CS & IT Engineering

Compiler Design

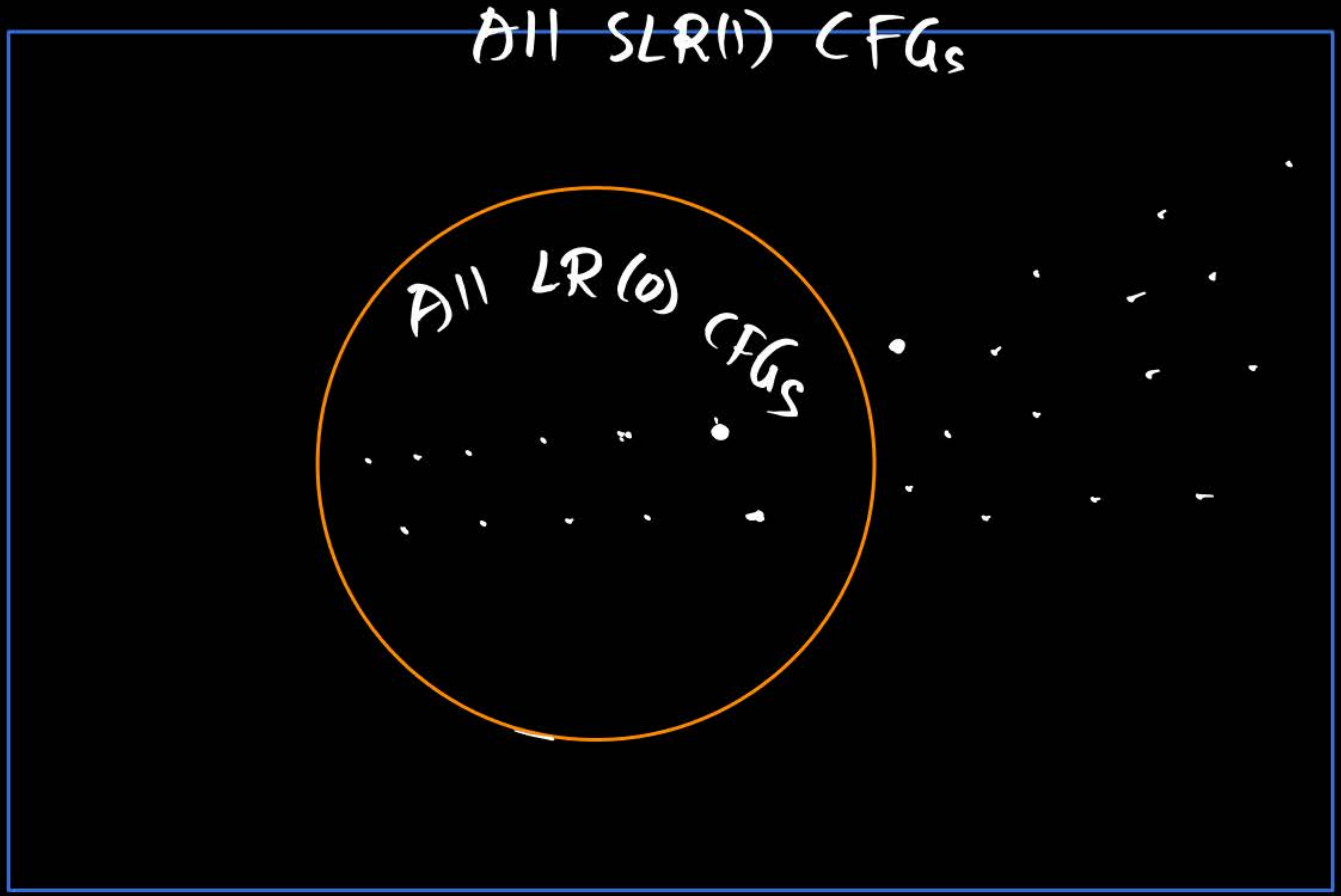
Lexical Analysis and Syntax Analysis

Lecture: 9

Deva sir

Topics to be covered:

- Bottom-up parser
    - LR (0) Parser
    - SLR parser
    - LALR and CLR parser
    - Operator precedence parsing


All SLR(1) CFGs
All LR(0) CFGs

Note :

I) Every LR(0) CFG is always SLR(1) CFG.

II) LR(0) CFG is need not be Subset of SLR(1) CFG.

III) Set of all LR(0) CFGs is Subset of Set of all SLR(1) CFGs.

LR(0) class is subset of SLR(1) class

① LR(0) DFA :

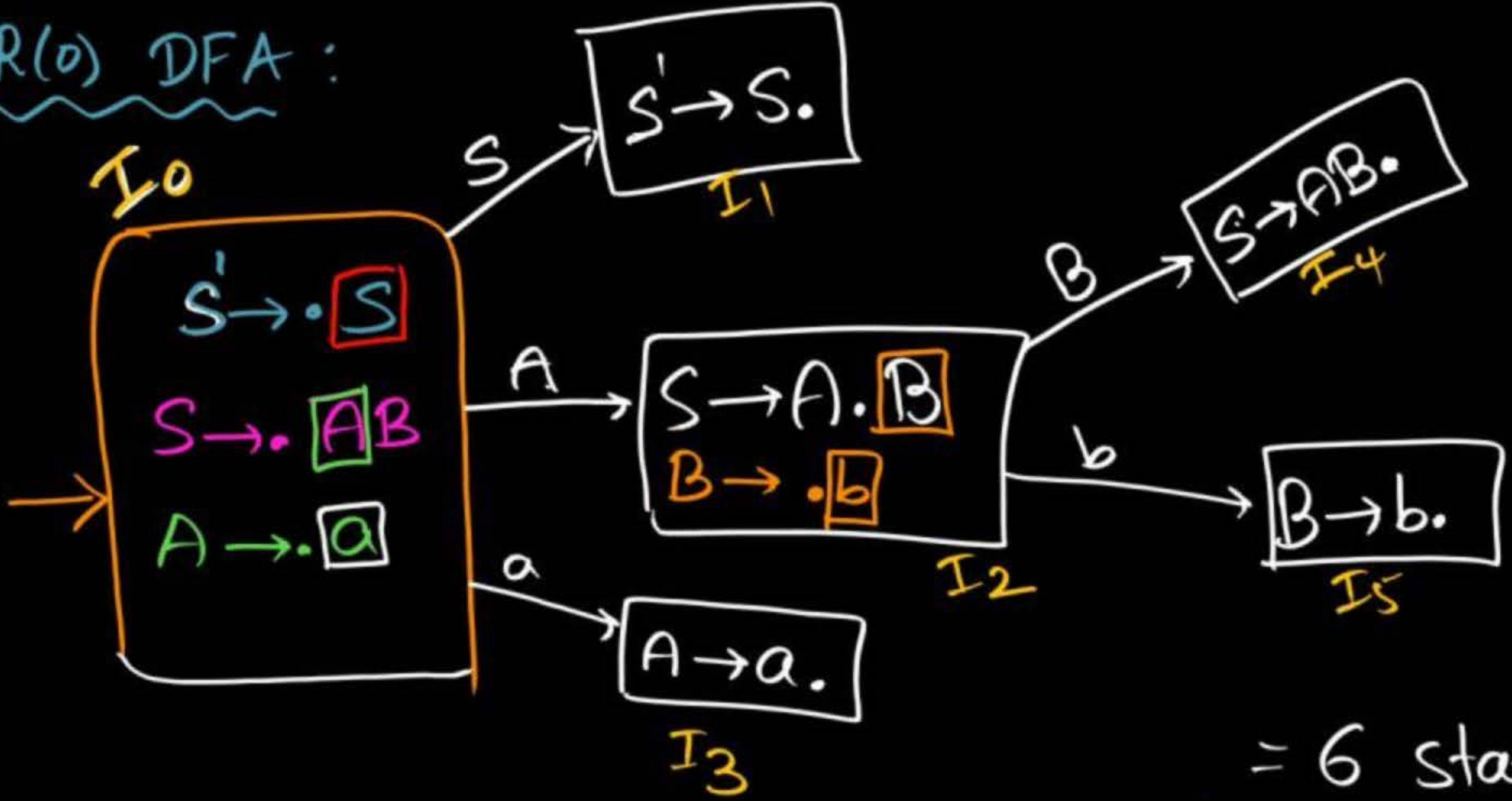
$S' \to S$

$S \to AB$

$A \to a$

$B \to b$

Augment CFG



= 6 states

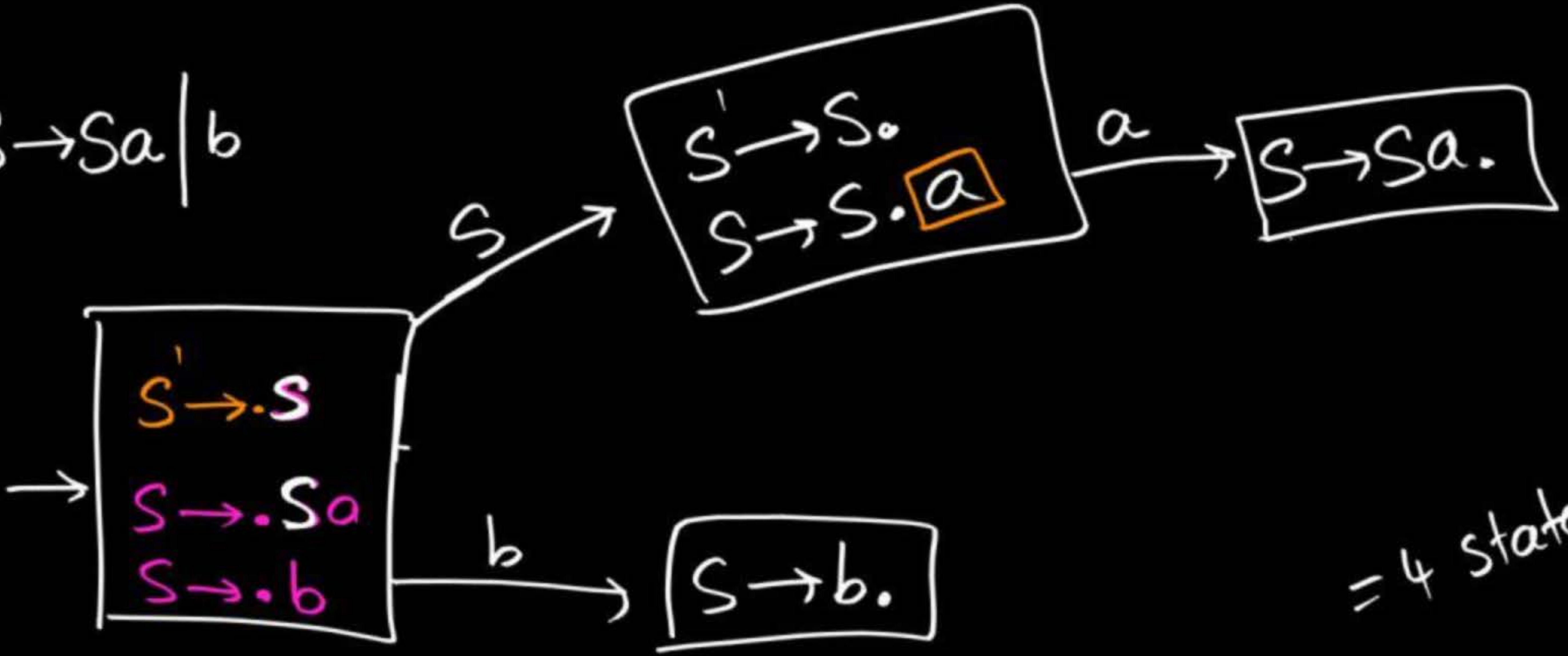
It is LR(0) CFG

So, SLR(1) CFG

Note : $S' \to S.$ is acceptance item
(It not involves in conflict)

②

$S \to Sa \mid b$



= 4 states

It is LR(0) CFG

So, It is SLR(1)

③ $S \rightarrow aA \mid aB$

$A \rightarrow f$

$B \rightarrow f$

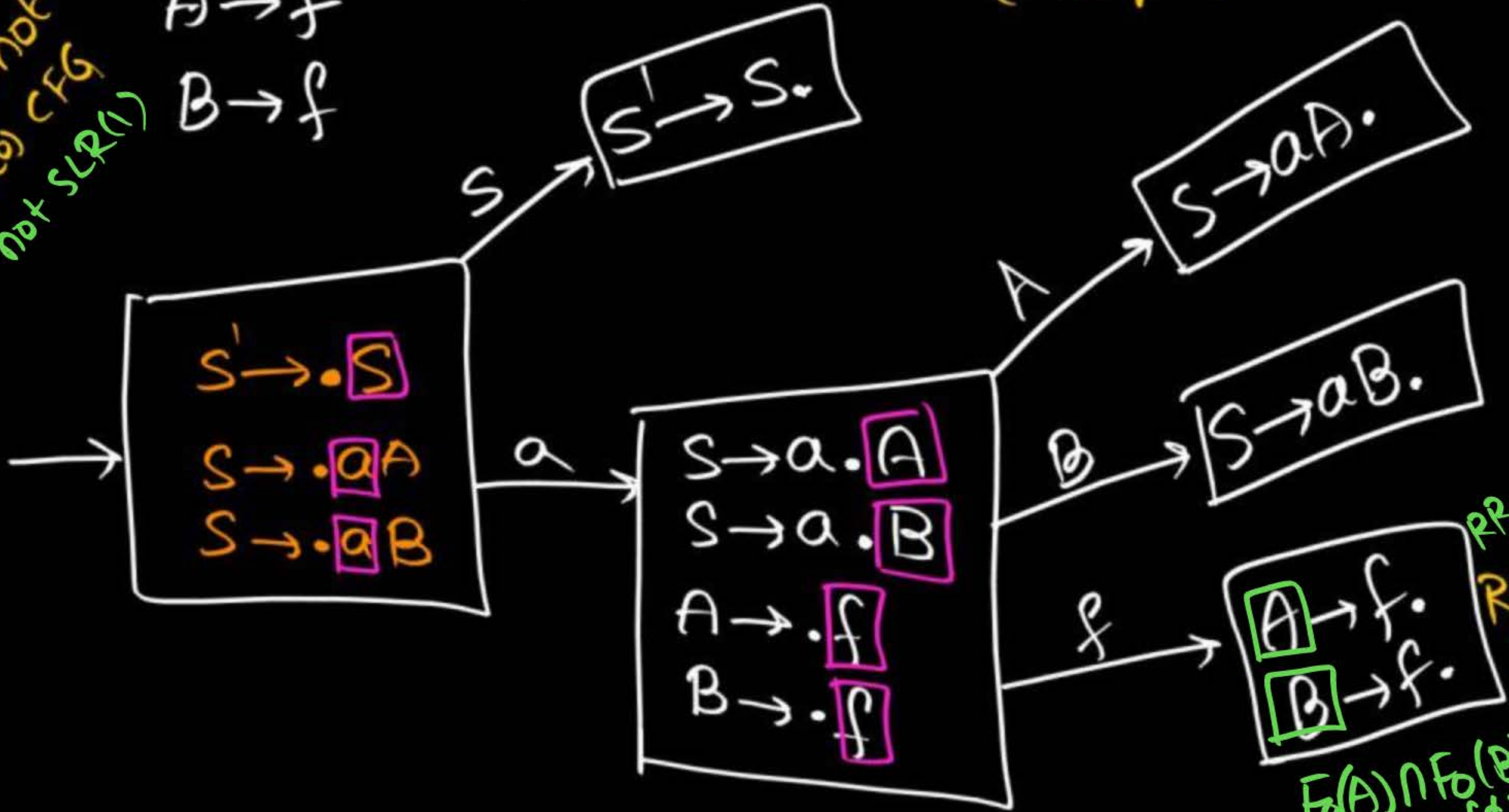
$Fo(A) = \{\$\}$

$Fo(B) = \{\$\}$

1 conflict state in LR(0)
(Inadequate)

It is not LR(0) CFG

It is not SLR(1)



RR conf in SLR(1)

RR conflict(s) in LR(0)

$Fo(A) \cap Fo(B) \neq \phi$

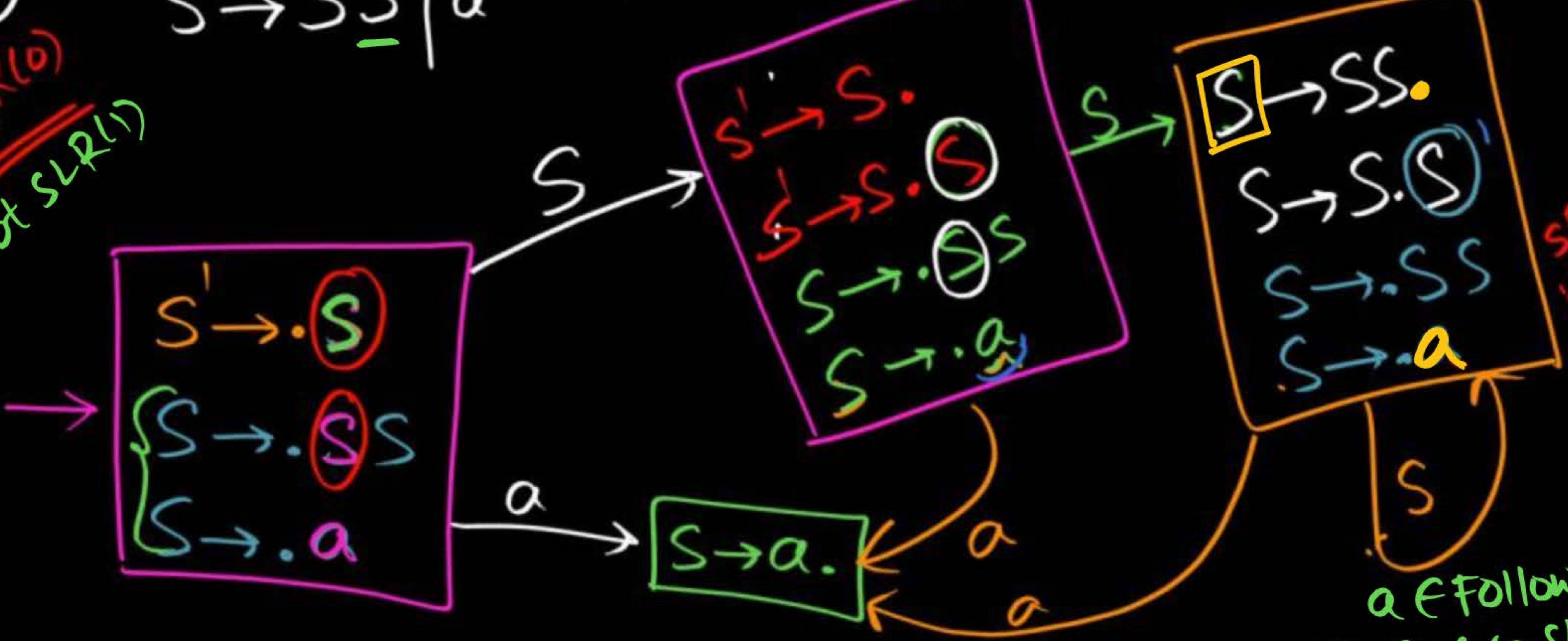
$\{\$\} \cap \{\$\} = \{\$\}$

④ $S \rightarrow SS \mid a$

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

not LR(0)

not SLR(1)



SR conflict in LR(0)

a ∈ Follow(S)
SR Conf in SLR(1)

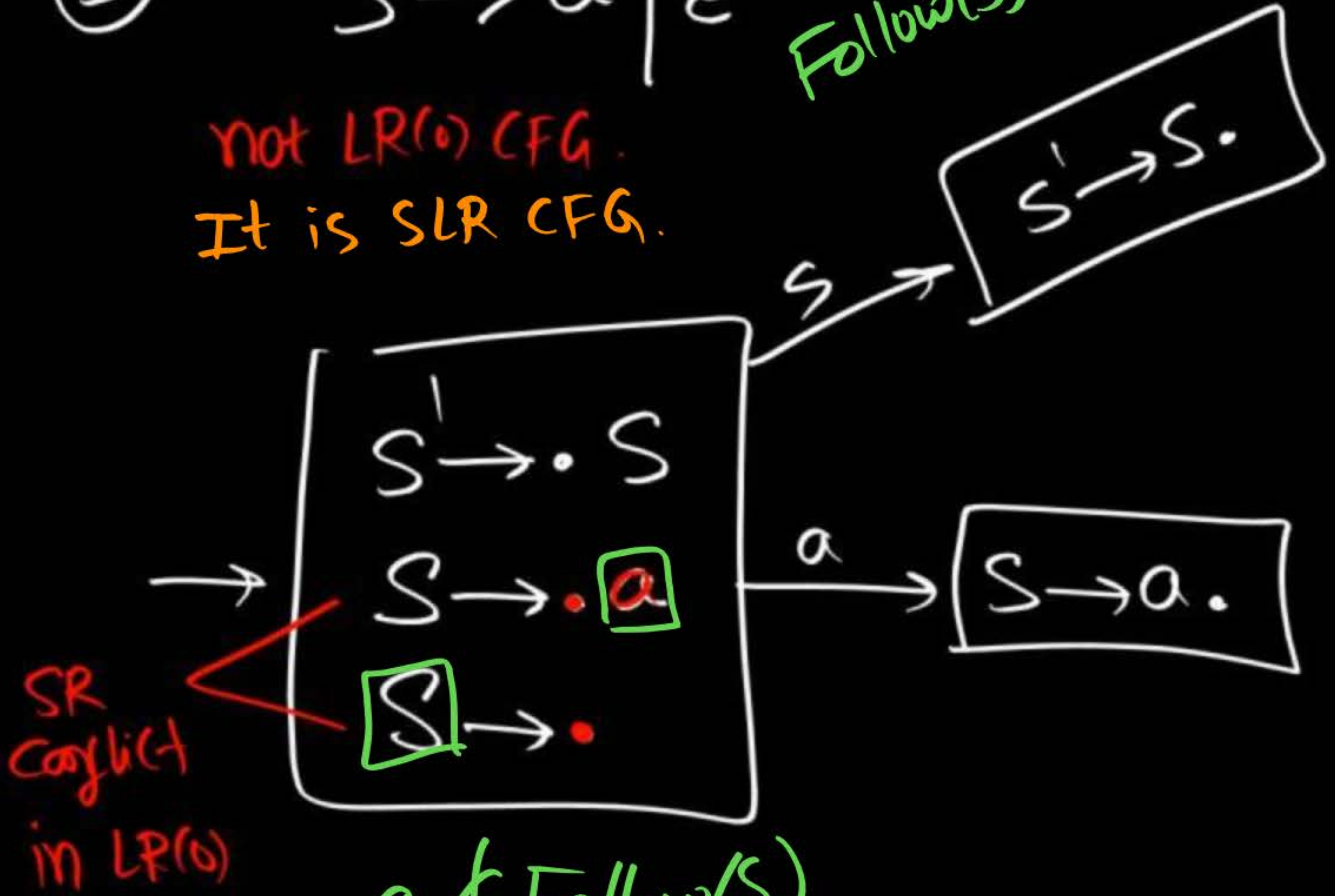
Note: State Item not parcipates in a conflict.

⑤ $S \to a \mid \varepsilon$

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

not LR(0) CFG.

It is SLR CFG.



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

$x\varepsilon = \varepsilon x = x$

$\cdot\varepsilon = \varepsilon\cdot = \cdot$

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

SR conflict

Shift Item
Reduced Item
⋮

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

RR conflict

Reduced $Item_1$
Reduced $Item_2$
⋮

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).

⑥ $S \rightarrow Aa \mid Bb$

$A \rightarrow f$

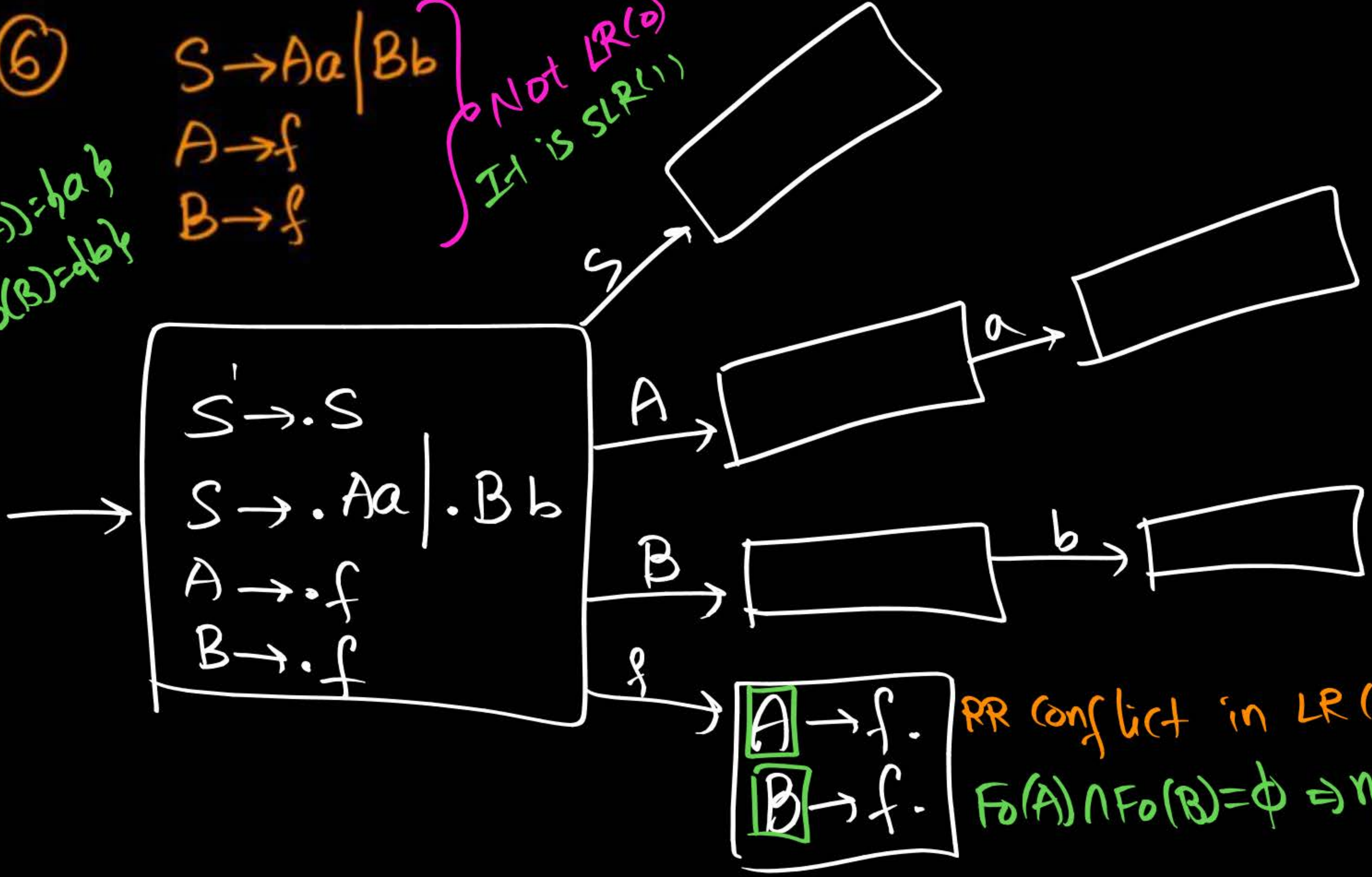
$B \rightarrow f$

Not LR(0)

It is SLR(1)

$Fo(A) = \{a\}$

$Fo(B) = \{b\}$



RR conflict in LR (0)

$Fo(A) \cap Fo(B) = \phi \Rightarrow$ no RR conf in SLR(1)

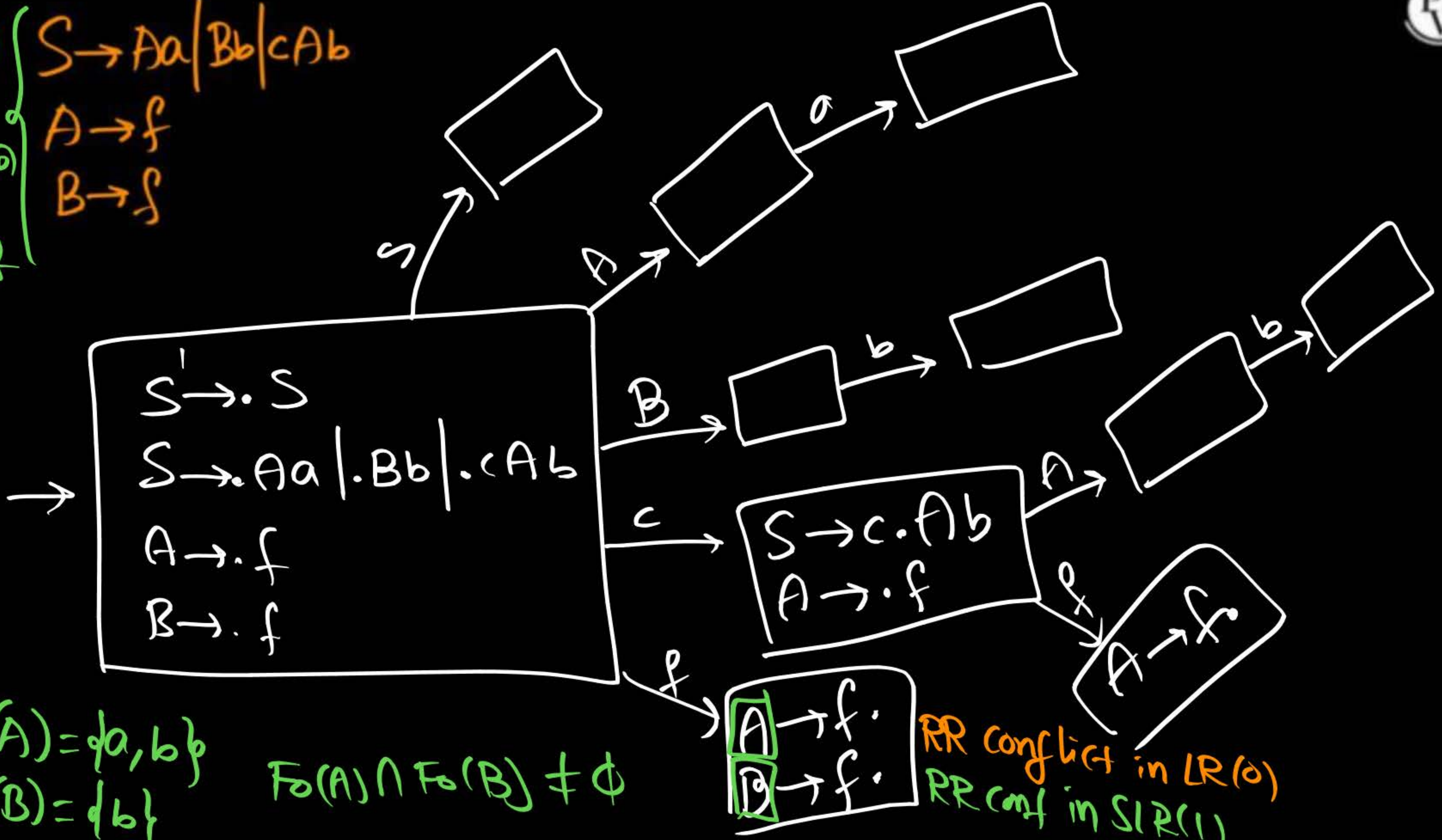
⑦ $S \to Aa \mid Bb \mid cAb$

$A \to f$

$B \to f$

Not LR(0)

Not SLR



RR conflict in LR(0)

RR conf in SLR(1)

$Fo(A) = \{a, b\}$

$Fo(B) = \{b\}$

$Fo(A) \cap Fo(B) \neq \phi$

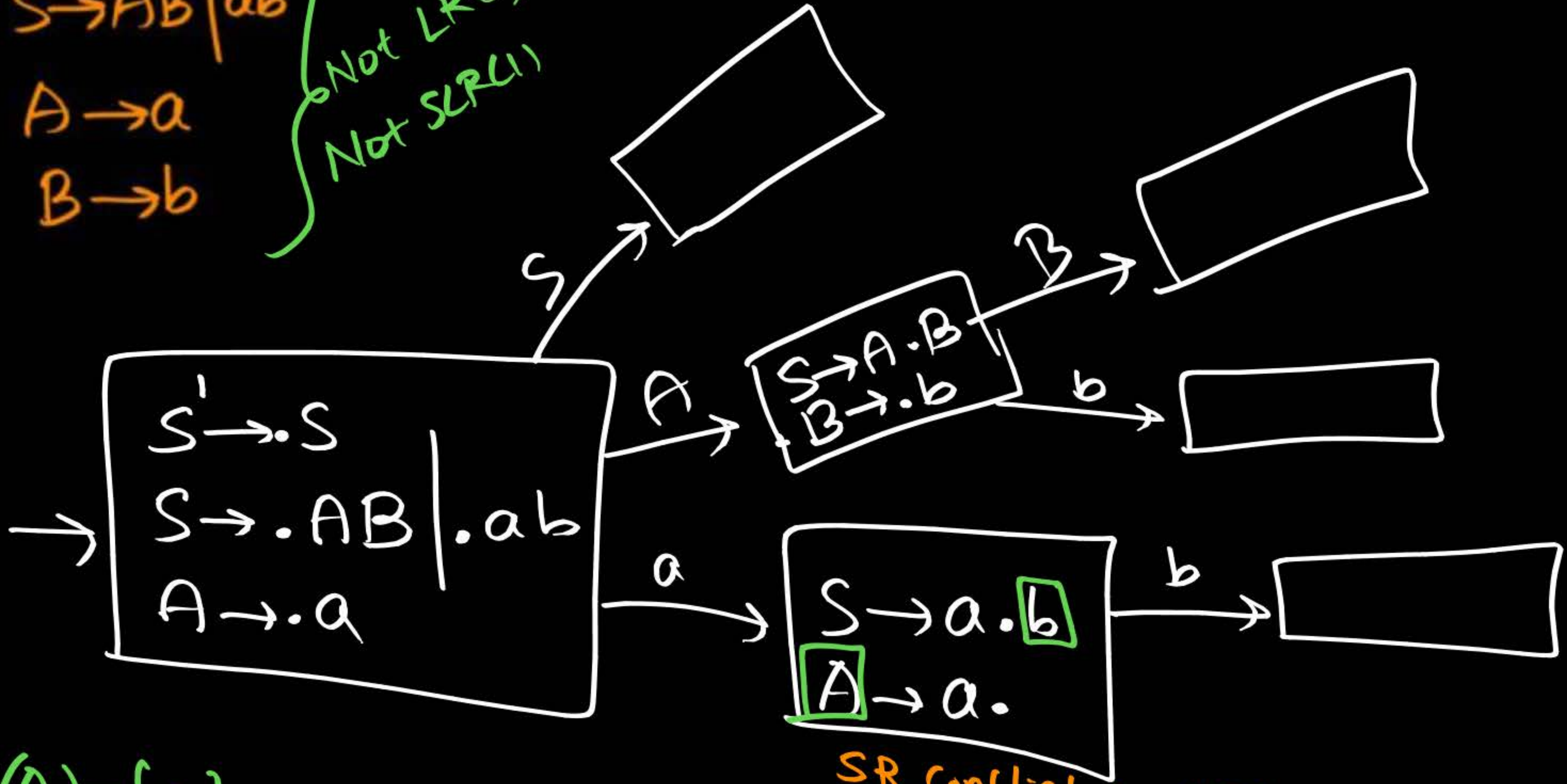
(8) $S \to AB \mid ab$

$A \to a$

$B \to b$

Not LR(0)

Not SLR(1)



FO(A) = {b}

SR conflict in LR(0)

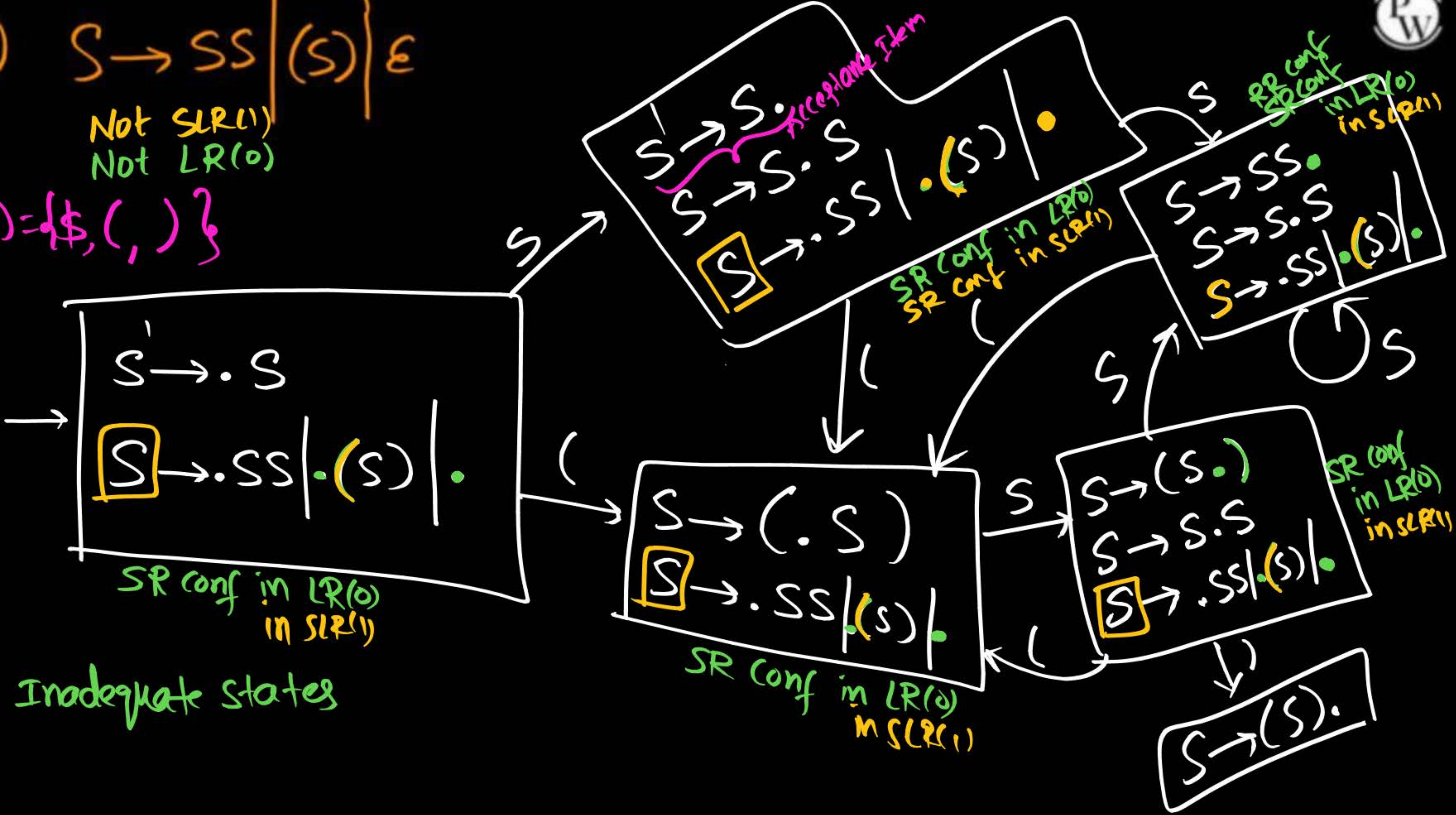
b ∈ FO(A) ⇒ SR conf in SLR(1)

⑨ $S \rightarrow SS \mid (S) \mid \varepsilon$

Not SLR(1)

Not LR(0)

$Fo(S) = \{\$, (,)\}$



5 Inadequate states

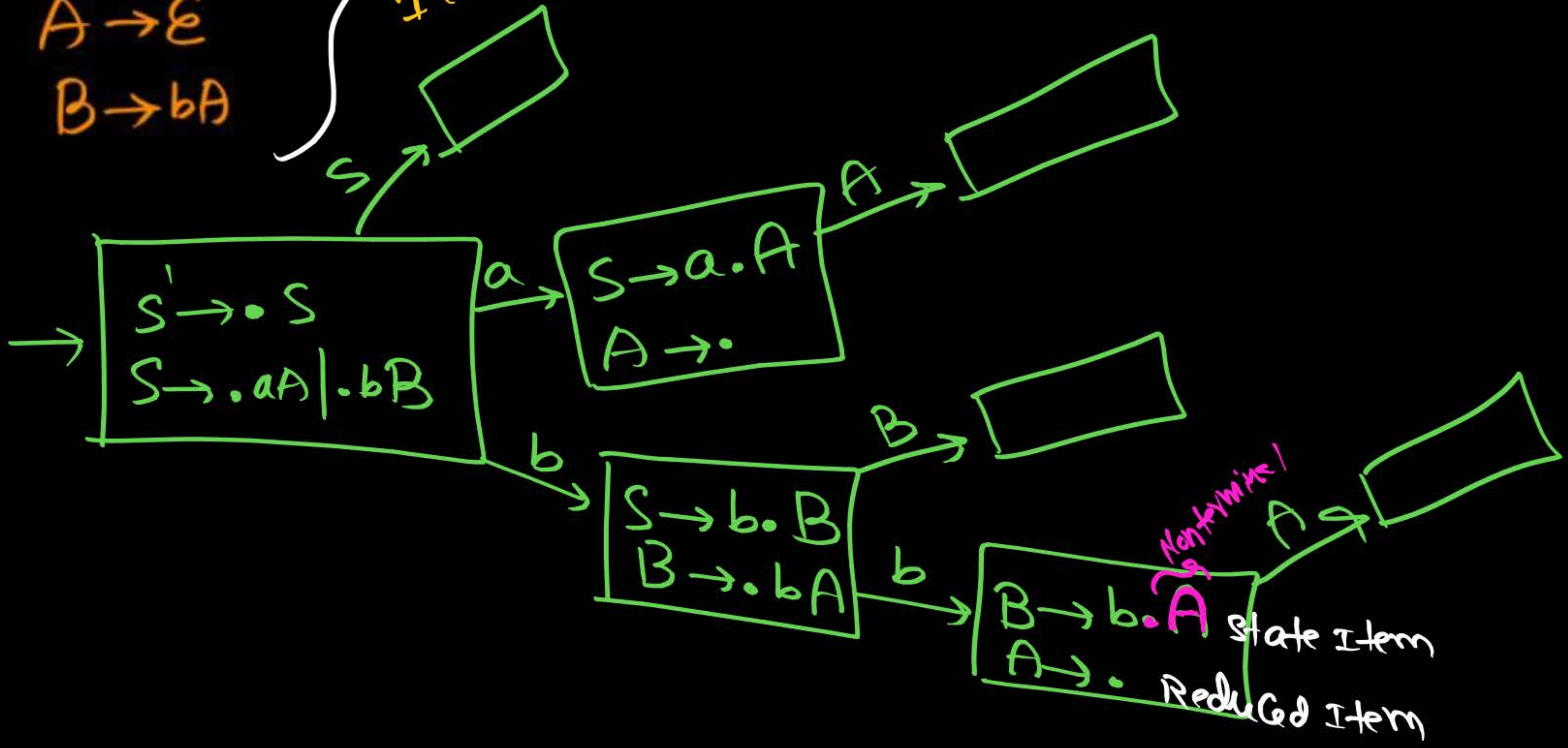
(10) $S \rightarrow aA \mid bB$

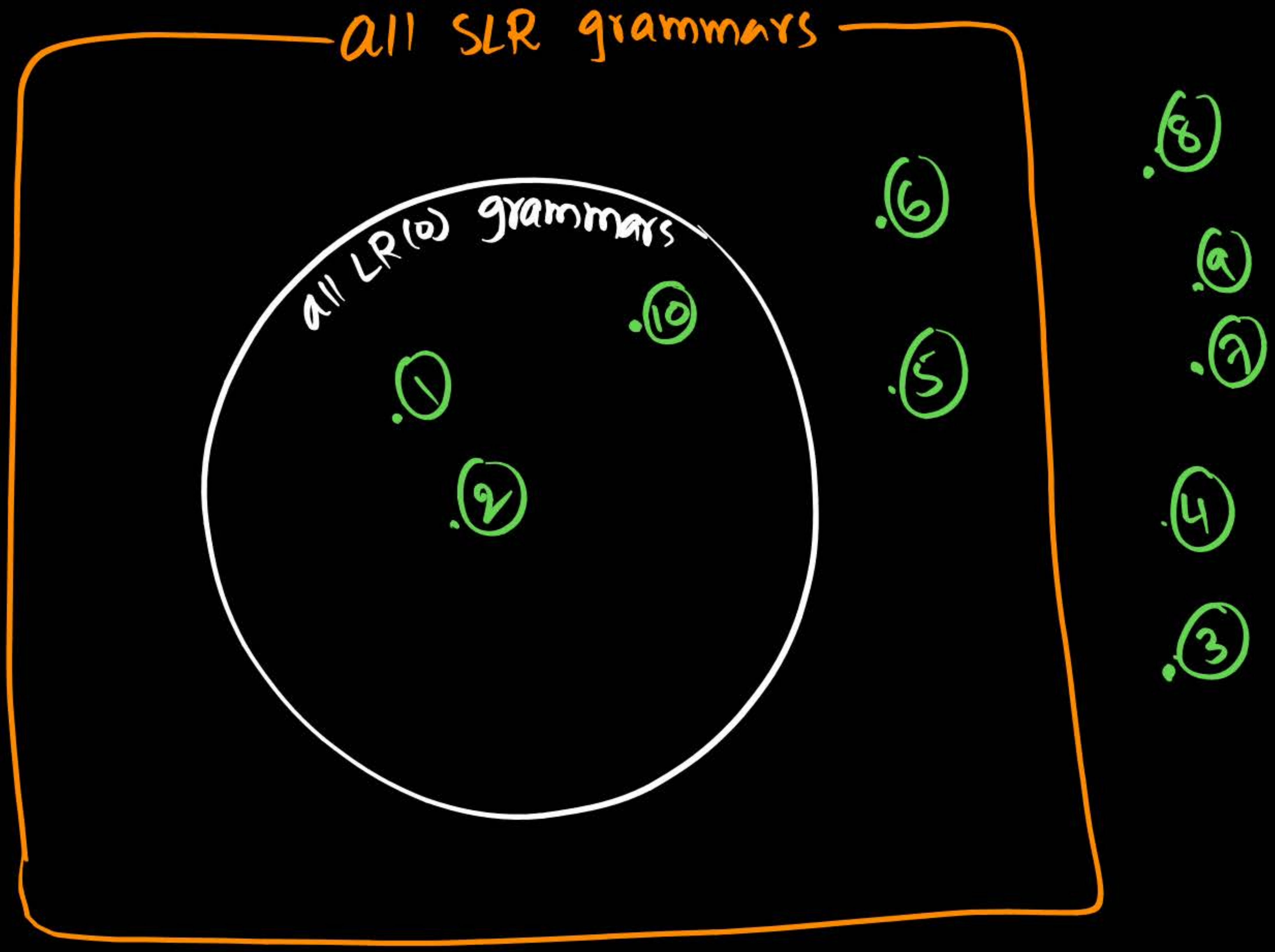
$A \rightarrow \epsilon$

$B \rightarrow bA$

It is LR(0)

It is SLR(1)




all SLR grammars
all LR(0) grammars
1
2
10
6
5
8
9
7
4
3

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

Note: SLR(1) also called as SLR.

→ Step1: Construct LR(0) DFA

Step2: check conflicts in SLR(1)

If there is no conflict then CFG is SLR(1).

S → Simple

L → Left to Right scan

R → Reverse of RMD

1 → one look-ahead

→ computed using whole CFG

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

SR conflict

Shift Item
Reduced Item
⋮

$X \to \alpha . \boxed{t} \beta$

$Y \to \alpha .$

If $t \in$ Follow(Y) then SR conflict in SLR(1)

⋮
$A \to b . \boxed{a}$
$\boxed{B} \to b$

If $a \in$ Fo(B)

RR conflict

Reduced Item_1,
Reduced Item_2
⋮

$\boxed{X} \to \alpha .$

$\boxed{Y} \to \alpha .$

If Follow(X) $\cap$ Follow(Y) $\neq \phi$ then RR conflict in SLR(1)

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

SR conflict in SLR(1):

"terminal of Shift Item" present in "follow of Reduced Item"
(t)

$t \in \text{Follow}(\underset{\text{LHS}}{\text{Reduced Item}})$

SR conflict in SLR(1)

$A \rightarrow b.a$

$D \rightarrow b.$

$\vdots$

If $a \in \text{Follow}(D)$

RR conflict in SLR(1)

$X \rightarrow a.$

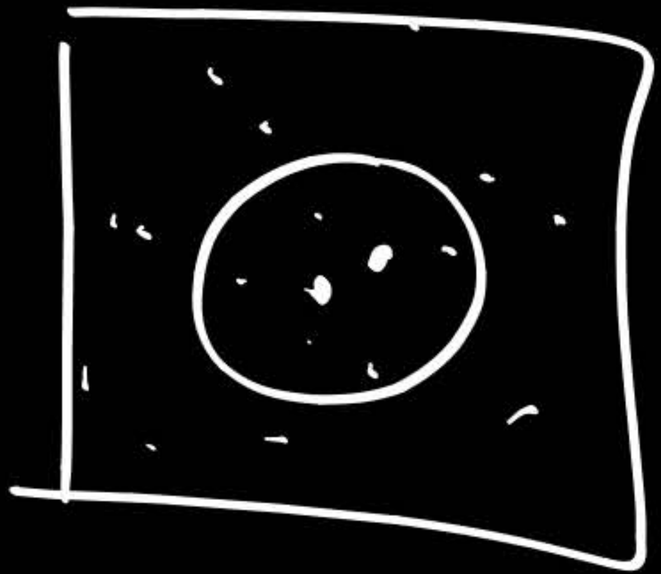
$Y \rightarrow a.$

$\vdots$

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

$$X = \left\{ \underset{S \rightarrow SS \mid a}{LR(0)\ CFG_1},\ LR(0)\ CFG_2,\ \ldots\ldots \right\}$$

$$Y = \left\{ \overset{S \rightarrow aA \mid b}{SLR(1)\ CFG_1},\ SLR(1)\ CFG_2,\ \ldots\ldots \right\}$$



$$X \subset Y$$

$$\left.\begin{array}{l}\text{LR(0) parser}\\[2ex] \text{SLR(1) parser}\end{array}\right\}\text{Depends on LR(0) Items}$$

$$X \to \alpha . \beta$$

LR(0) Item

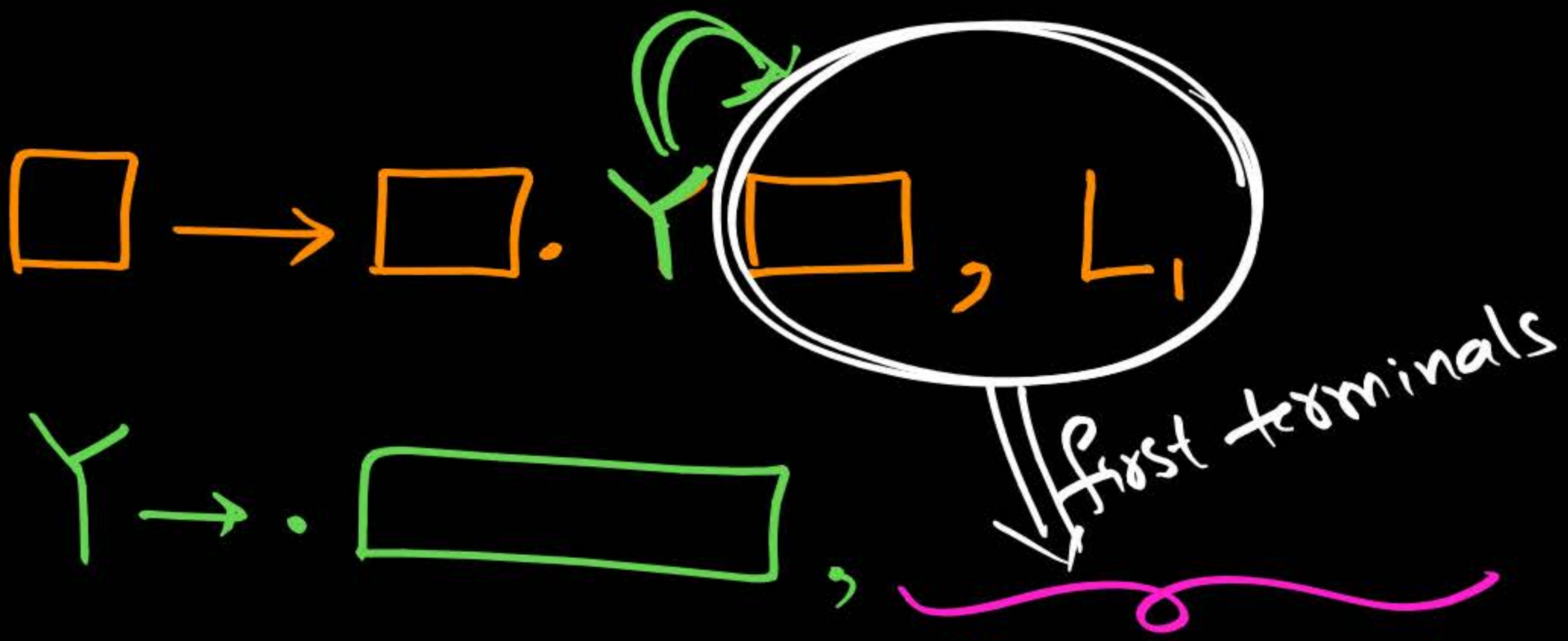
$$\left.\begin{array}{l}\text{CLR parser}\\[2ex] \text{LALR parser}\end{array}\right\}\text{Depends on LR(1) Items}$$

LR(1) Item :

LR(0) Item + Look-a-head set

$$\boxed{\underbrace{X \to \alpha . \beta}_{\text{Actual core production}} , \underbrace{L}_{\text{Look-a-head portion}}}$$

How to compute look-a-head set in LR(1) Item?



How to compute look-a-heads of Y?

$A \rightarrow a.\boxed{B}b, \{c, d\}$

$B \rightarrow .ef, \underline{\{b\}}$

$\text{FIRST}(\underline{b}c, \underline{b}d)$

$= \{b\}$

$A \rightarrow a.B, \{c, d\}$

$B \rightarrow .ef, \underline{\{c, d\}}$

$$A \rightarrow a \cdot b\ ,\ \{c, d, e\}$$

OR

$$A \rightarrow a \cdot b\ ,\ c/d/e$$

OR

$A \rightarrow a \cdot b\ ,\ L_1$ where $L_1 = \{c, d, e\}$

OR

$A \rightarrow a \cdot b L_1$ where L_1 is look-a-heads of A

$L_1 = \{c, d, e\}$

$X \to \alpha . Y\beta , L$

$Y \to . \gamma ,\quad \underline{\text{FIRST}(\beta L)}$

How to compute Look-a-heads of Y ?

C L R (1)

C → Canonical (exact)

L → Left to right scan

R → Reverse of RMD

1 → computed based on exact path

1 → Look-a-head

CLR(1) also called as LR(1)

CLR(1)

LR(1)

CLR

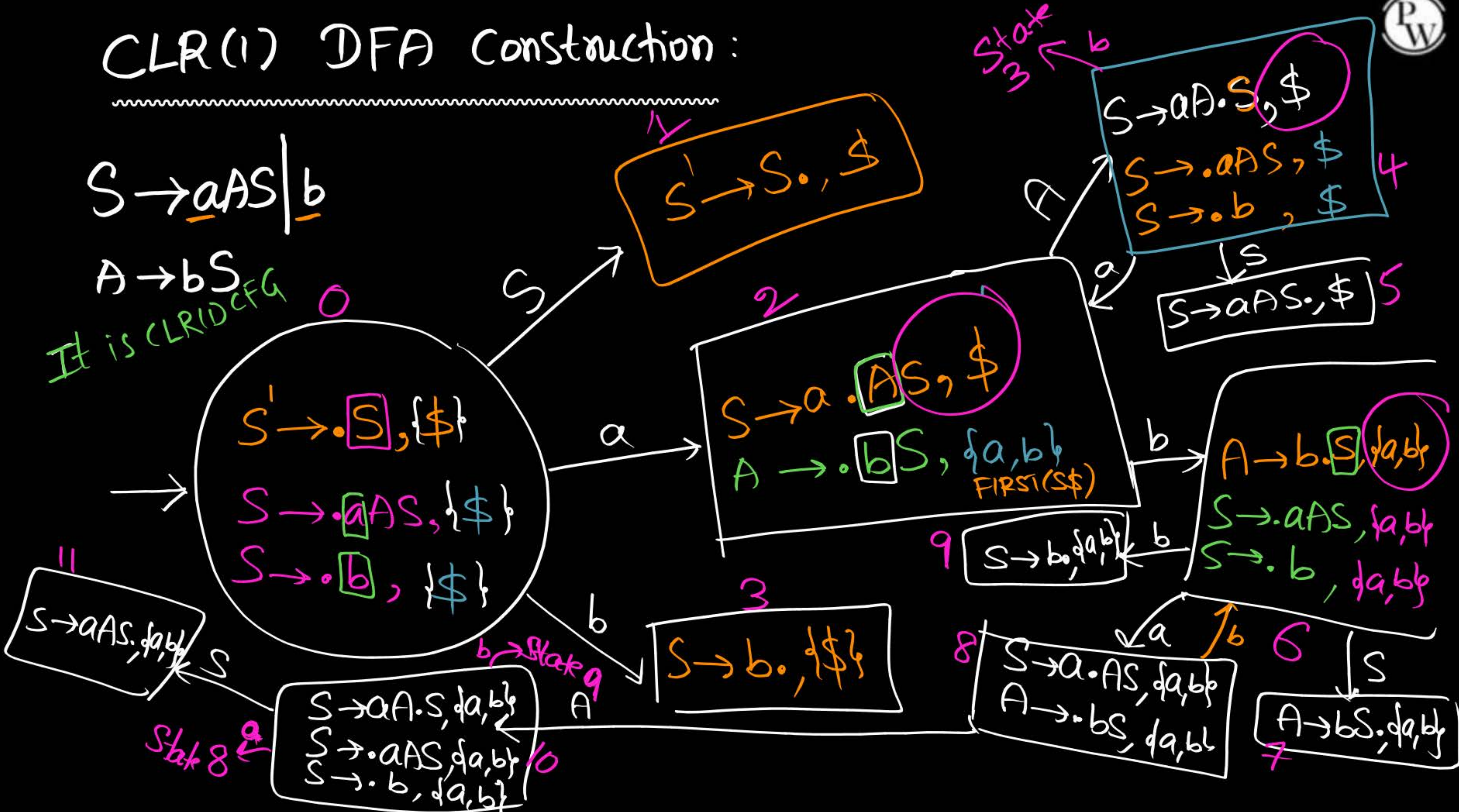
LR

CLR(1) DFA Construction:

$S \rightarrow aAS \mid b$

$A \rightarrow bS$

It is CLR(1) CFG



LALR(1) DFA

→ Step 1: Construct CLR DFA

Step 2: Merge state of CLR if they have same items but look-a-heads may be different.

LALR(1) DFA Construction :

$S \rightarrow aAS \mid b$

$A \rightarrow bS$

Merge(4 and 10) = 410

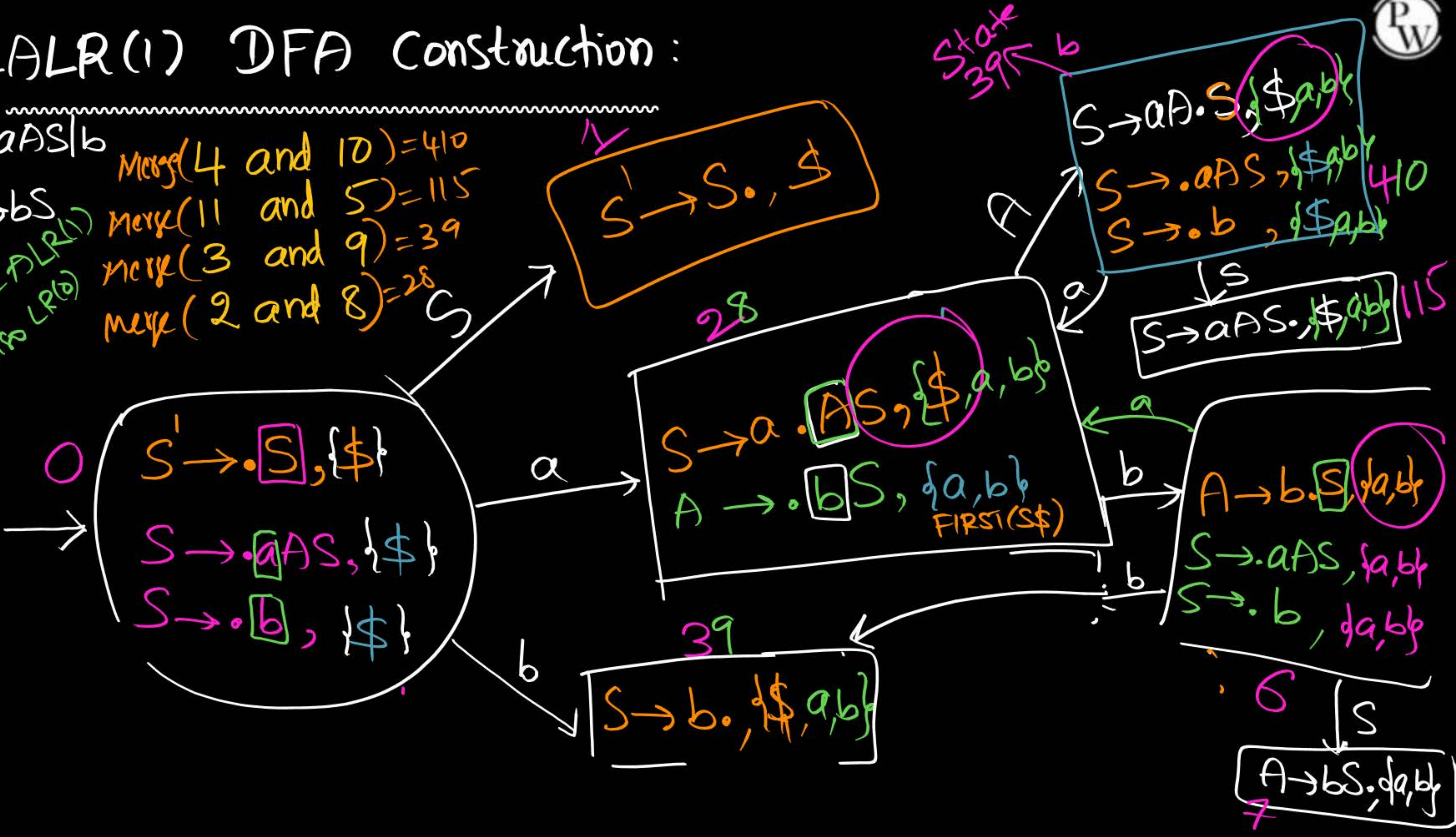
Merge(11 and 5) = 115

Merge(3 and 9) = 39

Merge(2 and 8) = 28

It is LALR(1)

It is also LR(0)



Conflicts checking in both CLR and LALR:

SR conflict

$$\begin{tikzcd} \text{Shift: } X \to \alpha . & t \arrow[d, bend right] & \beta , L_1 \\ \text{Reduce: } Y \to \alpha . , & \boxed{L_2} & \end{tikzcd}$$

⋮

If $t \in L_2$

RR conflict

$$\begin{tikzcd} \text{Reduce}_1 \quad X \to \alpha . , & \boxed{L_1} \arrow[d, leftrightarrow] \\ \text{Reduce}_2 \quad Y \to \alpha . , & \boxed{L_2} \end{tikzcd}$$

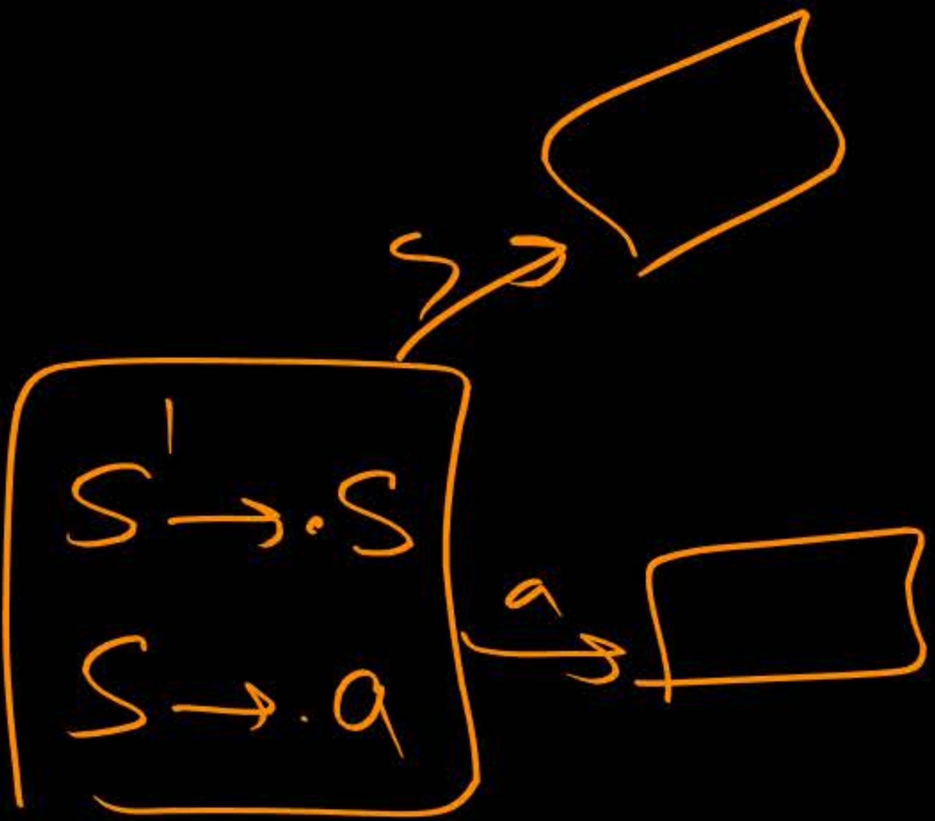
⋮

If $L_1 \cap L_2 \neq \phi$

No. of states

$$\left.\begin{array}{c}\text{no. of states in LR(0)}\\ = \\ \text{no. of states in SLR(1)}\\ = \\ \text{no. of states in LALR}\end{array}\right\} \leq \begin{array}{c}\text{no. of states}\\ \text{in CLR}\end{array}$$

(2) $S \to a$



[✓ A) LL(1)]

[✓ B) LR(0)]

[✓ C) SLR]

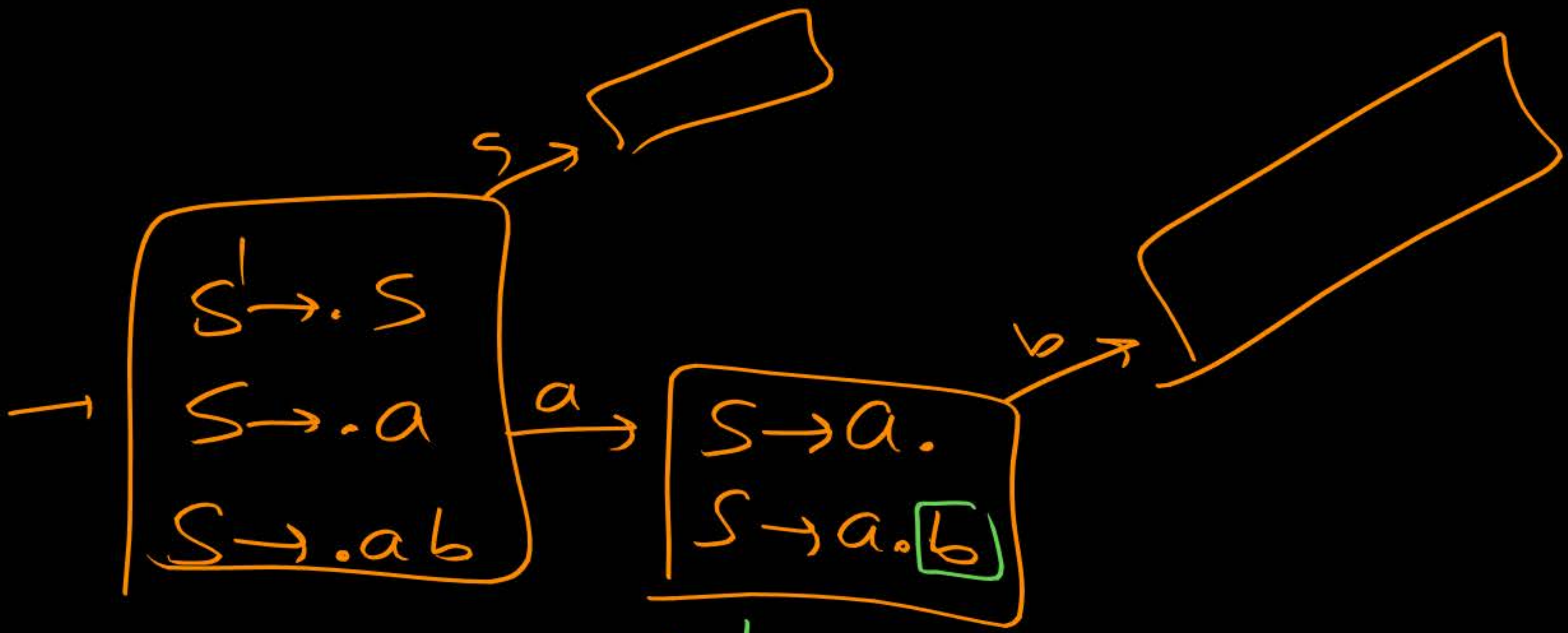
[✓ D) CLR]

[✓ E) LALR]

F) Ambiguous

③ $S \rightarrow a \mid ab$

$\{a\} \cap \{a\} \neq \phi \Rightarrow$ Not LL(1)



$b \notin Fo(S)$

~~A) LL(1)~~

~~B) LR(0)~~

[✓ C) SLR]

[✓ D) CLR]

[✓ E) LALR]

F) Ambiguous

4 $S \rightarrow SS \mid a$

~~A)~~ LL(1)

~~B)~~ LR(0)

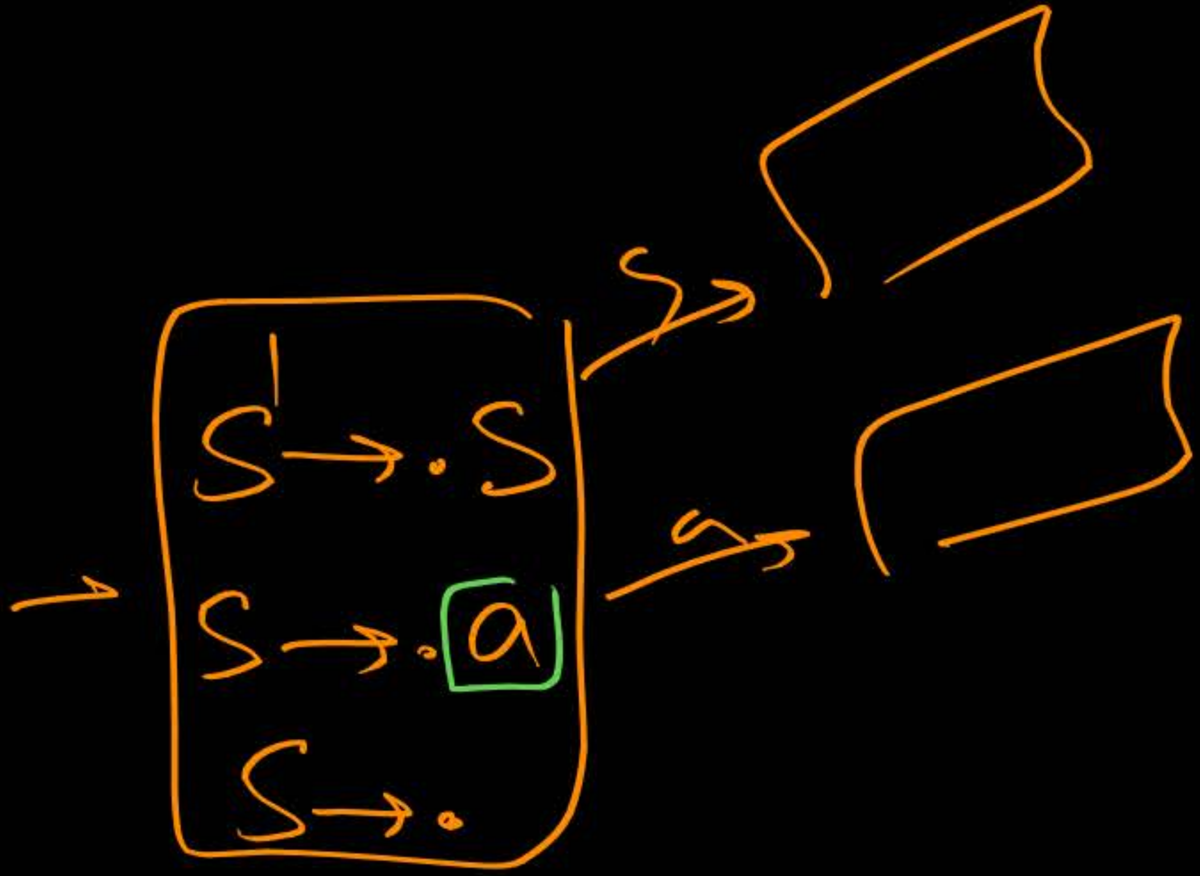
~~C)~~ SLR

~~D)~~ CLR

~~E)~~ LALR

[✓ F)] Ambiguous

⑤ $S \rightarrow a \mid \varepsilon$



$a \notin Fo(S)$

✓ A) LL(1)

✗ B) LR(0)

✓ C) SLR

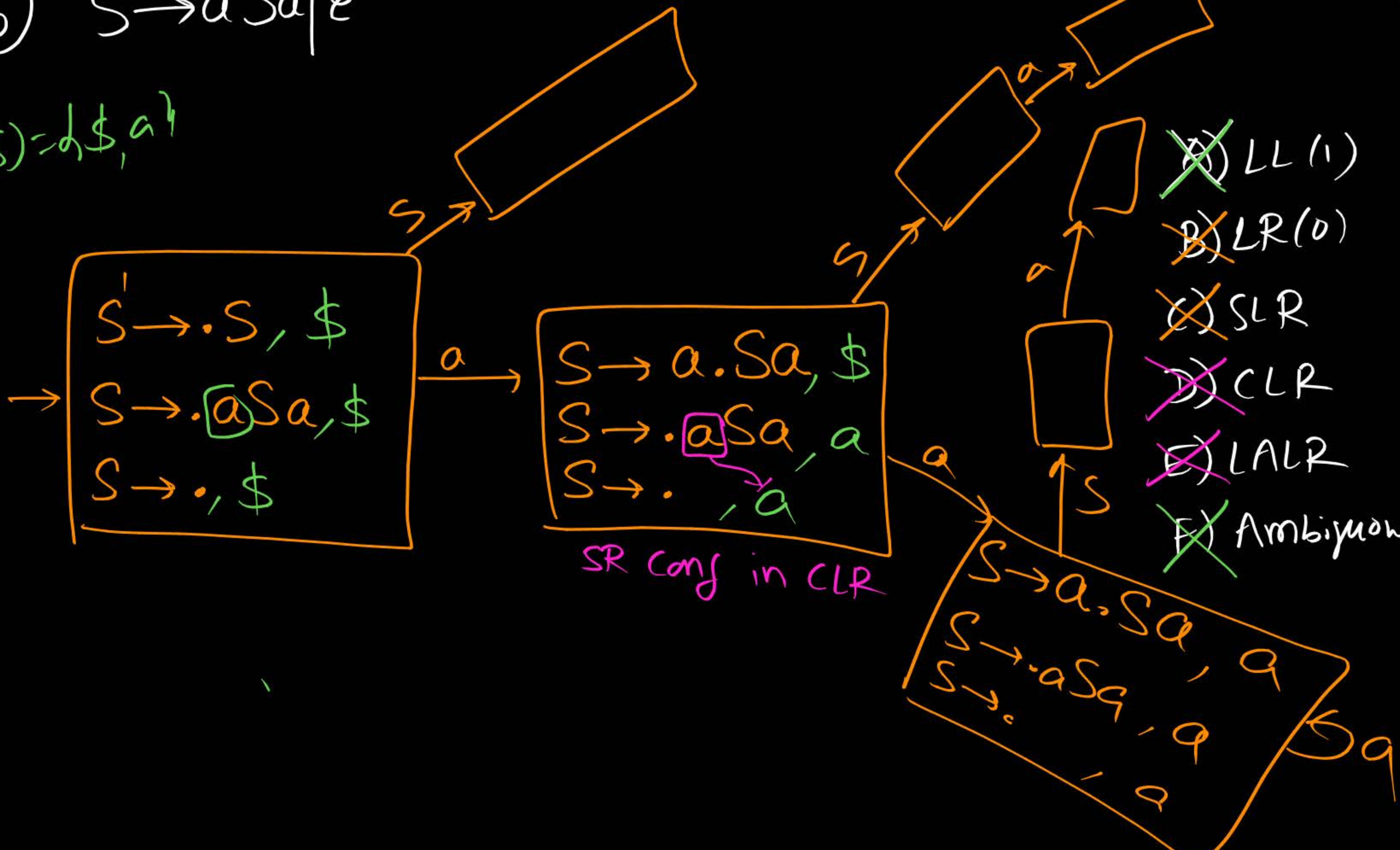
✓ D) CLR

✓ E) LALR

✗ F) Ambiguous

⑥ $S \rightarrow aSa \mid \varepsilon$

$Fol(S) = \{\$, a\}$



SR conf in CLR

~~A)~~ LL(1)

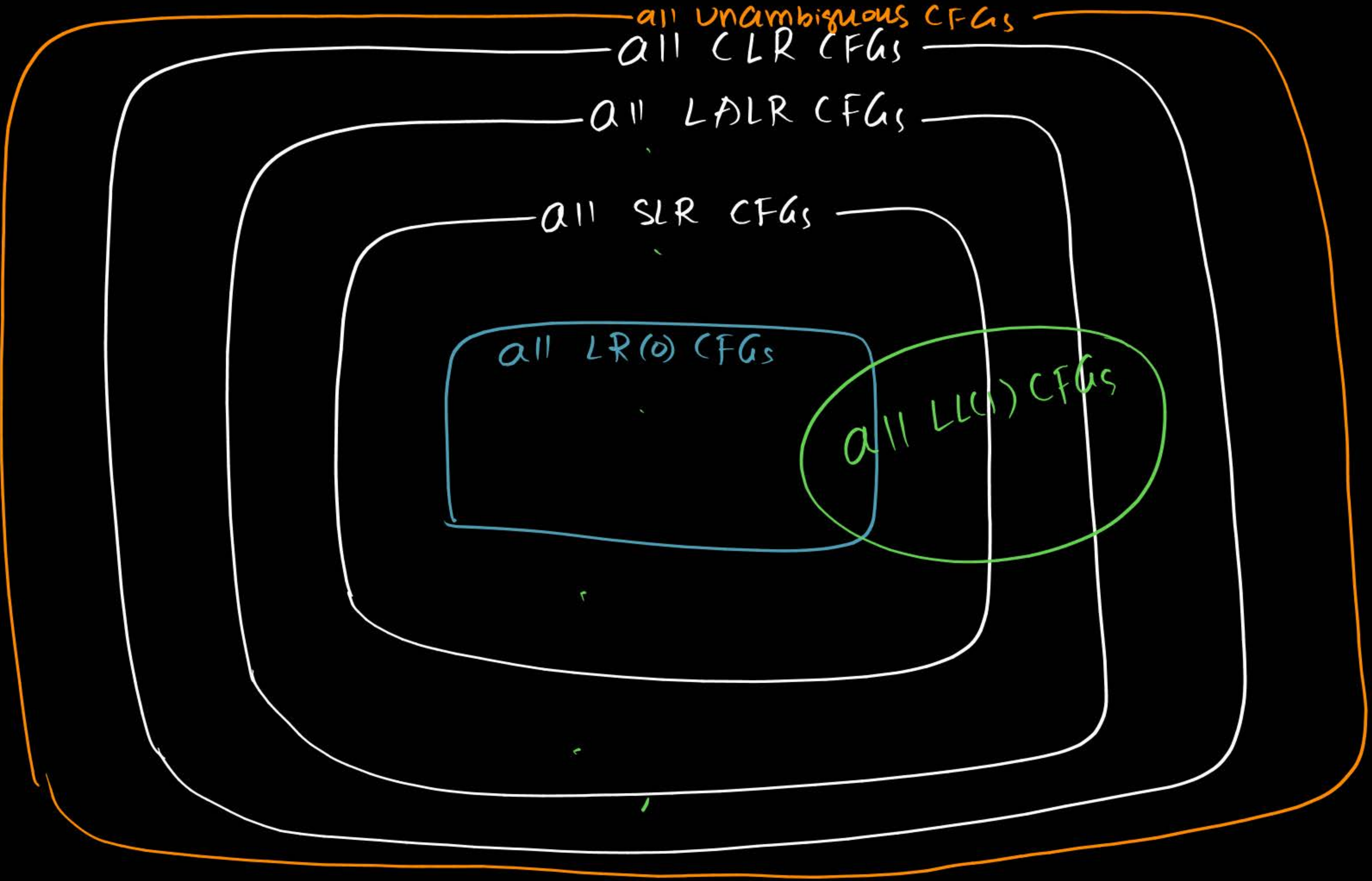
~~B)~~ LR(0)

~~C)~~ SLR

~~D)~~ CLR

~~E)~~ LALR

~~F)~~ Ambiguous


all Unambiguous CFGs
all CLR CFGs
all LALR CFGs
all SLR CFGs
all LR(0) CFGs
all LL(1) CFGs

Note:

I) Every LR(0) grammar is SLR, LALR, CLR, and Unambiguous

II) Every SLR grammar is LALR, CLR, and Unambiguous

III) Every LALR grammar is CLR, and Unambiguous

IV) Every CLR grammar is Unambiguous

V) Every not CLR grammar is not LALR, not SLR, not LR(0)

VI) Every not LALR grammar is not SLR, not LR(0)

VII) Every not SLR grammar is not LR(0).

LL(1) Vs LR(1)

I) Every LL(1) CFG is LR(1) CFG

II) No relation b/w LL(1) and SLR/LALR/LR(0)

III) If LL(1) CFG is free from null rules then always SLR(1).

IV) If LL(1) without unit rules then it is always LALR(1)

$S \rightarrow a \mid b$

[✓ A) LL(1)]

[✓ B) LR(1)]

C) not LL(1)

D) not LR(1)

NULL Rule

$X \to \varepsilon$

UNIT Rule

$X \to Y$

Summary

→ SLR CFG ✓

LALR CFG ✓

CLR CFG ✓

Next : Important relations b/w CLR and LALR

" " b/w parsers

LR Table constructions

Thank you
PW
Soldiers