

CS & IT ENGINEERING

Compiler Design

Lexical & Syntax Analysis



Lecture No. 8



By- DEVA Sir

TOPICS TO BE COVERED

01 closure() & goto()

02 LR(0) Diagram

03 LR(0) CFG ?

04 SLR(1) CFG ?

05

closure() : \Rightarrow It is "Set of items" state

I) \hookrightarrow It should be applied on every item in DFA.

II) If Nonterminal present just after dot then



We should add all X productions in same set
(Same state)
by placing dot at begin.

LR(0) DFA:

①

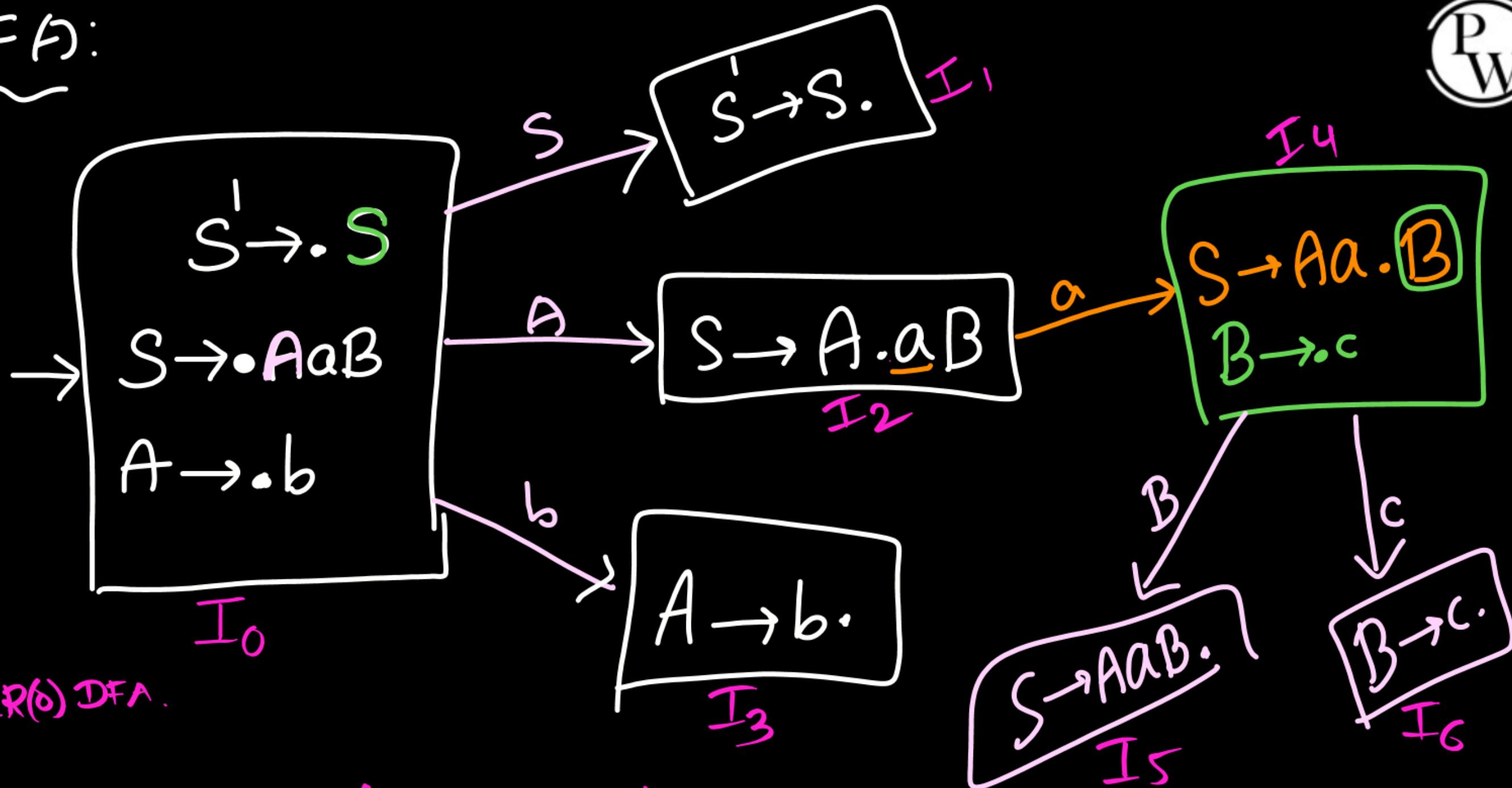
$$S \rightarrow AaB$$

$$A \rightarrow b$$

$$B \rightarrow c$$

It is LR(0)
CFG.
($S_0, SLR(1)$)

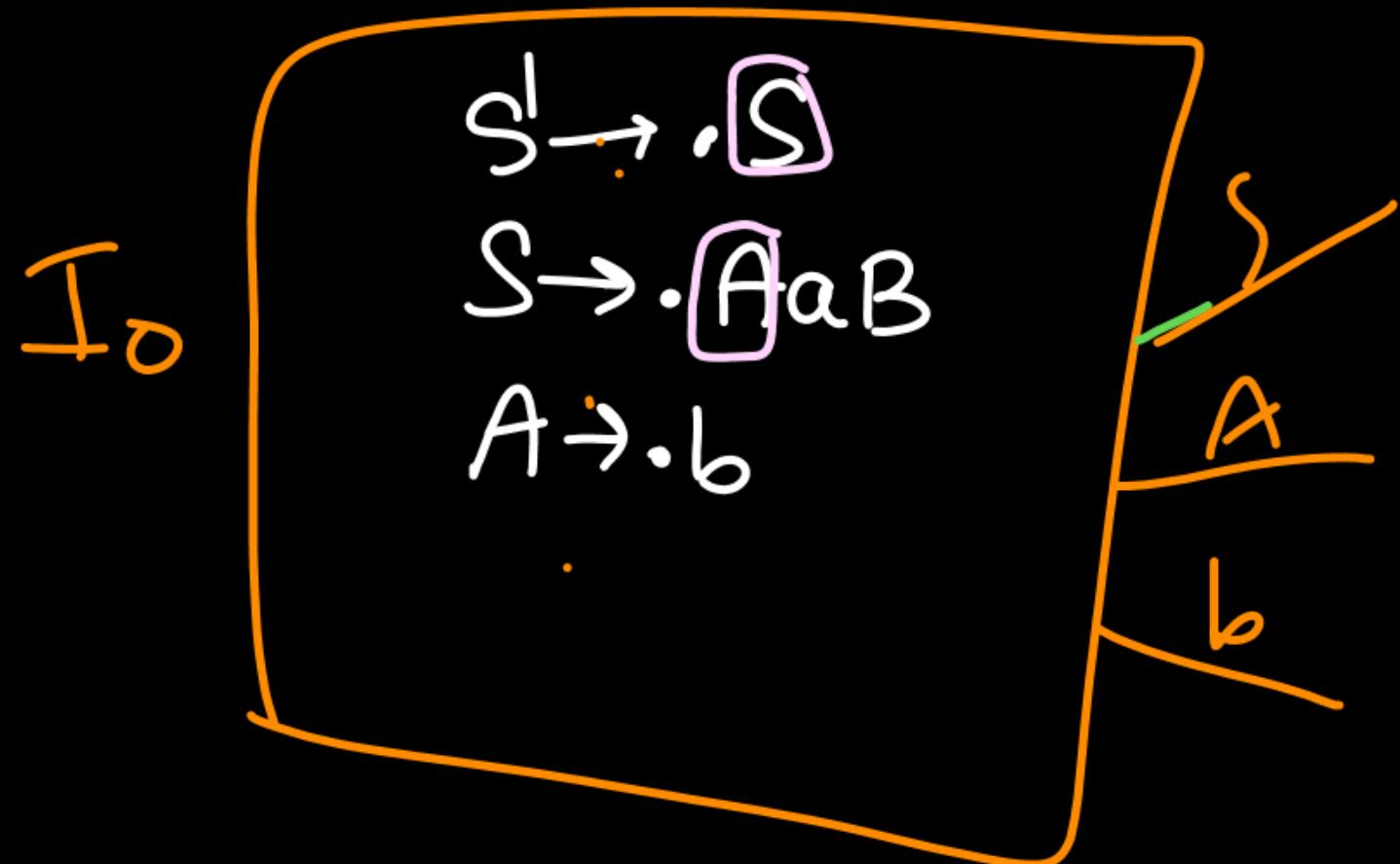
7 States in LR(0) DFA.



1 closure($S^l \rightarrow \cdot S$)
6 goto() functions

closure ($S' \rightarrow \cdot S$) = $\{ S' \rightarrow \cdot S, S \rightarrow \cdot AaB, A \rightarrow \cdot b \}$

= I_0

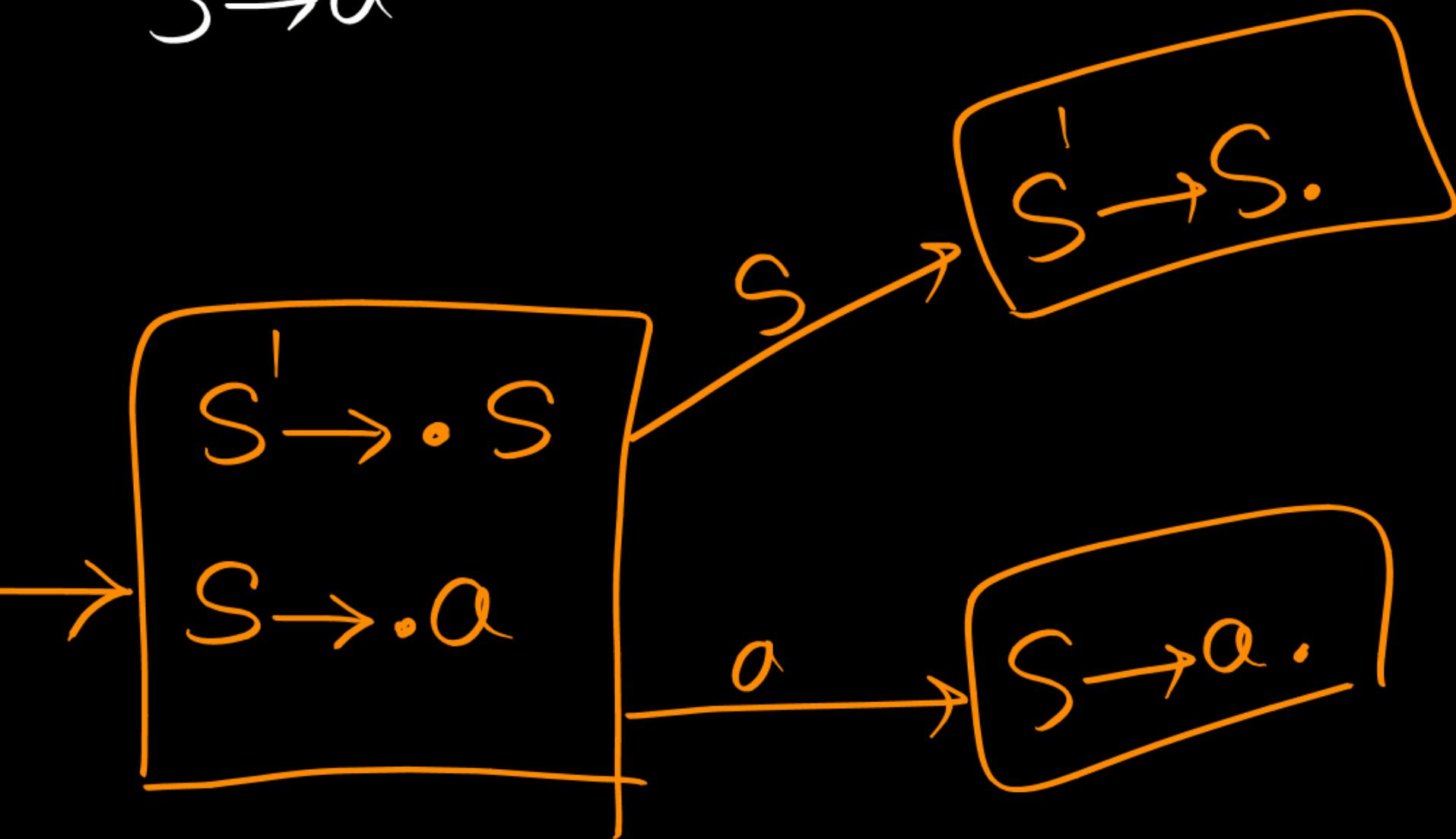


goto (I_0, S) = $\{ S' \rightarrow S \cdot \}$

goto (I_0, A) = $\{ S \rightarrow A \cdot aB \}$

goto (I_0, b) = $\{ A \rightarrow b \cdot \}$

②

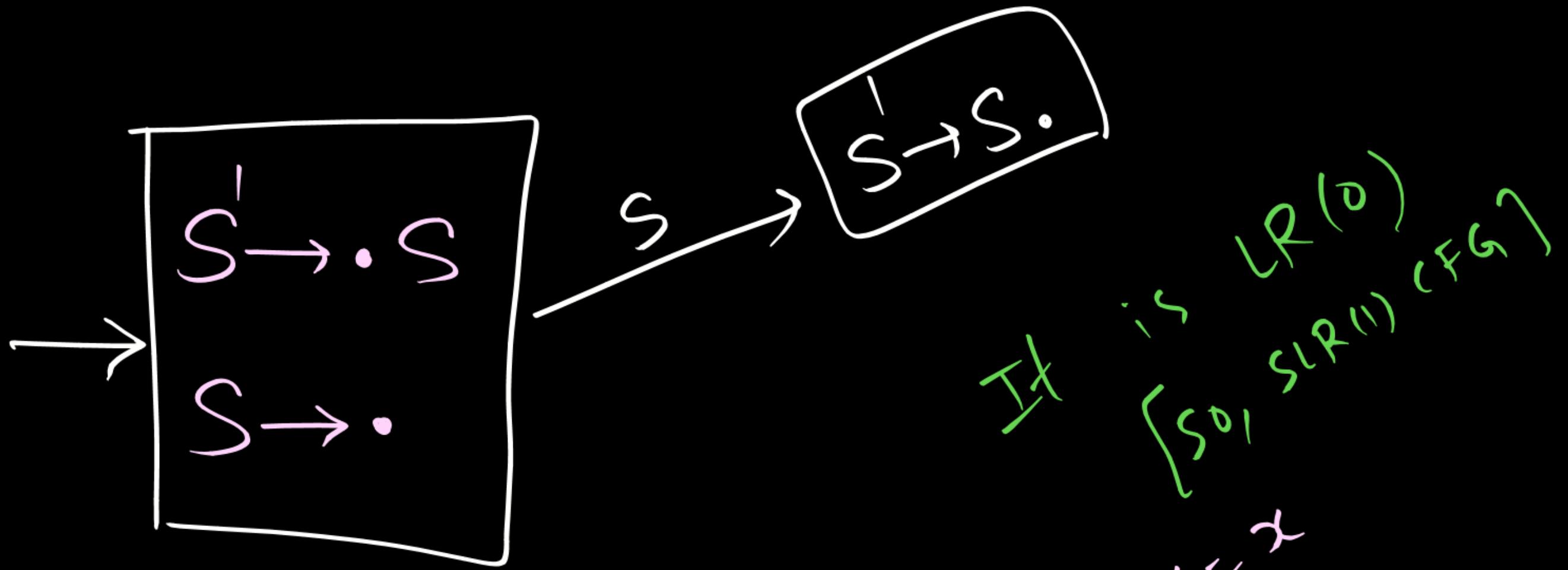
 $S \rightarrow a$ 

I^x is $(R^{(0)}, S_{R^{(1)}}, FG)$
 $\{S_0\}$

P
W

③ $S \rightarrow E$

P
W



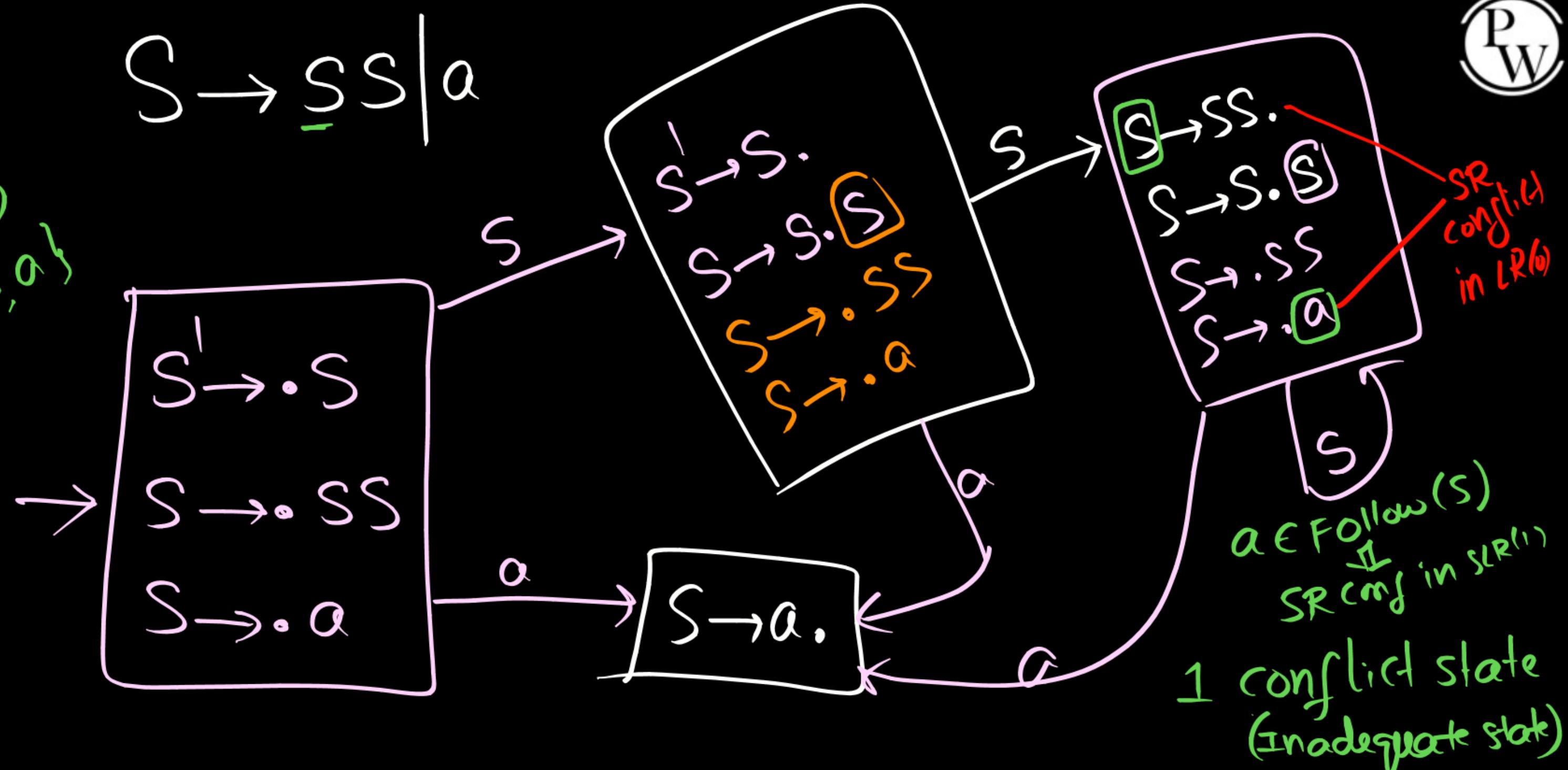
It is $\{S_0, S_{LR(0)}, LR(0)\}$ (FG)

$x^{\epsilon}, \epsilon^+, \epsilon^- = \cdot$
 $\cdot \epsilon = \epsilon^\cdot = \cdot$

④

$S \rightarrow SS | a$

$\text{Follow}(S) = \{\$, a\}$

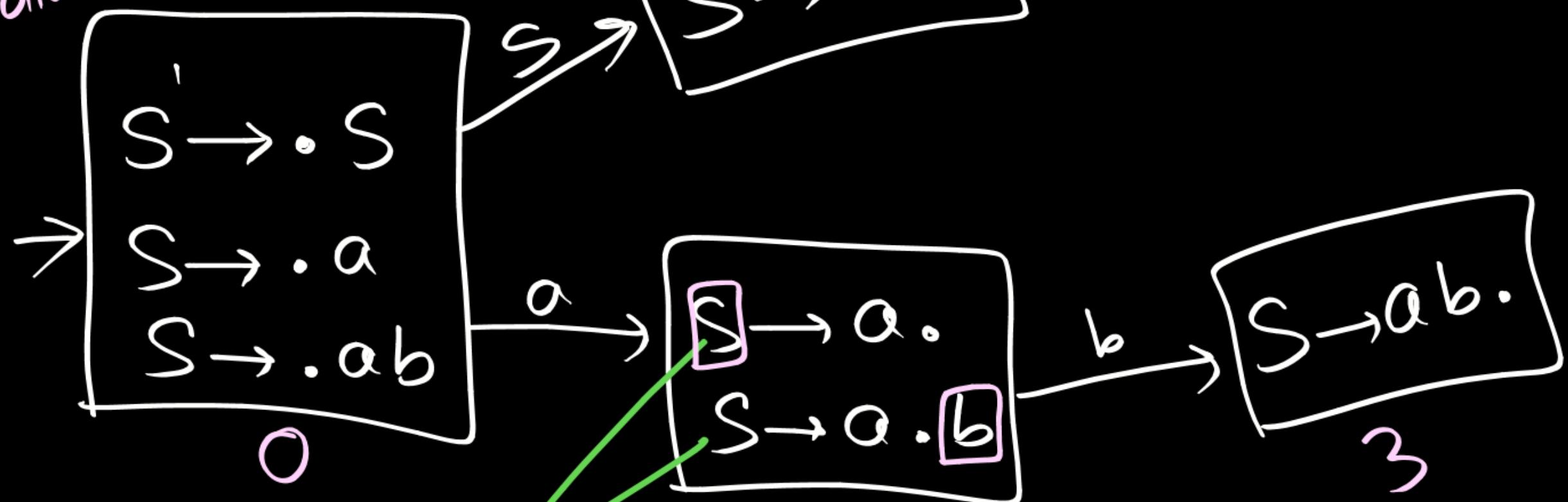


It is not LR(0) CFG
It is not SLR(1) CFG

⑤ $S \rightarrow a \mid ab$

P
W

$\text{Follow}(S) = \{\$\}$



SR conflict in LR(0)

$b \notin \text{Follow}(S)$

Not SR conflict in SLR(1)

It is not LR(0) (FG).
But it is SLR(1).
X States

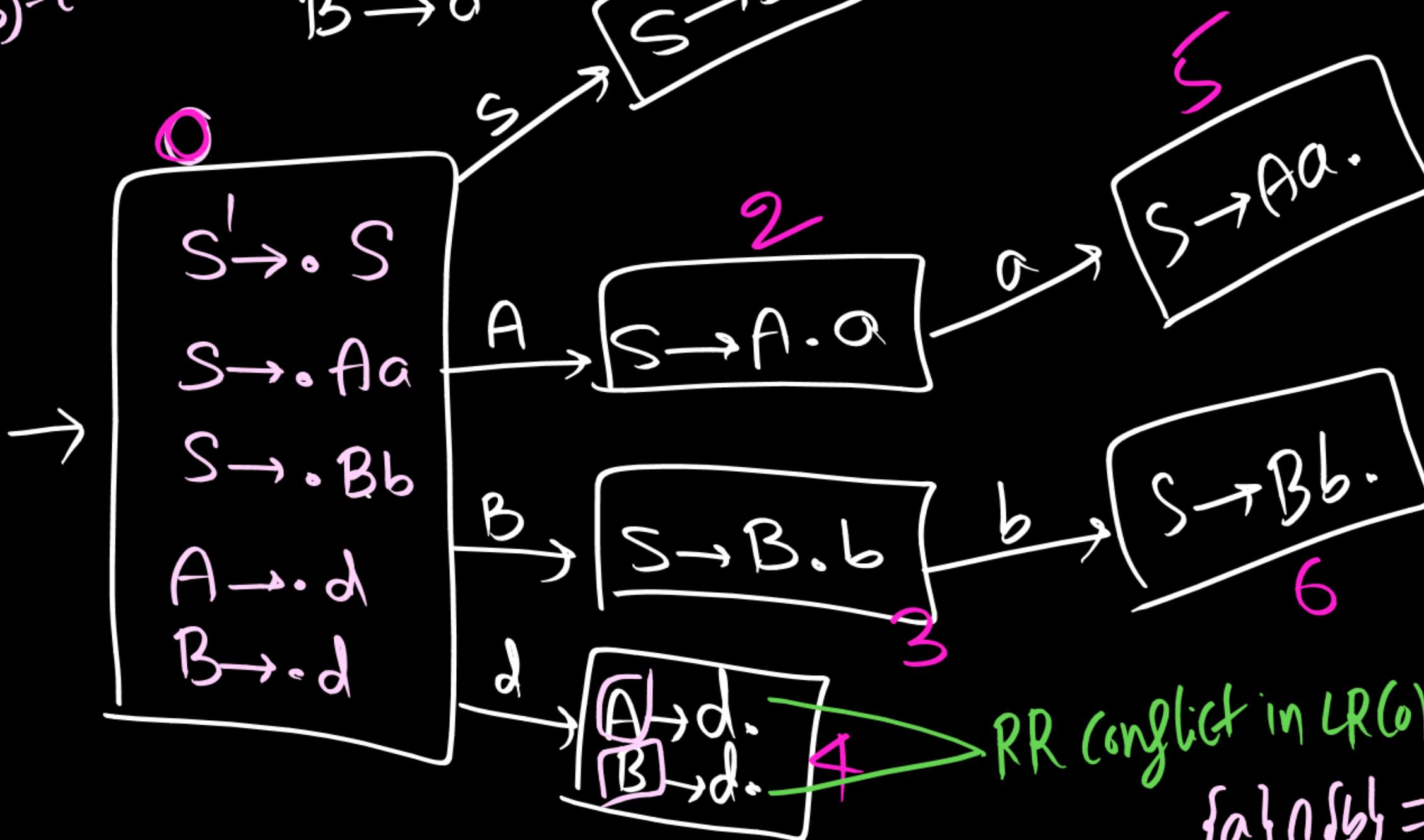
⑥

$$S \rightarrow Aa \mid Bb$$

$$A \rightarrow d$$

$$B \rightarrow d$$

$$\begin{aligned} F_0(A) &= \{a\} \\ F_0(B) &= \{b\} \end{aligned}$$



It is not LR(0) CFG,
It is SLR CFG

P
W

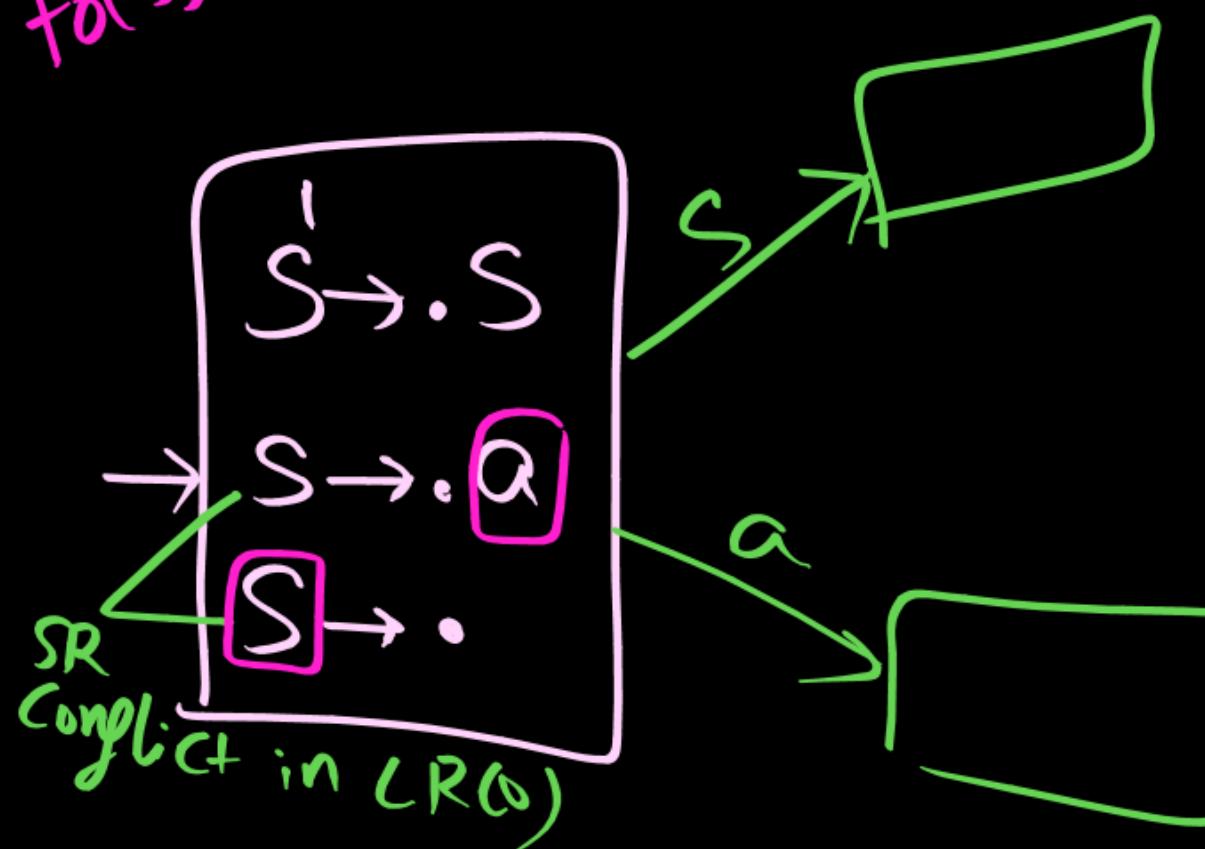
RR conflict in LR(0)

$\{a\} \cap \{b\} = \emptyset \Rightarrow$ no RR conf in SLR

↗ State
↗ State

⑦ $S \rightarrow a \mid \epsilon$

$$F_0(S) = \{ \$ \}$$

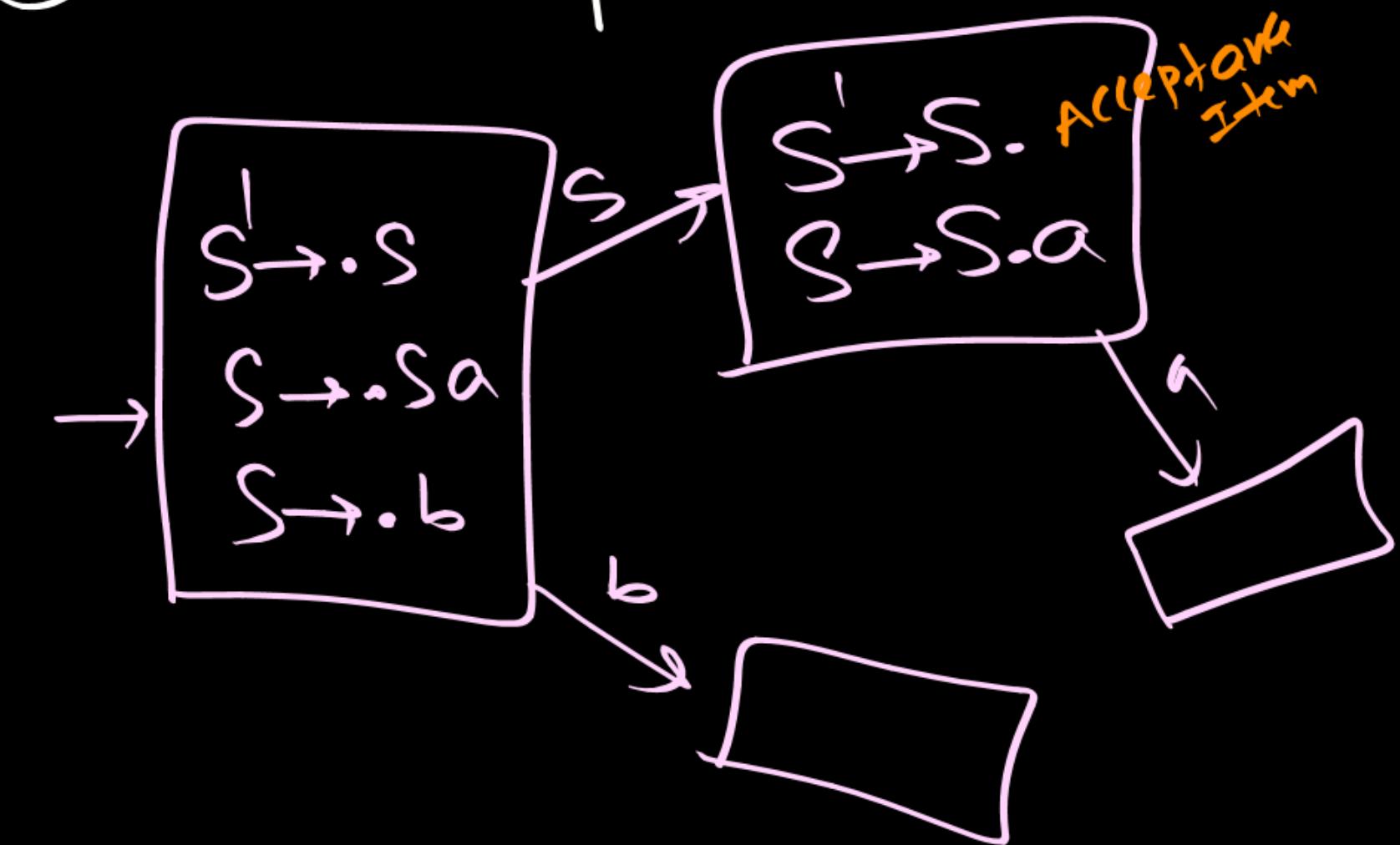


Not LR(0) CFG

$a \notin F_0(S) \Rightarrow$ No confg in SLR

So, CFG is SLR(1)

⑧ $S \rightarrow Sa \mid b$



It is LR(0) CFG
(It is SLR(1))

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

Step 1:
Construct LR(0) DFA

Step 2:
If no conflicts present in DFA
then CFG is LR(0).

SR conflict in LR(0)

Shift Item
Reduced Item

:

If both Shift item and Reduced item
are present in same state

then it produces SR conflict(s)

RR conflict in LR(0)

Reduced Item₁
Reduced Item₂

:

If two reduced items are
present in same state then
it produces RR conflict(s)

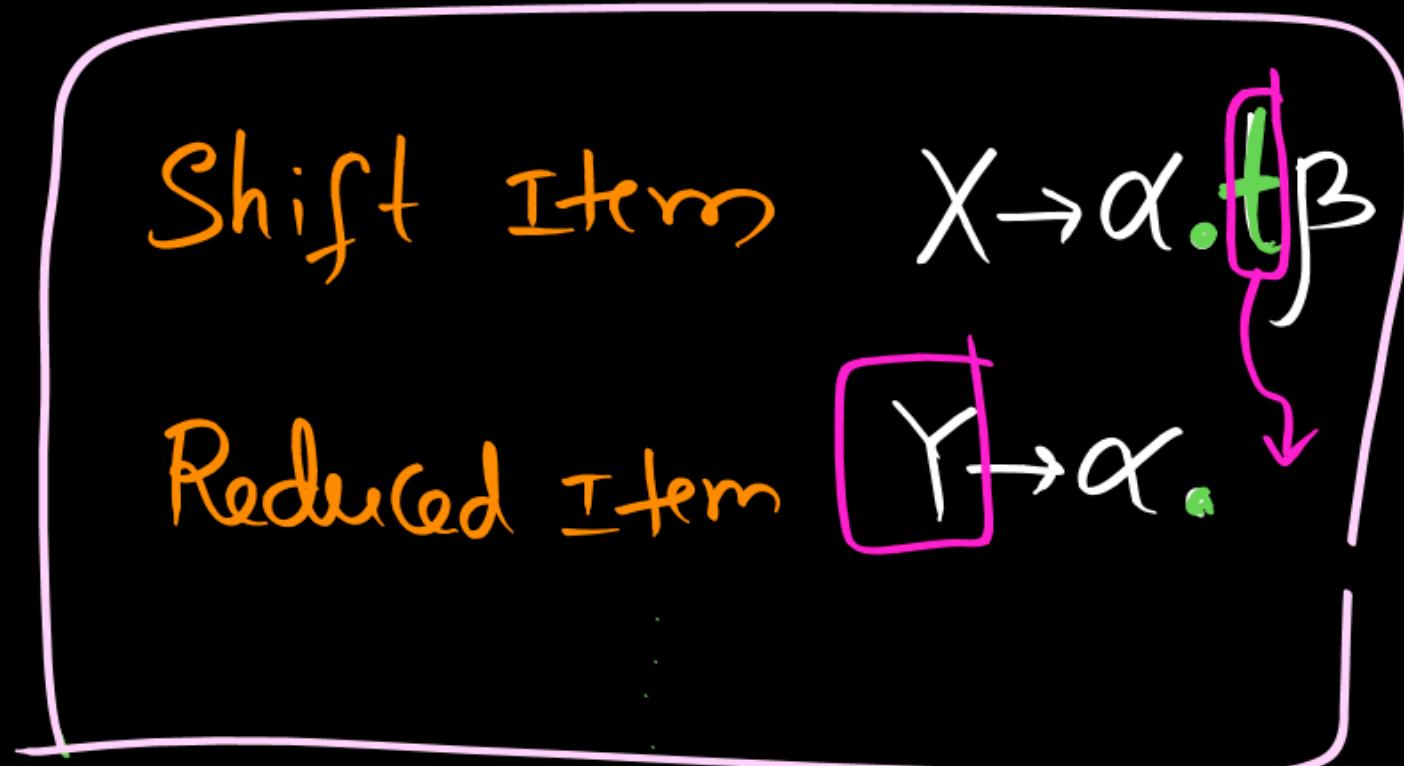
Note:

- I) If state has single item then it never produces conflicts.
- II) State item will not participate in any conflict
- * III) Acceptable Item ($s' \rightarrow s.$) not participates in any conflict
- IV) If state is not having Reduced Item then it never produces any conflict.

How to check CFG is SLR(1) or not ?

- Step 1: Construct LR(0) DFA
- Step 2: If no SLR(1) conflicts in DFA
then CFG is SLR.

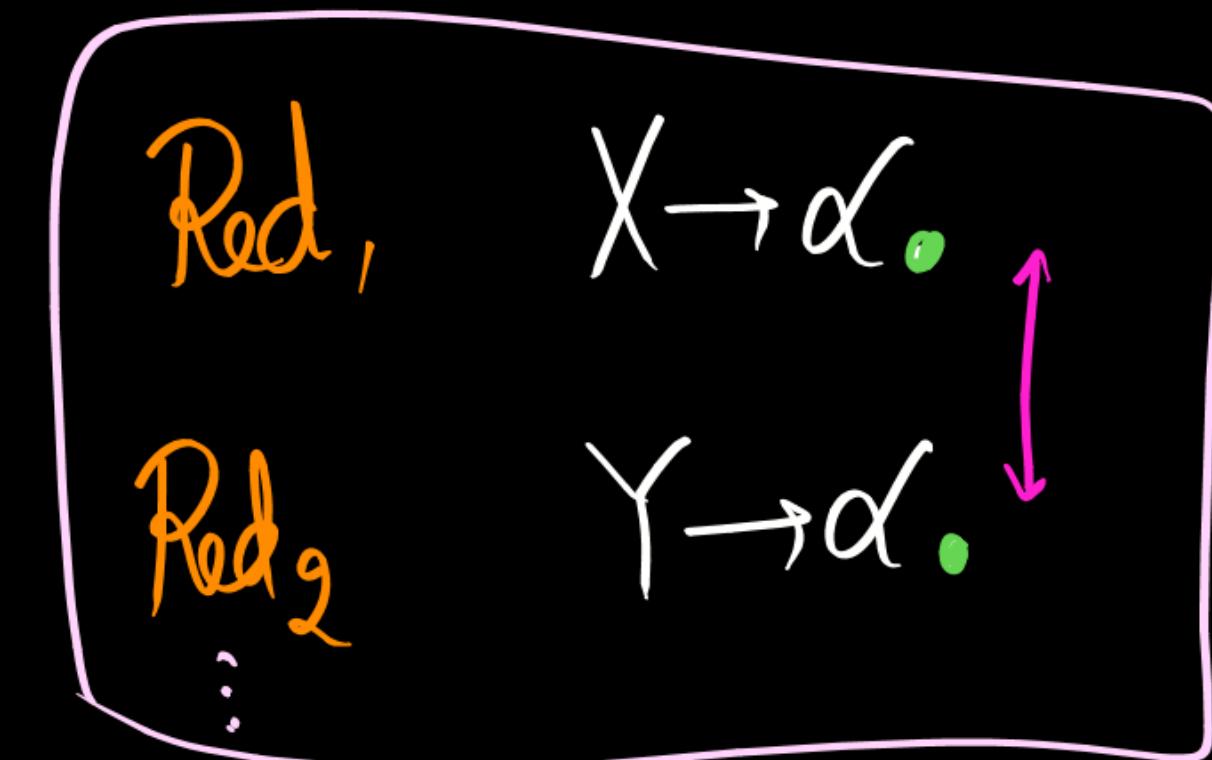
SR conflict in SLR(1)



If $t \in \text{Follow}(Y)$

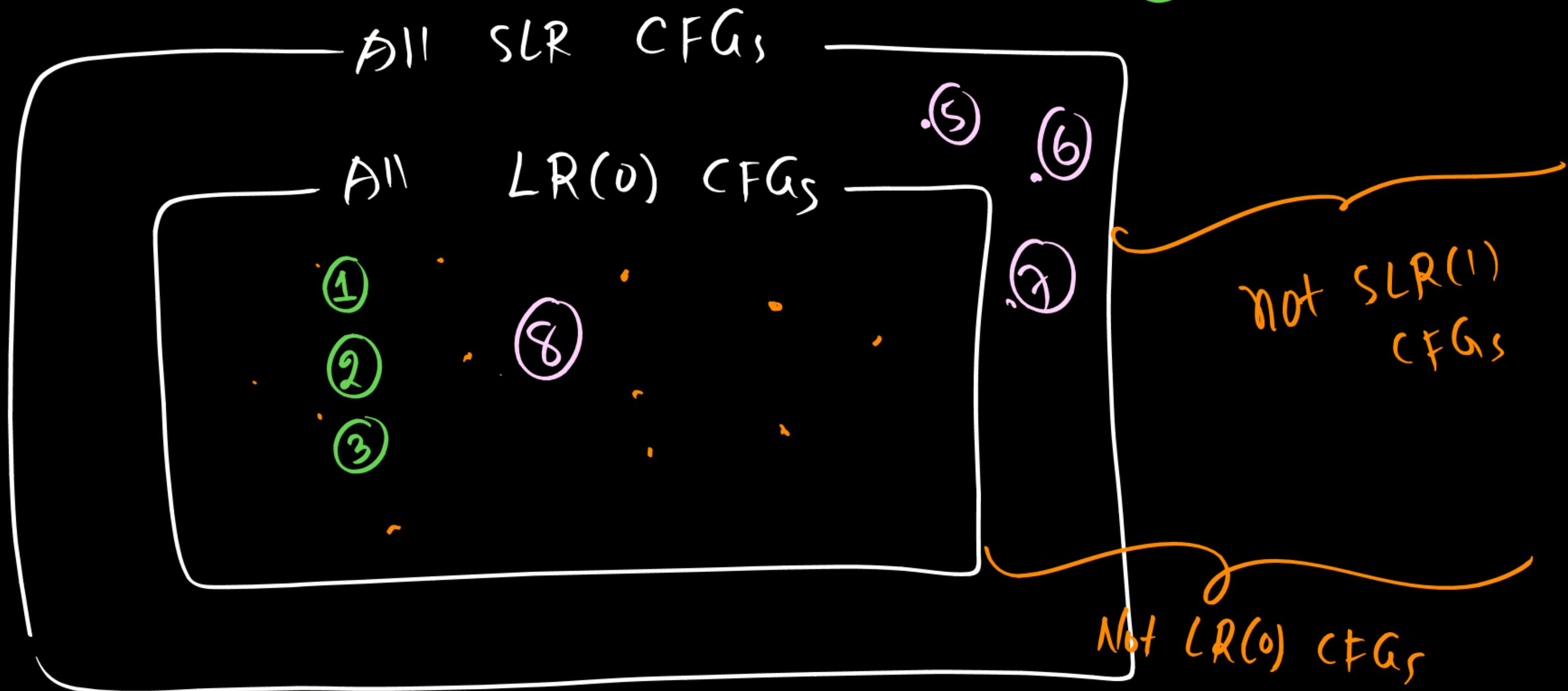
then SR conflict

RR Conflict in SLR(1)



If $\text{Follow}(X) \cap \text{Follow}(Y) \neq \emptyset$

then RR conflict



LR(0) Items



LR(0) parser

SLR(1) parser

Follow set

Computed

based on whole CFG

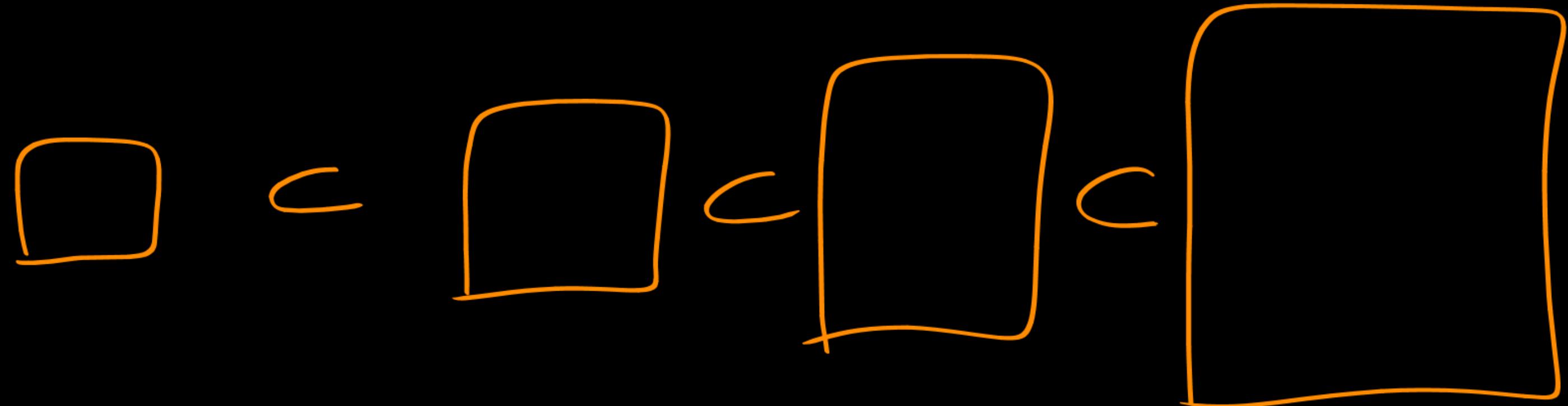
LR(1) Items



CLR parser [LR(1) parser]

LALR parser

LR(0) parser < SLR(1) < LALR(1) < CLR
LR(1)





$\hookrightarrow LR(0)$ CFGs } Using $LR(0)$ DFA .
 $\hookrightarrow SLR(1)$ CFGs } Requiring $LR(0)$ Items

