

CS & IT ENGINEERING

Compiler Design

Lexical & Syntax Analysis



Lecture No. 7

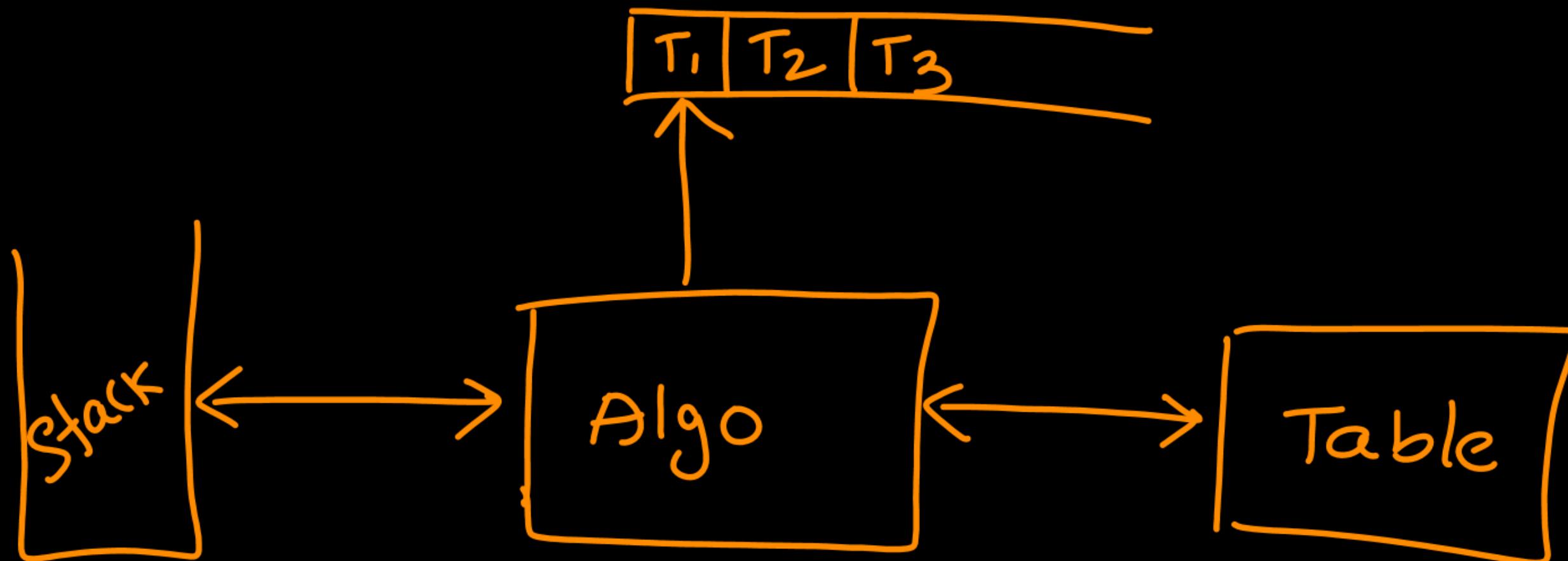


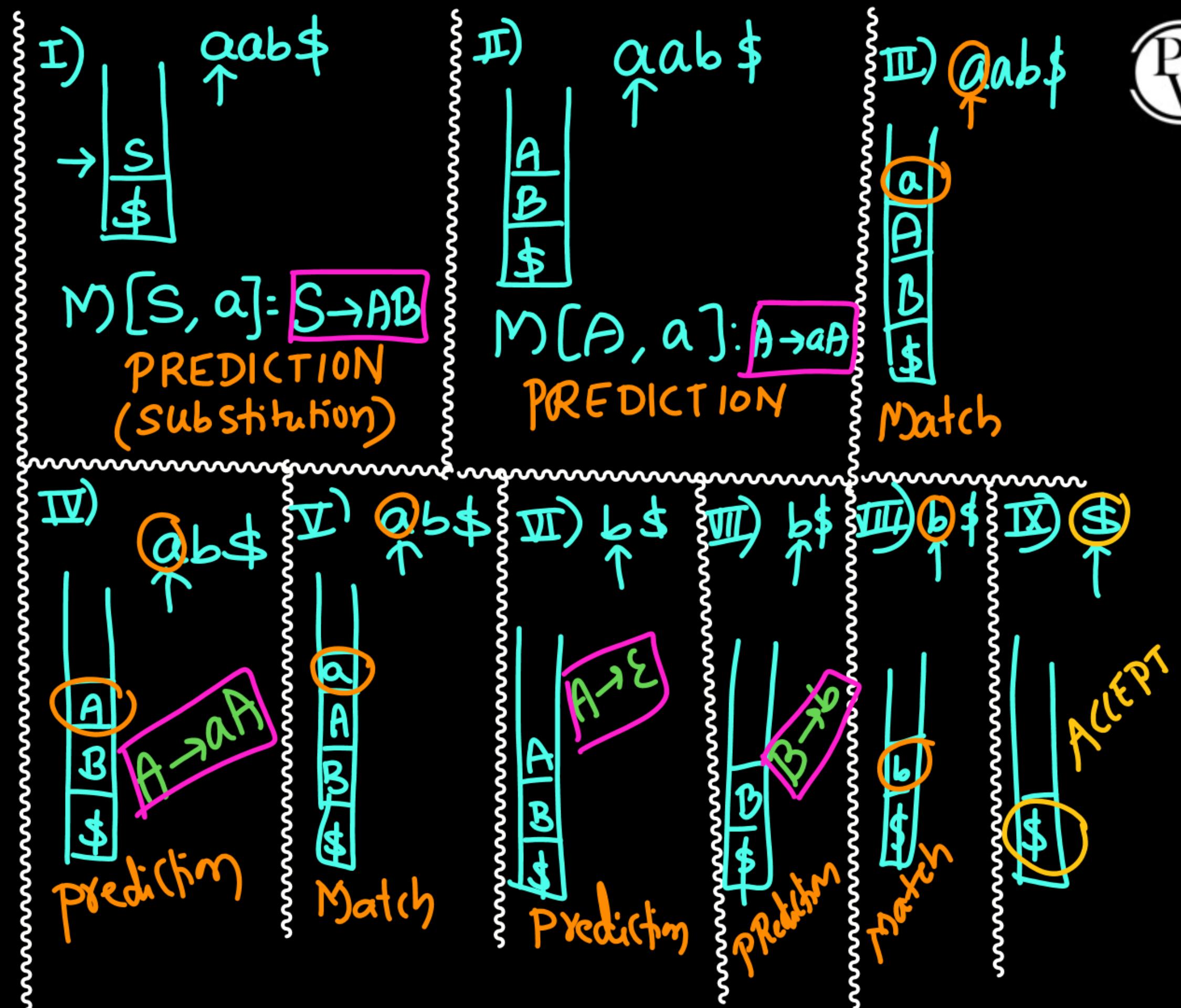
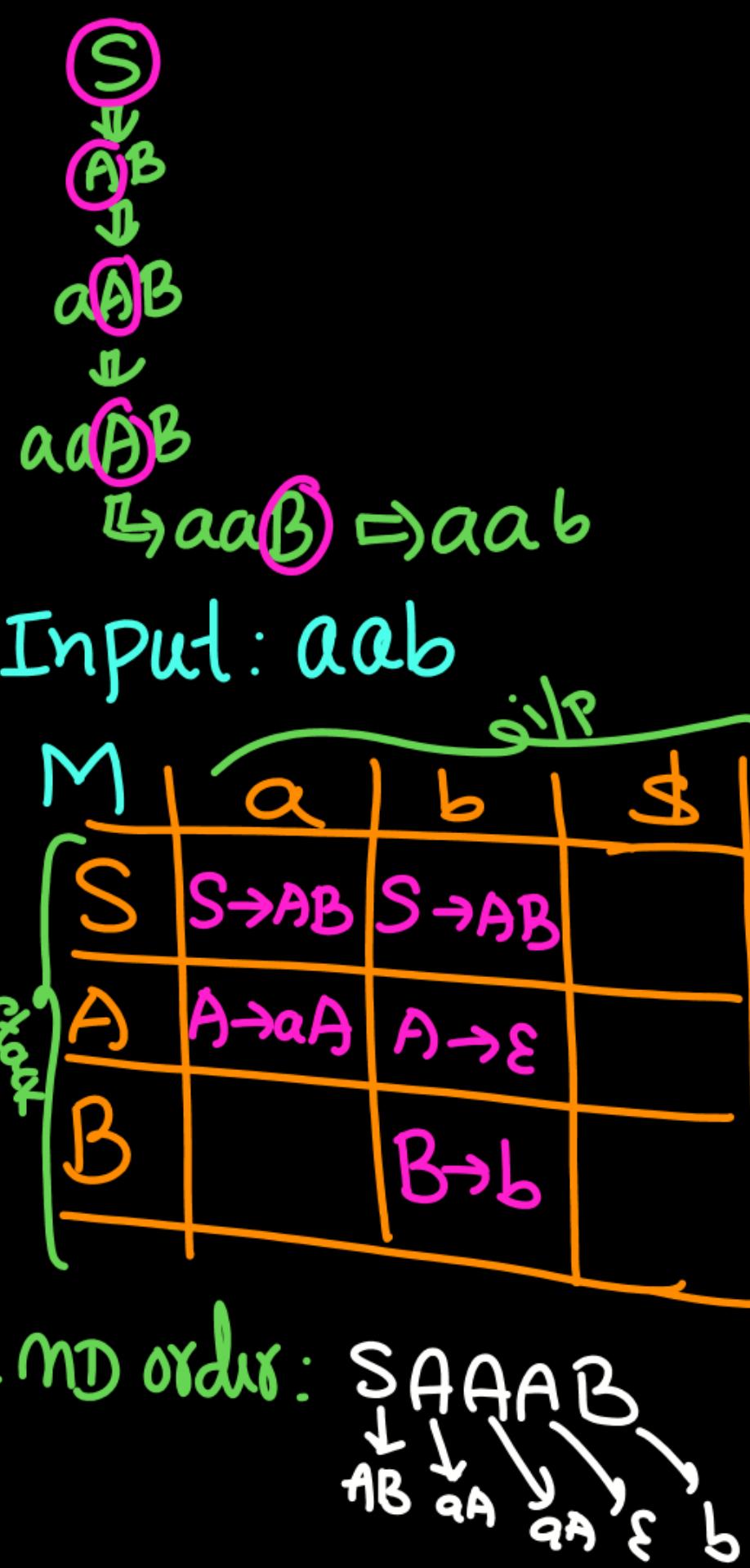
By- DEVA Sir



- 01 LL(0) Parsing Algo
- 02 LR Parsers
- 03 → Basics
- 04 → LR(0), SLR, LALR, CLR
- 05

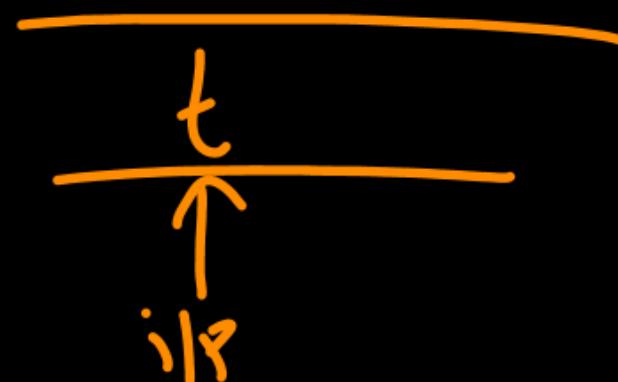
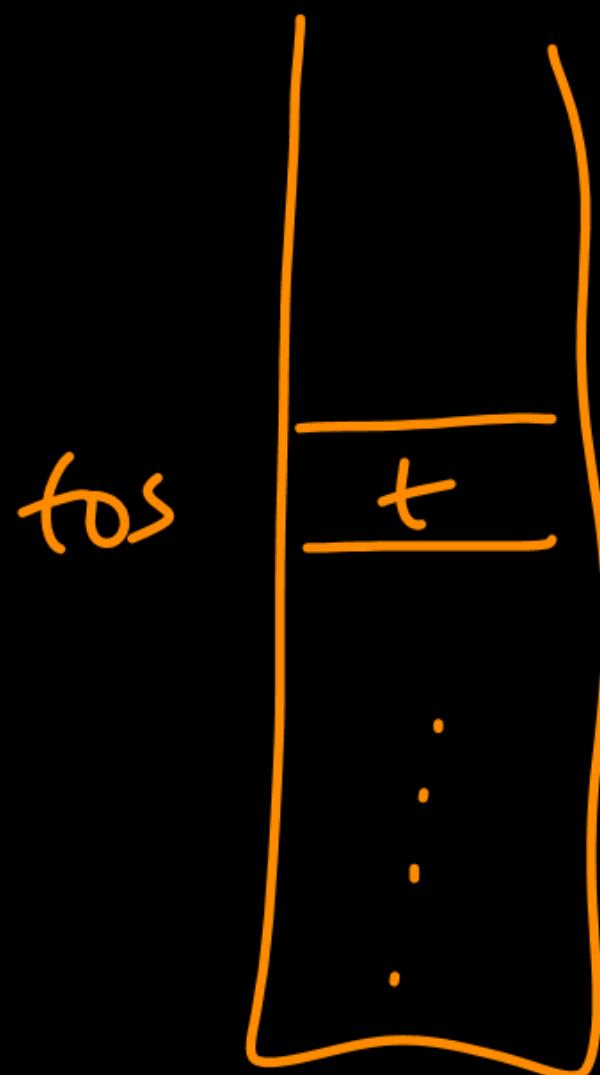
LL(1) Parser





$$A \rightarrow \epsilon$$

- i) POP A
- ii) PUSH ϵ
PUSH nothing



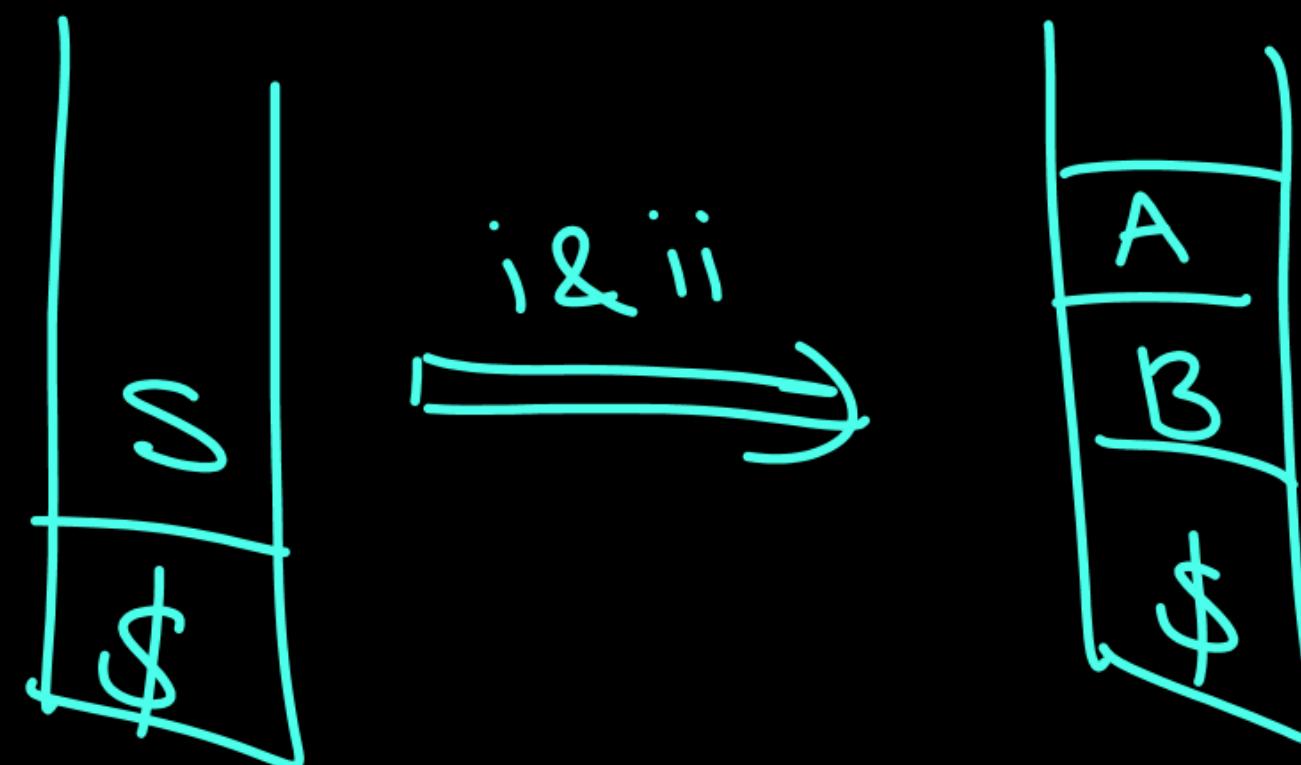
$(tos == i/p) \neq \$$

- i) Pop t
- ii) Inc i/p pointer

PREDICTION

$M[S, a] : S \rightarrow \underbrace{AB}_{\leftarrow}$

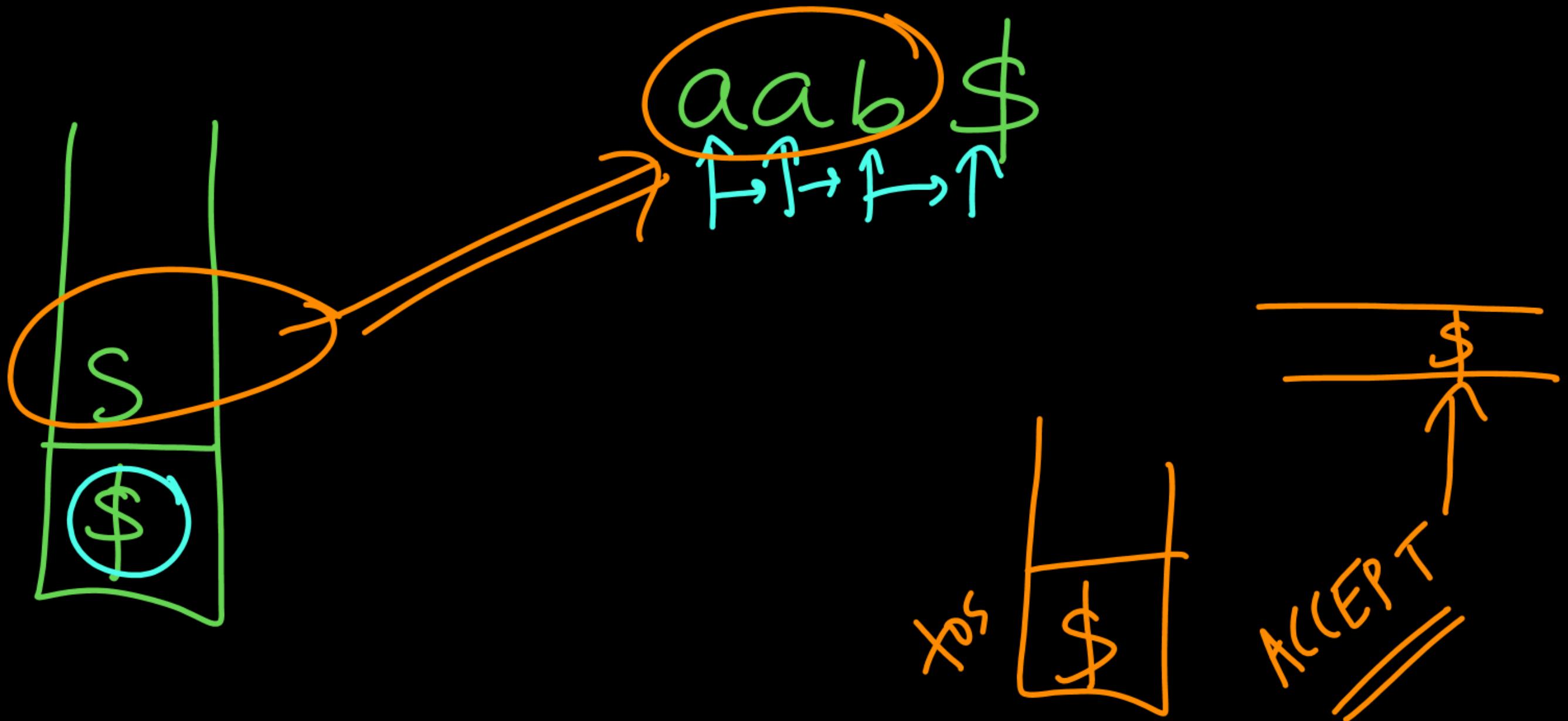
- i) POP LHS
- ii) PUSH RHS in reverse order



$$M[X, t] : X \rightarrow \alpha$$

i) POP ^{LHS} X

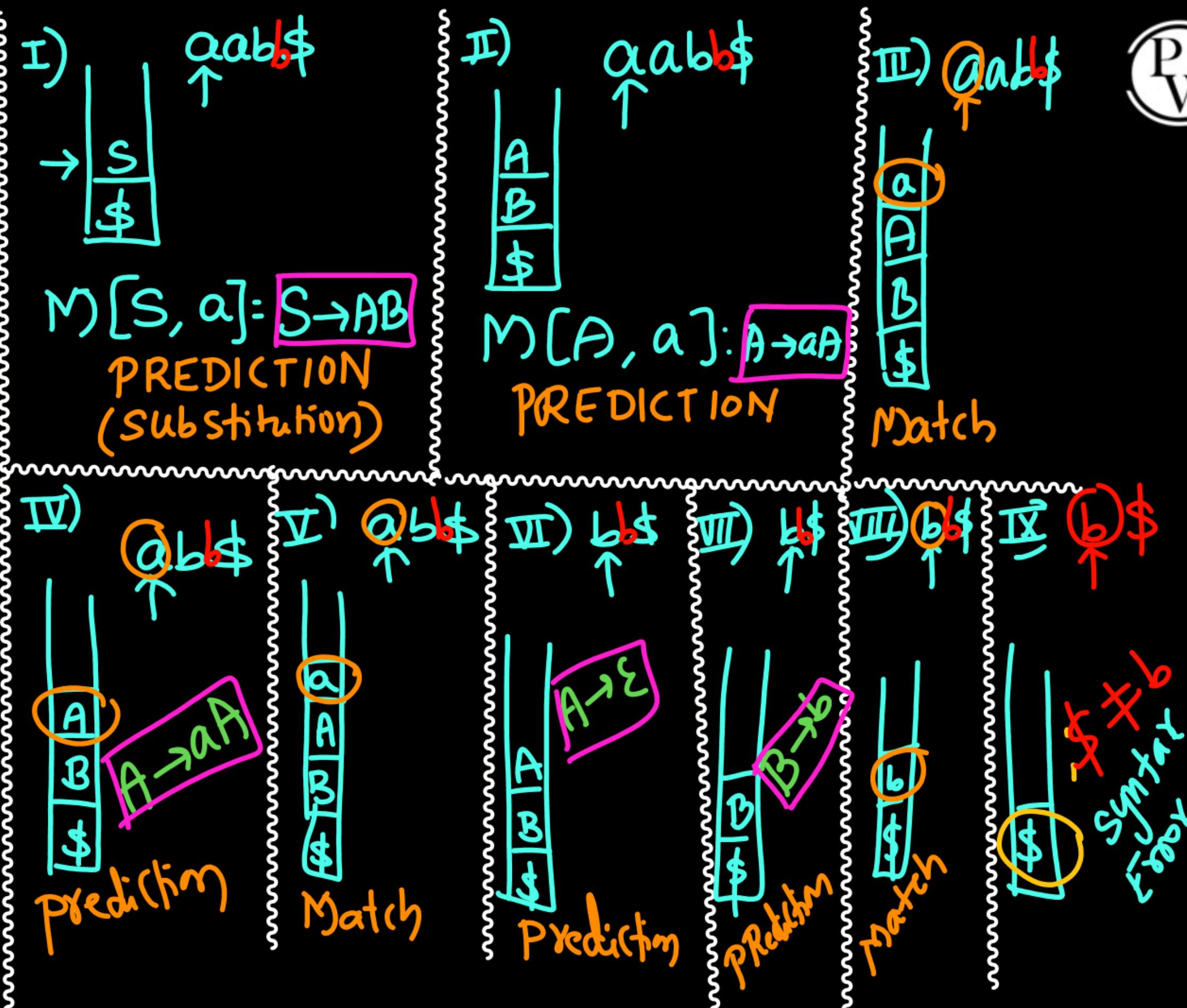
ii) Push α _{RHS} in reverse order



I): $aabb\$$

Input: aab

		a	b	$\$$
		$S \rightarrow AB$	$S \rightarrow AB$	
S	A	$A \rightarrow aA$	$A \rightarrow \epsilon$	
	B		$B \rightarrow b$	



I) T-conflict [Terminal conflict]

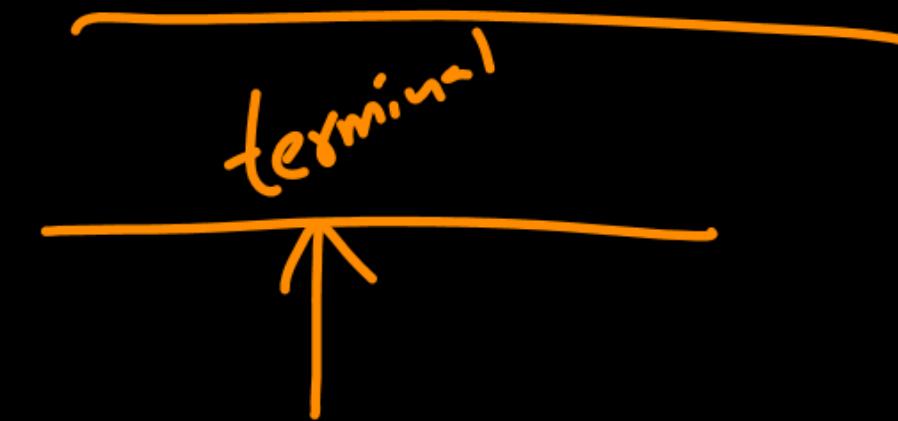
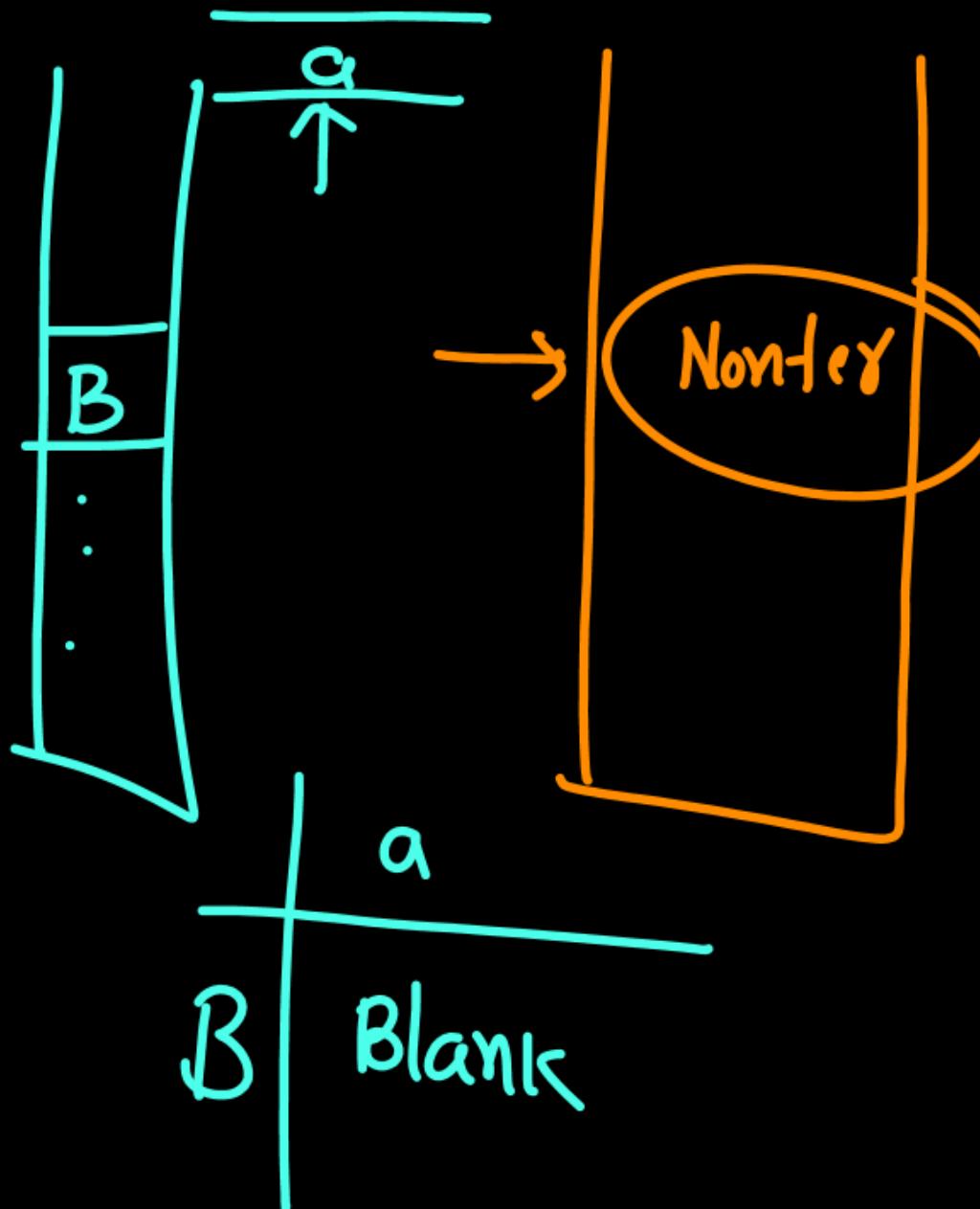
If tos is terminal

and not matching with i/p



T-conflict

II) N-conflict [Non-terminal Conflict]



If tos is Non-terminal
and there is no production
in table
↓
N-conflict

LL(1) Parsing Errors



- I) T-conflict ($\xrightarrow{\text{terminal}} \text{tos} \neq i/p$)
- II) N-conflict ($M[\text{tos}, i/p]$ is blank)

LL(1) parser

- First & Follow Set ✓
- LL(1) Table construction ✓
- Identify LL(1) ✓

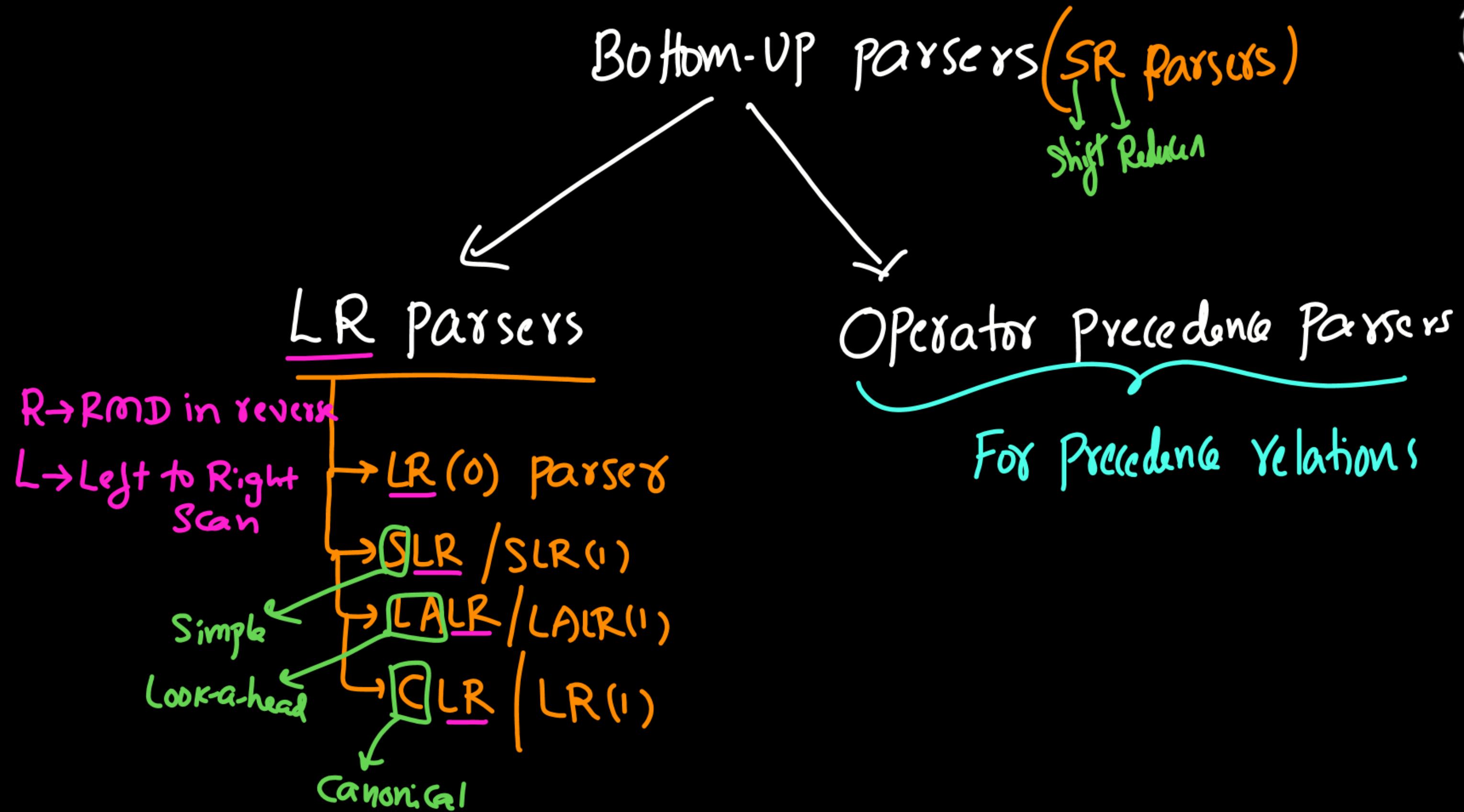
LL(0) \hookrightarrow G vs LL(1) vs LL(2) vs LL(3) vs ...

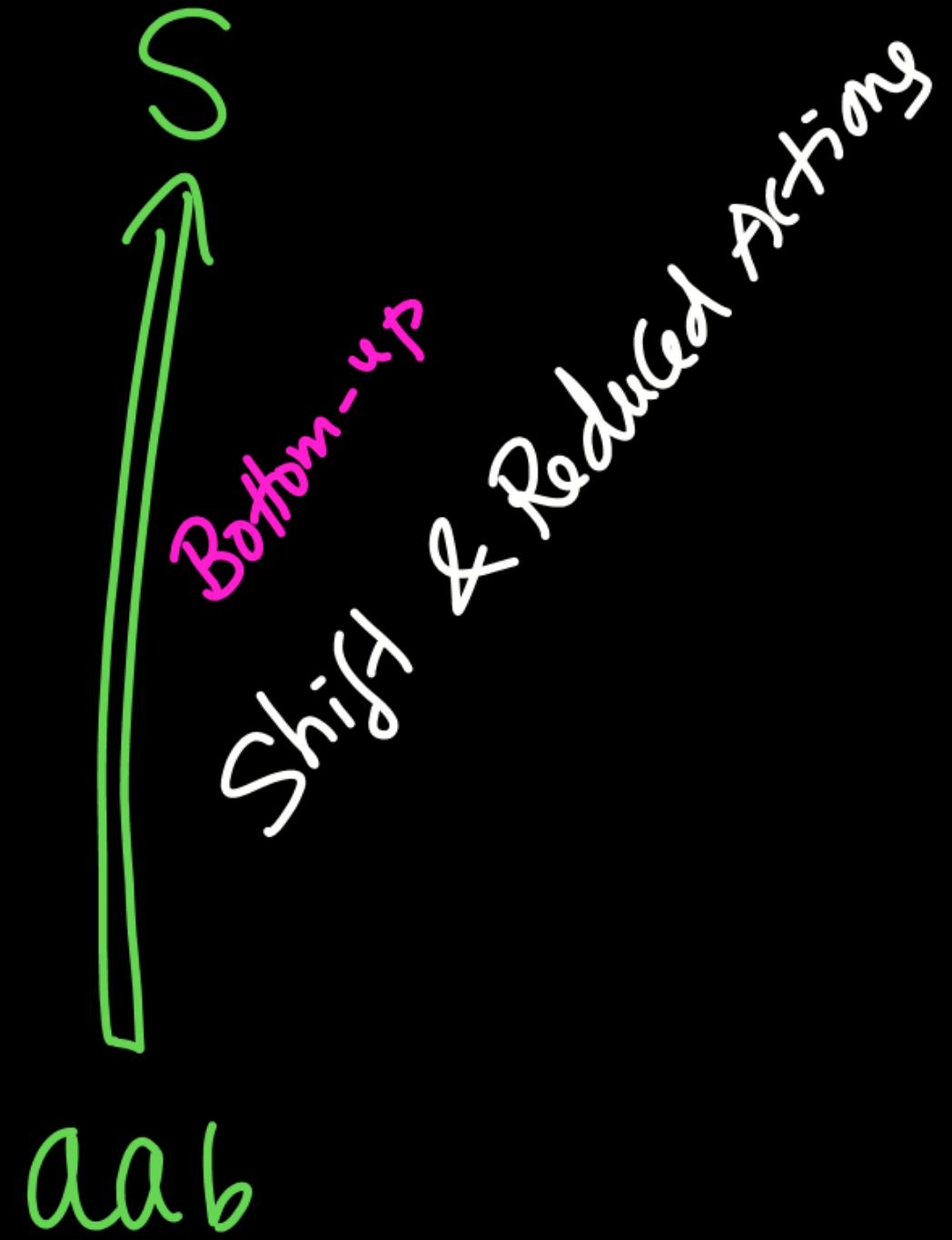
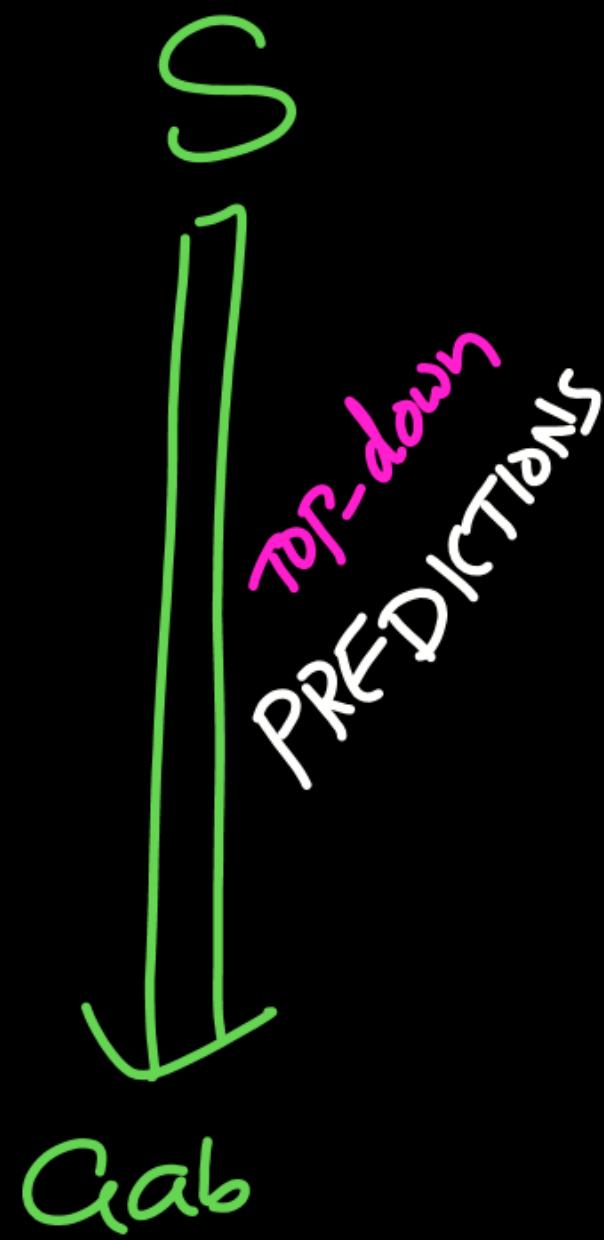
1) $S \rightarrow a$ ✓ ✓ ✓ ✓

2) $S \rightarrow aA$ ✓ ✓ ✓ ✓
 $A \rightarrow b$

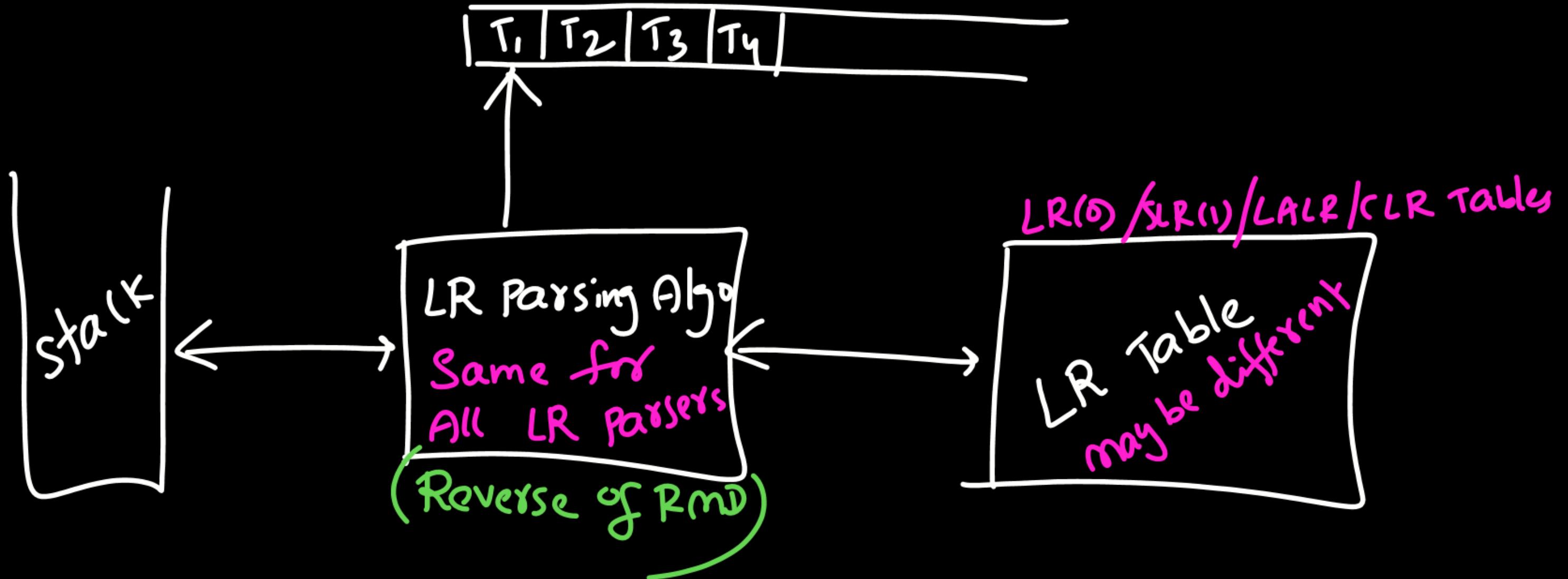
3) $S \rightarrow aA$ ✓ ✓ ✓ ✓
 $A \rightarrow bB$
 $B \rightarrow c$

	LL(0)	LL(1)	LL(2)	LL(3)
1) $S \rightarrow a b$	✗	✓	✓	✓
2) $S \rightarrow @ ab$	✗	✗	✓	✓
3) $S \rightarrow a$	✓	✓	✓	✓
4) $S \rightarrow ab abc gh$	✗	✗	✗	✓





LR Parser



LR(0) Parsing DFA

LR(0) Grammar Identification

SLR(1) .. "

} LR(0) Items
Required

~~~~~  
LR(1) Parsing DFA

LR(1) Grammar Identification

} LR(1) Items  
Required

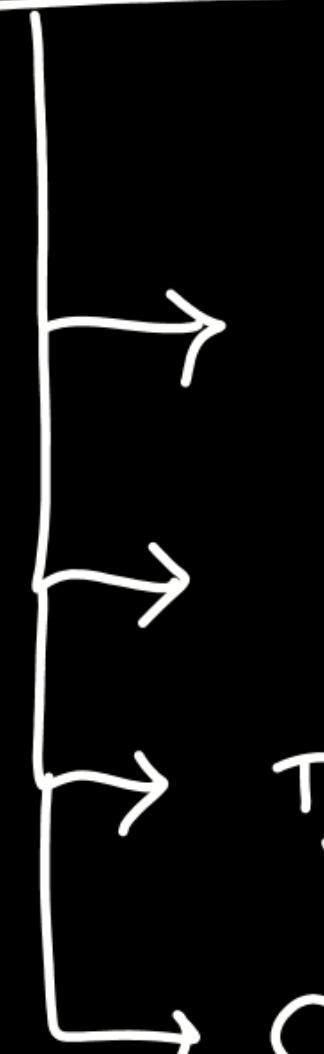
LALR(1) Parsing DFA

LALR(1) Grammar Identification

~~~~~  
LR(0) Table \rightarrow SLR Table \rightarrow LALR Table
 \rightarrow CLR Table

↓ LR Parsing Algo

Basics



Augmented CFG

Item

Types of Items

Closure()

and

goto()

} Helps to construct DFA

Augmented CFG

P
W

$$S \rightarrow S_a | S_b | c$$

CFG

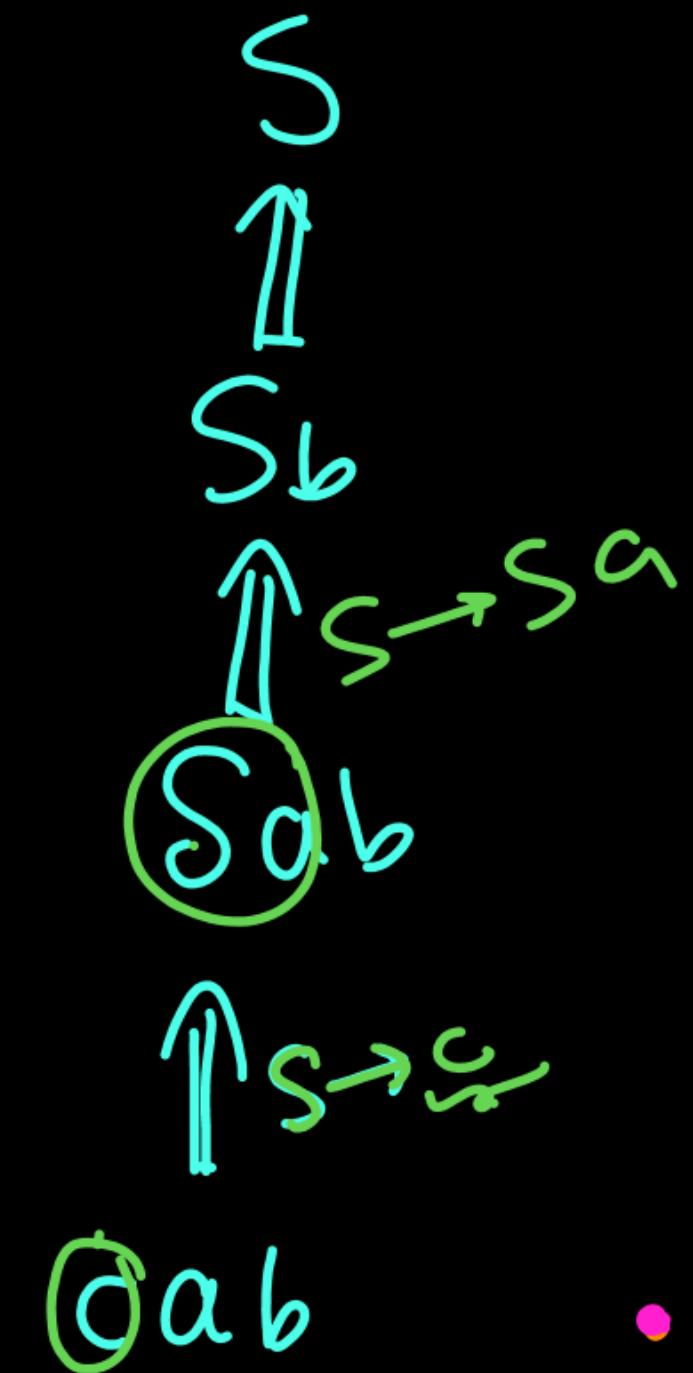
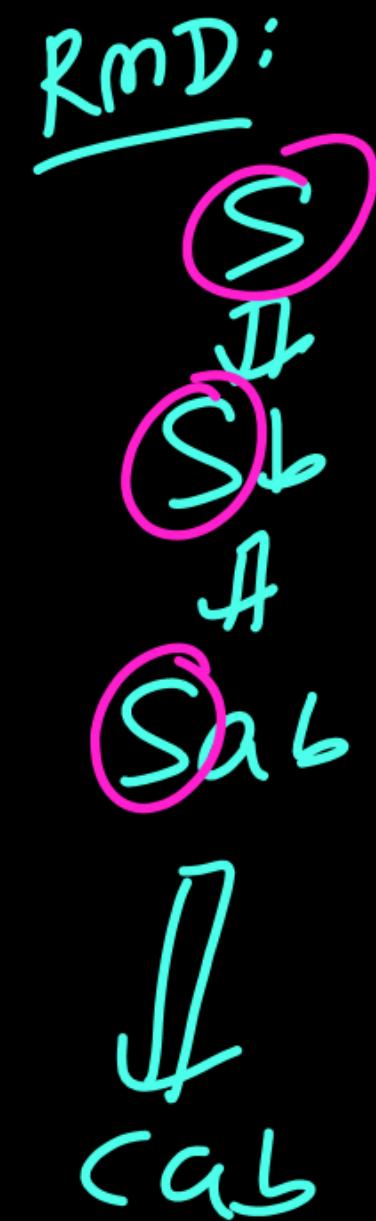
Append $S \rightarrow S$

$$\begin{array}{l} S \rightarrow S \\ S \rightarrow S_a | S_b | c \end{array}$$

Augmented CFG

$$S \rightarrow S_a | S_b | c$$

I/P: cab



• cab \Rightarrow @ab
 move dot
 Shift



Why we need a DOT ?



To perform Shift & Reduced Action

To keep track of progress

Why we need a symbol which is immediately after DOT ?

$$X \rightarrow \alpha . \boxed{\text{symbol}} \beta$$

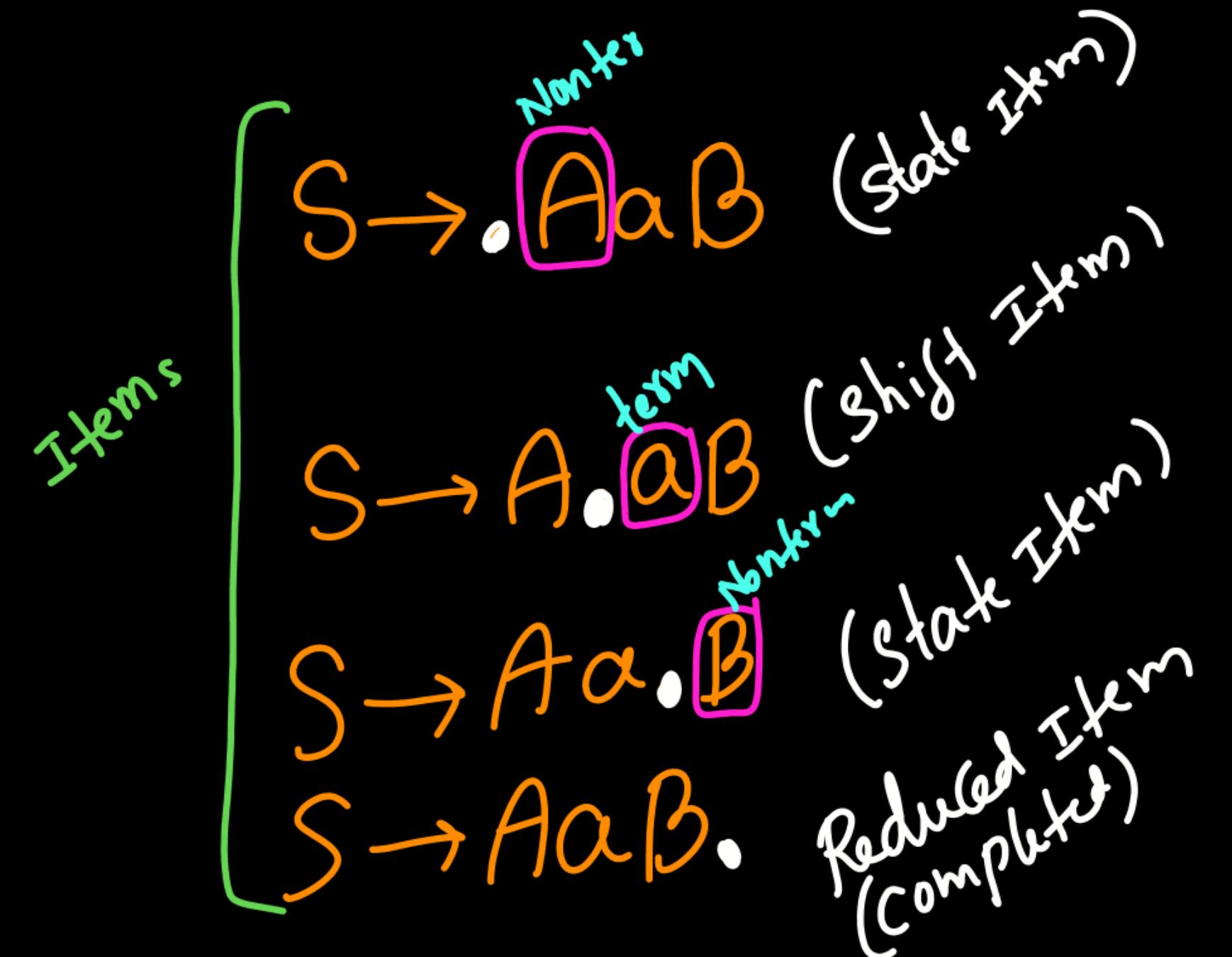
Item

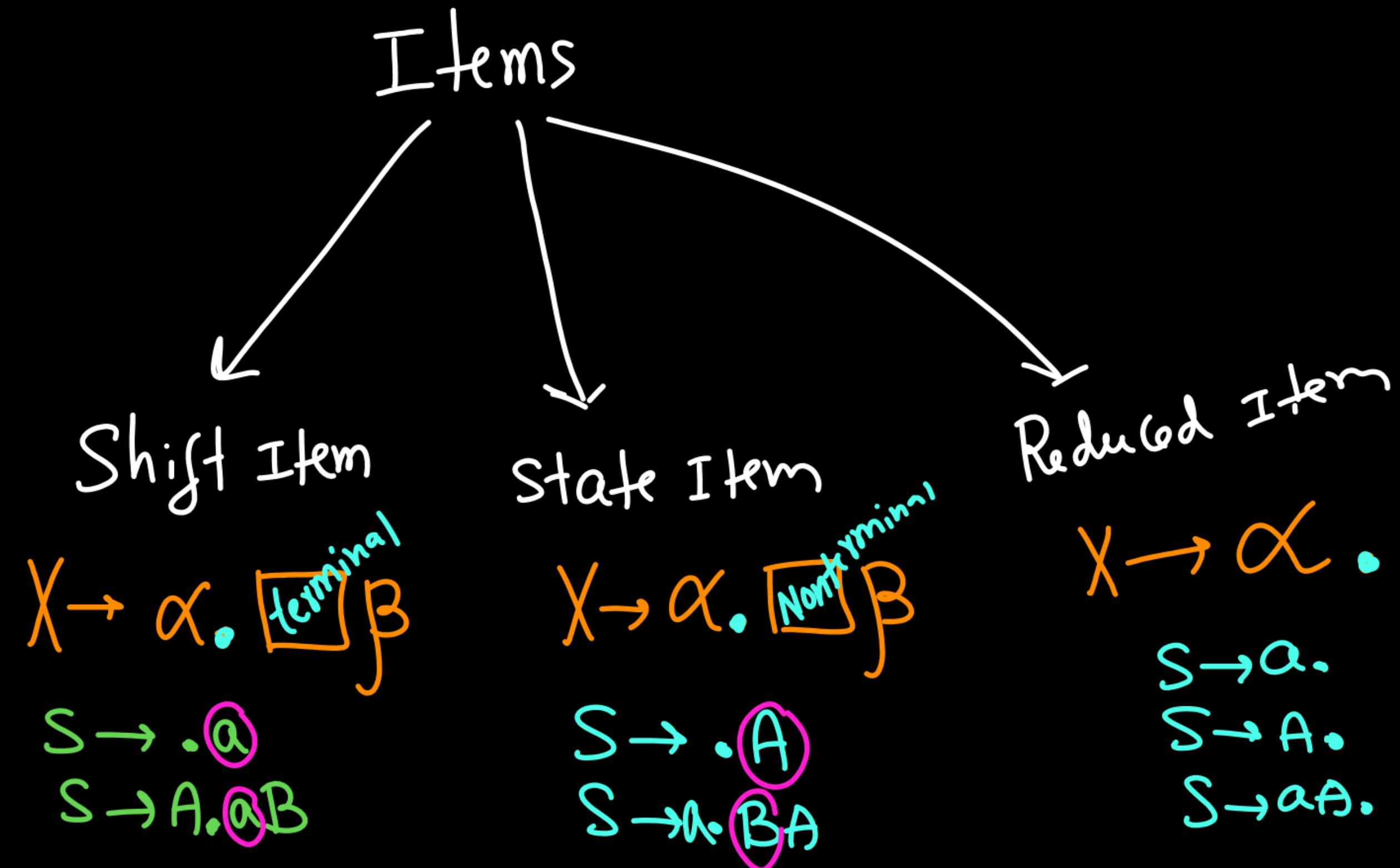
P
W

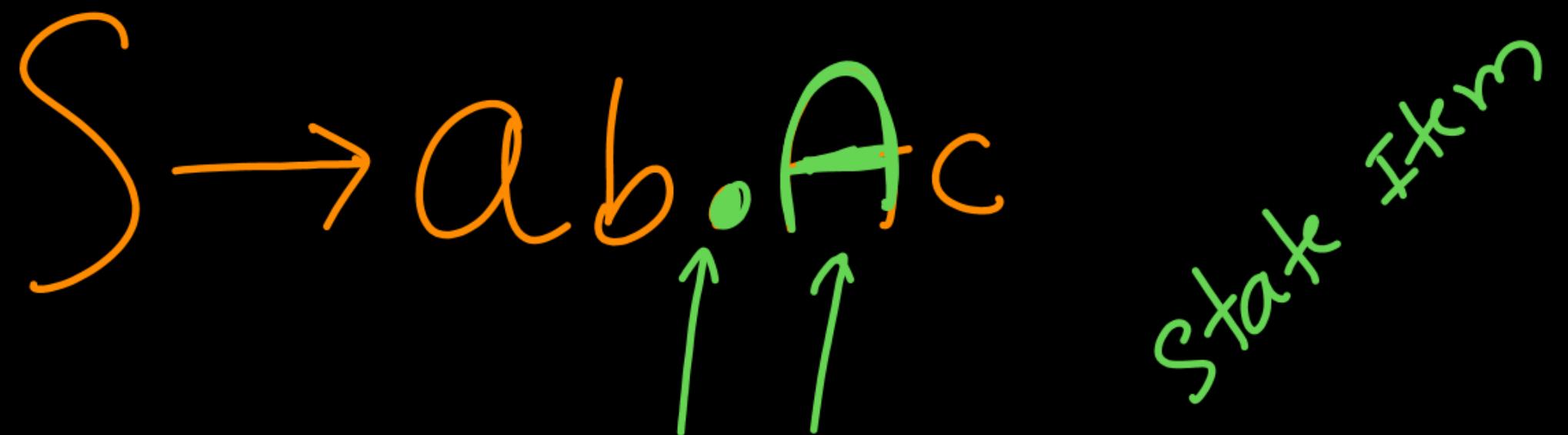
→ It is a production which has a dot.

$$S \rightarrow AaB$$

production







→ LL(1) Algo ✓

→ Augmented CFG , Item , type of Icms

