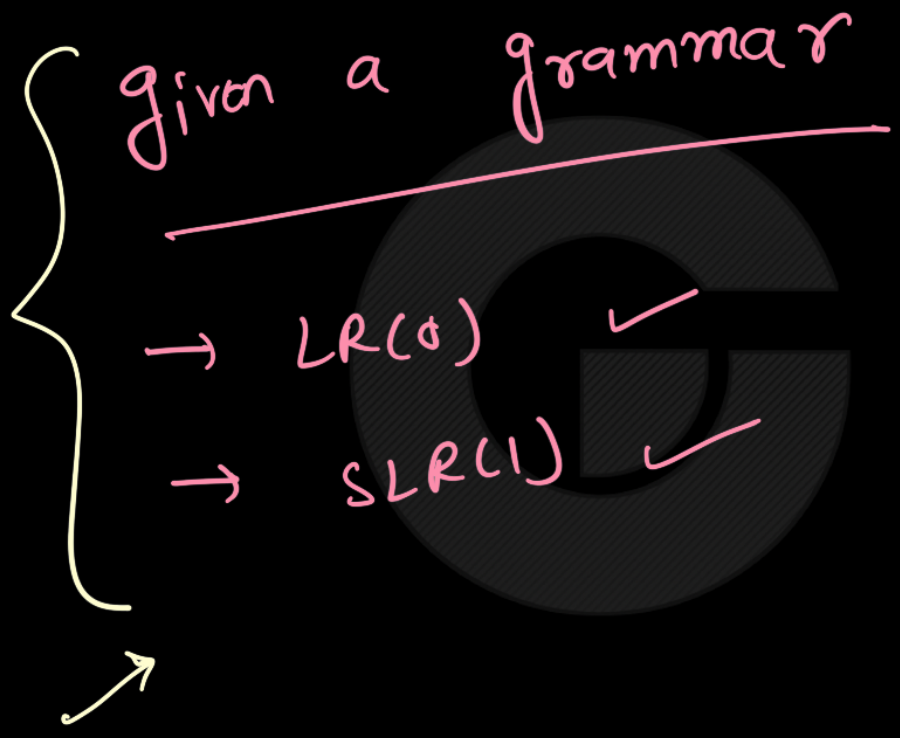
Most amazing and most needed

Re-Run of DFA

Bottom-up parsing will get CRYSTYL CLEAR

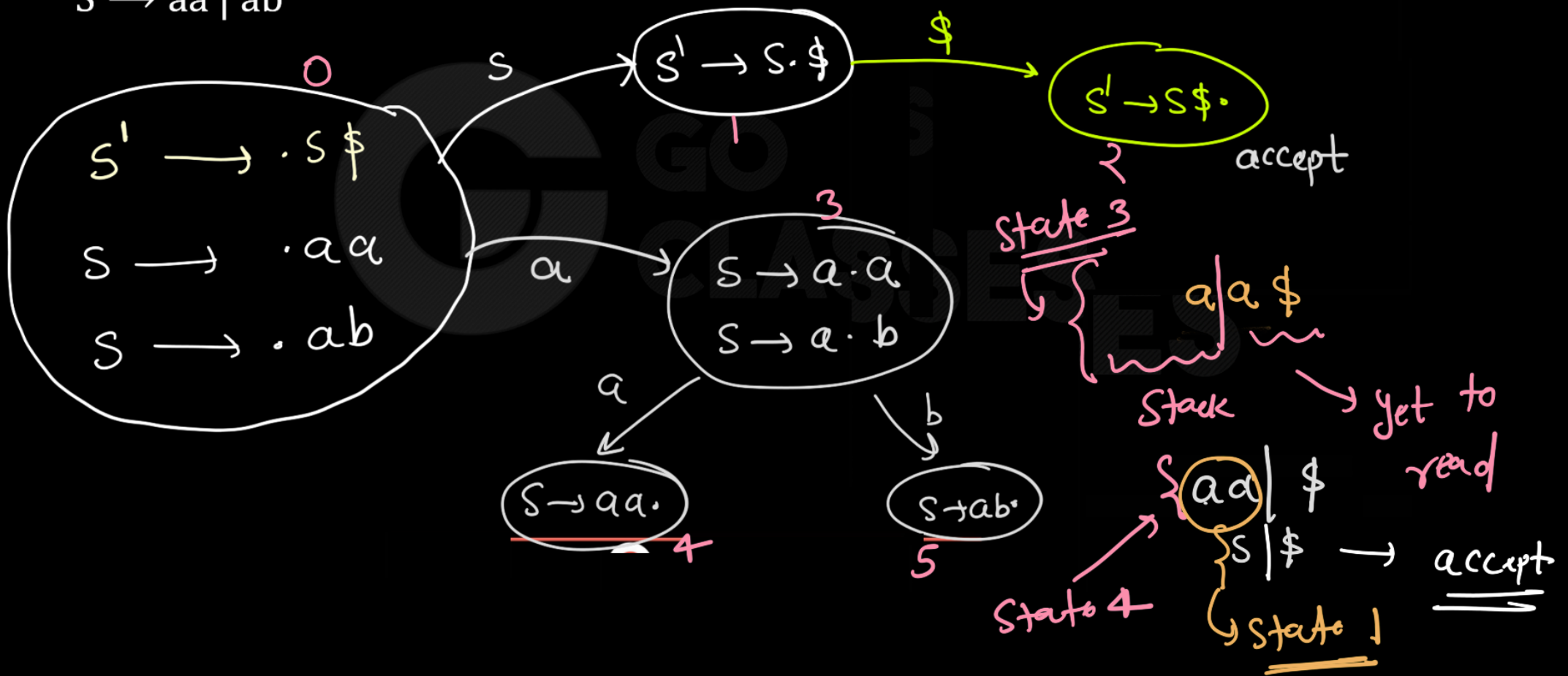
Most amazing

Re-Run of DFA


given a grammar
→ LR(0)
→ SLR(1)

Build LR(0) automaton

$S \rightarrow aa \mid ab$

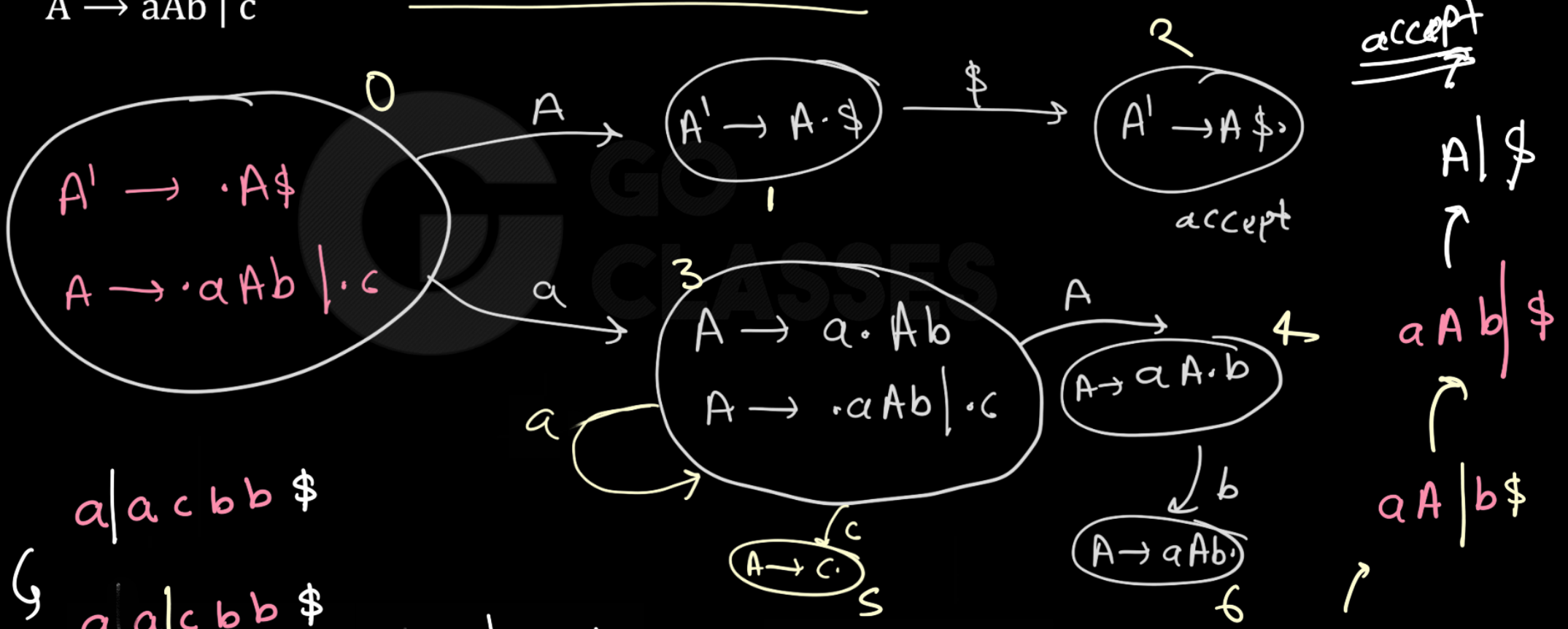


Q: which state the DFA is in?

⇒ run DFA on stack input.

Build LR(0) automaton

$A \longrightarrow aAb \mid c$



a|acbb$

↳ a|a|cbb$

↳ a|acb|b$ → aaA|bb$ → aaAb|b$

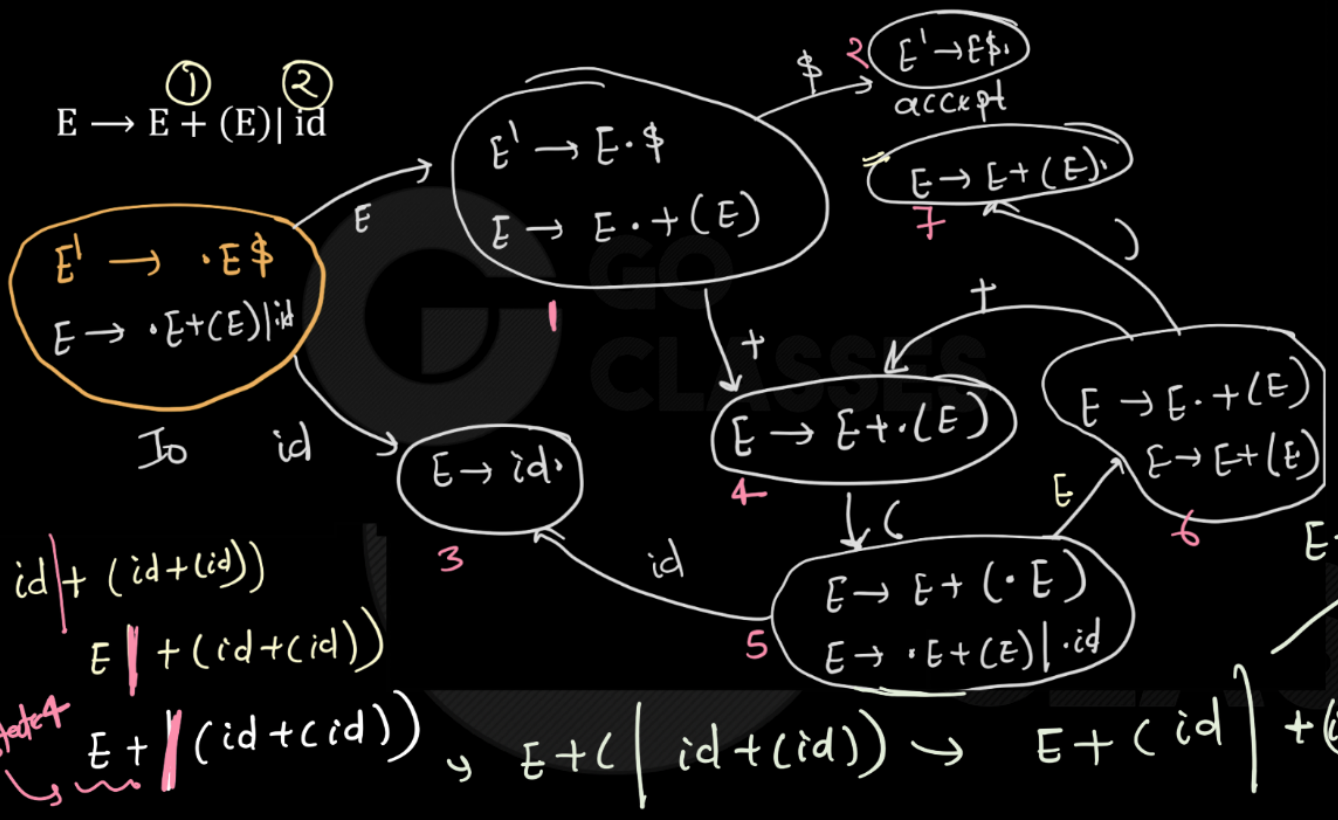
A|$

↑

aAb|$

↑

aA|b$



id | + (id+(id))

E | + (id+(id))

E + | (id+(id))

state 4

E+(| id+(id)) → E+(id | +(id))

E+(E | +(id)) → E+(E+(id |))

E+(E+(E |)) → E+(E+(E) |)

accept

E | $

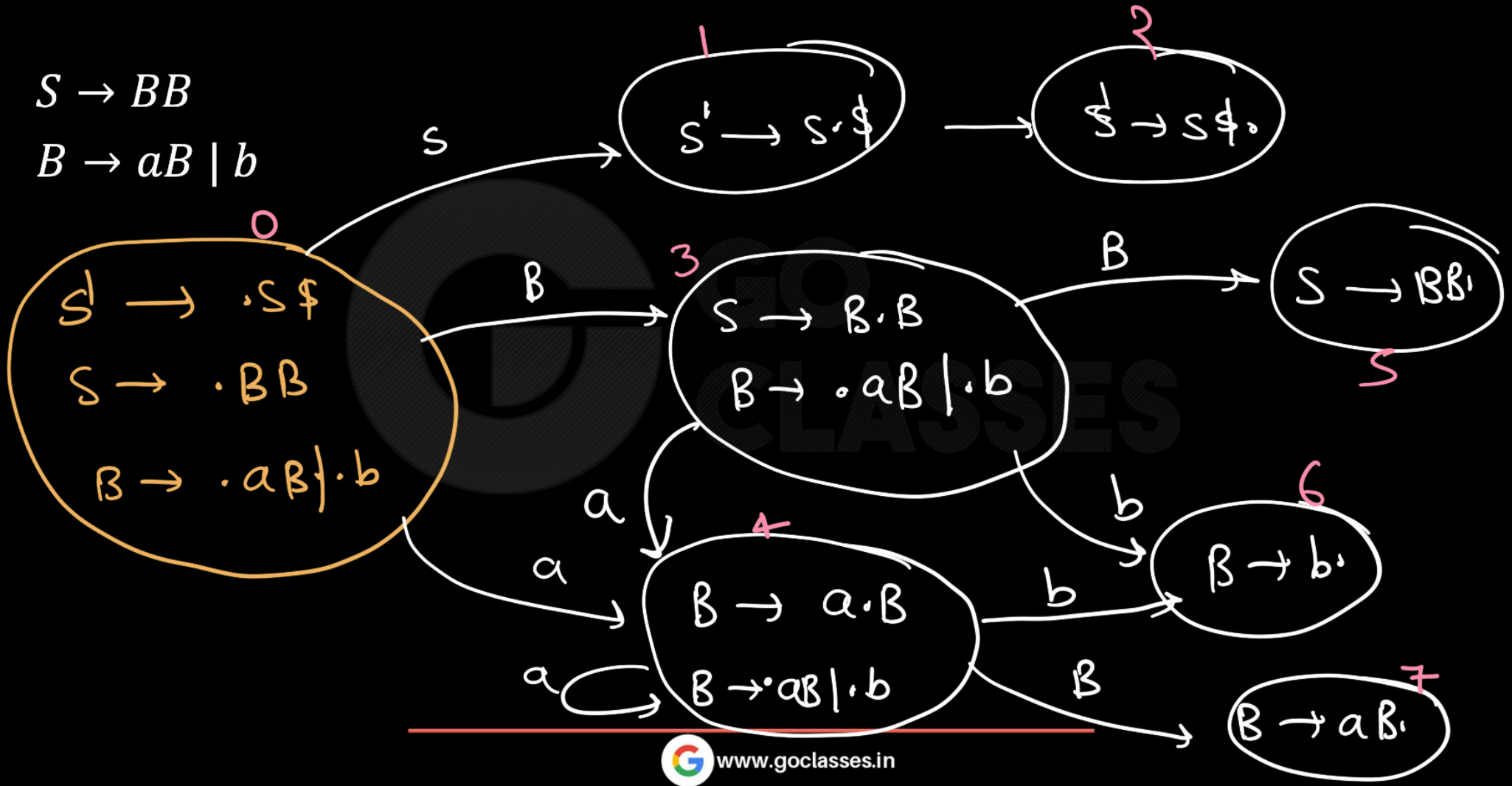
E+(E) | $

state 6

E+(E |) $

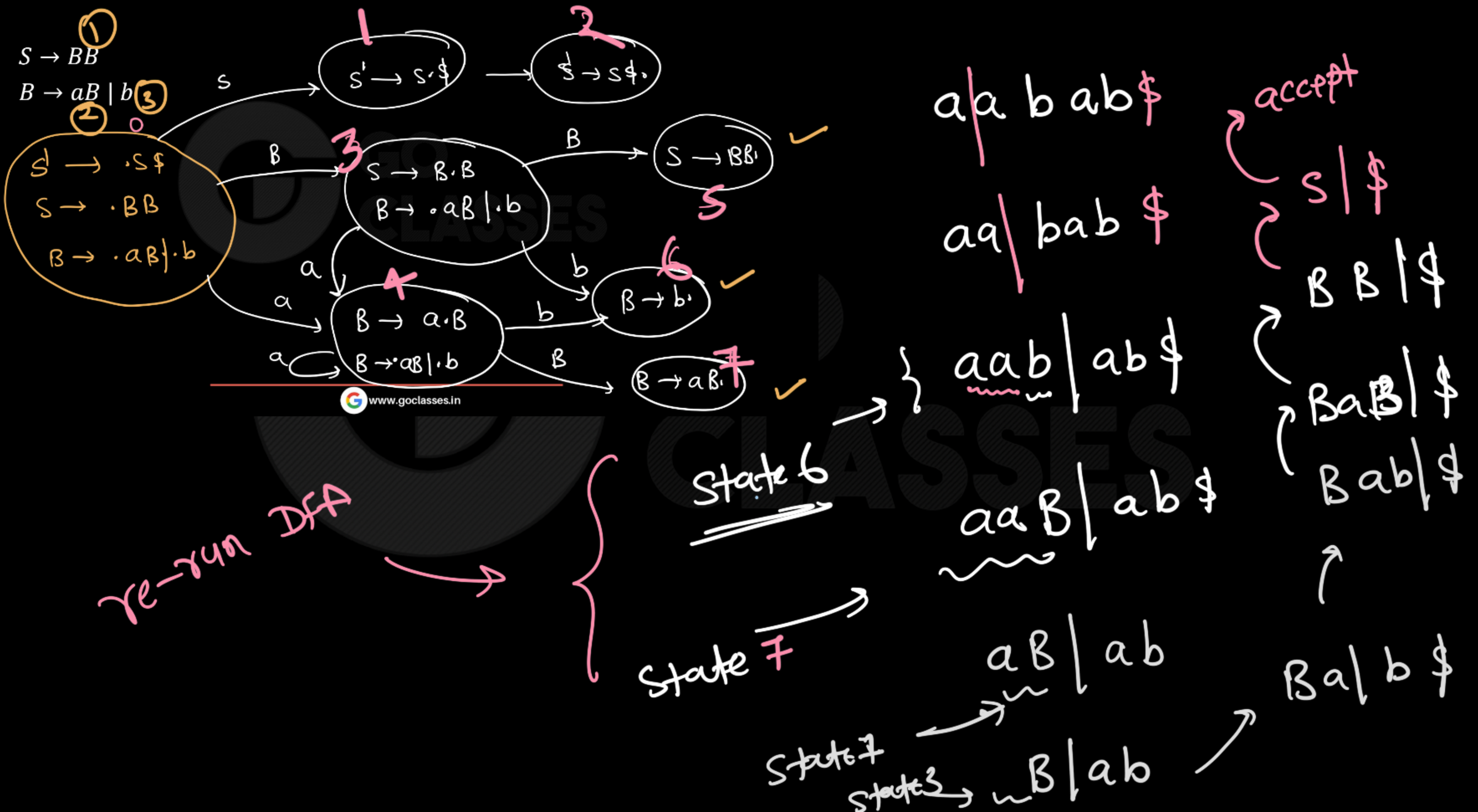
$S \rightarrow BB$

$B \rightarrow aB \mid b$



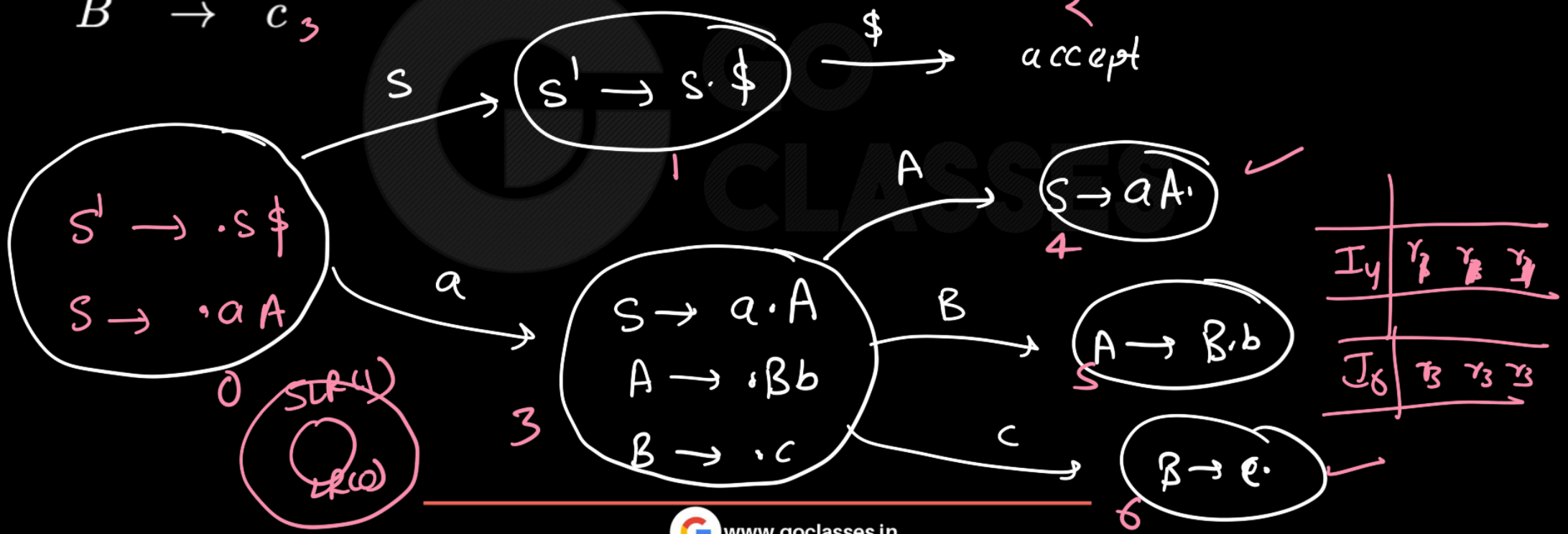
$S \rightarrow BB$

$B \rightarrow aB \mid b$



$S \rightarrow aA$ 1
$A \rightarrow Bb$ 2
$B \rightarrow c$ 3

Is this grammar LR(0) ?
Is this grammar SLR(1) ?



GO CLASSES

(1) (2)

E → E + T | T

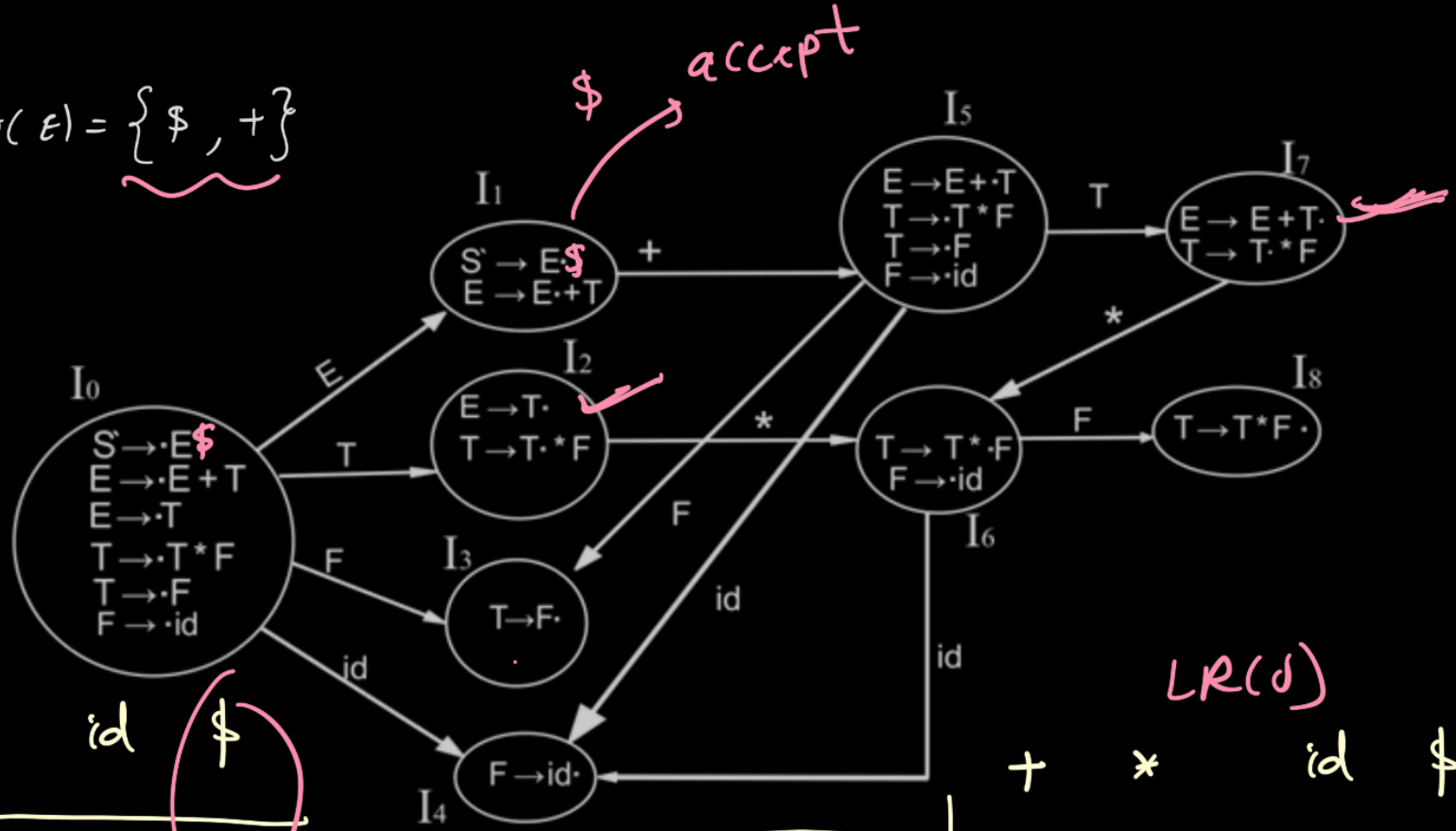
T → T * F | F

F → id

follow(E) = { $, + }

Is this grammar LR(0) ? ✗

Is this grammar SLR(1) ? ✓



SLR(1)

	+	*	id	$
I_2	r_2	S_6		r_2
I_7	r_1	S_6		r_1

LR(1)

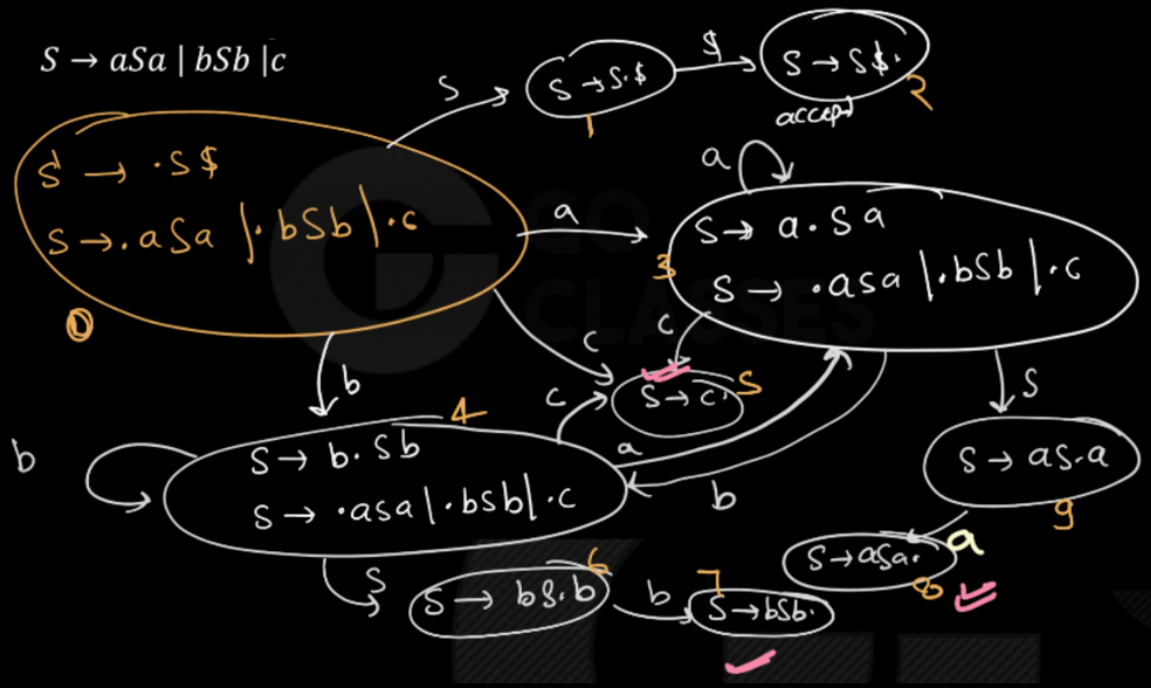
	+	*	id	$
I_2	r_2	r_2/S_6	r_2	r_2
I_7	r_1	r_1/S_6	r_1	r_1

GO CLASSES

Questions based on LR(0) Automata states

MSQ Question

Consider the LR(0) automaton given for the following grammar. Suppose that parser is in state 5 then what are the possible contents on stack ?



A. The stack content could be "bbbc" where 'c' is on the top of the stack.

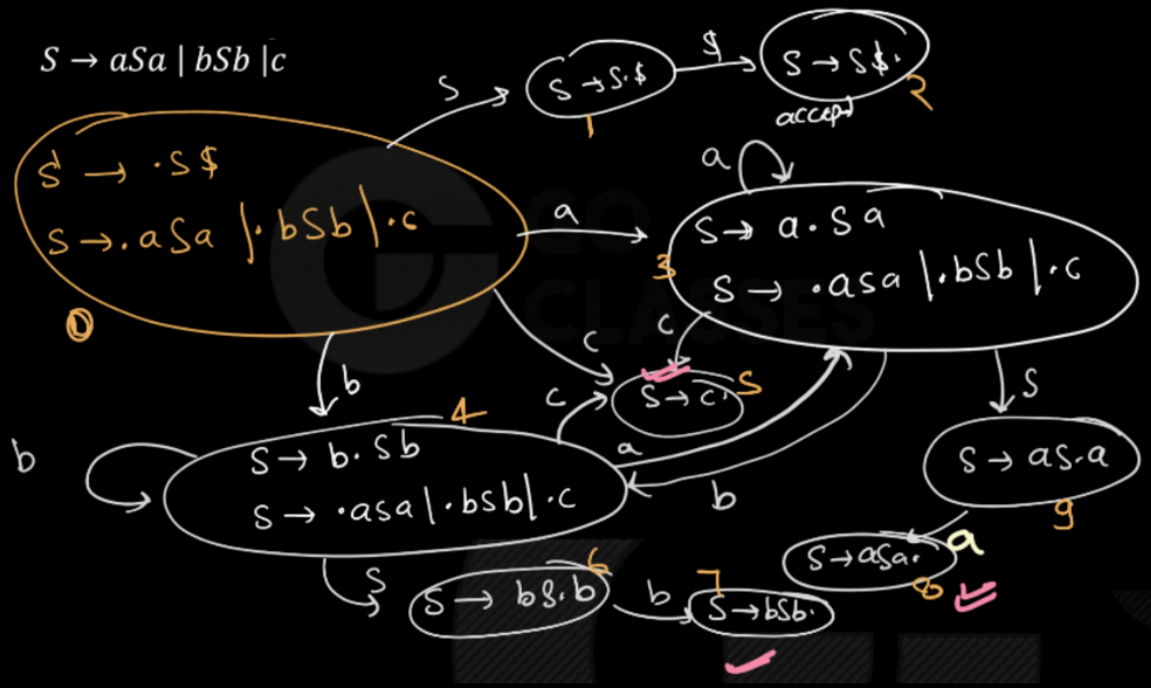
B. The top of the stack must be ‘c’.

C. The stack content could be ”bcbc" where 'c' is on the top of the stack.

D. The stack content could be ”bSc" where 'c' is on the top of the stack.

MSQ Question

Consider the LR(0) automaton given for the following grammar. Suppose that parser is in state 5 then what are the possible contents on stack ?



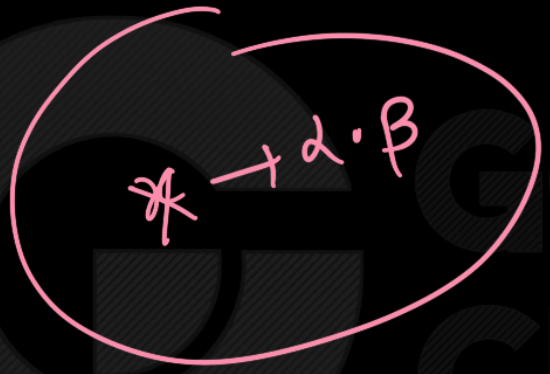
A. The stack content could be "bbbc" where 'c' is on the top of the stack.

B. The top of the stack must be 'c'.

C. The stack content could be "bcbc" where 'c' is on the top of the stack.

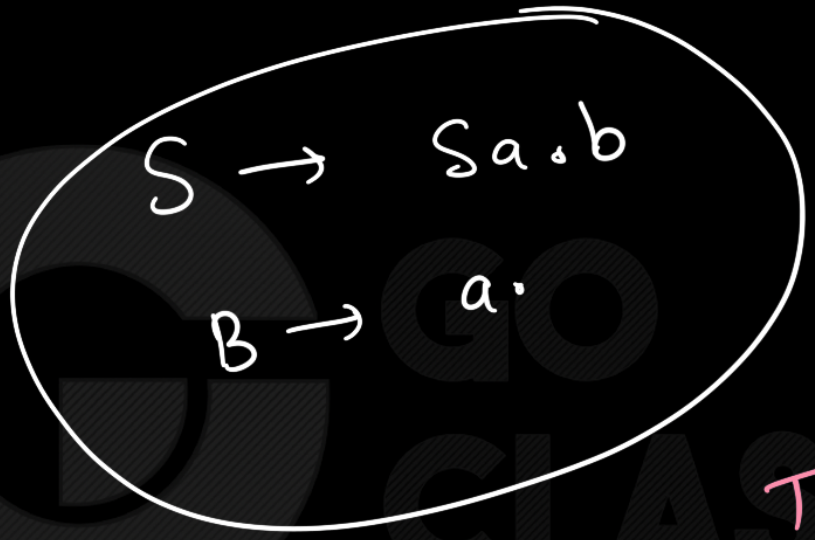
D. The stack content could be "bSc" where 'c' is on the top of the stack.

$Y \rightarrow \alpha \cdot \beta$



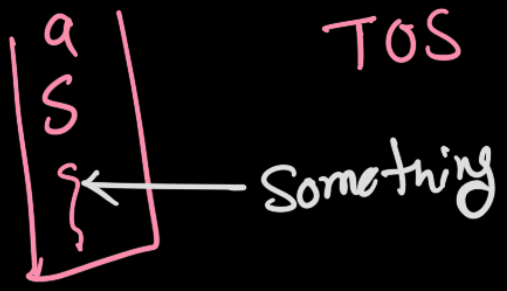
what can you say about this state?

$\Rightarrow \alpha$ must be on TOS



← what is on TOS ?

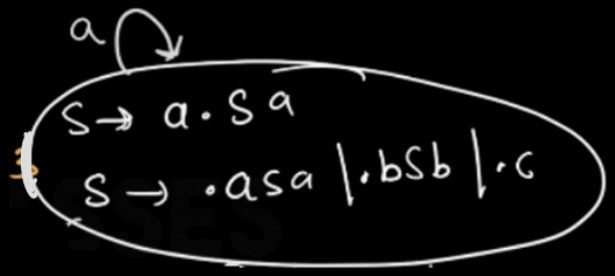
TOS is "a"



TOS is "Sa"

MSQ Question

Consider the LR(0) automaton state of some unknown grammar.
If parser is in given state then what can we say about content of the stack ?



A. We can not say anything about top of stack

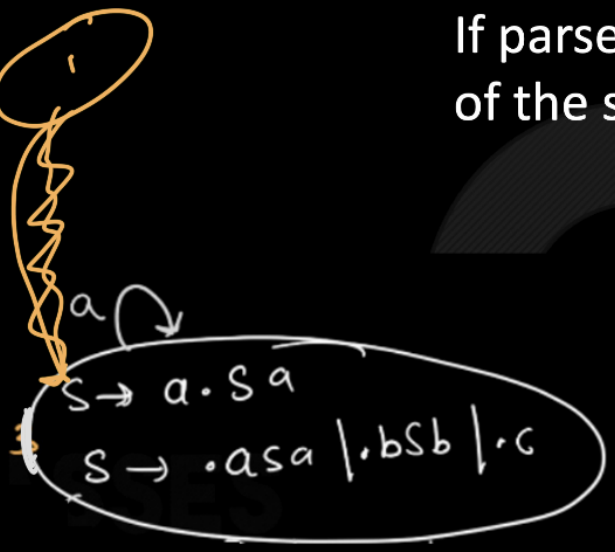
B. The top of the stack must be 'a'.

C. The top of the stack must be 'b'.

D. The top of the stack must be 'c'.

MSQ Question

Consider the LR(0) automaton state of some unknown grammar.
If parser is in given state then what can we say about content of the stack ?



A. We can not say anything about top of stack

B. The top of the stack must be 'a'.

C. The top of the stack must be 'b'.

D. The top of the stack must be 'c'.

E. stack contain only 'a'

MSQ Question

Consider the three LR(0) items given of some unknown grammar.
What can we say about these LR(0) items ?

1. $S \longrightarrow a.A$
2. $A \longrightarrow B.b$
3. $B \longrightarrow c.$

A. Item 1 and Item 3 could be part of same LR(0) automaton state.

B. Item 1 and Item 2 could be part of same LR(0) automaton state.

C. Item 2 and Item 3 could be part of same LR(0) automaton state.

D. These 3 items must belong to different-different LR(0) automaton states.

MSQ Question

Consider the three LR(0) items given of some unknown grammar.
What can we say about these LR(0) items ?

1. $S \longrightarrow a.A$
2. $A \longrightarrow B.b$
3. $B \longrightarrow c.$

A. Item 1 and Item 3 could be part of same LR(0) automaton state.

B. Item 1 and Item 2 could be part of same LR(0) automaton state.

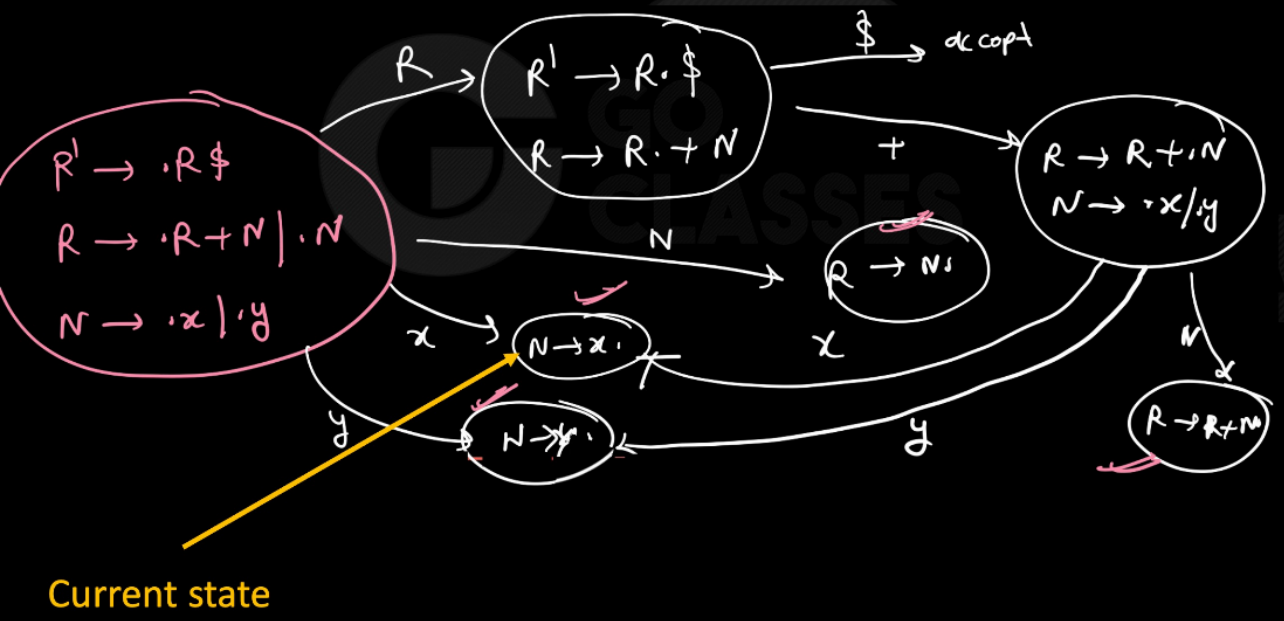
C. Item 2 and Item 3 could be part of same LR(0) automaton state.

D. These 3 items must belong to different-different LR(0) automaton states.

true

MSQ Question

Consider the LR(0) automaton given for the following grammar. Suppose that parser is in the given current state then what are the possible contents on stack ?



A. The stack content could be "R+x" where 'x' is on the top of the stack.

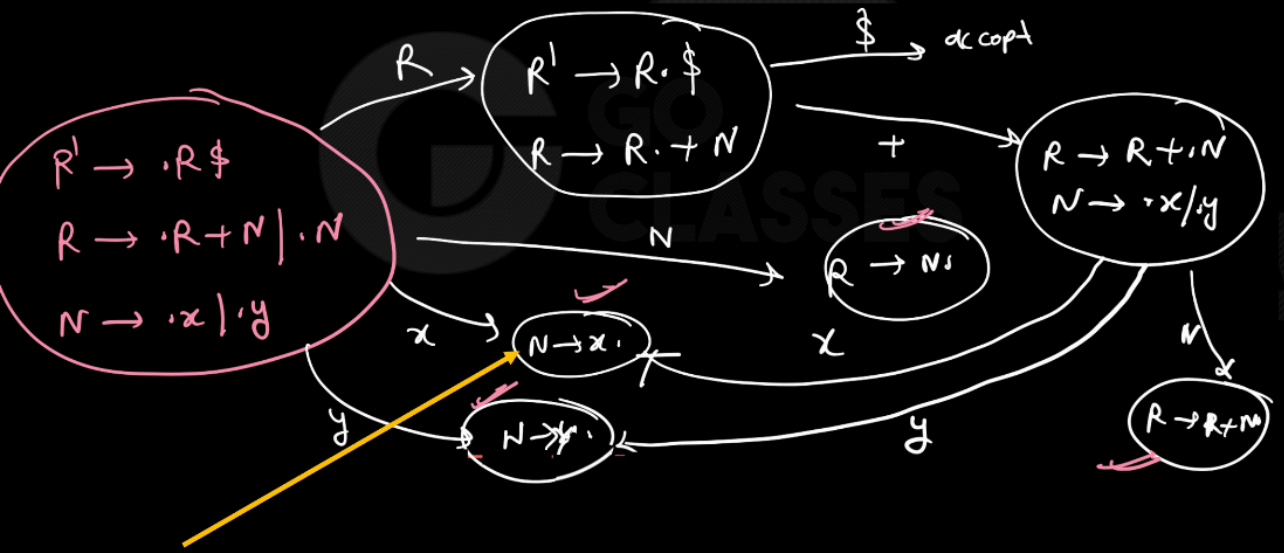
B. The stack content could be "R+x+R+x" where 'x' is on the top of the stack.

C. The stack content could be "x"

D. The top of the stack must be "x"

MSQ Question

Consider the LR(0) automaton given for the following grammar. Suppose that parser is in the given current state then what are the possible contents on stack ?



A. The stack content could be "R+x" where 'x' is on the top of the stack.

B. The stack content could be "R+x+R+x" where 'x' is on the top of the stack.

C. The stack content could be "x"

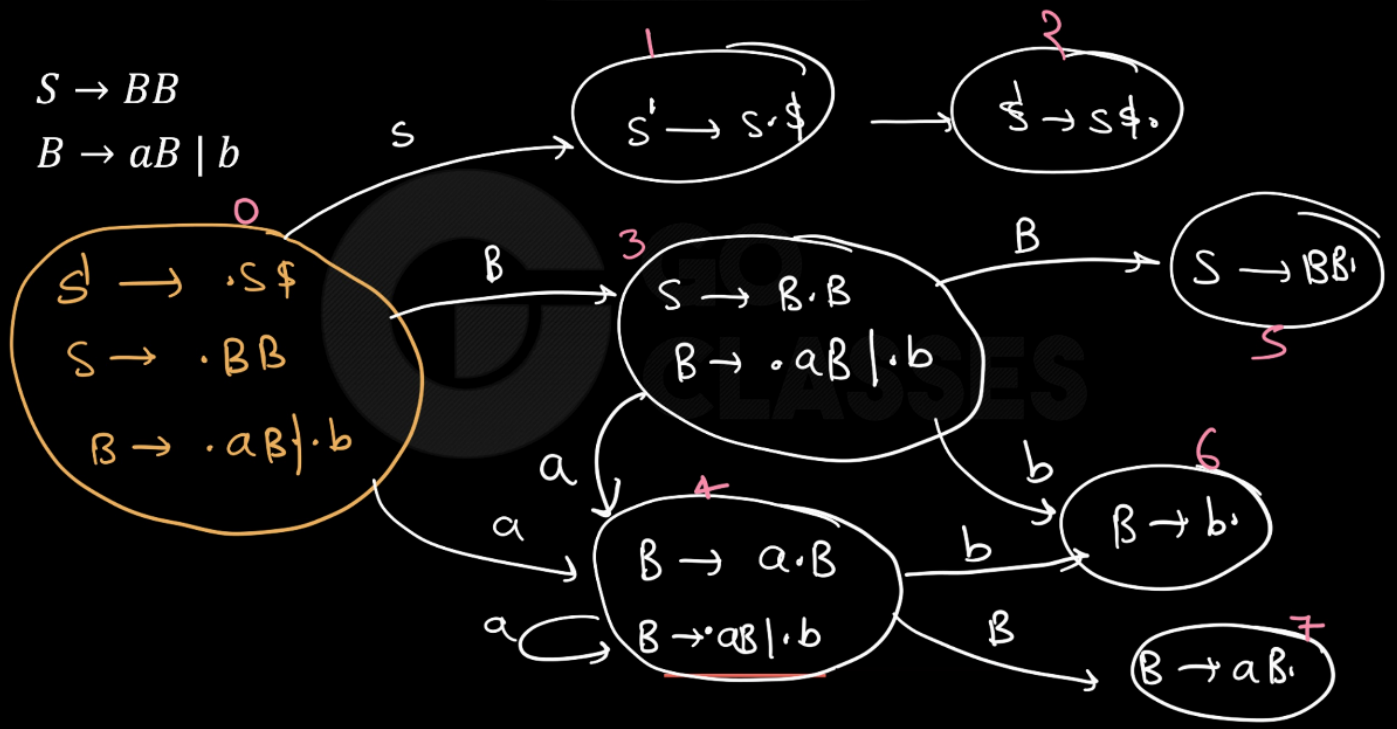
D. The top of the stack must be "x"

MSQ Question

Consider the LR(0) automaton given for the following grammar. Which of the followings can NEVER be content of the stack in parsing any string ?

$S \rightarrow BB$

$B \rightarrow aB \mid b$



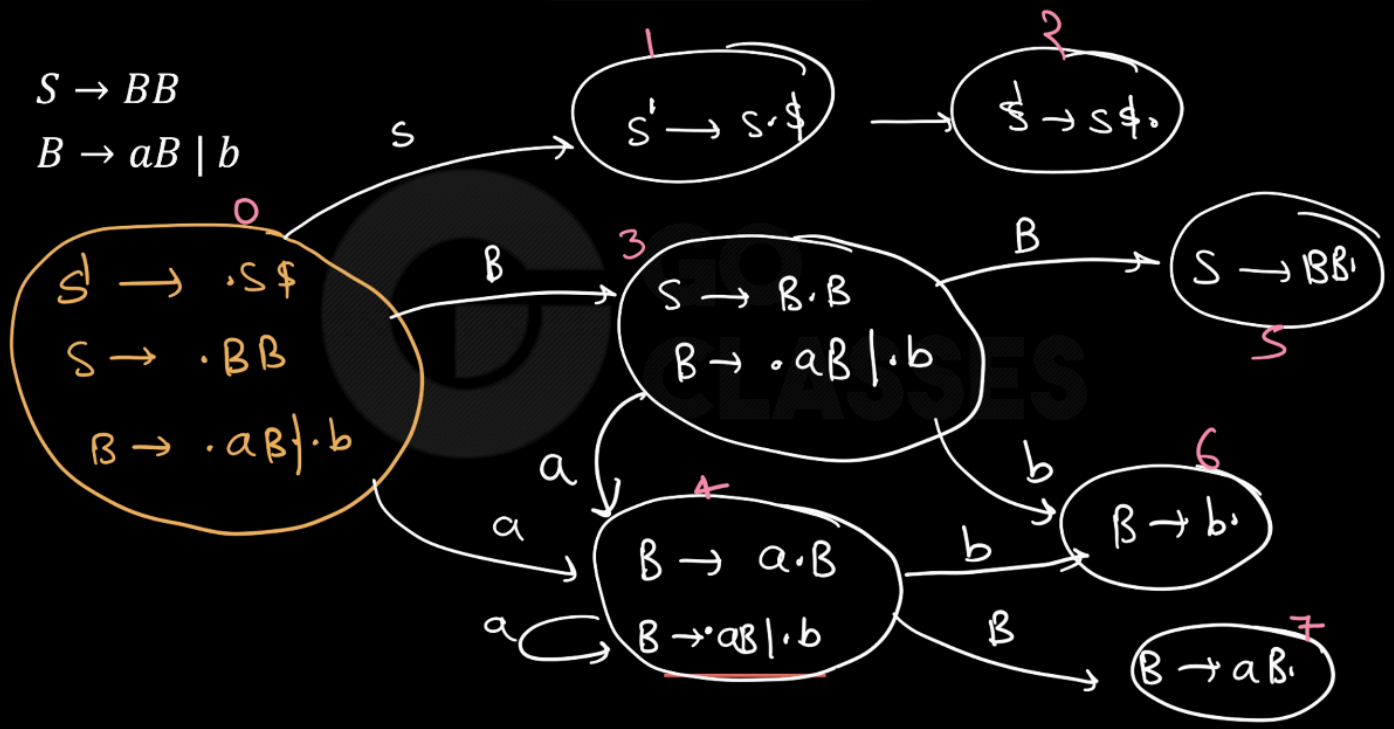
a) BB
b) BBb
c) Bab
d) aB

MSQ Question

Consider the LR(0) automaton given for the following grammar. Which of the followings can NEVER be content of the stack in parsing any string ?

$S \rightarrow BB$

$B \rightarrow aB \mid b$



a) BB

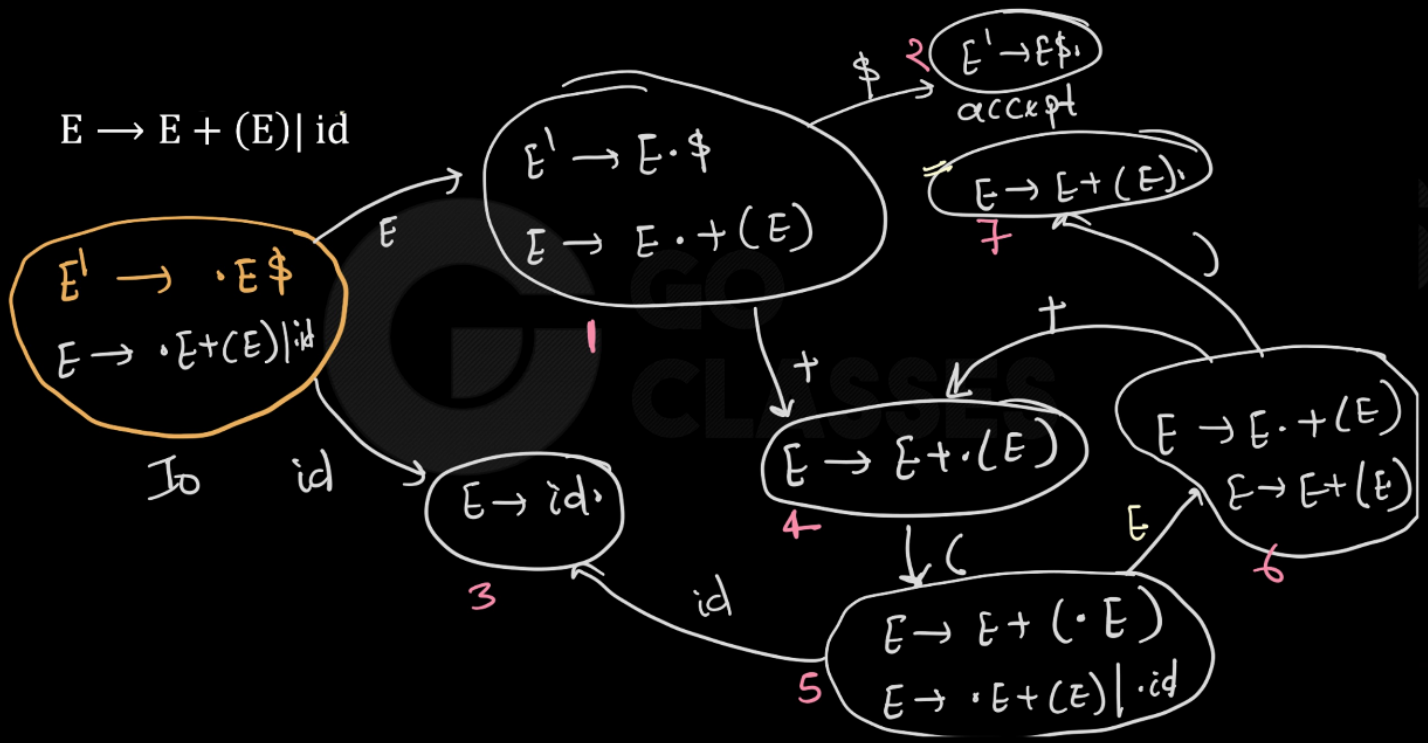
b) BBb

c) Bab

d) aB

MSQ Question

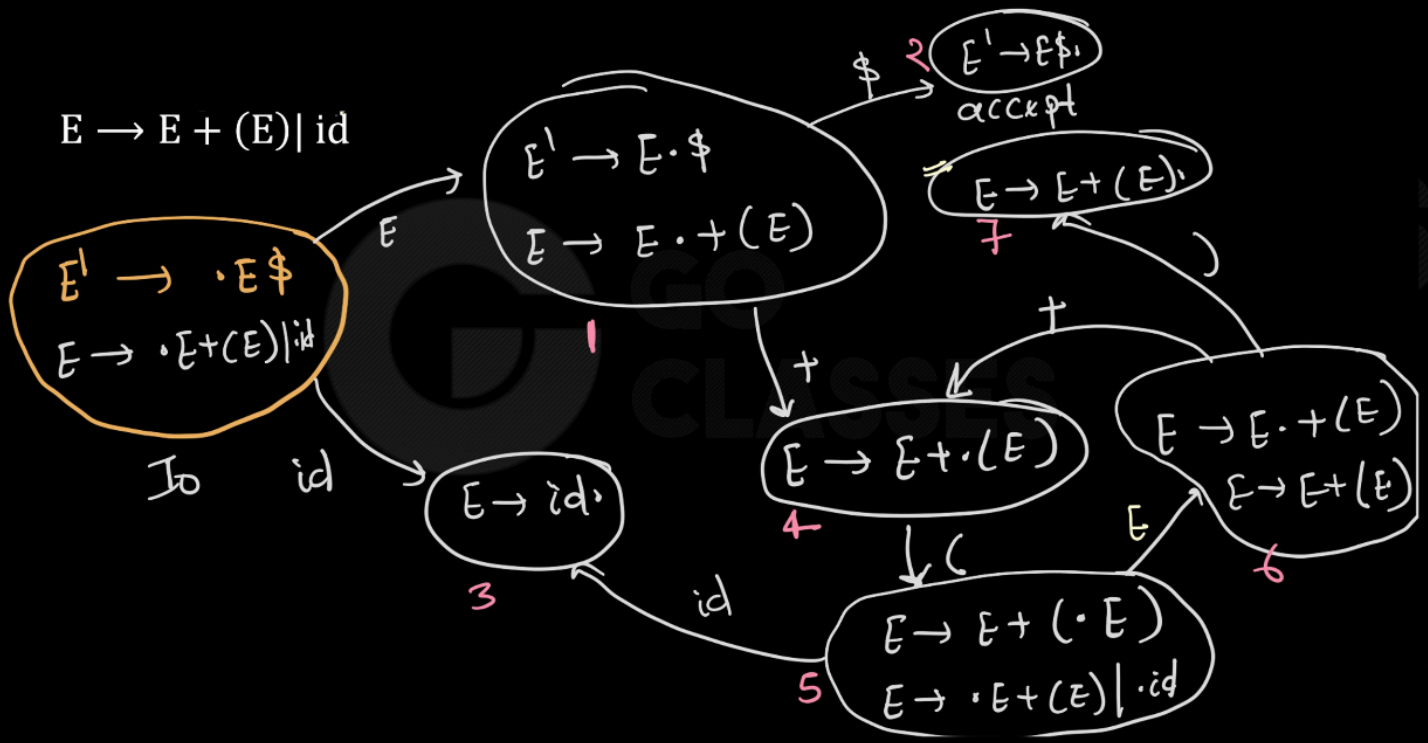
Consider the LR(0) automaton given for the following grammar. If parser is state 4 then what can we say about possible contents on the stack ?



1. Stack contain could be “E+” where + is on top of stack
2. Stack contain could be “id+” where + is on top of stack
3. Stack contain could be E+(E+ where + is on top of stack
4. Stack contain could be id+(id+ where + is on top of stack

MSQ Question

Consider the LR(0) automaton given for the following grammar. If parser is state 4 then what can we say about possible contents on the stack ?



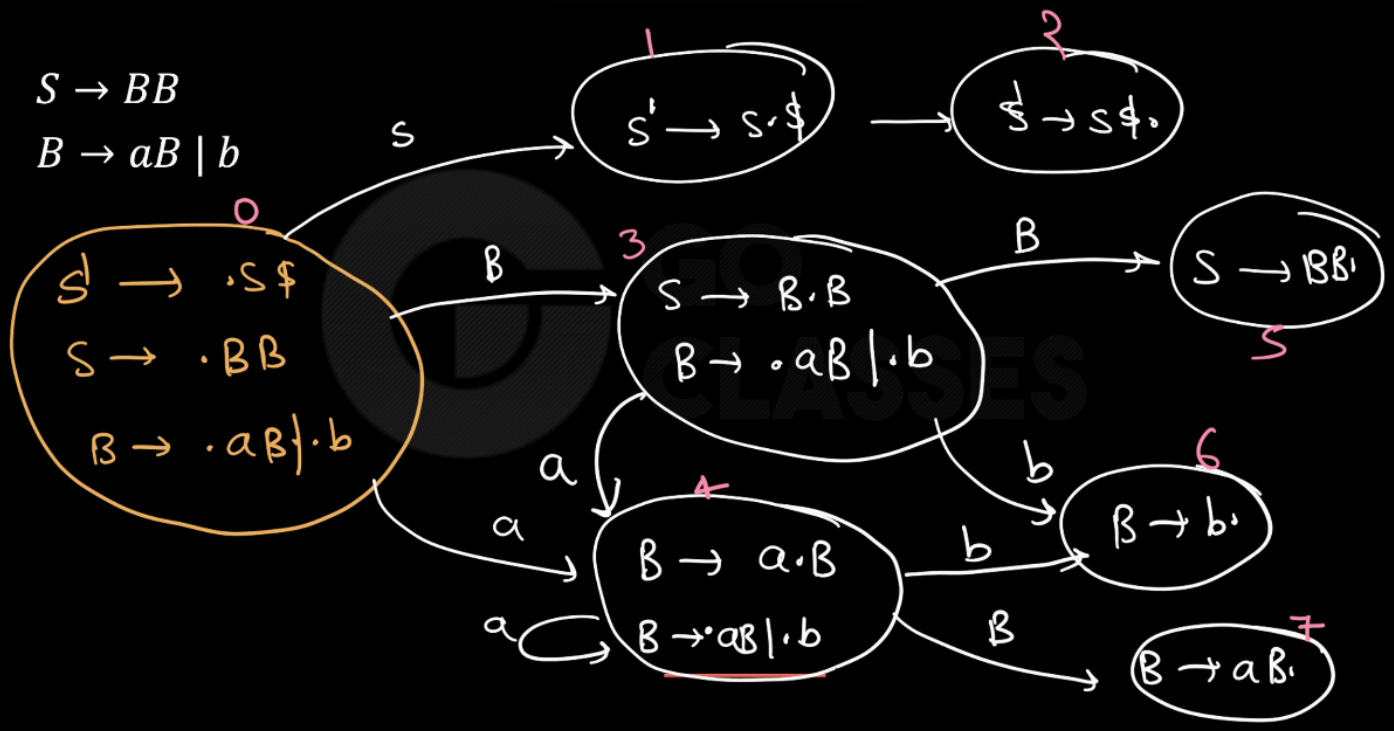
1. Stack contain could be "E+" where + is on top of stack ✓
2. Stack contain could be "id+" where + is on top of stack ✗
3. Stack contain could be E+(E+ where + is on top of stack ✓
4. Stack contain could be id+(id+ where + is on top of stack ✗

MSQ Question

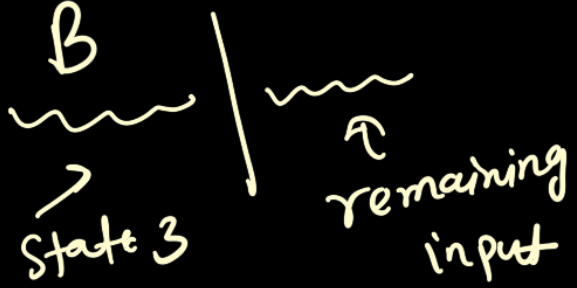
Consider the LR(0) automaton given for the following grammar. If parser is state 3 after reading a part of input then what should be remaining input such that input string finally get accepted ?

$S \rightarrow BB$

$B \rightarrow aB \mid b$



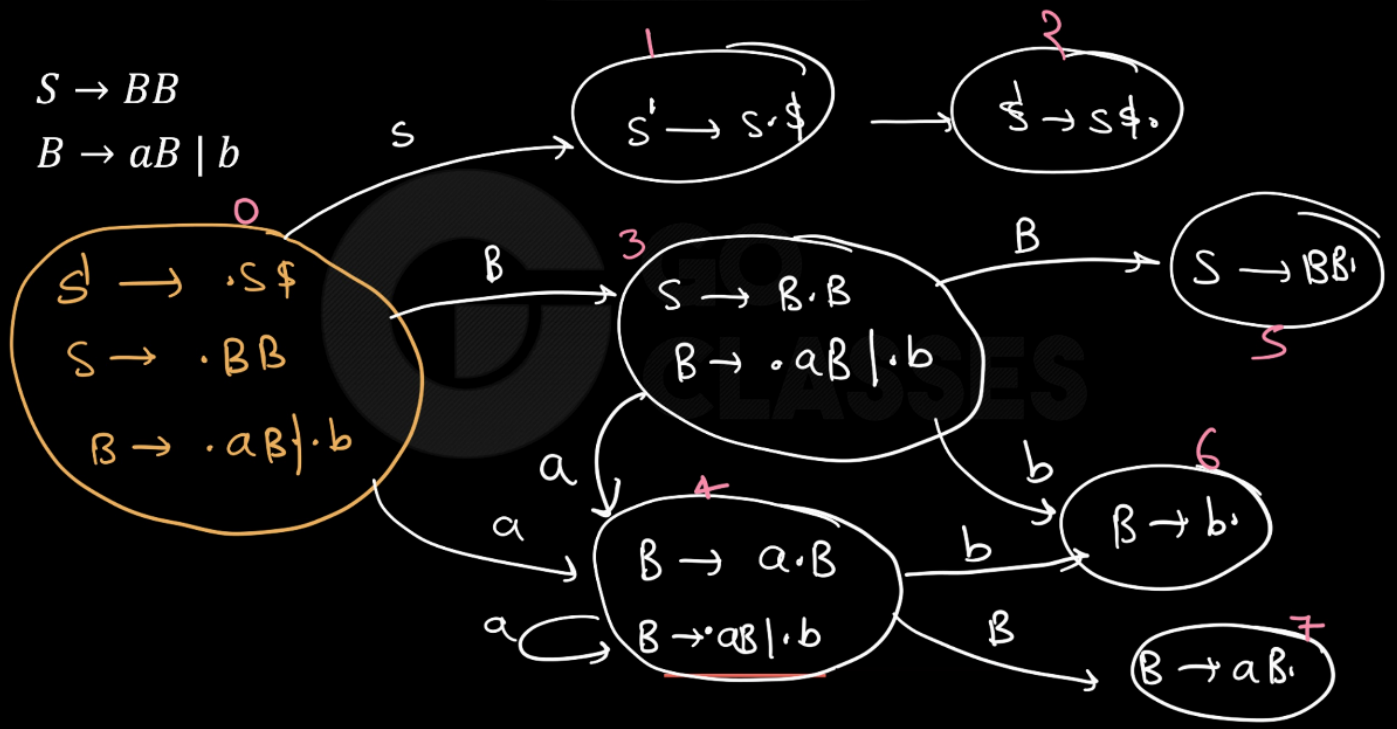
a) b
b) ab
c) abb
d) aab



MSQ Question

Consider the LR(0) automaton given for the following grammar. If parser is state 3 after reading a part of input then what should be remaining input such that input string finally get accepted ?

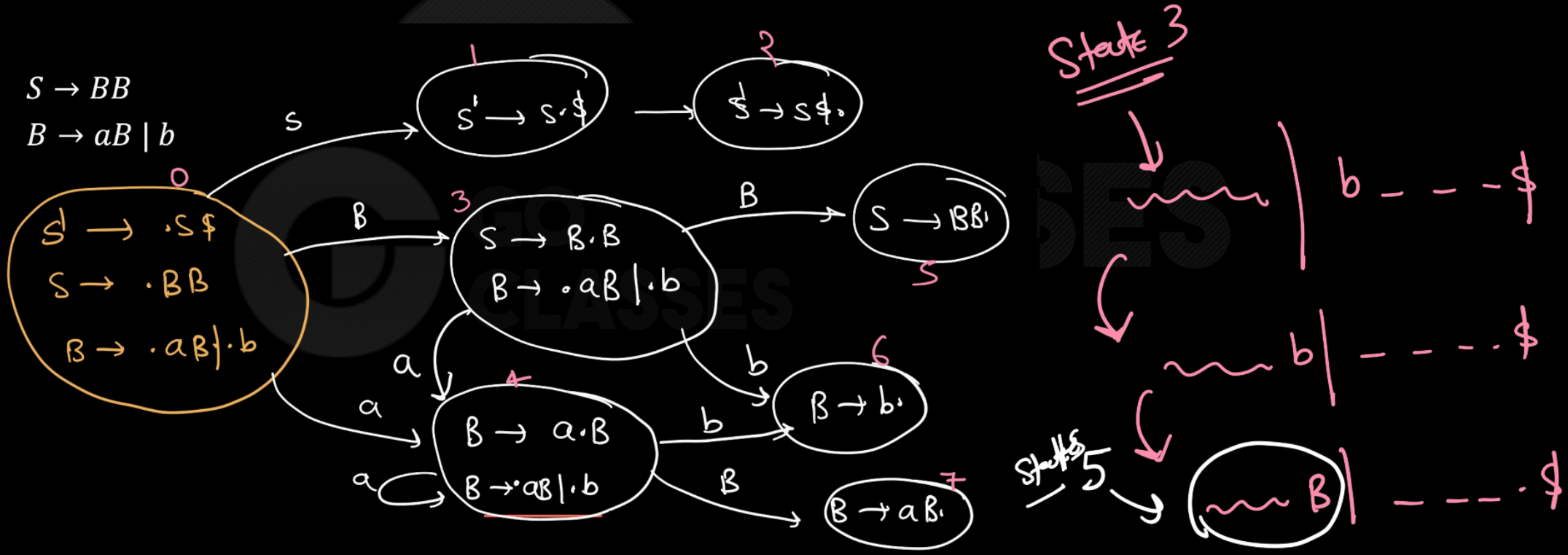
$S \rightarrow BB$
$B \rightarrow aB \mid b$



a) b
b) ab
c) abb
d) aab

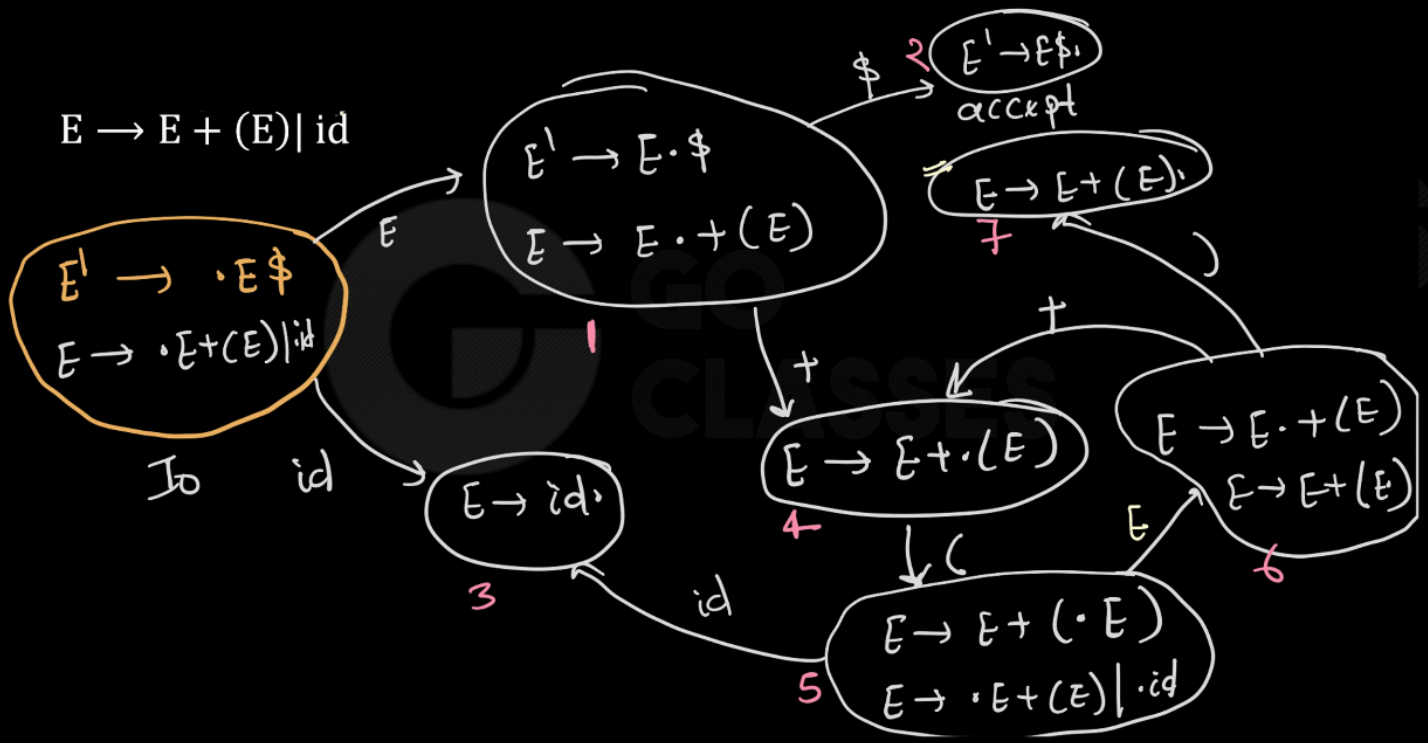
MSQ Question

Consider the LR(0) automaton given for the following grammar. If parser is state 3 after reading a part of input then what should be remaining input such that input string finally get accepted ?



MSQ Question

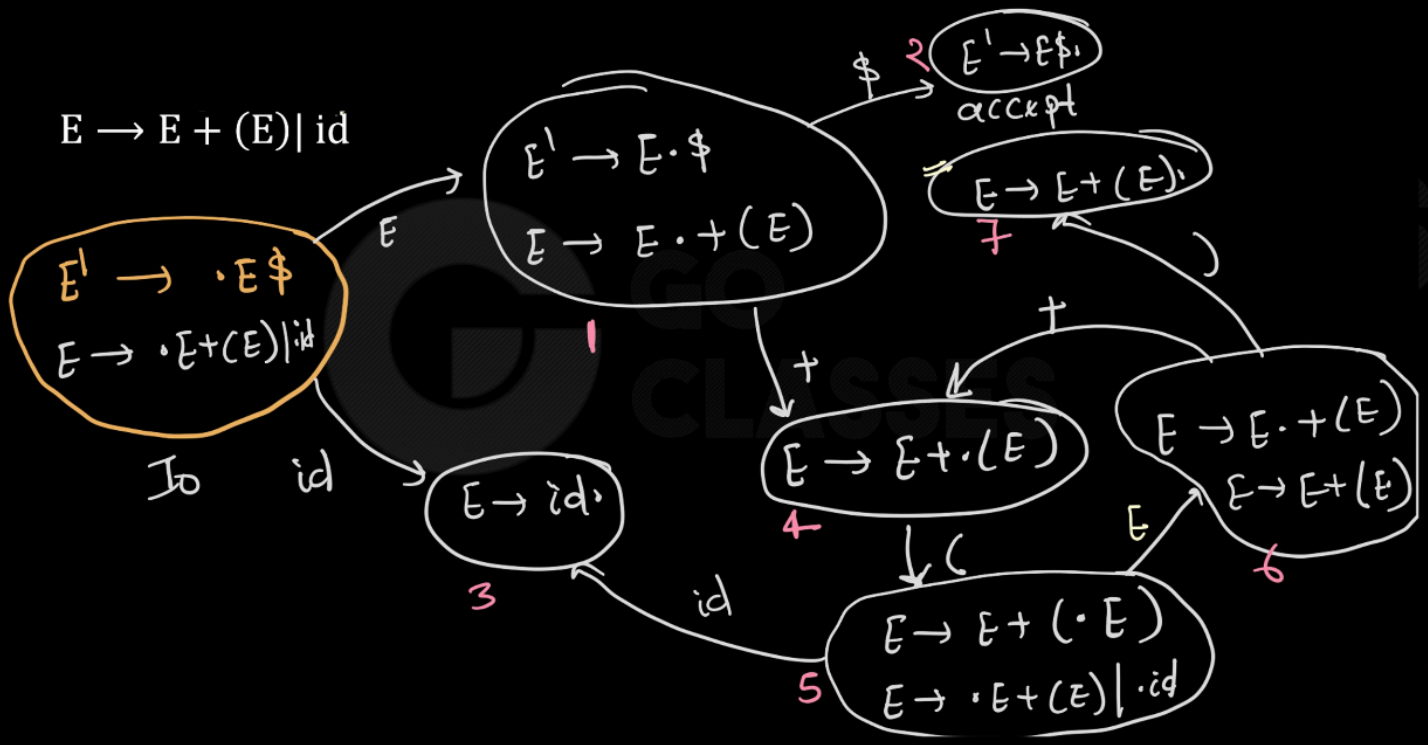
Consider the LR(0) automaton given for the following grammar. If parser is state 4 then what can we say about possible contents on the stack ?



1. Stack contain could be “E+” where + is on top of stack
2. Stack contain could be “id+” where + is on top of stack
3. Stack contain could be E+(E+ where + is on top of stack
4. Stack contain could be id+(id+ where + is on top of stack

MSQ Question

Consider the LR(0) automaton given for the following grammar. Which of the followings are possible contents of stack in parsing ?



a) id
b) Id+ (here '+' is on top)
c) E+((here '(' is on top)
d) E+E

GATE CSE 2006 | Question: 7

29

Consider the following grammar

- $S \rightarrow S * E$
- $S \rightarrow E$
- $E \rightarrow F + E$
- $E \rightarrow F$
- $F \rightarrow id$

Consider the following LR(0) items corresponding to the grammar above

i. $S \rightarrow S*. E$
ii. $E \rightarrow F. +E$
iii. $E \rightarrow F+. E$

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

A. i and ii
B. ii and iii
C. i and iii
D. None of the above

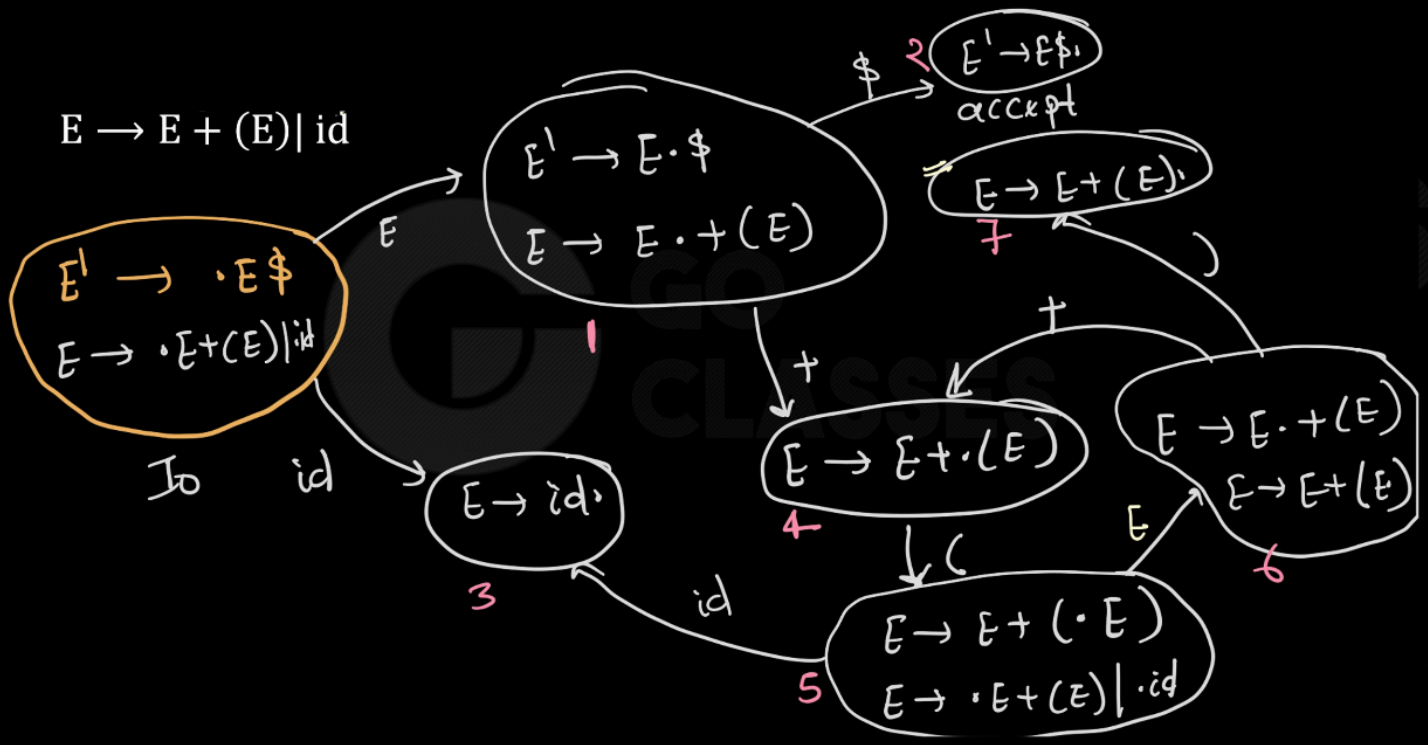
gatecse-2006 compiler-design parsing normal

https://gateoverflow.in/886/gate-cse-2006-question-7

i. $S \rightarrow S*. E$
ii. $E \rightarrow F. +E$
iii. $E \rightarrow F+. E$

MSQ Question

Consider the LR(0) automaton given for the following grammar. Which of the followings are possible contents of stack in parsing ?



a) id
b) Id+ (here '+' is on top)
c) E+((here '(' is on top)
d) E+E

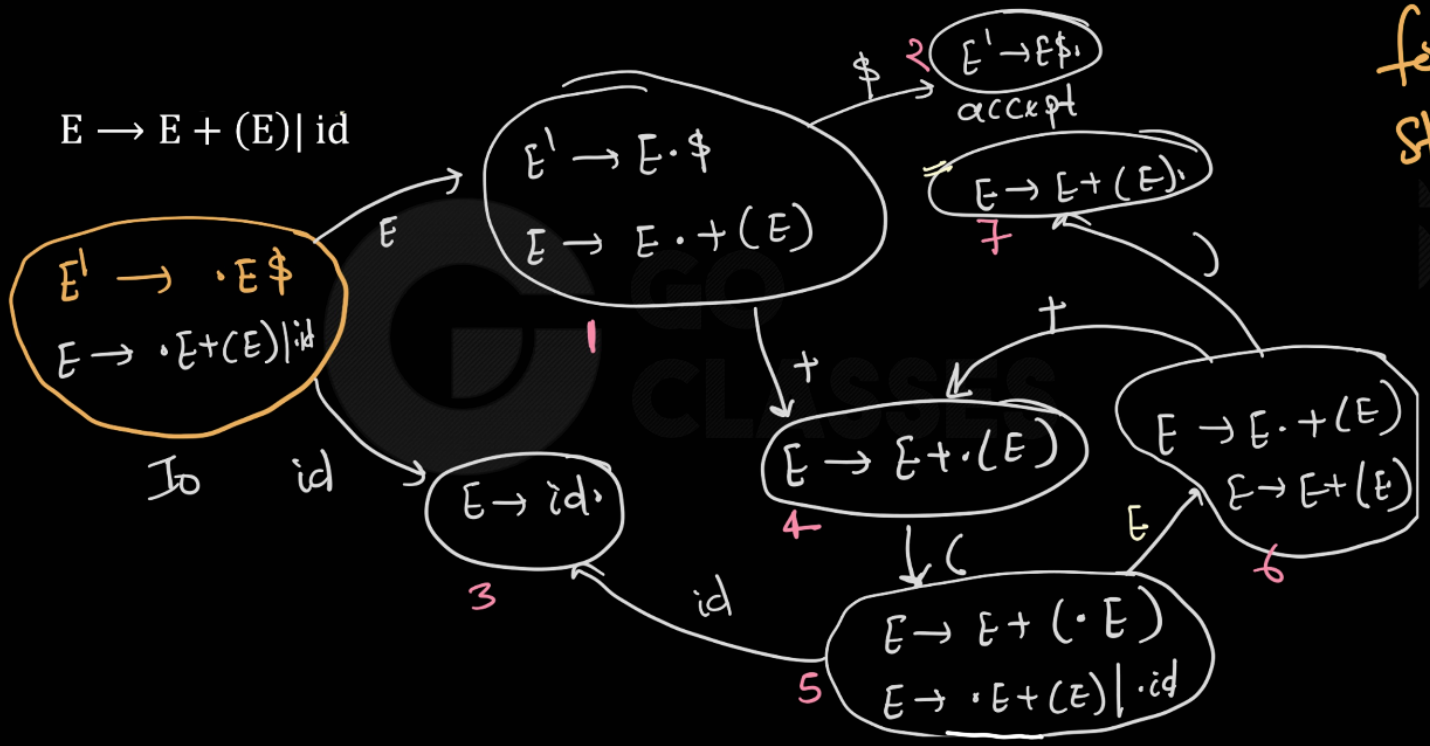
✗ id+

Viable Prefix

Viable Prefix = Content of stack

GO CLASSES

$E \rightarrow E + (E) \mid id$



Which of the following can be stack content.

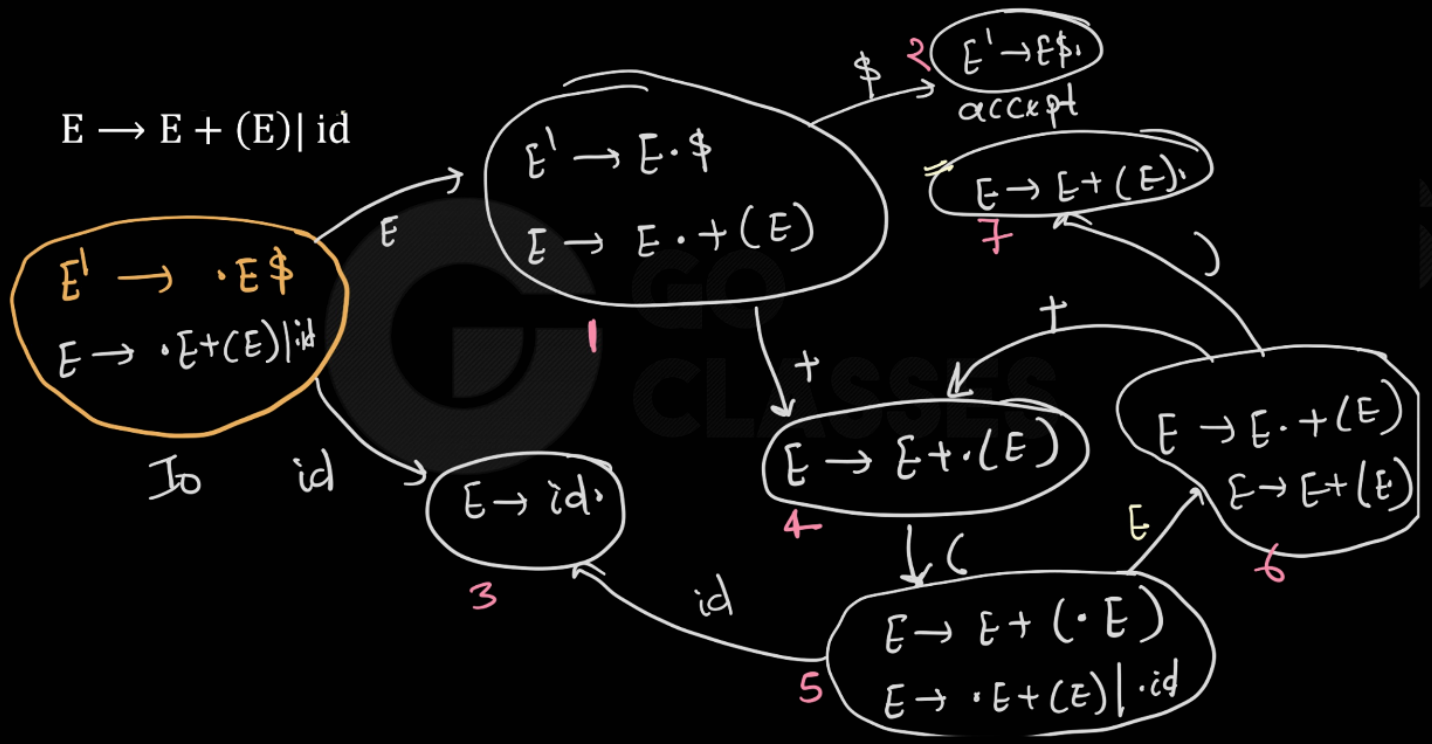
1. ~~E~~ ✗ 7.)
2. ~~id~~ ✗ 8.) E
3. ~~E+~~
4. ✗ E+ +
5. ~~E+(~~
6. ~~E+(E~~

GO CLASSES

if Someone ask you what can appear in stack

$\Rightarrow$ we just need to check whether it is leading to valid stack or not.

$E \longrightarrow E + (E) \mid id$

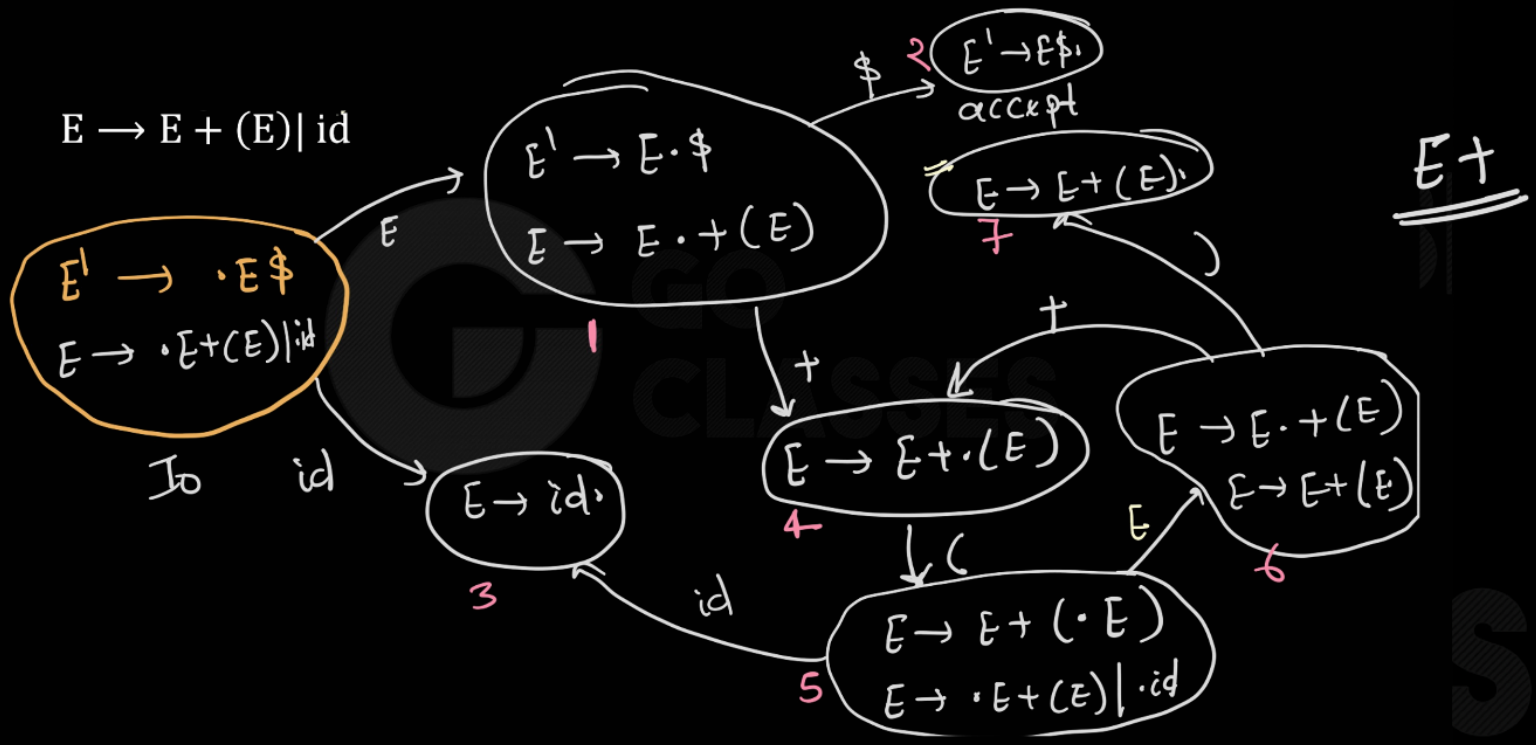


Which of the following can be the viable prefix

a) E b) id c) id+ d) E+ e) E+E f) E+(

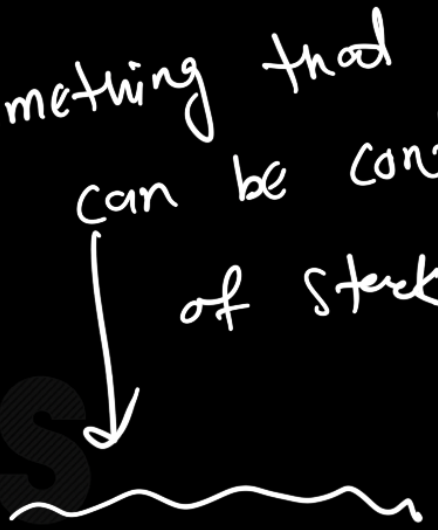
g) +(h) id+(i) E+(E

$E \longrightarrow E + (E) \mid id$



E+

Something that can be content of stack



viable prefix

Which of the following can be the

a) E b) id c) id+ d) E+ e) E+E f) E+(

g) +(h) id+(i) E+(E

MSQ

Stack content is : $\alpha\ \beta\ \gamma\ \delta$

then what are the possible viable prefix?

a) $\alpha\beta\gamma\delta$

b) $\beta\gamma$

c) $\alpha\beta$

d) $\alpha\beta\gamma$

GO CLASSES

MSQ

Stack content is : $\alpha\beta\gamma\delta$

anything that can be stack content

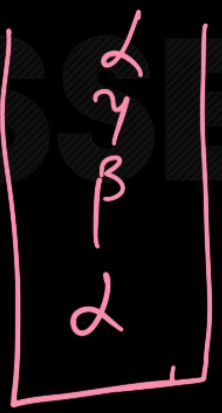
then what are the possible viable prefix?

a) $\alpha\beta\gamma\delta$

b) $\beta\gamma$

c) $\alpha\beta$

d) $\alpha\beta\gamma$



GO CLASSES

MSQ

Stack content is : $\alpha\beta\gamma\delta$

then what are the possible viable prefix? (anything that can be stack content)

a) $\alpha\beta\gamma\delta$

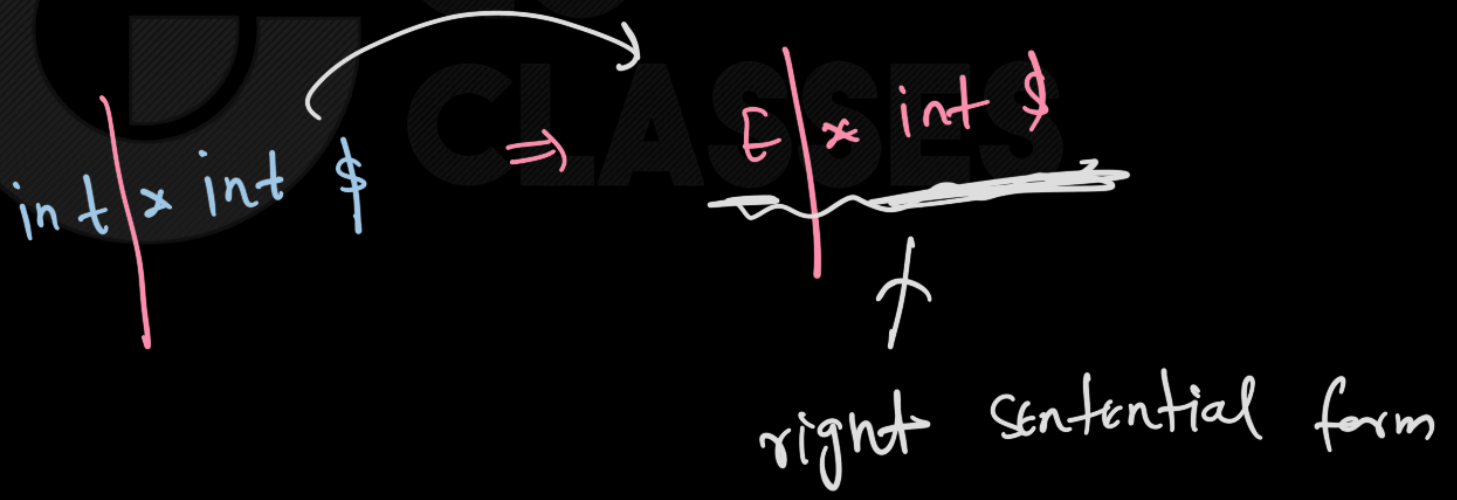
b) $\beta\gamma$

c) $\alpha\beta$

d) $\alpha\beta\gamma$

Prefix of stack content was the earlier stack content.

Definition (viable prefix): The prefixes of right sentential forms that can appear on the stack of a shift-reduce parser are called viable prefixes.



The set of viable prefixes correspond to the possible stack states in a shift-reduce parser.

http://people.cs.uchicago.edu/~jhr/courses/w2004-51086/docs/notes-04.pdf

Ullman Question

Exercise 4.6.1 : Describe all the viable prefixes for the following grammars:

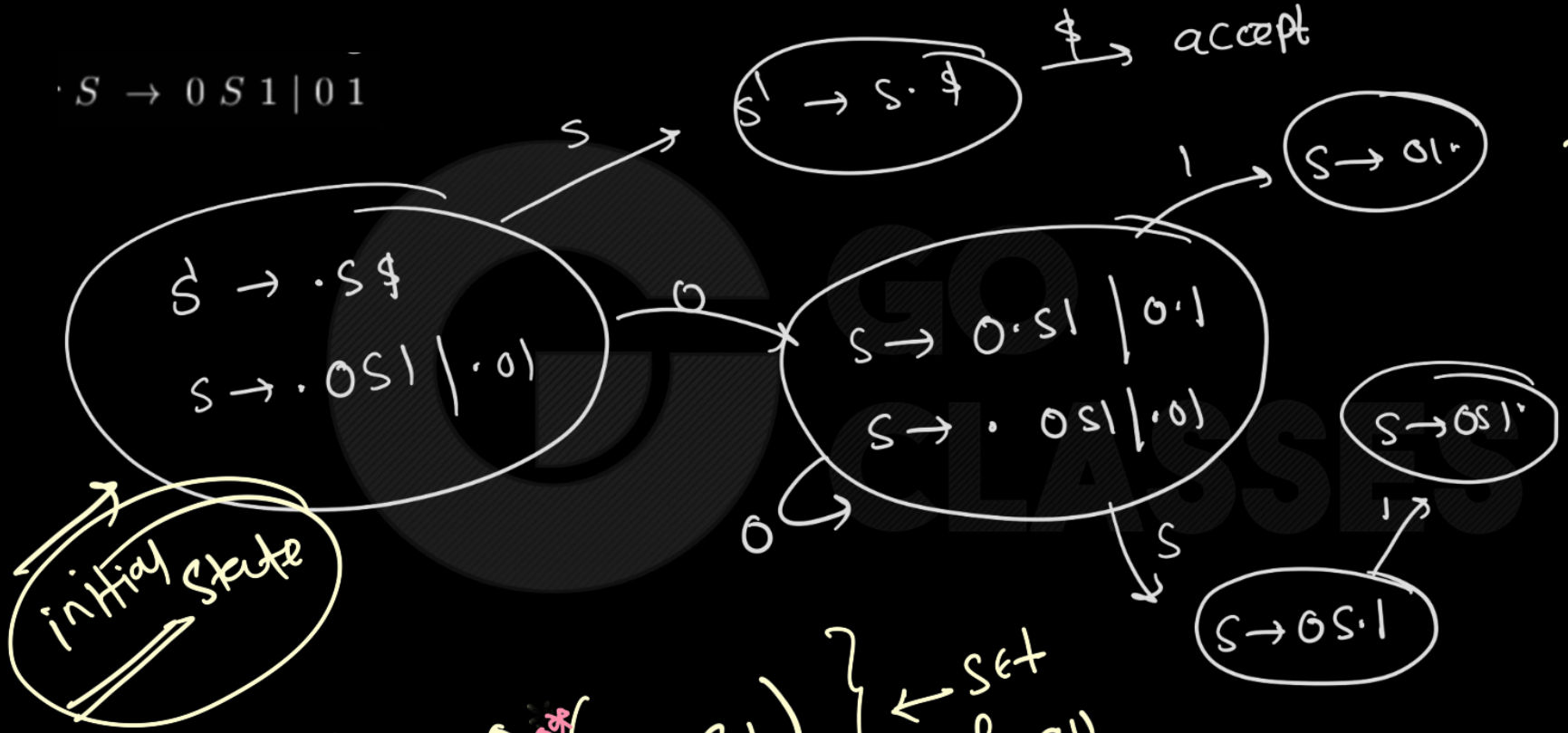
a) The grammar $S \rightarrow 0\ S\ 1 \mid 0\ 1$

b) The grammar $S \rightarrow S\ S\ + \mid S\ S\ * \mid a$

c) The grammar $S \rightarrow S\ (\ S\ ) \mid \epsilon$

GO CLASSES

$S \rightarrow 0S1 \mid 01$



for viable prefix, treat every state as a final state

$\epsilon + S + 00^*(1 + S1)$ ← set of all viable prefixes

S, 0, 00, 0^*S, 01

True/False Question

The set of all viable prefixes forms a regular language

True/False Question

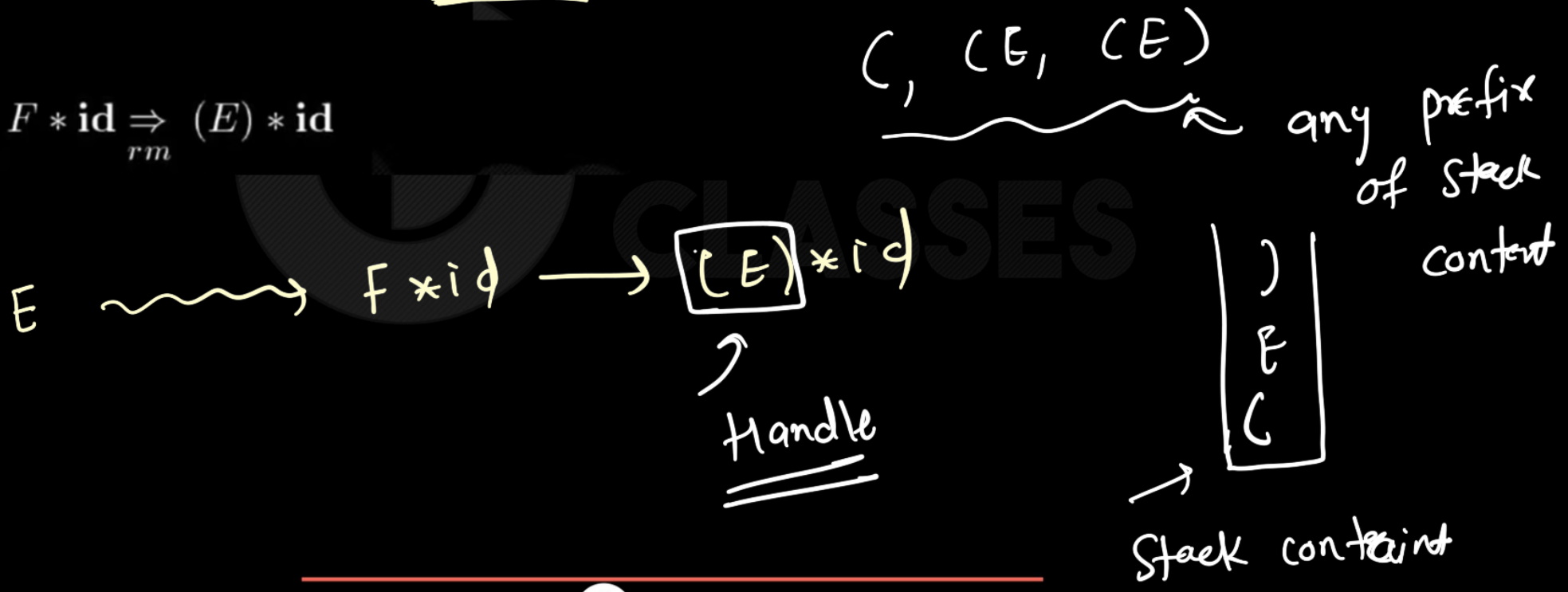
The set of all viable prefixes forms a regular language

true

Question

What are the viable prefixes of $(E) * id$ in following rightmost derivation?

$$E \underset{rm}{\overset{*}{\Rightarrow}} F * \mathbf{id} \underset{rm}{\Rightarrow} (E) * \mathbf{id}$$

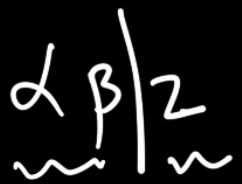


Question

Consider the given derivation, What is viable prefix of the right sentential form $\alpha\beta z$?

$$S \Rightarrow^* \alpha Az \Rightarrow \alpha\beta z$$

A. Any prefix of β is viable prefix
B. Any prefix of $\alpha\beta$ is viable prefix
C. Any prefix of $\alpha\beta z$ is viable prefix
D. Any prefix of βz is viable prefix



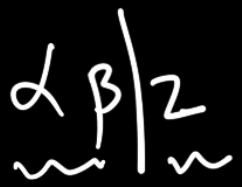
stack content

Question

Consider the given derivation, What is viable prefix of the right sentential form $\alpha\beta z$?

$$S \Rightarrow^* \alpha Az \Rightarrow \alpha\beta z$$

A. Any prefix of β is viable prefix
B. Any prefix of $\alpha\beta$ is viable prefix
C. Any prefix of $\alpha\beta z$ is viable prefix
D. Any prefix of βz is viable prefix



stack content

Question

A. If α is a viable prefix then we can always append set of terminals to get some right sentential form.

B. If α is a viable prefix then we can always append set of non-terminals to get some right sentential form .

C. If α is a viable prefix then we need to append mix of terminals and nonterminals (alone terminals or non terminals won't do) to get some right sentential form.

D. If α is a viable prefix then we can append mix of terminals and nonterminals to get some right sentential form.

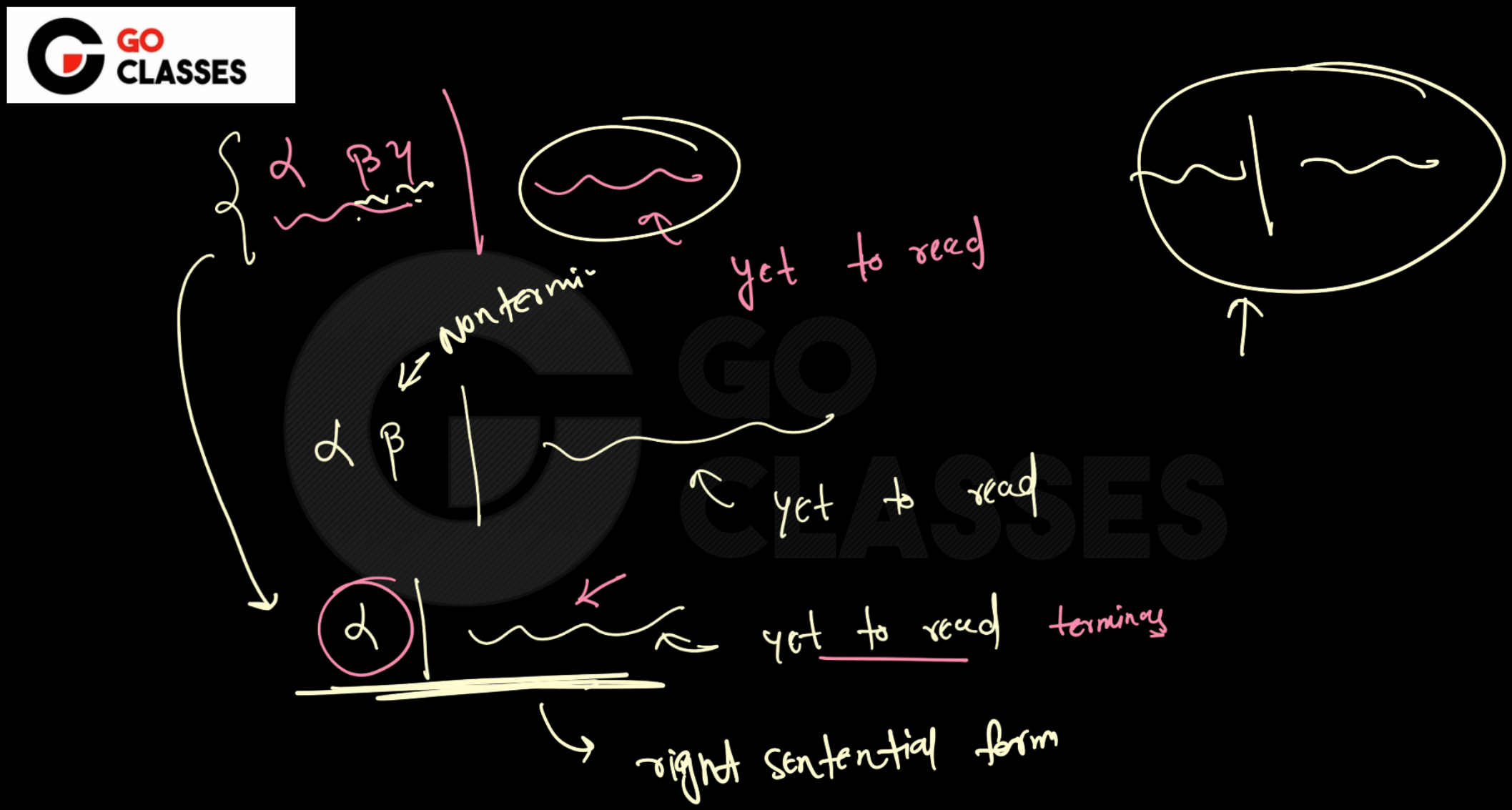
Question

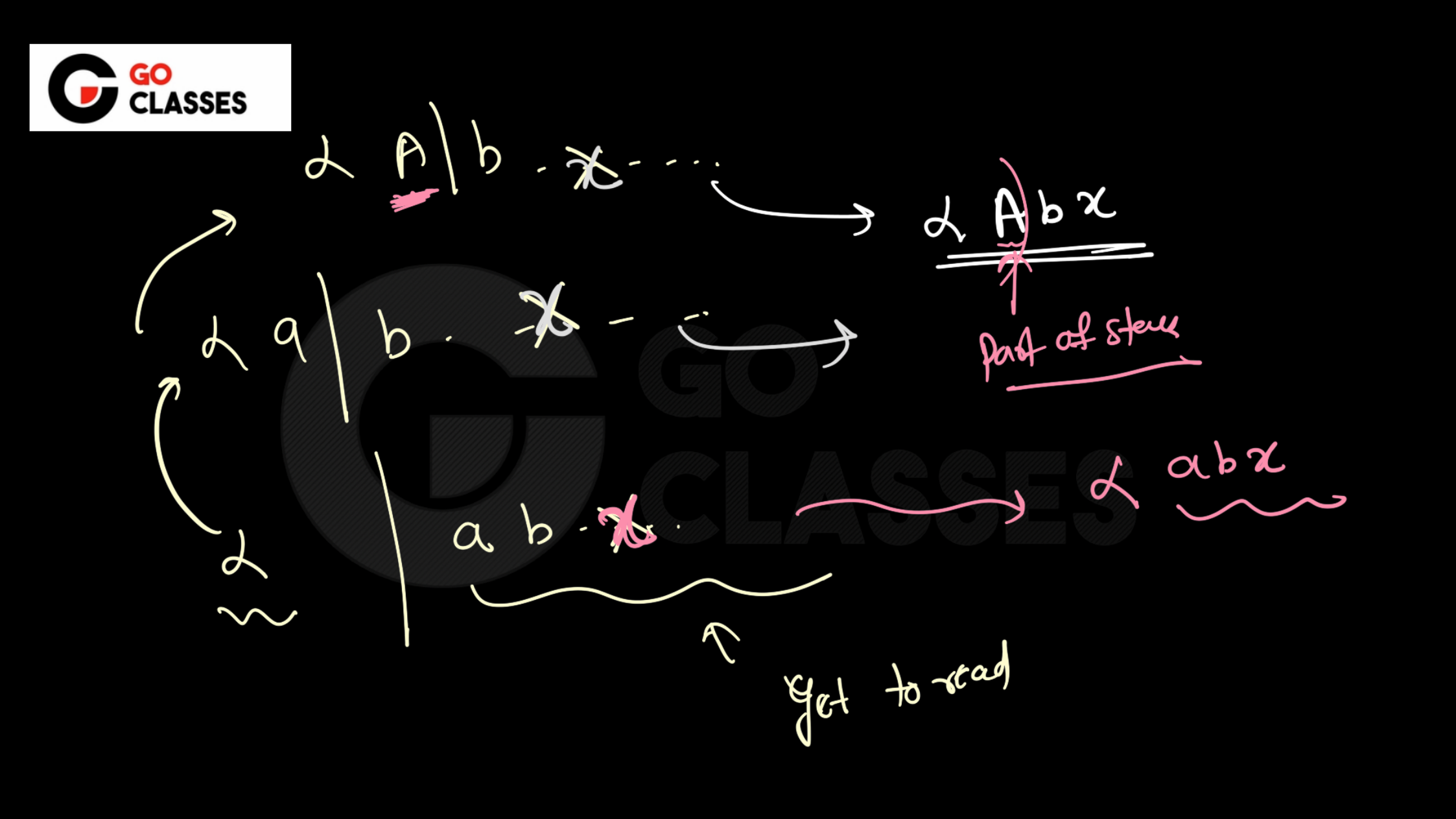
A. If α is a viable prefix then we can always append set of terminals to get some right sentential form.

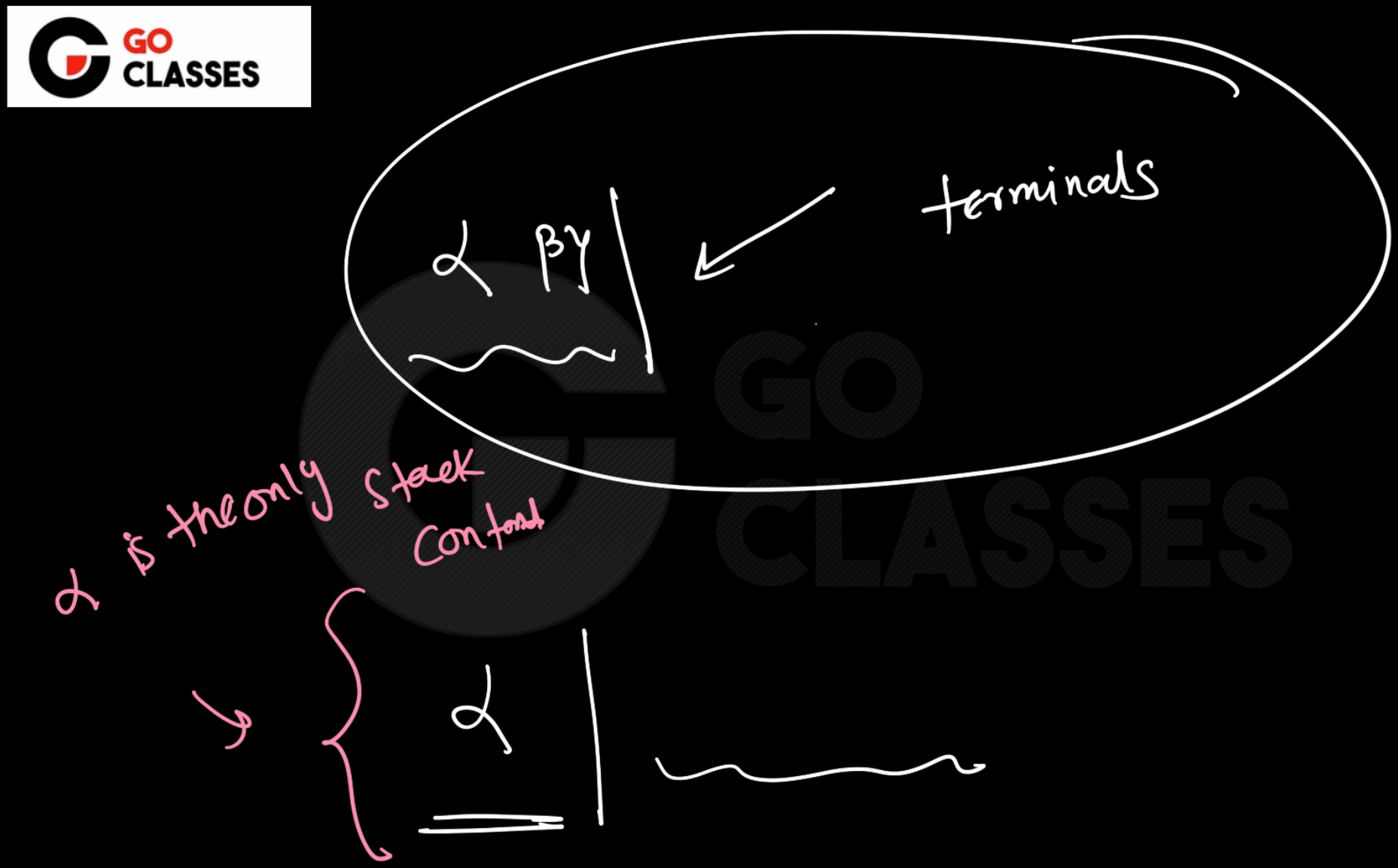
B. If α is a viable prefix then we can always append set of non-terminals to get some right sentential form .

C. If α is a viable prefix then we need to append mix of terminals and nonterminals (alone terminals or non terminals won't do) to get some right sentential form.

D. If α is a viable prefix then we can append mix of terminals and nonterminals to get some right sentential form.


α β γ
yet to read
α β
yet to read
α
yet to read terminas
right sentential form


GO CLASSES
α A | b . x . . . .
α A b x
α a | b . x . . . .
Part of stem
| a b . x .
α a b x
α
Yet to read


GO CLASSES
α β γ
terminals
α is the only stack content
α

Question

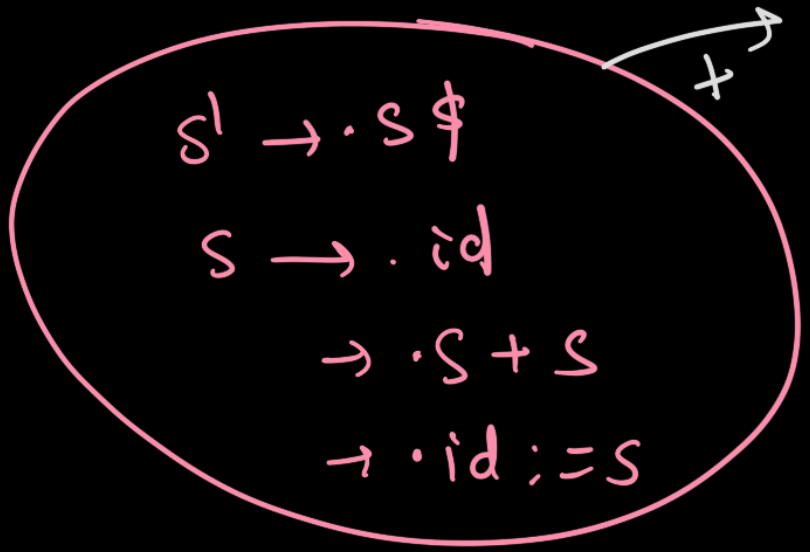
```
S -> id | S + S | id := S
```

Which of the following strings are viable prefixes? For each string that is a viable prefix, give a suffix that produces a right sentential form.

i. + S +

ii. S + id :=

id



https://tbp.berkeley.edu/exams/2315/download/

given this viable prefix, bottom of stack = α, αβ, αβγ

Answer

```
S -> id | S + S | id := S
```

Which of the following strings are viable prefixes? For each string that is a viable prefix, give a suffix that produces a right sentential form.

i. `+ S +`

No.

ii. `S + id :=`

Yes: suffix: id $

GO CLASSES

Viable prefix

Prefix of stack content is viable prefix

$\alpha\ \beta\gamma \mid abc$

α is a viable ~~pre~~ fix, what we need to append to get right sentential form ?

$\beta\gamma\ abc$

$\alpha \beta\gamma \mid abc$

$\alpha\beta$ is a viable prefix, what we need to append to get right sentential form ?

γabc

$\alpha\ \beta\gamma \mid abc$

$\alpha\beta\gamma$ is a viable prefix, what we need to append to get right sentential form ?

abc

GO CLASSES

$a|bc$

$\alpha \beta \gamma \mid a b c$ → Right sentential form

viable prefixes

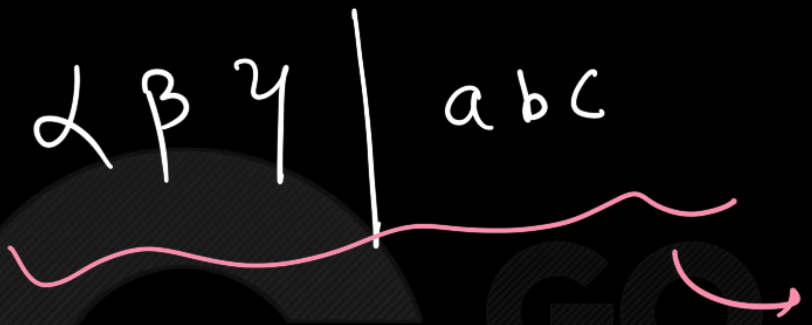
α, $\alpha\beta$, $\alpha\beta\gamma$

$AB|\$$

$a|bc \Rightarrow A|bc \Rightarrow Abc|\$$

RMD

$S \to AB \to AbC \to Abc \to abc$



Right sentential form

(AB | $)

Viable prefixes

α, αβ, αβγ

αβγa ✗

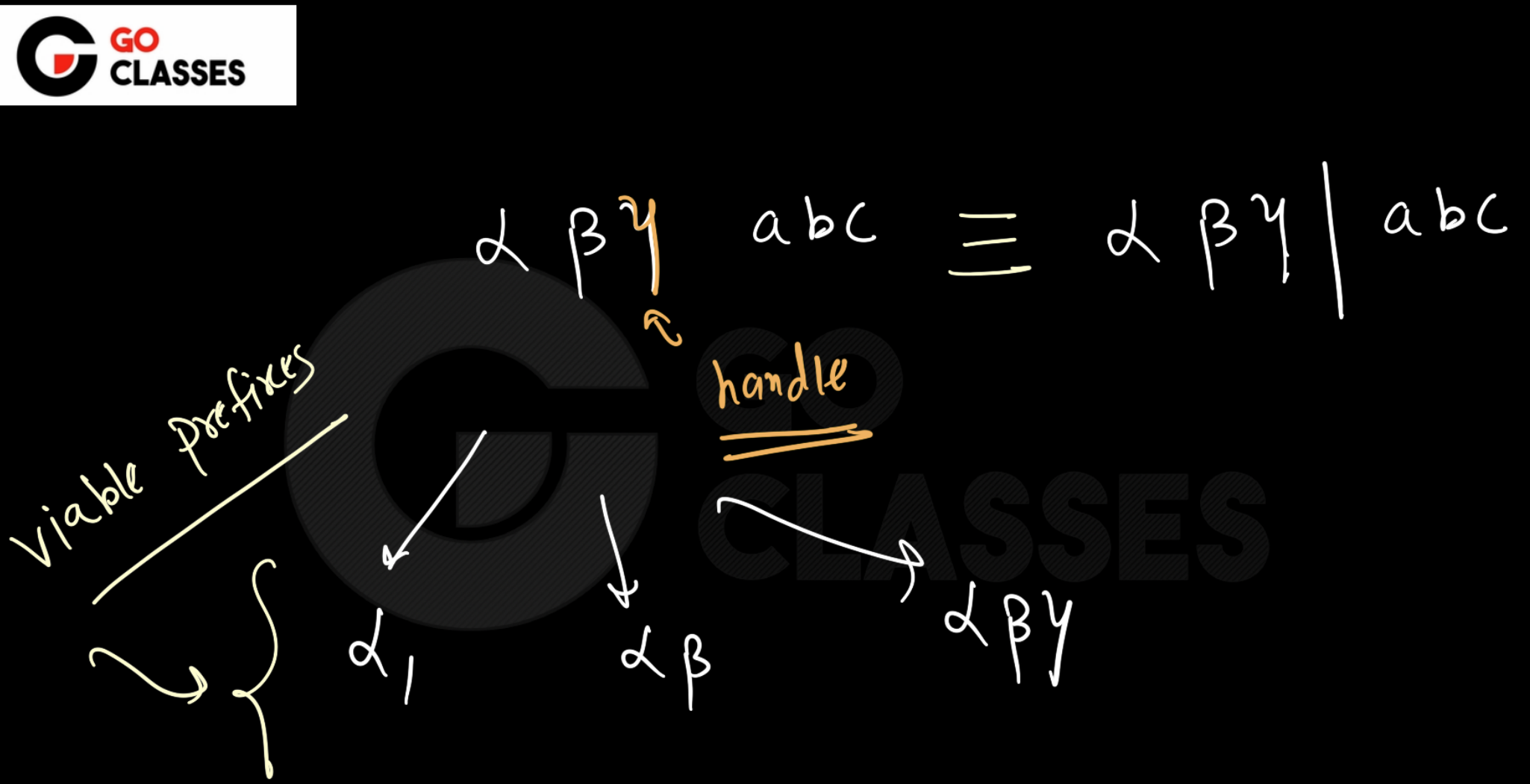
Possible handles ← TOS

✗a) β ✗c) αβ e) βγ

✗b) α d) γ ✗f) a

$\alpha\beta\gamma \mid abc$

$\alpha, \alpha\beta, \alpha\beta\gamma$ — viable prefixes


αβγ abc ≡ αβγ | abc
handle
Viable Prefixes
α1
αβ
αβγ

GO CLASSES

T/F

Viable prefix is a prefix of any right sentential form where we do not go past by handle.

GO CLASSES

$S \rightarrow AB \longrightarrow AbC \longrightarrow Abc \rightarrow abc$

$abc\$ \Rightarrow a|bc\$ \Rightarrow A|bc\$ \Rightarrow Ab|c\$ \Rightarrow$

$S|\$ \Leftarrow AB|\$ \Leftarrow AbC|\$ \Leftarrow Abc|\$$

ab

abc $\rightsquigarrow$ viable prefix $\Rightarrow$ a

$w_1\ w_2\ w_3\ w_4\ w_5$

if w_3 is handle then which of the following are viable prefix -

w_1 ✓

$w_1 w_2$ ✓

$w_1 w_2 w_3$ ✓

$w_2 w_3$ ✗

$w_1\ w_2\ w_3 \mid w_4\ w_5$

GO CLASSES

Set of all viable prefixes

↓

draw LR(0) automaton and treat every state as final state initial state is fixed.

Given a string ⇒ what is viable prefix?

↳ locate the handle, put a bar and tell prefixes

viable prefix

add some terminals to get right sentential form

GATE CSE 2015 Set 1 | Question: 13

52

Which one of the following is TRUE at any valid state in shift-reduce parsing?

A. Viable prefixes appear only at the bottom of the stack and not inside
B. Viable prefixes appear only at the top of the stack and not inside
C. The stack contains only a set of viable prefixes
D. The stack never contains viable prefixes

gatecse-2015-set1 compiler-design parsing normal viable-prefix

https://gateoverflow.in/8187/gate-cse-2015-set-1-question-1-3

GATE CSE 2015 Set 1 | Question: 13

52

Which one of the following is TRUE at any valid state in shift-reduce parsing?

A. Viable prefixes appear only at the bottom of the stack and not inside
B. Viable prefixes appear only at the top of the stack and not inside
C. The stack contains only a set of viable prefixes
D. The stack never contains viable prefixes

gatecse-2015-set1 compiler-design parsing normal viable-prefix

https://gateoverflow.in/8187/gate-cse-2015-set-1-question-1-3

True/False Question

Shift is allowed if after shifting, the content of STACK is still a viable prefix.

True/False Question

Shift is allowed if after shifting, the content of STACK is still a viable prefix.

✓ TRUE

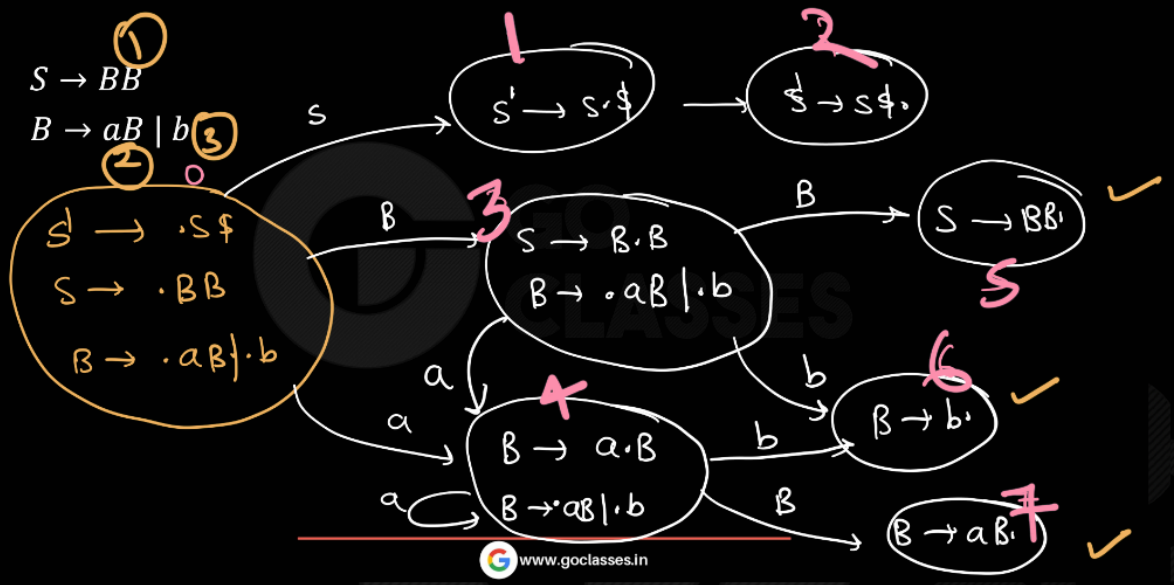

a
b

optimised version of rerun of DFA.

Bottom-up parsing Algorithm

$S \to BB$

$B \to aB \mid b$



Re-run DFA

State 6

State 7

aa b ab $

aa | bab $

aa b | ab $

aaB | ab $

aB | ab

State 7

State 3 → B | ab

accept

S | $

BB | $

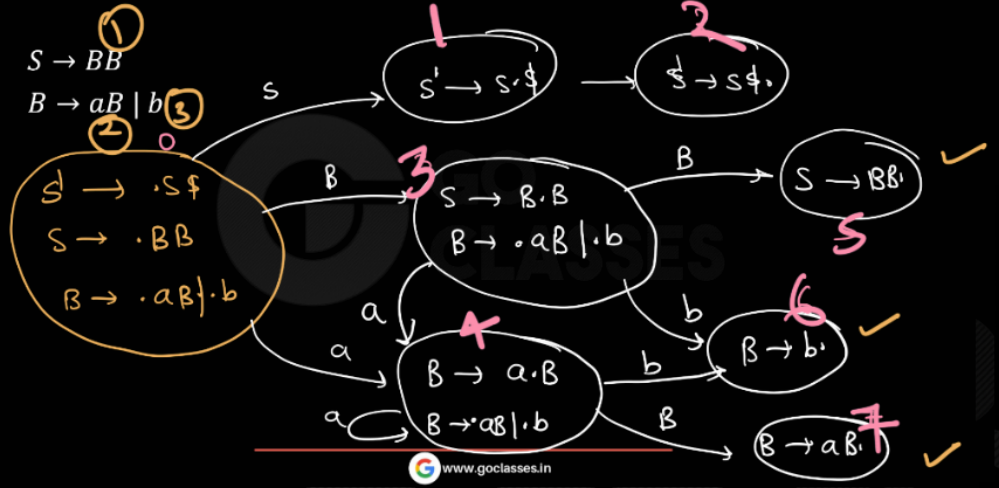
BaB | $

Bab | $

Ba | b $

$S \to BB$ (1)

$B \to aB \mid b$ (2) (3)



Stack contain

State number on top

aabab

0a| abab

0a4| abab

0a4a4| bab

0a4a4b6| ab

0a4a4B7| ab

0a4B7| ab

0B3| ab

0B3a4| b

0B3a4b6| $

0B3a4B7| $ ⇒ 0B3B5| $ ⇒ 0S1| $

ab c

accept

Stack: contains configuration of the form $(s_0 X_1 s_1 X_2 \cdots X_m s_m)$
X_i: grammar symbol, s_i: state

State No-

Symbol

at Start : Stack will have 0 ← State No.

Goal

→ re run of DFA

rerun DFA

→ given any state of LR(0) automaton we can tell possible stack contents

→ viable prefix

→ Bottom parsing algorithm (optimised version of re-run of DFA)