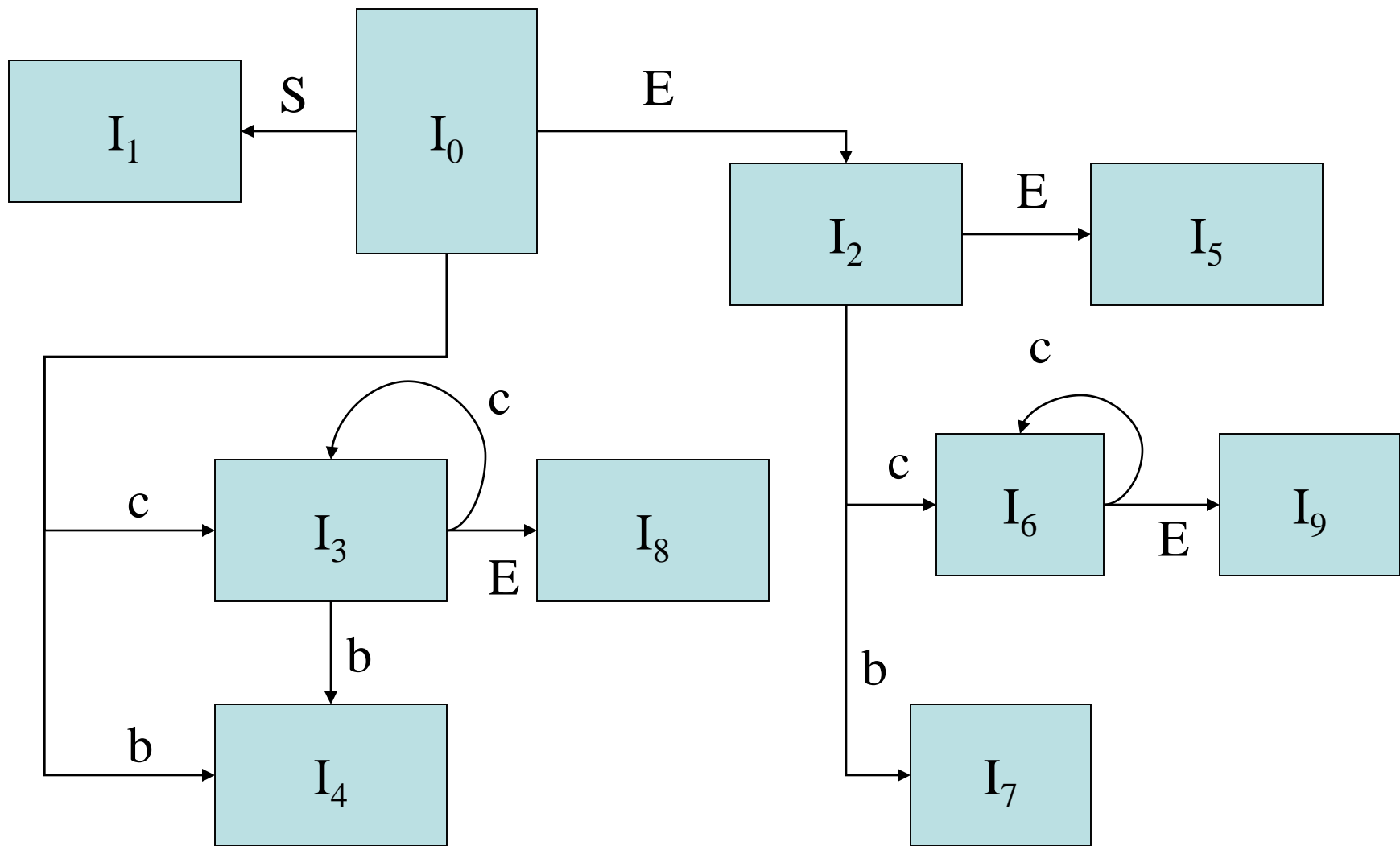
- a) GOTO( $I_2, E$ )  
 $I_5: \{ S \rightarrow EE \cdot, \$ \}$
- b) GOTO( $I_2, c$ )  
 $I_6: \{ E \rightarrow E \cdot E, \$ \}$   
 $E \rightarrow \cdot cE, \$$   
 $E \rightarrow \cdot b, \$ \}$
- c) GOTO( $I_2, b$ )  
 $I_7: \{ E \rightarrow b \cdot, \$ \}$

3.  $I_3$

- a) GOTO( $I_3, E$ )  
 $I_8: \{ E \rightarrow cE \cdot, c/b \}$
- b) GOTO( $I_3, c$ )  
 $I_3: \{ E \rightarrow c \cdot E, c/b \}$
- c) GOTO( $I_3, b$ )  
 $I_4: \{ E \rightarrow b \cdot, c/b \}$

4.  $I_6$

- a) GOTO( $I_6, E$ )  
 $I_9: \{ E \rightarrow cE \cdot, \$ \}$
- b) GOTO( $I_6, c$ )  
 $I_6: \{ E \rightarrow c \cdot E, \$ \}$
- c) GOTO( $I_6, b$ )  
 $I_7: \{ E \rightarrow b \cdot, \$ \}$



LR(1) **ITEM** 集合之Transition Diagram

# Canonical Parsing Table 建構的方法

1. 建立Augment Grammar  $G'$  的所有LR(1)的項目集合  $I_0 I_1 I_2 , \dots I_n$
2. 每一個LR(1)項目集合是唯一新的**State**;即將 $I_i$ 視為**State i**. 至於每個**state** 的動作則根據以下規則決定
  - 若 $[A \rightarrow \alpha \bullet a\beta]$  是在  $I_i$  中,而且 $GOTO(I_i, a) = I_j$ , 則  $ACTION[I, a] = \text{Shift } j$ . ( $a$  : Terminal Symbol)
  - 若 $[A \rightarrow B\bullet, a]$  是在  $I_i$  中,且 $A \neq S'$ ,則 $ACTION[i, a] = \text{"Reduce } A \rightarrow B"$ .
  - 若 $[S' \rightarrow S\bullet, \$]$ 是在  $I_i$  中,則令 $ACTION[i, \$] = \text{Accept}$ .

# Canonical Parsing Table 建構的方法

3. 對於Non-terminal Symbol  $A$ , 若有  $GOTO(I_i, A) = I_j$ , 則  $GOTO[i, A] = j$ .
4. 其他未加以定義的欄位則表示錯誤(Error).
5. 令包含有  $[S' \rightarrow \bullet S, \$]$  的 State 為起始狀態(Initial State).

## ■ Example

6. 對某一文法  $G$  所建構的  $LR(1)$  的 Parsing Table 中若沒有任一個欄位具有重複定義, 則此文法  $G$  為  $LR(1)$  文法
7. 每一  $SLR(1)$  文法是  $LR(1)$  文法; 即某一文法  $G$  若為  $SLR(1)$  文法 則必為  $LR(1)$  文法.

G:

$S \rightarrow EE$

$E \rightarrow cE|b$

$E \rightarrow b$

1.  $I_0: \{ S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot EE, \$$

$E \rightarrow \cdot cE, c/b$

$E \rightarrow \cdot b, c/b \}$

$\text{ACTION}[0, c] = \text{Shift } 3$

$\text{ACTION}[0, b] = \text{Shift } 4$

$\text{GOTO}(0, S) = 1$

$\text{GOTO}(0, E) = 2$

2.  $I_1$

$\text{ACTION}[1, \$] = \text{Accept}$

.....

自行練習

State	ACTION			GOTO	
	c	b	\$	S	E
0	S3	S4		1	2
1			Acc		
2	S6	S7			5
3	S3	S4			8
4	R3	R4			
5			R1		
6	S6	S7			9
7			R3		
8	R2	R2			
9			R2		

## LR(1) Parsing Table