

CS & IT Engineering



Compiler Design

Lexical Analysis and Syntax Analysis

Lecture: 8



Deva sir

Topics to be covered:

- $LL(0)$ vs $LL(1)$ vs $LL(2)$ vs ...
- Bottom-up parser
 - LR(0) parser
 - SLR parser
 - LALR and CLR parser
 - Operator precedence parsing

$$\textcircled{1} \quad S \rightarrow a$$

LL(0) CFG ✓

[LL(1), LL(2), ...]

$$\textcircled{2} \quad S \rightarrow aA$$

$$A \rightarrow bB$$

$$B \rightarrow c$$

LL(0) ✓

[LL(1), LL(2), ...]

③ $S \rightarrow \underline{a} \mid \underline{b}$

LL(0) X

LL(1) ✓

[LL(2), LL(3), ...]

④ $S \rightarrow \boxed{a} \mid \boxed{ab}$

LL(0) X

LL(1) X

LL(2) ✓

a\$

ab...

⑤ $S \rightarrow \boxed{a} \mid \boxed{f} \mid \boxed{abc} \mid \boxed{abcde}$

4 length prefix has no problem

3 length common

LL(0) X

LL(1) X

LL(2) X

LL(3) X

LL(4) ✓

Note:

I) Every $LL(k)$ CFG is $LL(k+1)$ CFG

II) $\left. \begin{array}{l} \text{Every } LL(2) \text{ CFG is convertible to } LL(1) \text{ CFG} \\ \text{Every } LL(3) \text{ " " " " } LL(1) \text{ CFG} \\ \text{" } LL(4) \text{ " " " " } LL(1) \text{ CFG.} \end{array} \right\}$

III) Set of languages generated by $LL(1)$ CFGs

Set of languages $\stackrel{=}{=}$ generated by $LL(2)$ CFGs

TDP

- 1) Predictive parser
(LL(1) parser)
- 2) predictions
(LR(0) substitutions)
- 3) Uses LR(0) "to verify
syntax" (to produce parse tree)
(to derive input)

BRP

- 1) LR parser
- 2) Shift and Reduced
Action
- 3) Uses "Reverse of LR(0)"

LR(0) Parser:

~~~~~ No look-ahead  
 ↳ RMD in reverse  
 ↳ Left to Right Scanning

1) LR(0) parsing Diagram  
[LR(0) DFA]

2) Conflicts checking for LR(0) CFG

- ↳ SR conflict
- ↳ RR conflict

3) LR(0) Table construction



# LR(0) DFA :

- ① Augmented CFG
- ②  $LR(0)$  Item and types of items
- \*\*\* ③ closure() and goto() functions
- ④ How to construct LR(0) DFA?
- ⑤ How to check SR and RR conflicts in LR(0)?



What is Augmented CFG?


$$\begin{array}{l} S \rightarrow Sa | Sb | Aa \\ A \rightarrow b \end{array}$$

CFG

Append  $S'$   $\nearrow S$

$\Rightarrow$

$S'$  is new  
Start  
in Augmented  
CFG

$$\begin{array}{l} S' \rightarrow S \\ S \rightarrow Sa | Sb | Aa \\ A \rightarrow b \end{array}$$

Augmented CFG

Why we add  $S' \rightarrow S$  in CFG?

How to accept given input?

$S' \rightarrow \bullet S$

$\Downarrow$   
 $S' \rightarrow S \bullet$   
Acceptance item

$\uparrow$   
 $S$   
Bottom-up parsing  
 $ab$

$S \bullet$   
 $\uparrow$   $S \rightarrow AB$  Reduce  
 $AB \bullet$   
 $\uparrow$   $B \rightarrow b$  Reduced Action

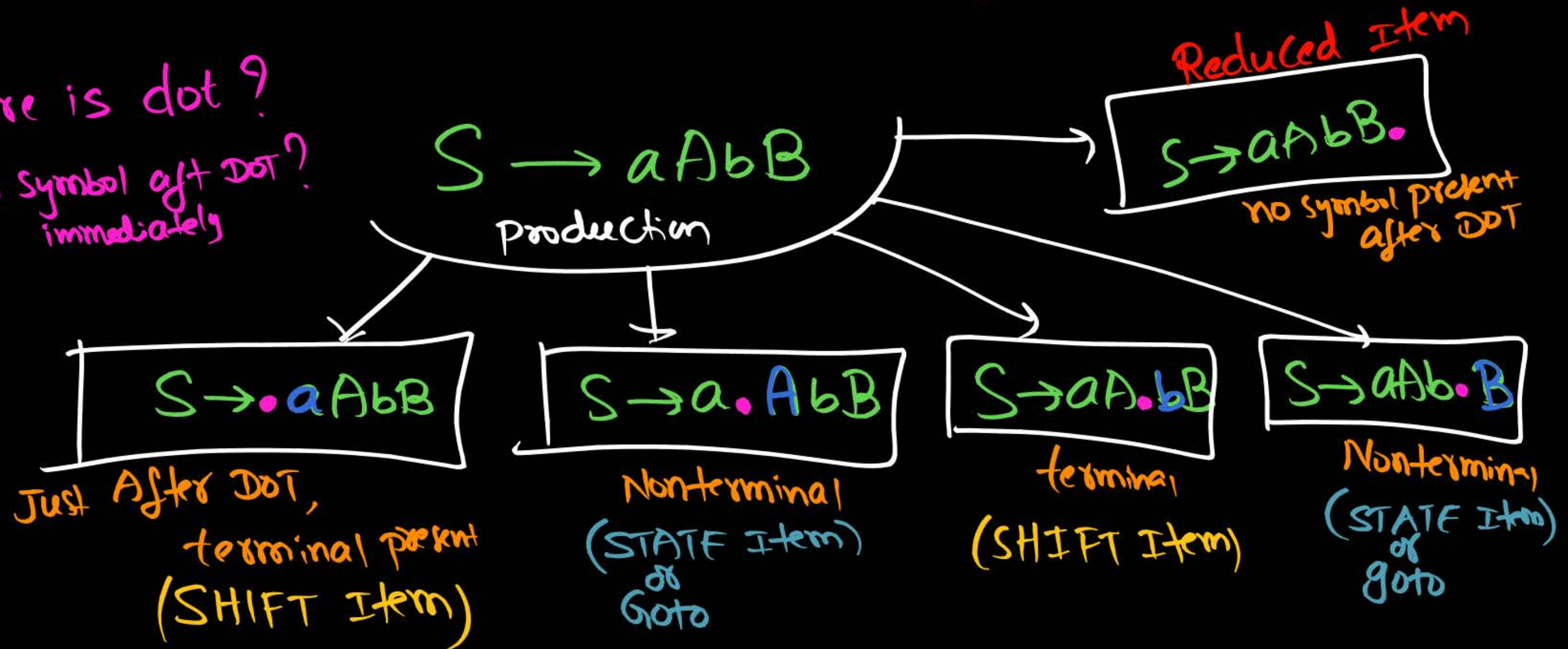
$A \bullet b$  Shift  $A \underline{b} \bullet$   
Handle  
Reduce  $\uparrow$   $A \rightarrow a$   
 $a$  is reduced to  $A$   
 $\bullet ab$  Shift  $\underline{a} \bullet b$   
Handle

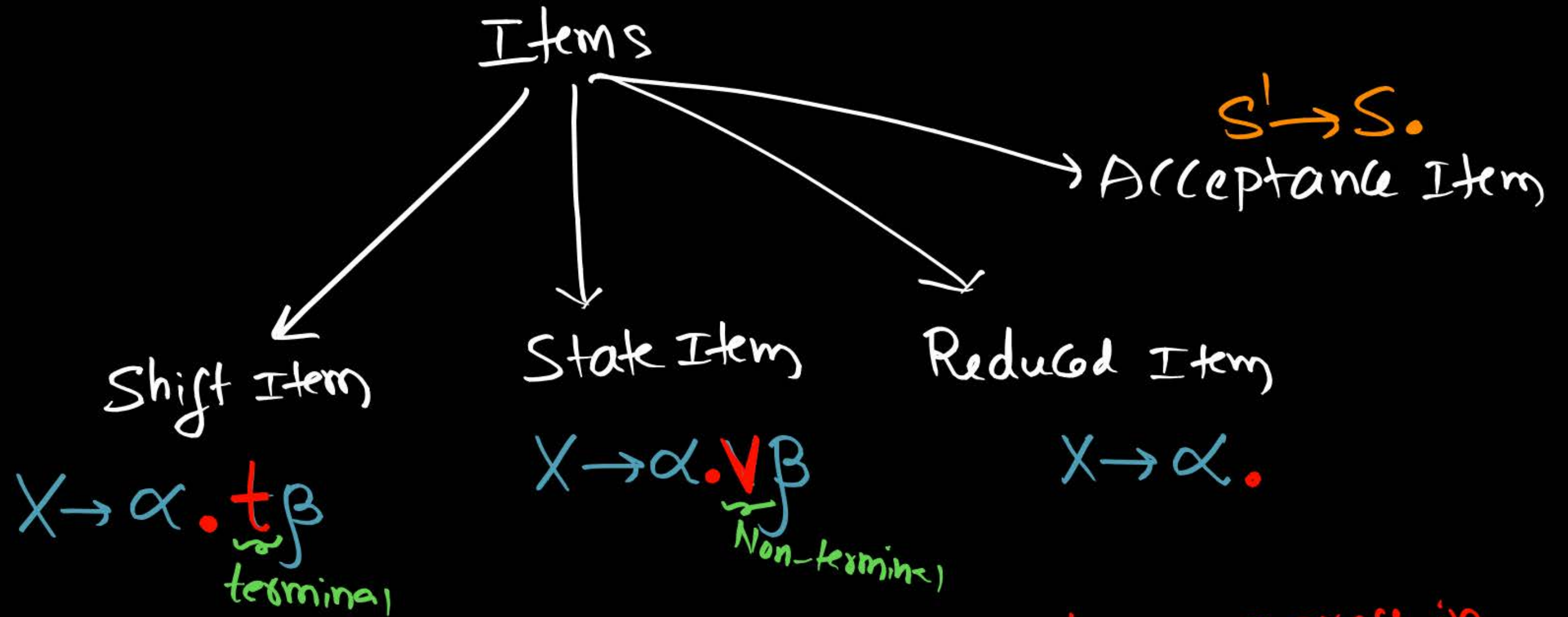


# LR(0) Item?

↳ It is a production along with a DOT.

Where is dot?  
Which symbol aft DOT?  
immediately





Note: DOT is used to track progress in derivation  
 (to perform Shift and Reduced Actions)



Note :

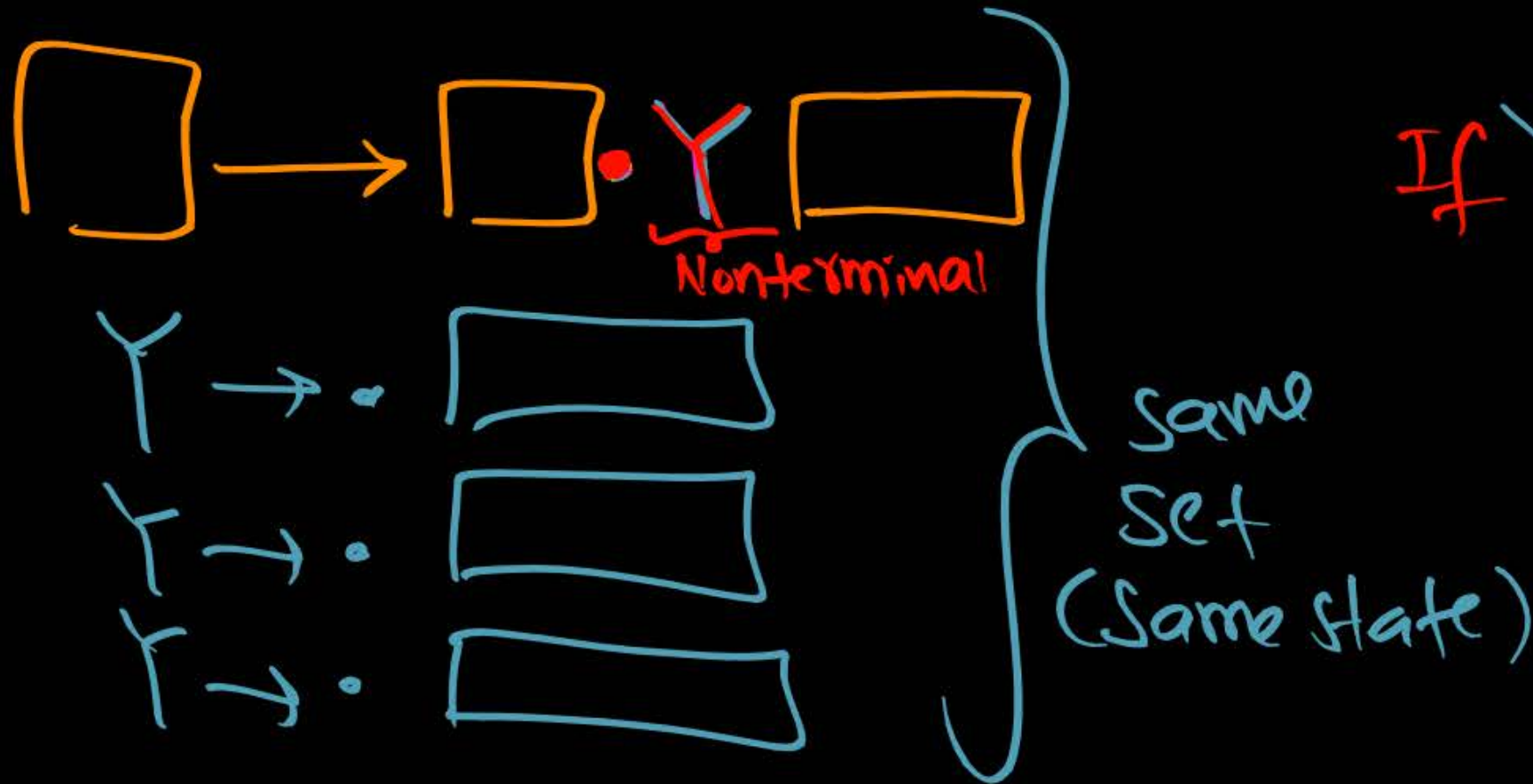
We must focus on **DOT**

and **NEXT symbol after DOT**.

Note:

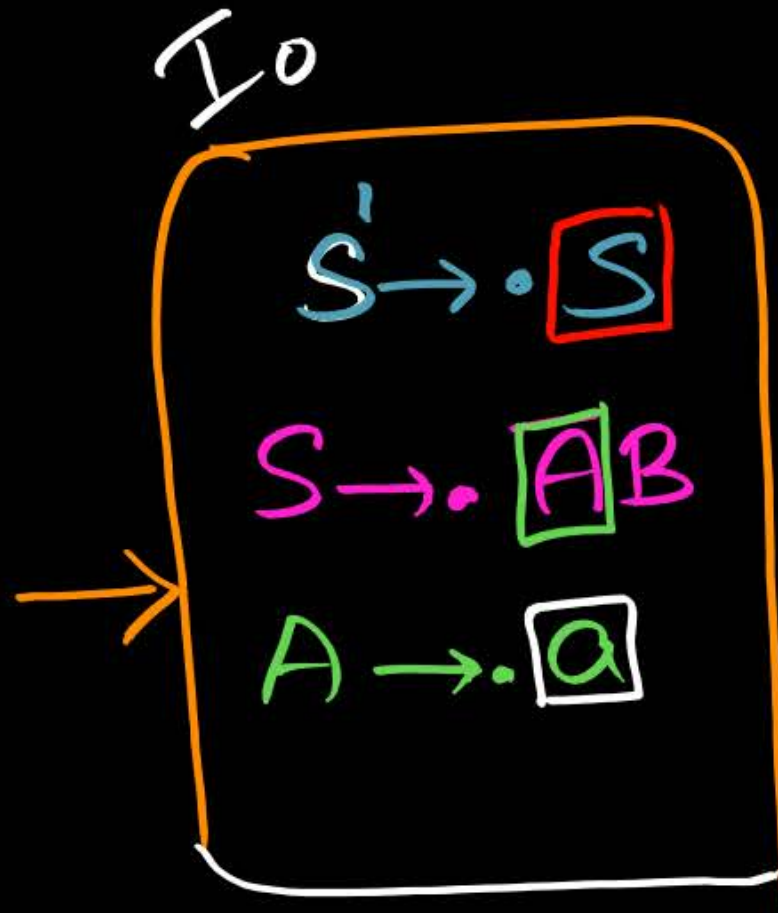
Closure(): If Nonterminal present just after DOT then we must add all productions in the same state.

→ Recursive function  
→ Applied on every item in state.



$S' \rightarrow S$ 
 $S \rightarrow AB$ 
 $A \rightarrow a$ 
 $B \rightarrow b$ 

Augment CFG



Let  $X = \text{closure}(S \rightarrow A \cdot B)$ .

Then  $|X| = ?$

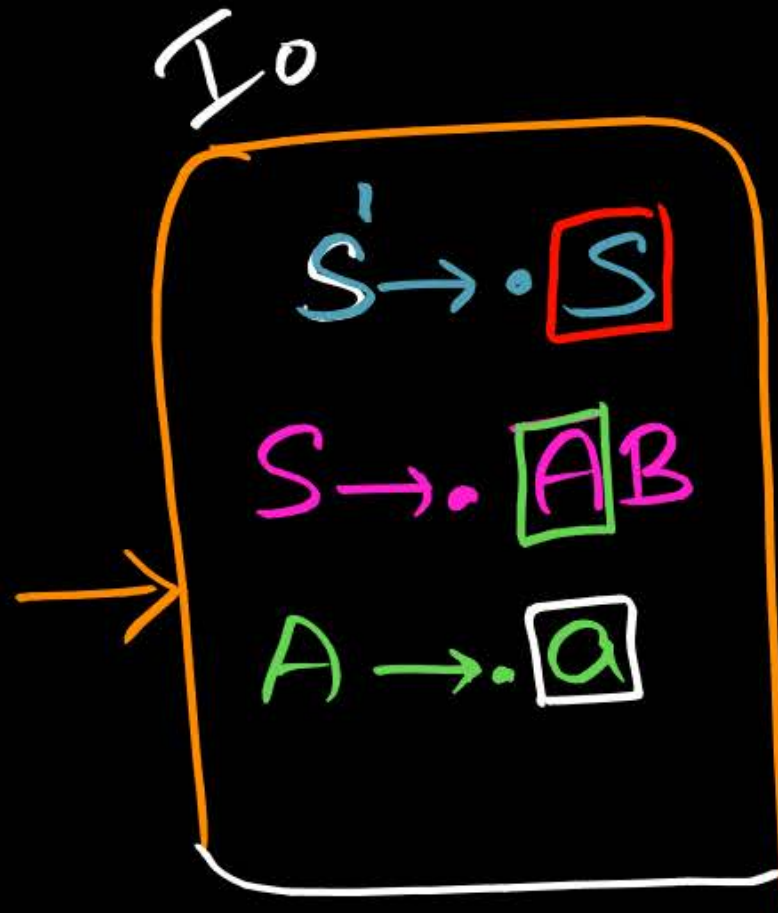
$X = \{S \rightarrow A \cdot B, B \rightarrow \cdot b\}$

$S \rightarrow A \cdot \boxed{B}$   
 $B \rightarrow \cdot \boxed{b}$

$|X| = 2 //$

$S' \rightarrow S$ 
 $S \rightarrow AB$ 
 $A \rightarrow a$ 
 $B \rightarrow b$ 

Augment CFG


 $I_0 = \text{closure}(S' \rightarrow \cdot S) =$ 

$$\{ S' \rightarrow \cdot S, \\ S \rightarrow \cdot AB, \\ A \rightarrow \cdot a \}$$



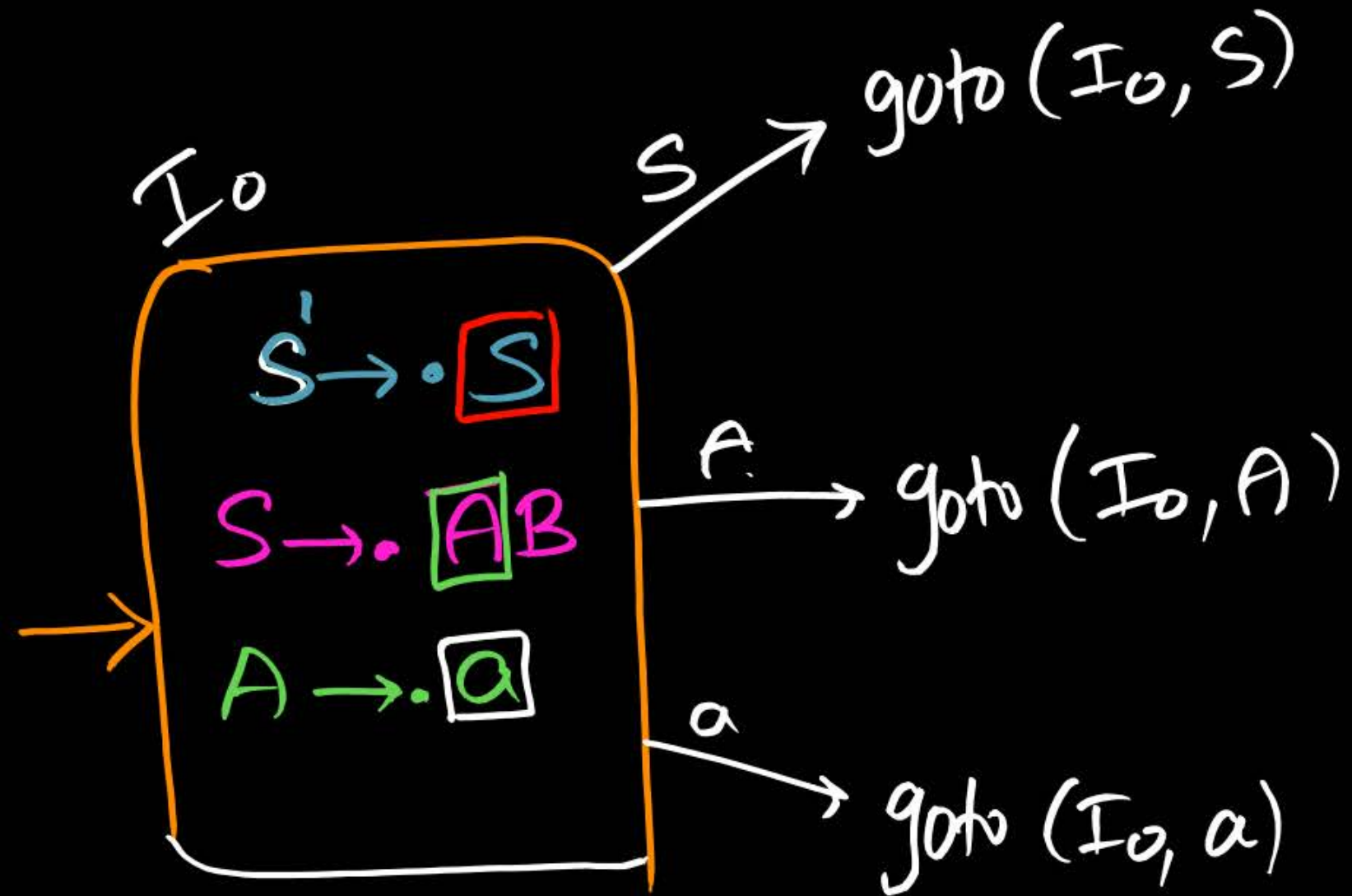
$$S' \rightarrow S$$

$$S \rightarrow AB$$

$$A \rightarrow a$$

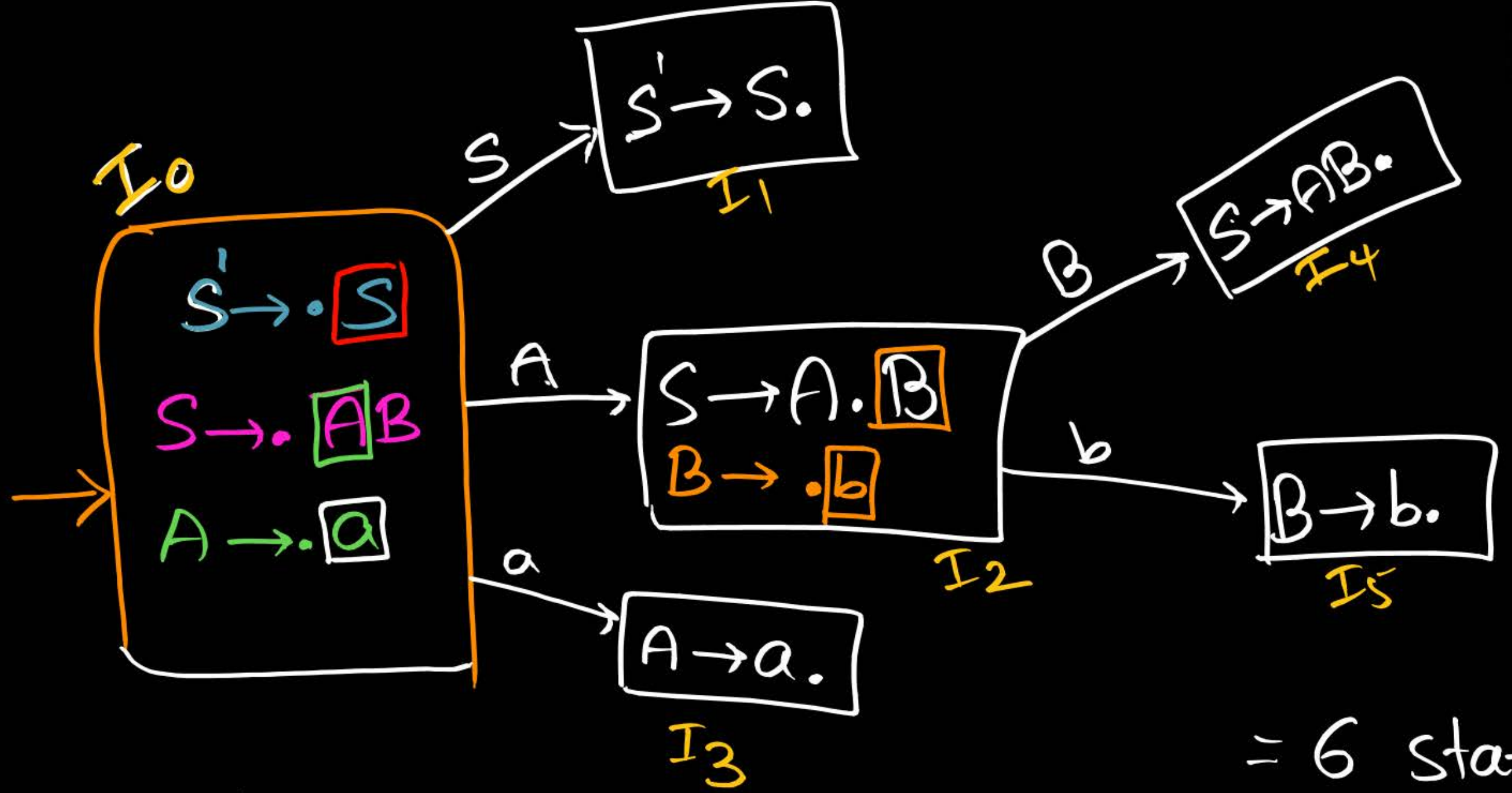
$$B \rightarrow b$$

Augment CFG



$S' \rightarrow S$   
 $S \rightarrow AB$   
 $A \rightarrow a$   
 $B \rightarrow b$

Augment CFG



= 6 states

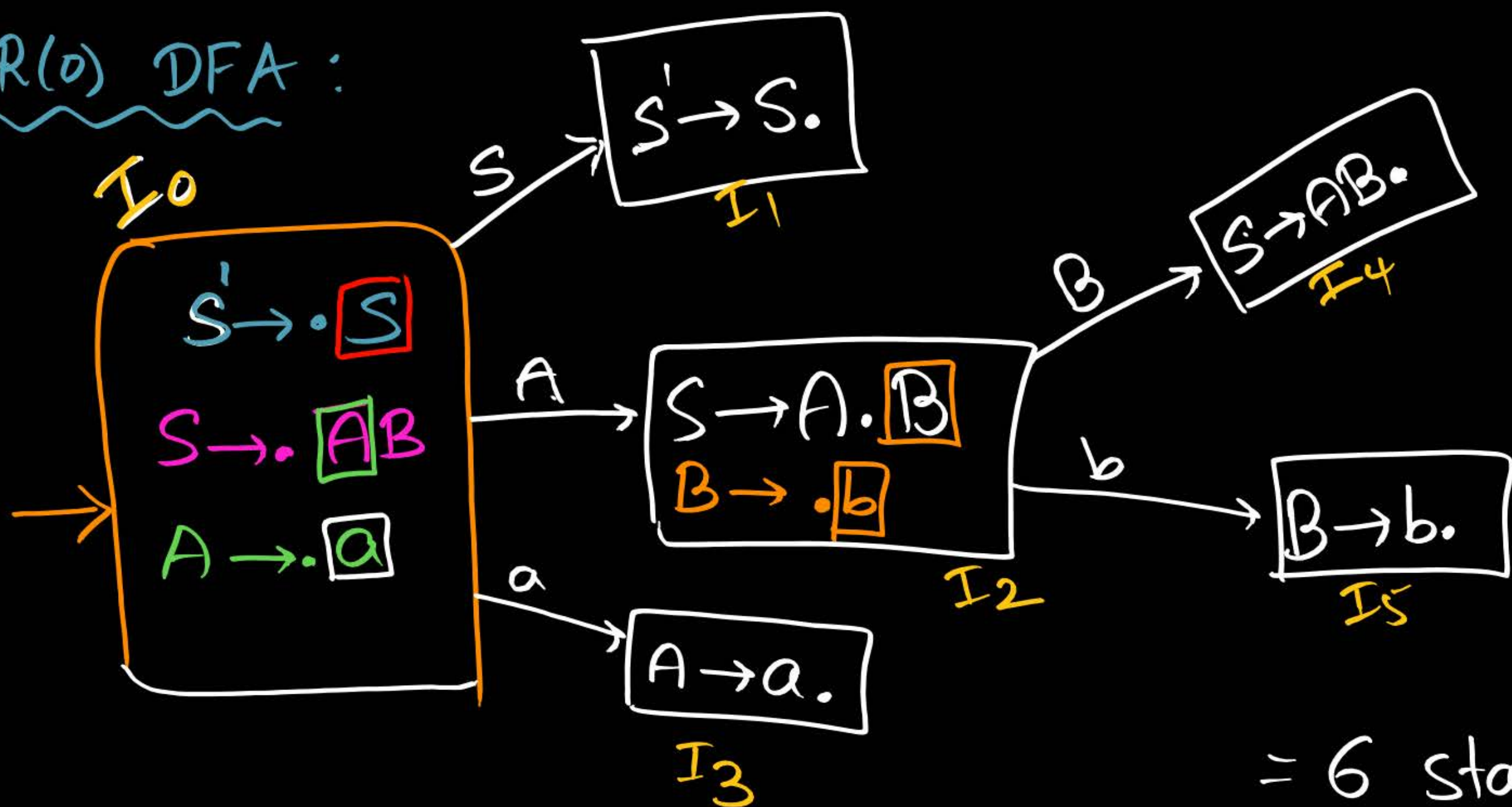
Note:  $I_0 = \text{closure}(S' \rightarrow \cdot S)$

Find  $|\text{goto}(I_0, A)| = ? = |\{S \rightarrow A.B, B \rightarrow \cdot b\}| = 2$

①

LR(0) DFA : $S' \rightarrow S$  $S \rightarrow AB$  $A \rightarrow a$  $B \rightarrow b$ 

Augment CFG



= 6 states

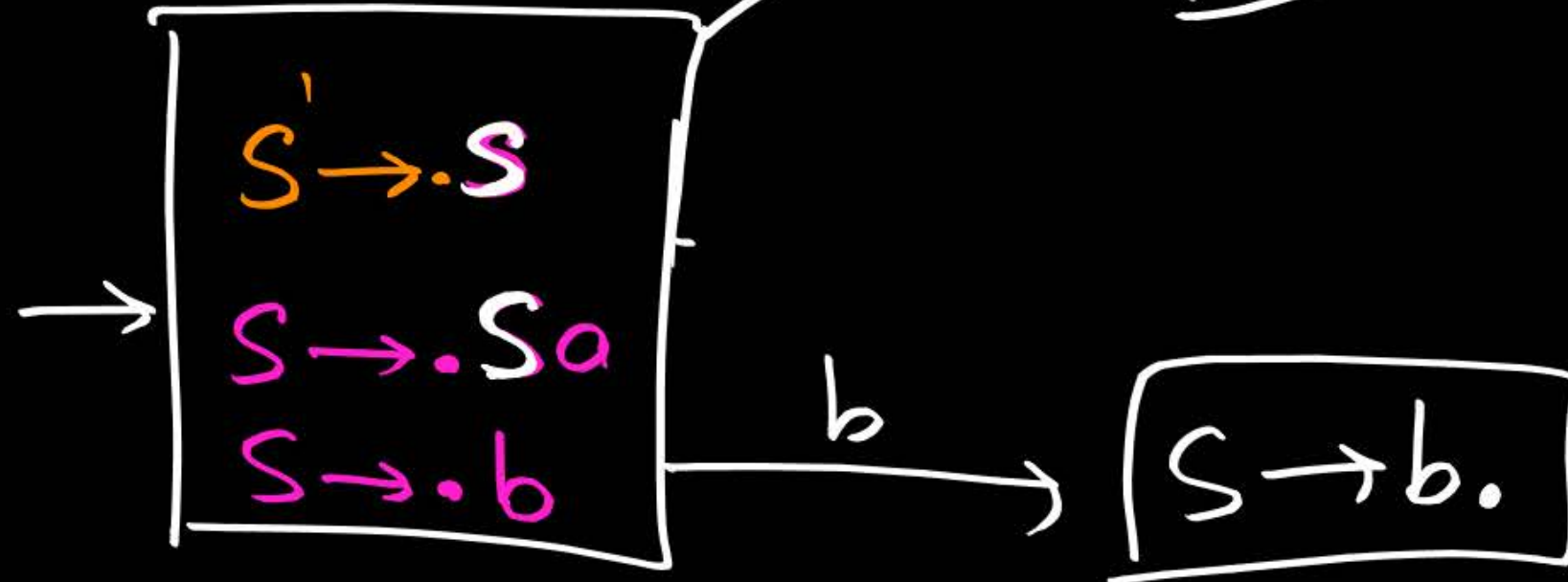
It is LR(0) CFG.

Note:  $S' \rightarrow S \cdot$  is an accepting item  
(It not involves in conflict)



②

$S \rightarrow Sa \mid b$



= 4 states

It is LR(0) CFG



③

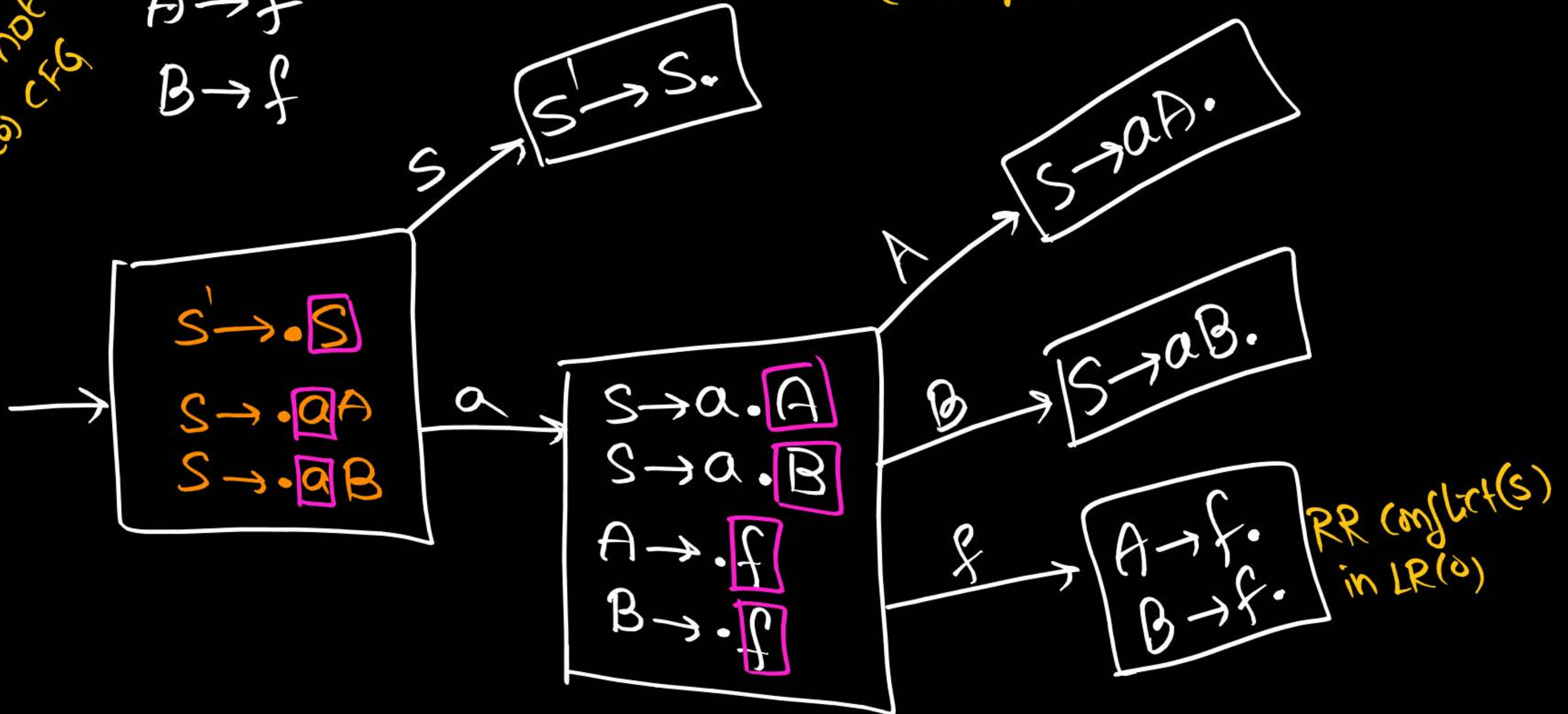
$S \rightarrow aA \mid aB$

$A \rightarrow f$

$B \rightarrow f$

It is not LR(0) CFG

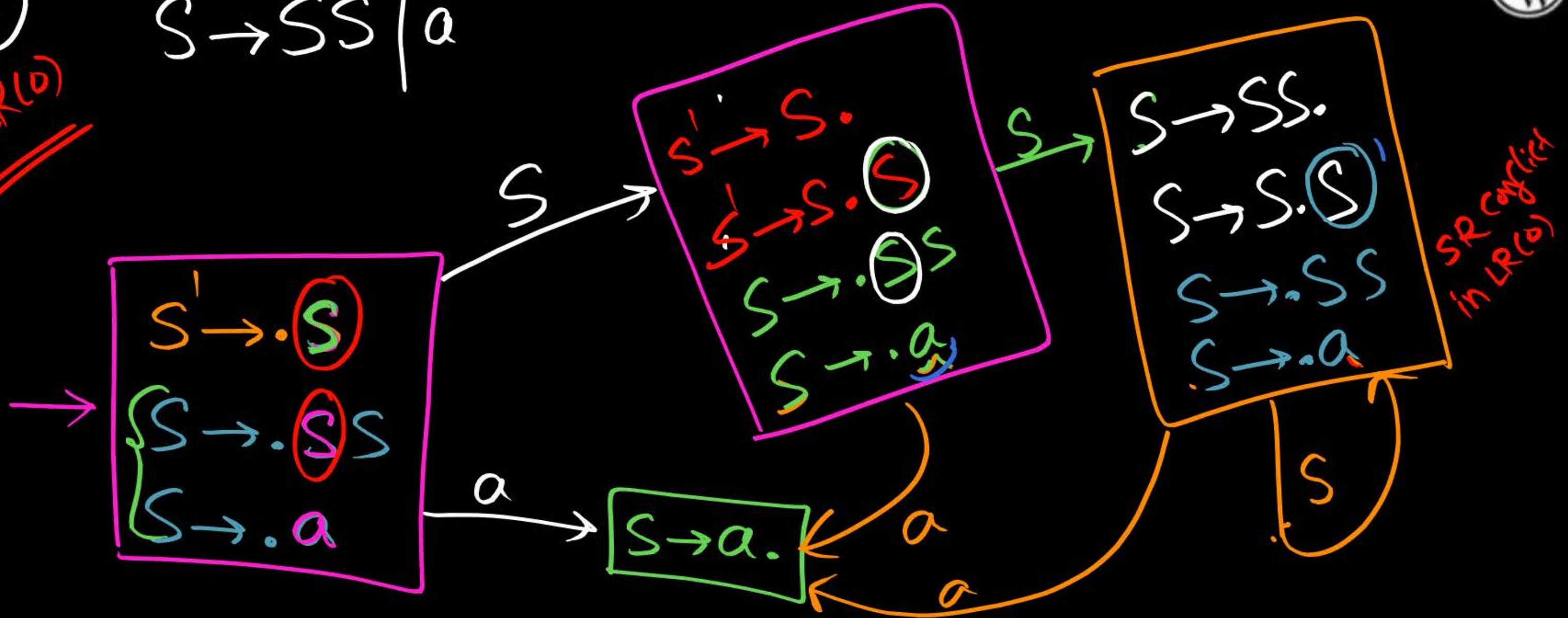
1 conflict state in LR(0)  
(Inadequate)



④

$S \rightarrow SS/a$

not LR(0)

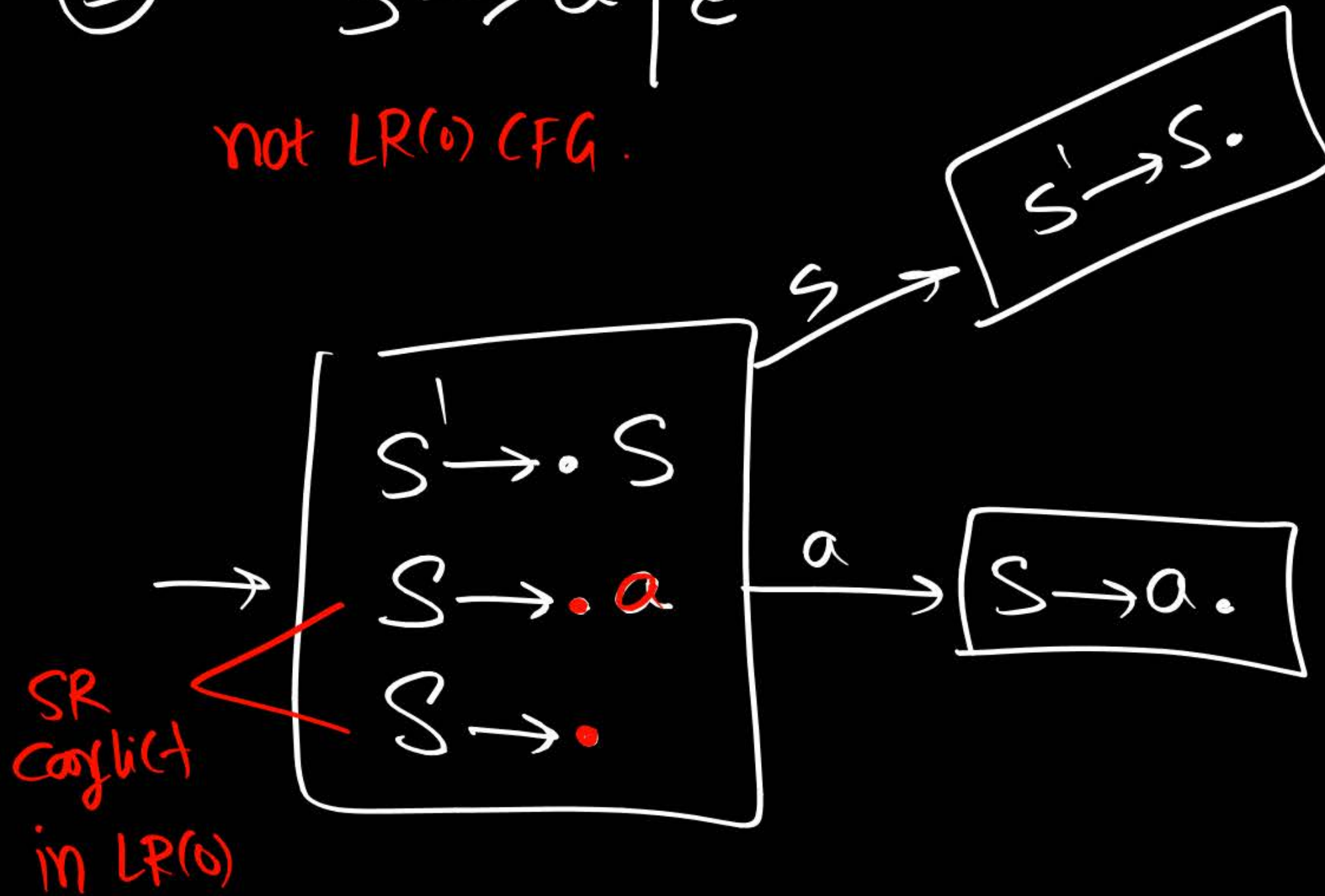


Note: State Item not participates in a conflict.



⑤  $S \rightarrow a / \epsilon$

not LR(0) CFG.



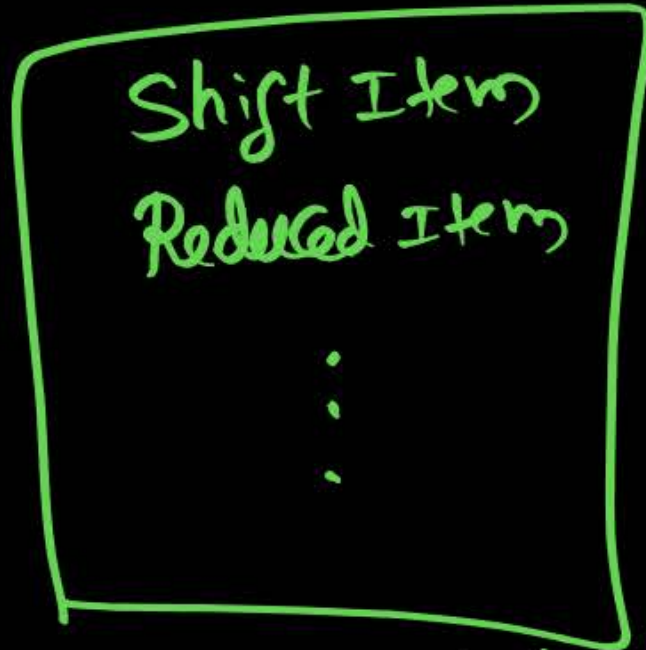
$$x\epsilon = \epsilon x = x$$

$$\cdot \epsilon = \epsilon \cdot = \cdot$$

How to check given CFG is LR(0) or not?

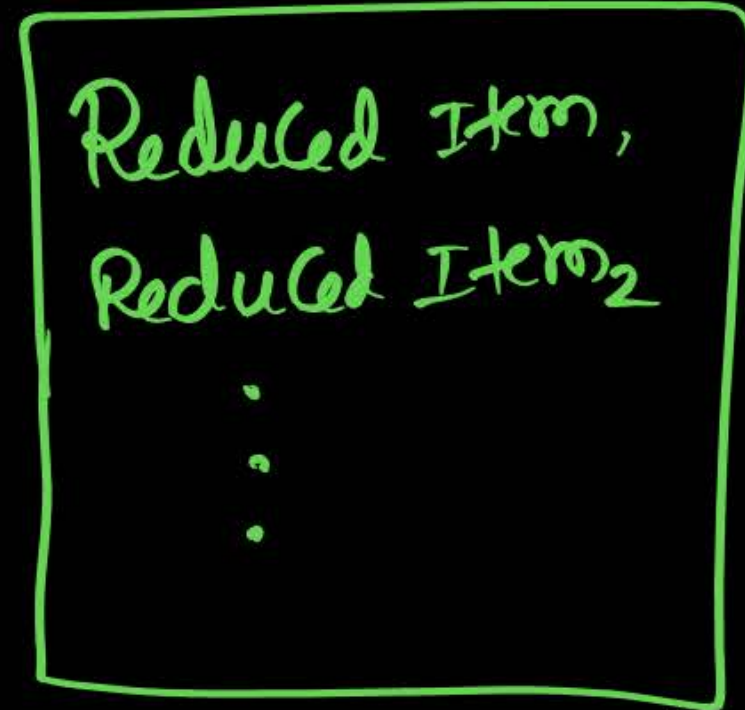


SR conflict



If state has both S Item  
and R Item then it  
produces SR conflict

RR conflict



If state has 2 reduced Items then  
it produces RR conflict(s)

If LR(0) DFA has no conflicts then given CFG is LR(0).



Note:

- I) Acceptance Item not participates in any conflict  
 $S' \rightarrow S.$
- II) State Item also not participates in any conflict
- III) State with only one item never produces any conflict.
- IV) If state has Reduced Item then only there is a possibility  
 - to produce either RR or SR  
 - conflict
- V) If state has no reduce item then that state never produces conflict.

⑥

$$S \rightarrow Aa \mid Bb$$
$$A \rightarrow f$$
$$B \rightarrow f$$

$$(7) S \rightarrow Aa/Bb/cAb$$

$$A \rightarrow f$$

$$B \rightarrow g$$



$$(8) S \rightarrow AB \mid ab$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$\textcircled{9} \quad S \rightarrow SS \mid (S) \mid \epsilon$$

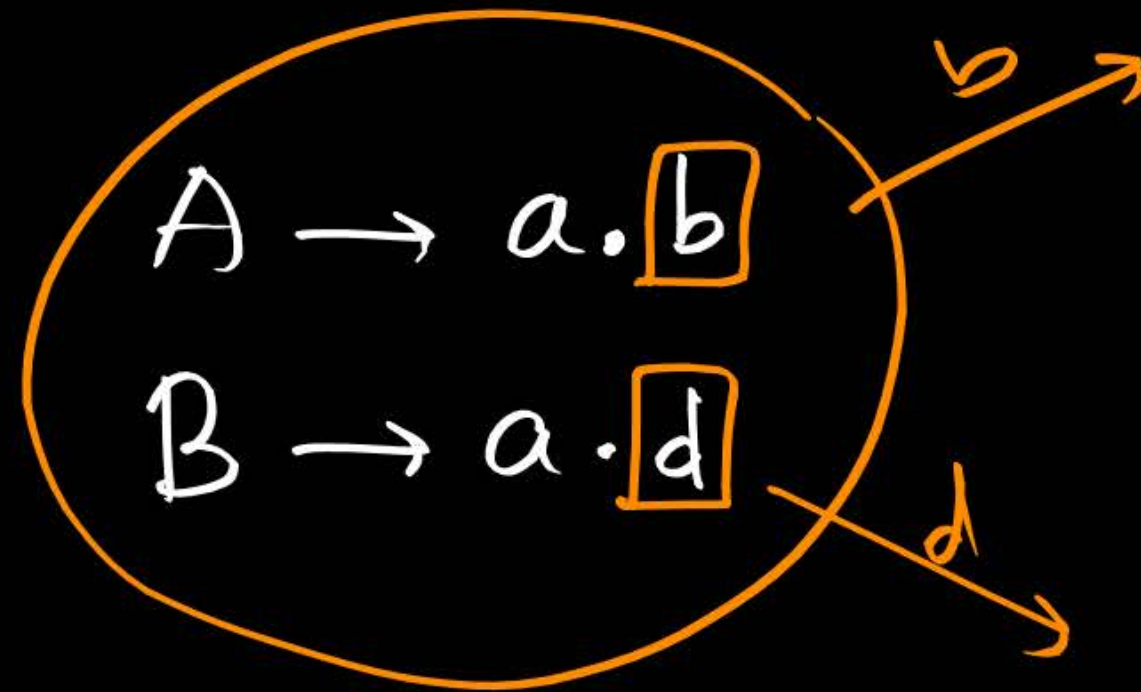
$$(10) \quad S \rightarrow aA \mid bB$$

$$A \rightarrow \epsilon$$

$$B \rightarrow bA$$



Why SS conflict not here?



$A \rightarrow a.b$

$B \rightarrow a.$

# Summary

→ LR(0) DFA construction

→ Augmented CFG?

→ LR(0) Item?

→ Closure() and goto() functions

→ LR(0) DFA

→ SR and RR conflicts checking for LR(0).

Thank you  
PW  
Soldiers

