

CLR(1) and LALR(1)

→ LR(0) and SLR(1) uses LR(0) items

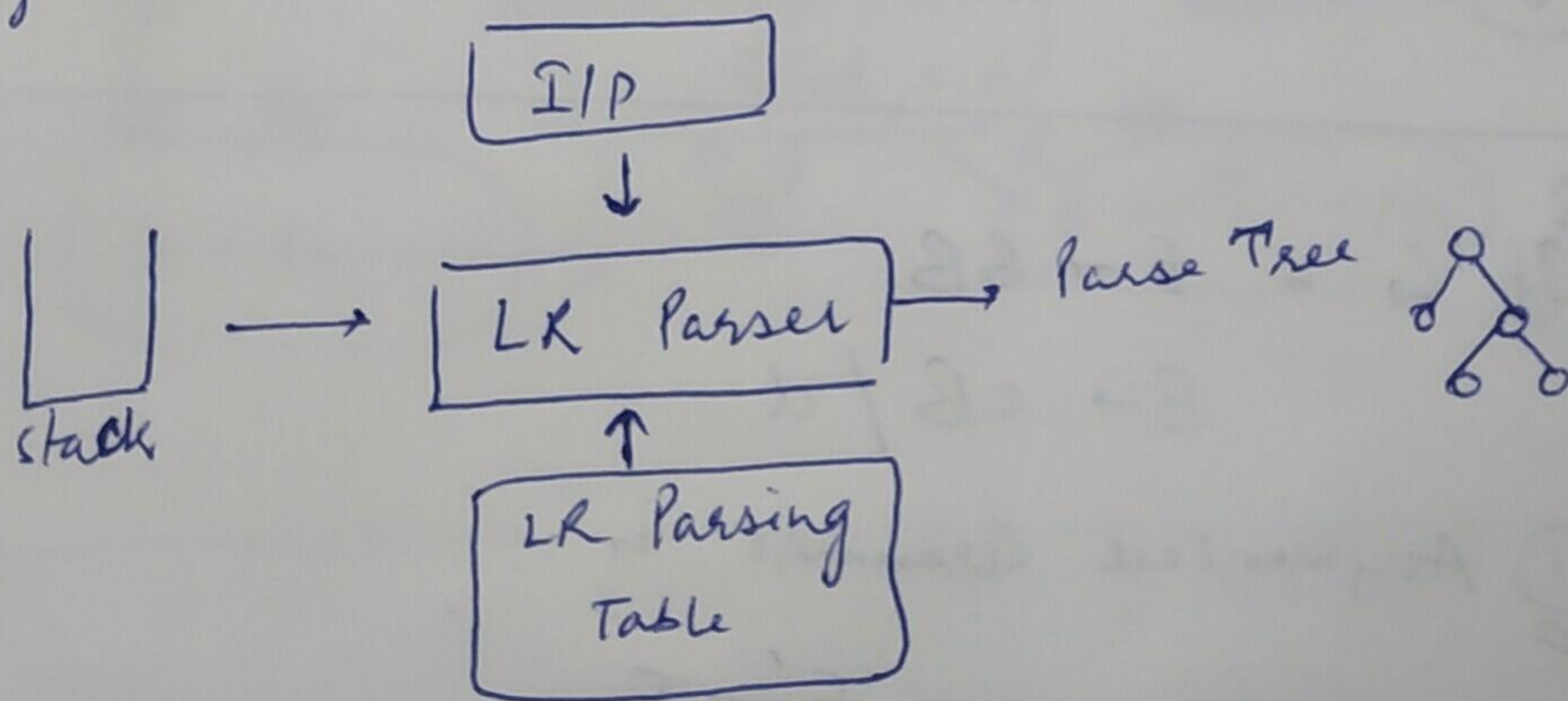
→ CLR(1) and LALR(1) uses LR(1) items

Major difference b/w LR(0) items and LR(1) items

✓ LR(0) items are productions with "." at any position in the body of ^{the} production.

✓ LR(1) items are LR(0) + Lookahead

LR Algorithm (revisited)



(using LR(0) item or LR(1) item)

Steps to solve LR(1) items

CLR(1)

②

- ① Augment the given Grammar
- ② Draw Canonical Collection of LR(1) items/
DFD/DFA
- ③ Number the Productions
- ④ Create Parsing Table
- ⑤ Stack Implementation
- ⑥ Draw Parse Tree

Ex

If G is $E \rightarrow BB$
 $B \rightarrow cB / d$

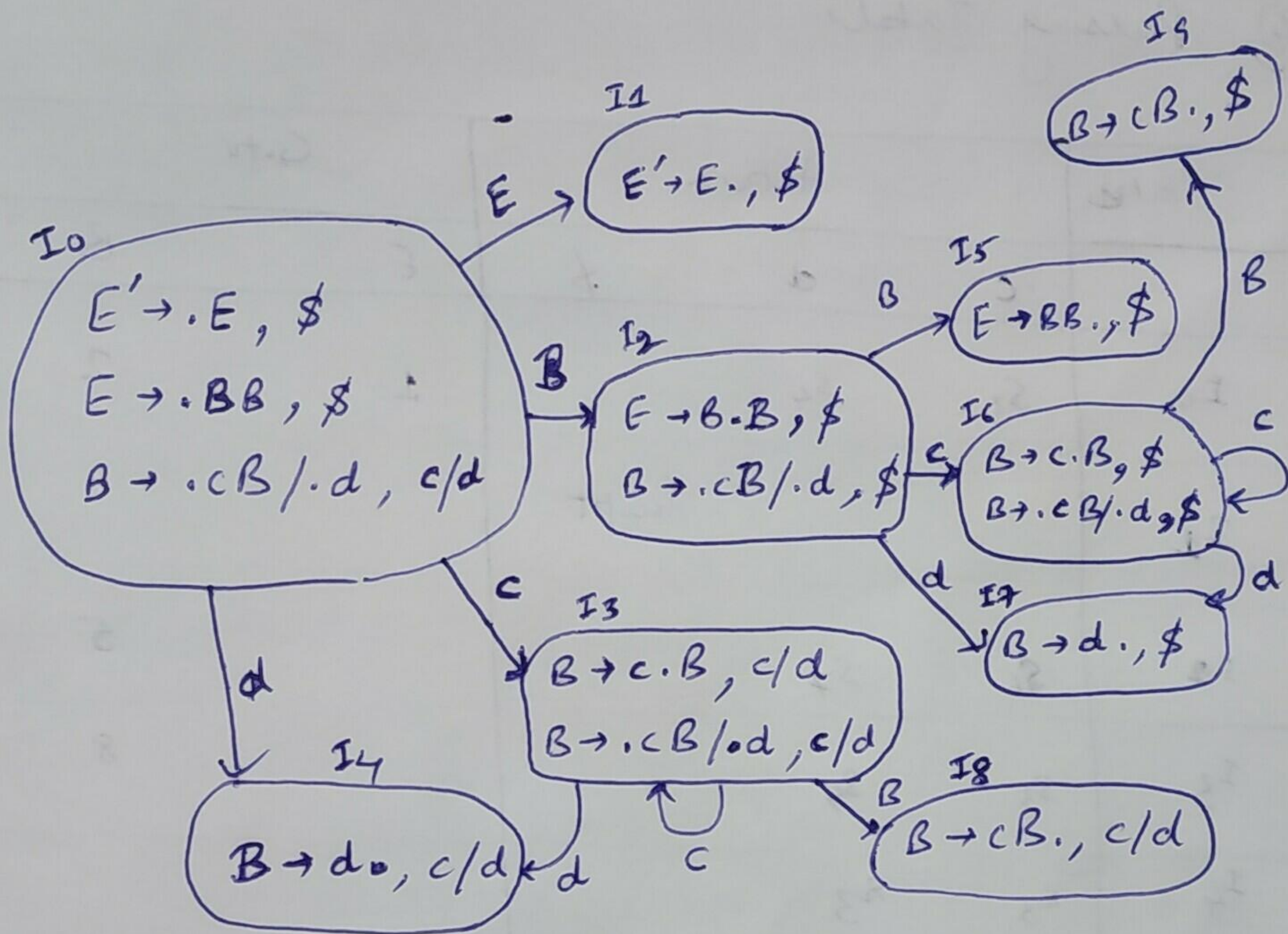
① Augmented Grammar is

$$E' \rightarrow E$$

$$E \rightarrow BB$$

$$B \rightarrow cB / d$$

② Draw Canonical Collection of LR(1) item/DFD (closures, goto, lookaheads)



* CLK(1) states will be more than SLR(1) states

- * No change in lookahead while goto
- * There will be change in the lookahead while closure.

③ Number the Production

④

$$E' \rightarrow E$$

$$E \rightarrow BB^{(1)}$$

$$B \rightarrow CB/d^{(2)} \quad B \rightarrow CB/d^{(3)}$$

④ Parsing Table

State	Action			Goto	
	c	d	\$	E	B
I ₀	S ₃	S ₄		1	2
I ₁			Accept		
I ₂	S ₆	S ₇			5
I ₃	S ₃	S ₄			8
I ₄	R ₃	R ₃			
I ₅			R ₁		
I ₆	S ₆	S ₇			9
I ₇			R ₃		
I ₈	R ₂	R ₂			
I ₉			R ₂		

LALR(1)

(5)

⇒ Same steps upto Parsing table, then modify it

Merge same type of production with different lookahead, and create a new state.

For ex:

$$I_3 + I_6 = I_{36}$$

$$I_4 + I_7 = I_{47}$$

$$I_8 + I_9 = I_{89}$$

State	Action			Cntr	
	c	d	⋄	E	B
I ₀	S ₃₆	S ₄₇	Accept	1	2
I ₁					
I ₂	S ₃₆	S ₄₇			5
I ₃₆	S ₃₆	S ₄₇			89
I ₄₇	(R ₃)	(R ₃)	(R ₃)		
I ₅			(R ₁)		
I₃₆	S₃₆	S₄₇			89
I₄₇			(R₃)		
I ₈₉	(R ₂)	(R ₂)	(R ₂)		
I₈₉			(R₂)		

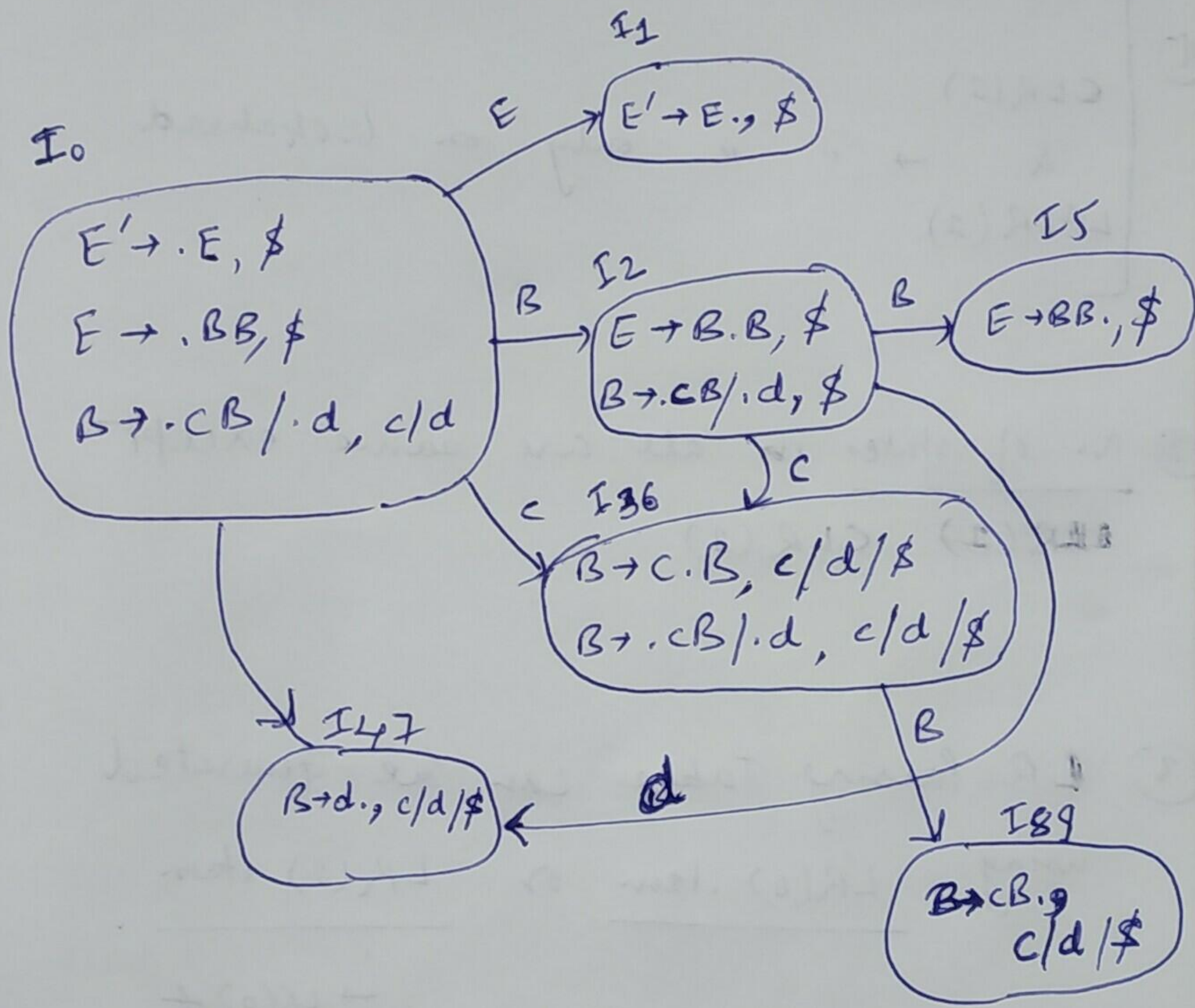
Rewrite the table

⑥

State	Action			Goto	
	c	d	\$	E	B
I ₀	S ₃₆	S ₄₇		1	2
I ₁			Accept		
I ₂	S ₃₆	S ₄₇			5
I ₃₆	S ₃₆	S ₄₇			89
I ₄₇	r ₃	r ₃	r ₃		
I ₅			r ₁		
I ₈₉	r ₂	r ₂	r ₂		

update the DFD

(7)



Summary of parsing type

8

- ①
- LR(0) → put reduce in full row
 - SLR(1) → " " " follow of P
 - CLR(1)
& → " " only on lookahead
 - LALR(1)

② No. of states in all are same except
~~LR(1)~~ CLR(1)

③ LR parsing Table can be generated
using LR(0) item or LR(1) item

↳ LR(0) +
lookahead

① Nothing at the end
put \$ for start symbol or other
symbols

② ~~NT~~ First of NT

(First of Remaining Part)