# Theory of Computation

**Grammar**

**Lecture 19_20**

**Gaurav Raj**

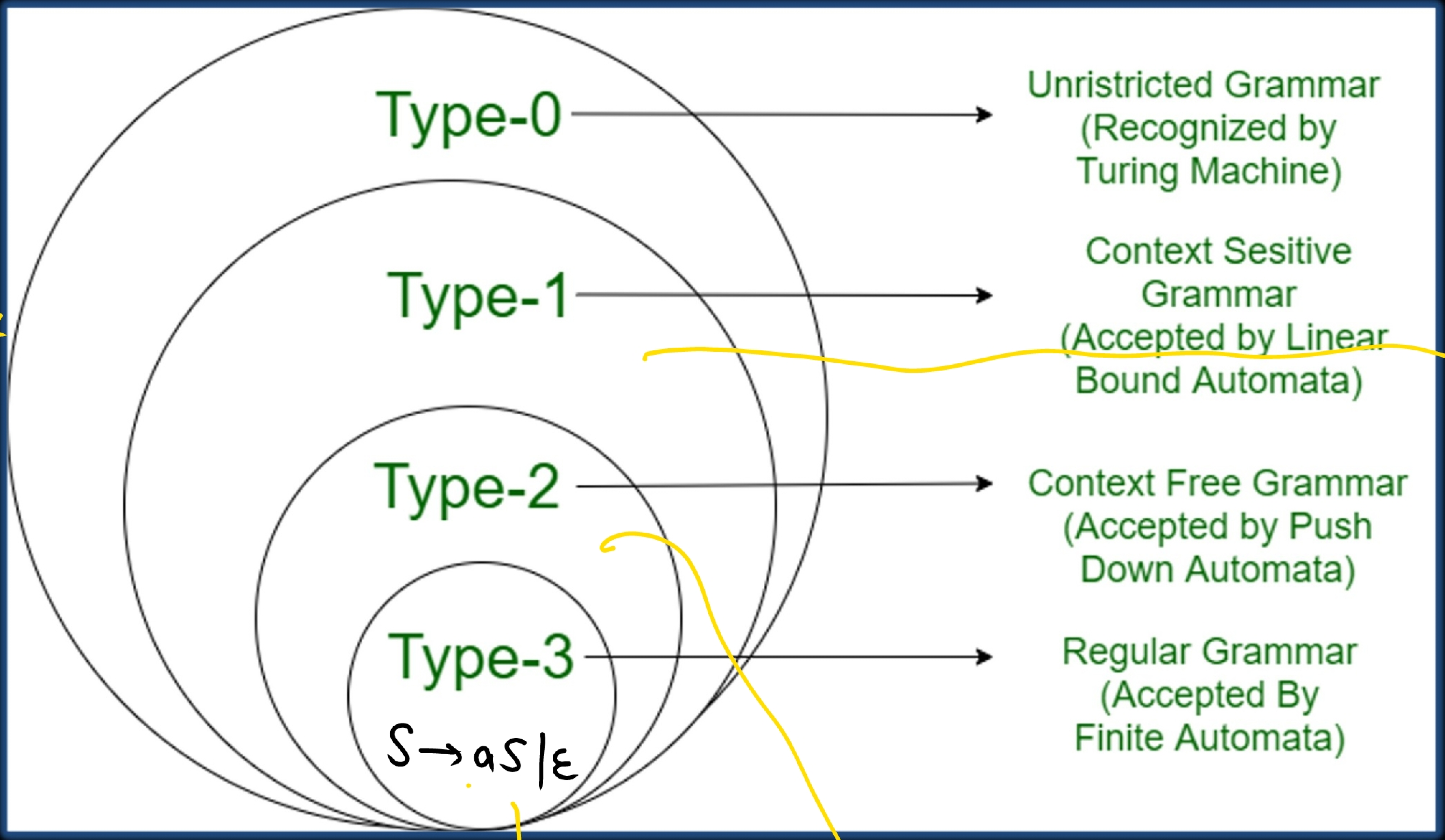
Grammar

$(V \cup T)^*$

$X \rightarrow Y$

$(V \cup T)^+$

at least one Variable




$aS \rightarrow bb$

$a \rightarrow A$

$|X| \leq |Y|$

$S \rightarrow A$
$A \rightarrow a$
→ Type 2

→ Type-2

$aS \rightarrow b$ → Type-0

$S \rightarrow aS \mid \varepsilon$ → ,,

$S \rightarrow AB$
$A \rightarrow \varepsilon$
$B \rightarrow \varepsilon$
→ X

$VT^*$ OR $T^*V$

$V \in \{A, B\}$

$V \in \{\underline{AB}, CD\}$

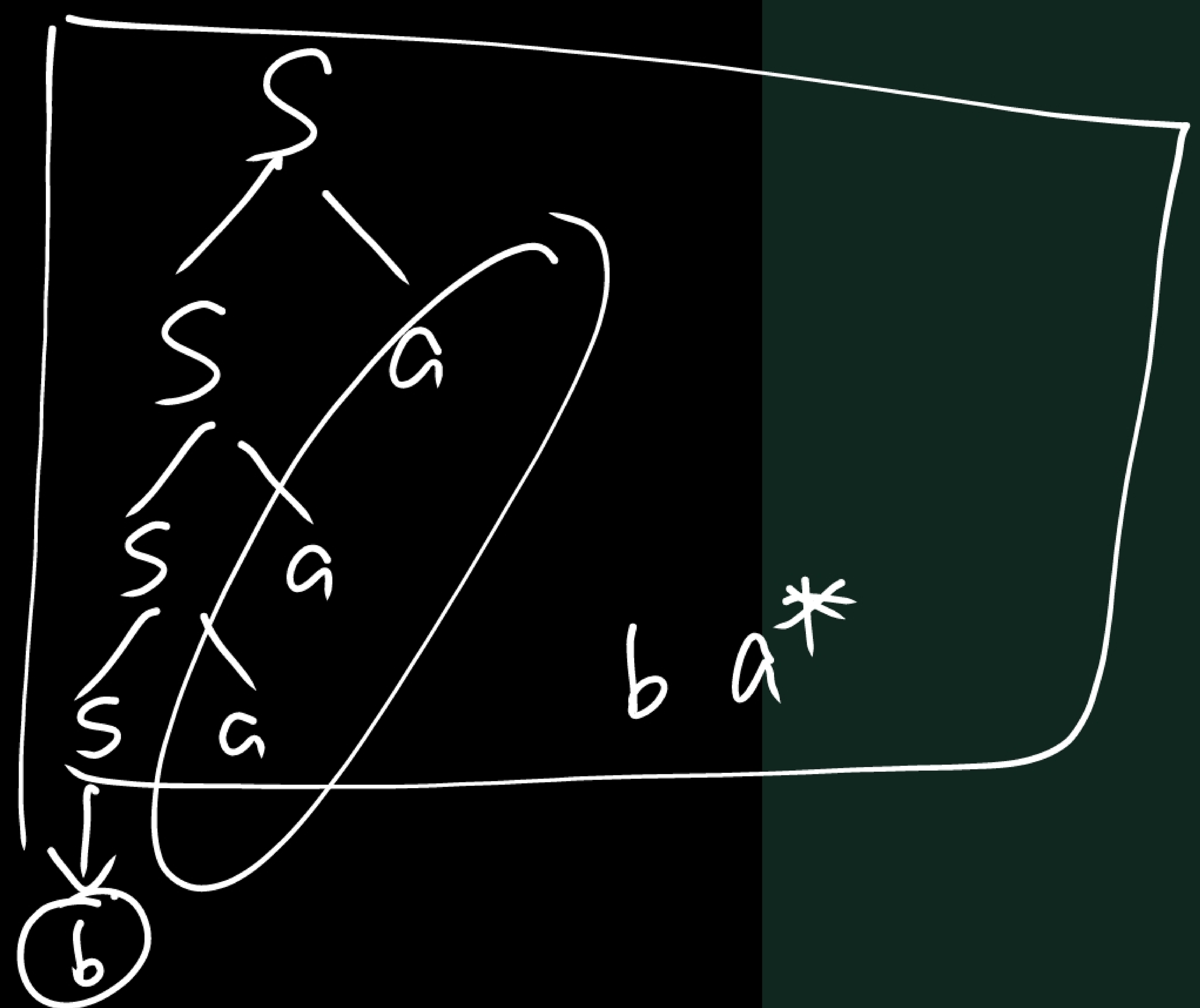
$S \to AB \longrightarrow$ not RLG or LLG

| Type | Grammar Name | Production Rule Form | Restrictions | Recognized By | Language Class |
|---|---|---|---|---|---|
| **Type 0** | Unrestricted Grammar | **α → β** | α must contain at least one non-terminal | Turing Machine | Recursively Enumerable |
| **Type 1** | Context Sensitive Grammar (CSG) | **α → β** | **\|α\|<= \|β\|**<br>RHS\| ≥ \|LHS\|<br>(No shrinking) | Linear Bounded Automaton (LBA) | Context Sensitive |
| **Type 2** | Context Free Grammar (CFG) | **α → β** | Single non-terminal on LHS | Pushdown Automaton (PDA) | Context Free |
| **Type 3** | Regular Grammar | **A → aB** or **A → a** (Right Linear)<br>**A → Ba** or **A → a** (Left Linear) | Only one non-terminal and it must be at one end | Finite Automaton (DFA/NFA) | Regular |

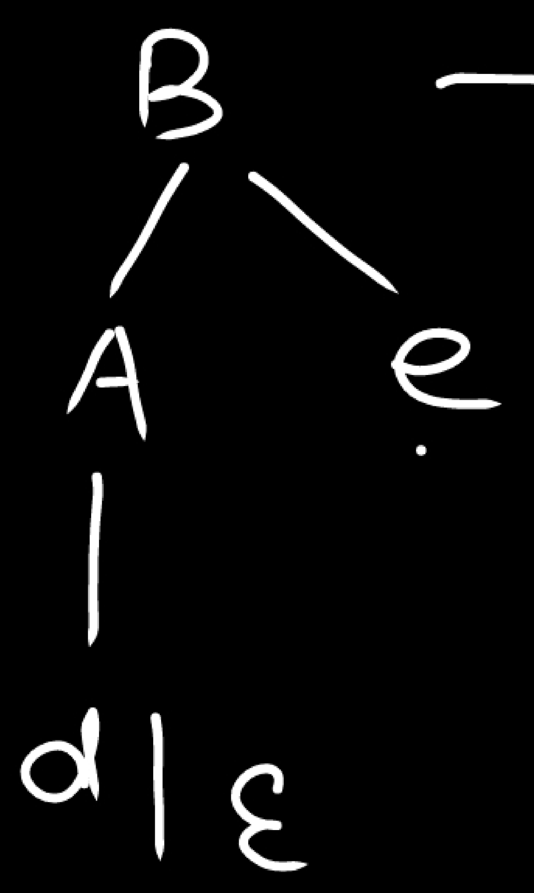
not R.G

1. $S \rightarrow AB$
   $A \rightarrow a \mid \epsilon$
   $B \rightarrow b \mid \epsilon$

$L = \cancel{(ab)^*}$ $\{\varepsilon, a, b, ab\}$



2. $S \rightarrow Aa \mid Ab \mid Bc$
   $A \rightarrow d \mid \epsilon \longrightarrow \{\varepsilon, d\}$
   $B \rightarrow Ae \mid \epsilon \longrightarrow$



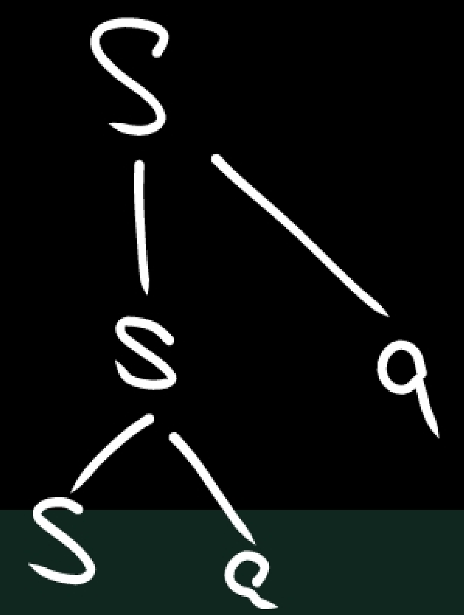
$B \rightarrow \{e, de, \varepsilon\}c$

$L = \{\varepsilon, d\}a + \{\varepsilon + d\}b + \{\varepsilon, e, de\}c$

3. $S \rightarrow Sa \mid b$

4. $S \rightarrow Sa$ $\quad \emptyset$

$S \rightarrow Sa$
$S \rightarrow b$



$ba^*$

3. $S \rightarrow Sa \mid \epsilon$ $\longrightarrow$ $L = a^*$

4. $S \rightarrow Saa \mid \epsilon$ $\longrightarrow$ $L = (aa)^*$

5. $S \rightarrow Saaa \mid \epsilon$ $\longrightarrow$ $(aaa)^*$

6. $S \rightarrow Sa \mid Sb \mid c$ $\longrightarrow$

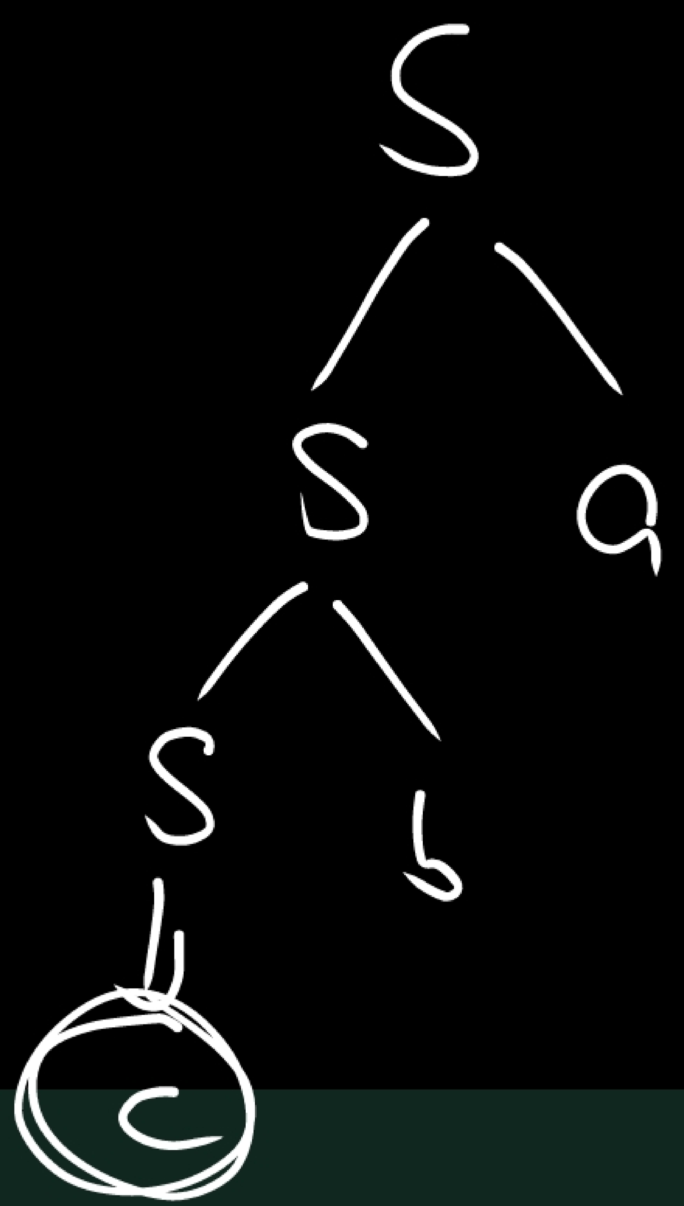
$S \rightarrow Sa$

$S \rightarrow Sb$

$S \rightarrow c$

$c(a+b)^*$

7. $S \rightarrow Sa \mid b \mid c \mid d$ $\longrightarrow$ $(b+c+d)a^*$



8. S → aS | ϵ

9. S → aaS | ϵ  $(aa)^*$

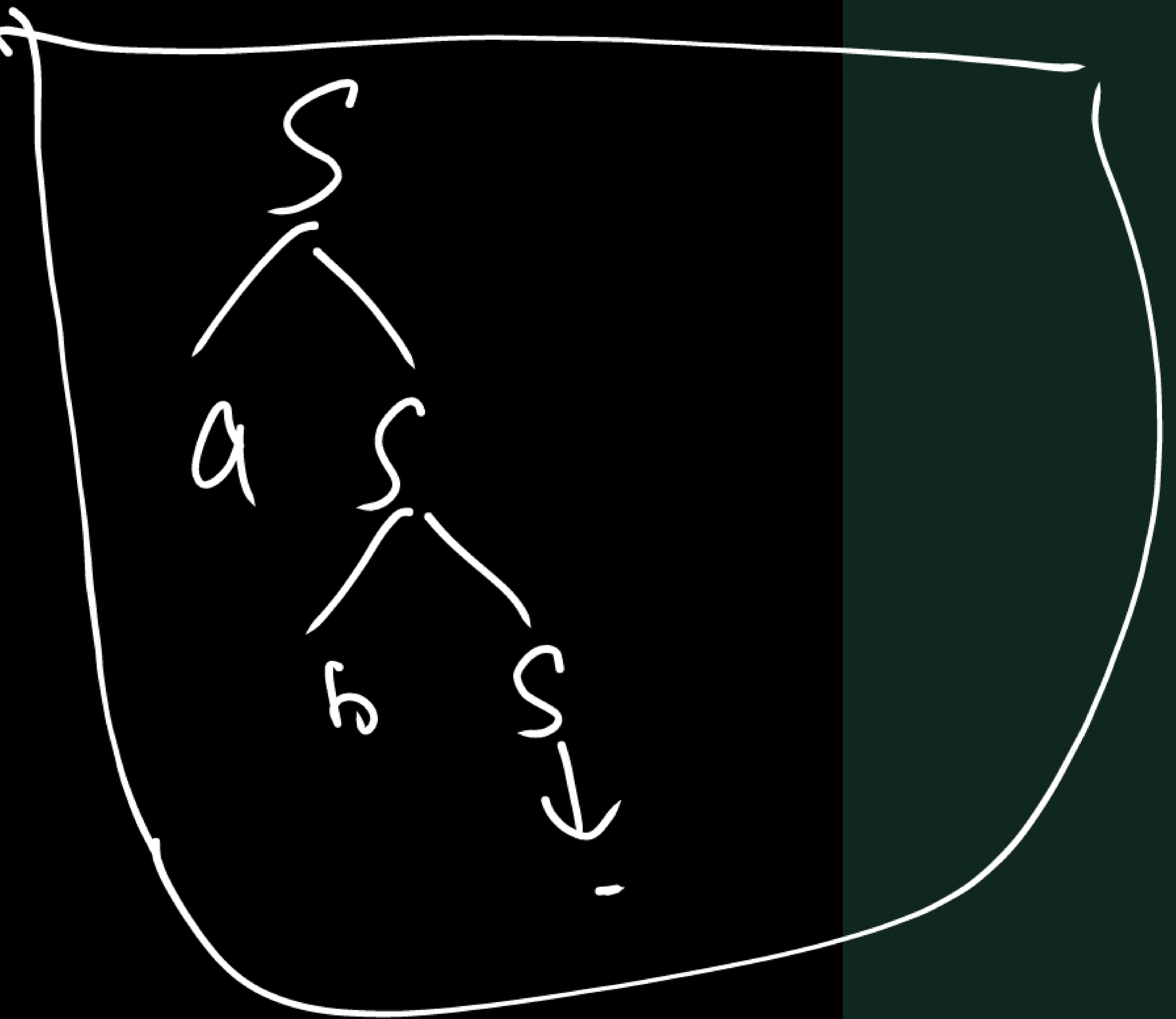
10. S → aS | a  $a^+$

11. S → aS | bS | ϵ  $(a+b)^*$

12. S → aS | bS | a | b

$L = (a+b)^+$

$(a+b)(a+b)^*$



Sa

13. S → Sa | Sb | A
A => Ba
B => Ba | Bb | ϵ

14. S → Sa | Sb | A  $(a+b)^* . a . (a+b)^*$
A => Ba  $(a+b)^* . a$
B => Ba | Bb | ϵ  $(a+b)^*$

1. What is the highest type number that can be assigned to the following grammar?

$\underbrace{S \rightarrow Aa}_{LLG}$, $\underbrace{A \rightarrow Ba}_{LLG}$, $\underbrace{B \rightarrow abc}_{both}$ LLG

MCQ

A. Type 0

B. Type 1

C. Type 2

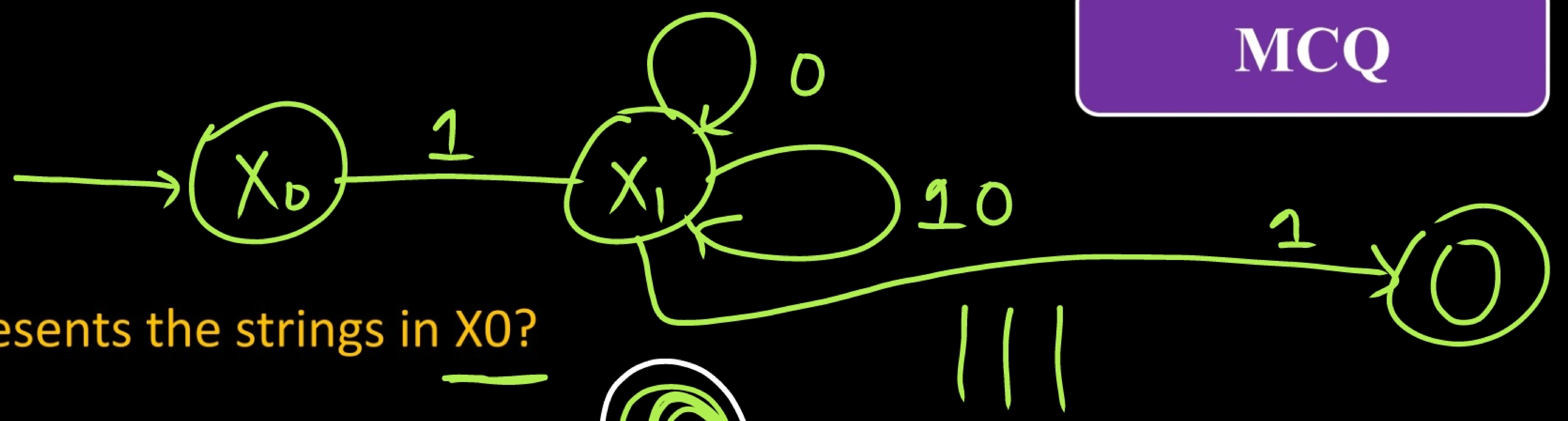
D. Type 3 ✓

$V \in \{A, B, S\}$

MCQ

2. Consider the alphabet Σ={0, 1}, the null/empty string λ and the sets of strings X0, X1, and X2 generated by the corresponding non-terminals of a regular grammar. X0, X1, and X2 are related as follows.

X0 = 1X1
X1 = 0X1 + 1X2
X2 = 0X1 + {λ}



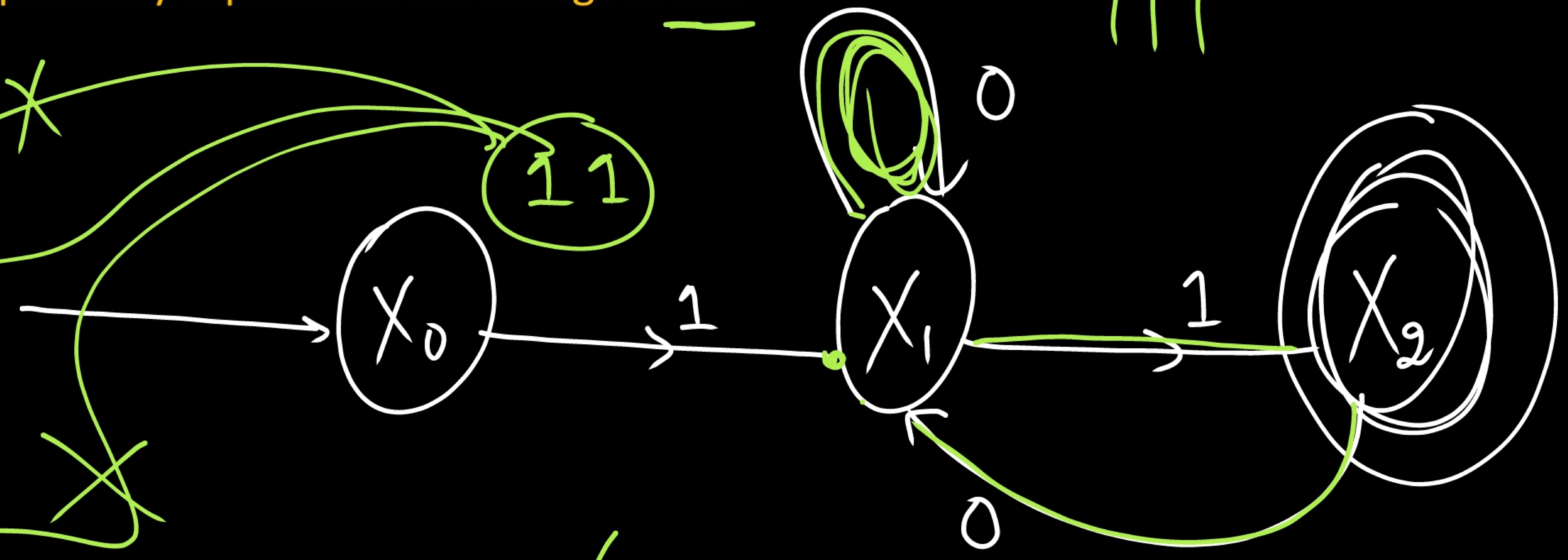

Which one of the following choices precisely represents the strings in X0?

A. 10(0* + (10)*)1  101

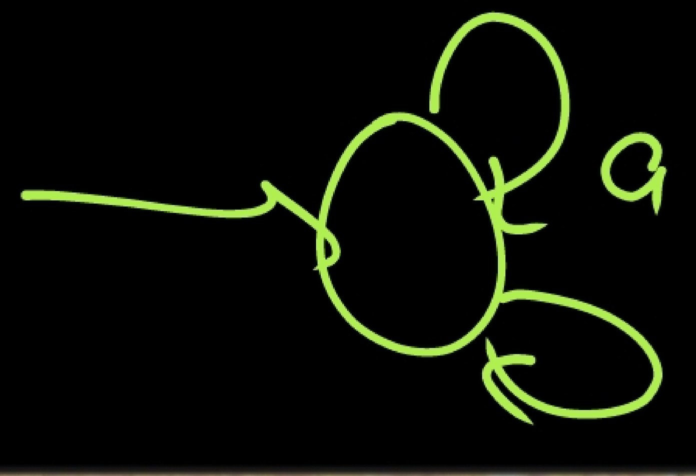
B. 10(0* + (10)*)*1

C. $\boxed{\text{1(0 + 10)*1}}$

D. 10(0 + 10)*1 + 110(0 + 10)*1




$1(0+10)^*1$




$(a+b)^*$

3. Let us Consider the regular grammar below

S→ bS | aA | ϵ

A→ aS | bA

The language generated by the states S and A by the above grammar are

MCQ

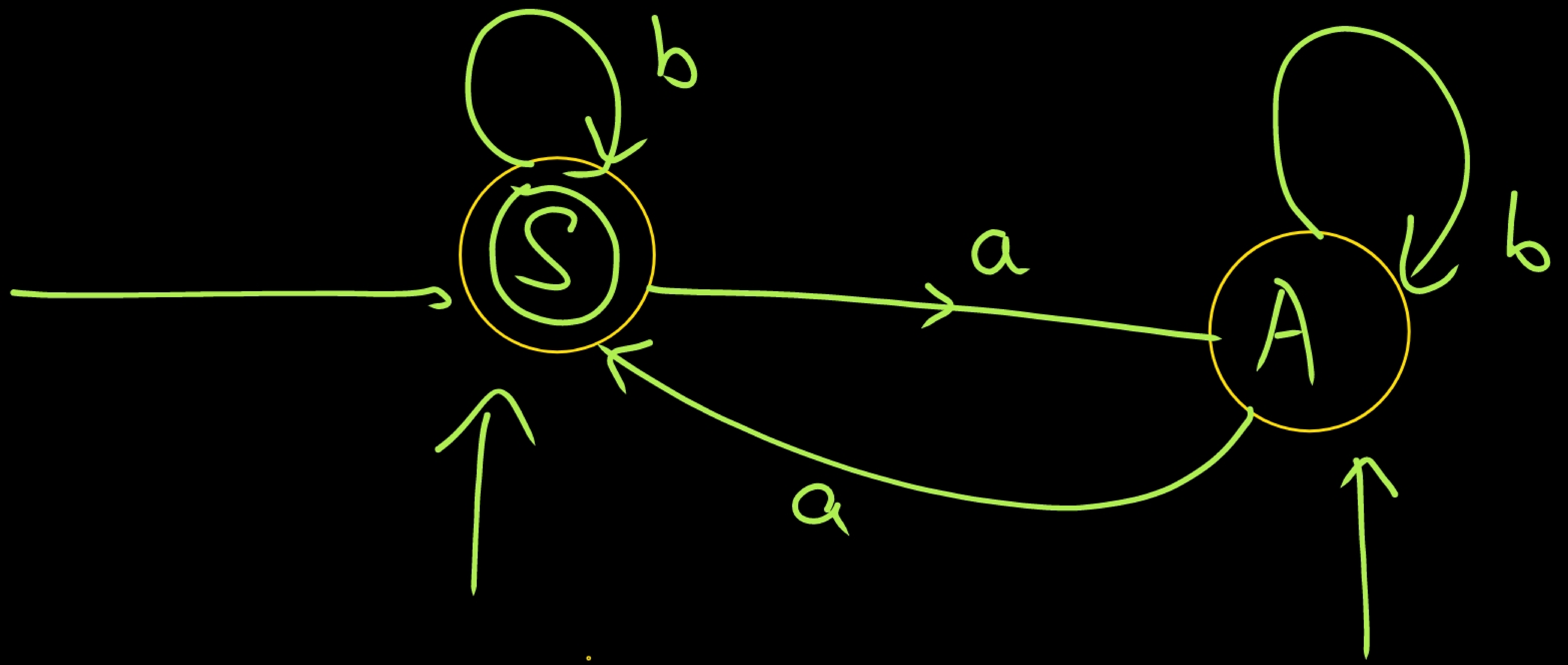
lock

A. {w ∈ (a+b)* | #a(w) is even } and {w ∈ (a+b)* | #a(w) is odd}

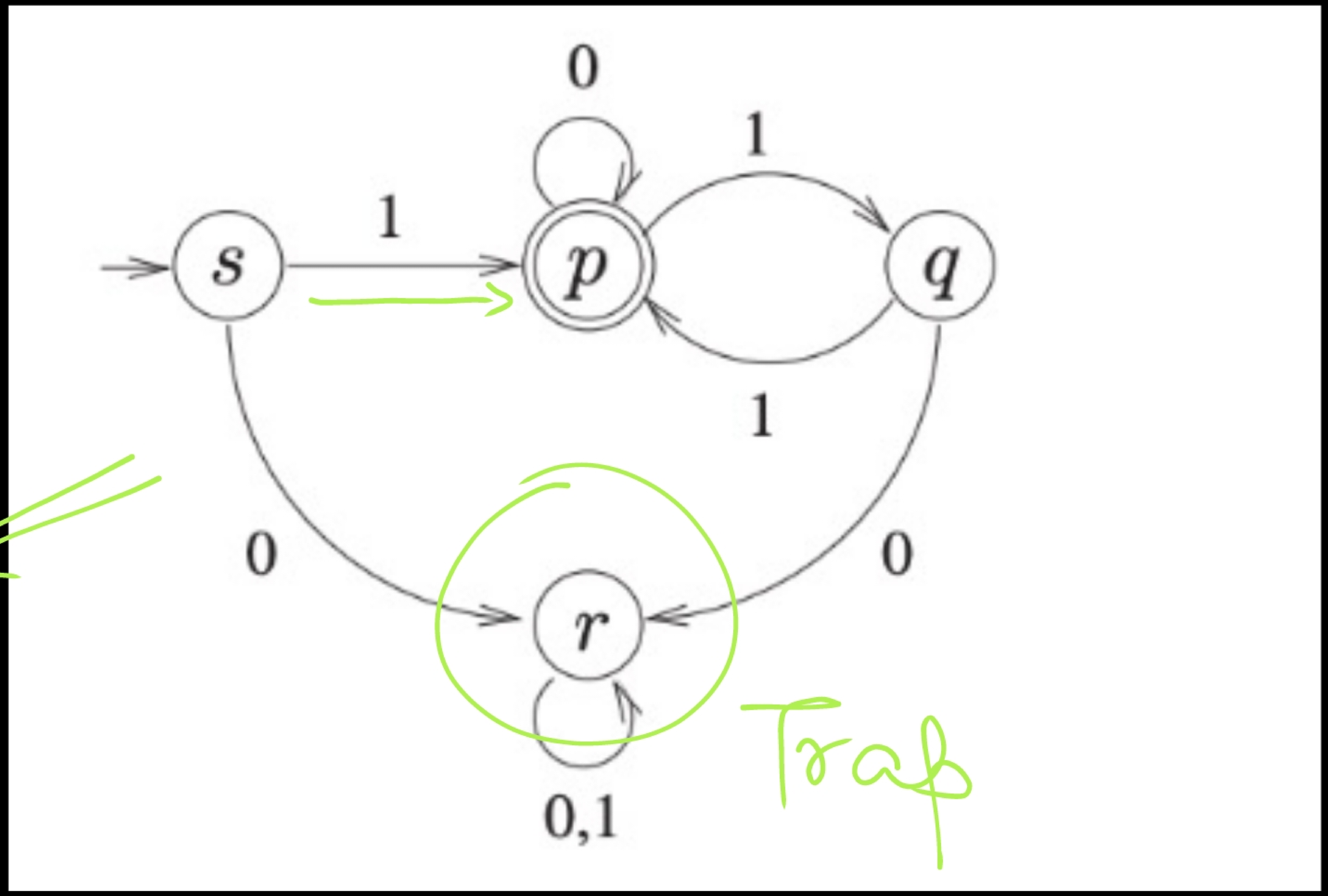
B. {w ∈ (a+b)* | #a(w) is even } and {w ∈ (a+b)* | #b(w) is odd}

C. {w ∈ (a+b)* | #a(w) = #b(w) } and {w ∈ (a+b)* | #a(w) != #b(w) }

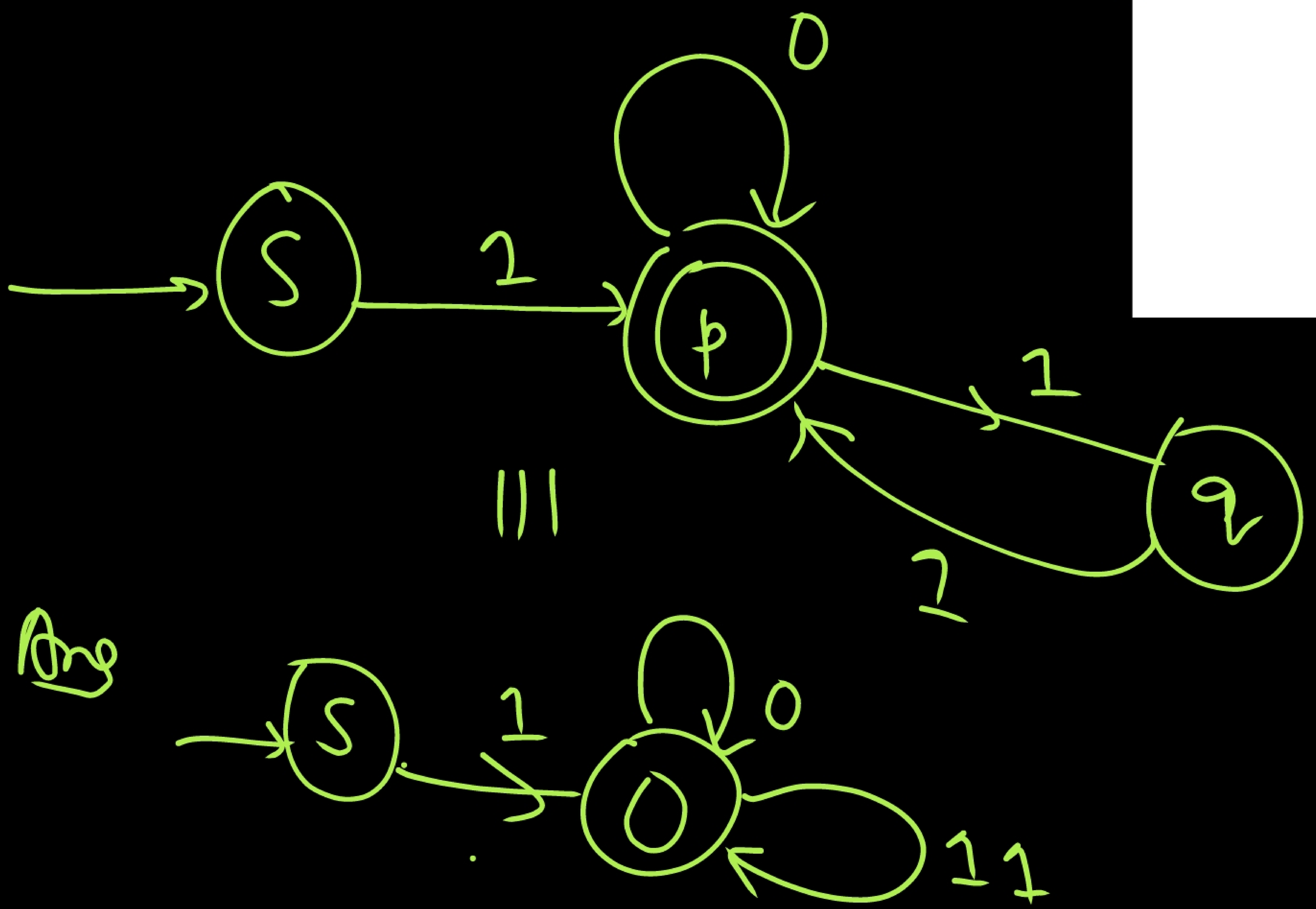
D. {ϵ}, {wa |w ∈ (a+b)* and {wb | w ∈ (a+b)* }



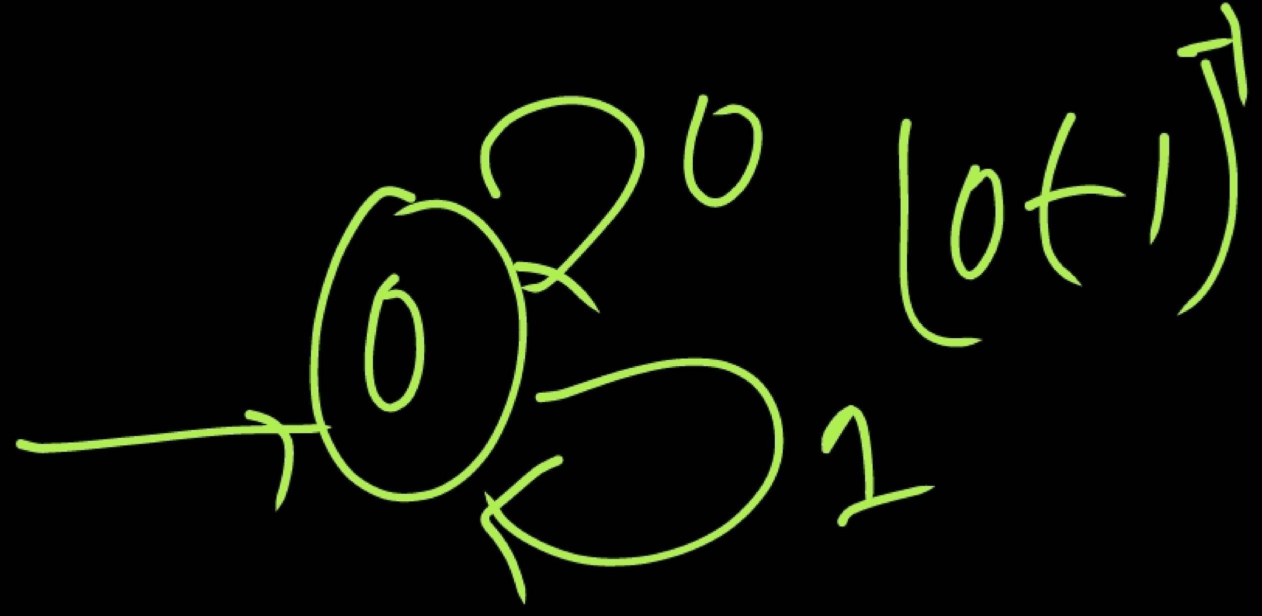
4. Consider the Deterministic Finite-state Automaton (DFA) shown below. The DFA runs on the alphabet {0, 1}, and has the set of states {s, p, q, r}, with s being the start state and p being the only final state. Which one of the following regular expressions correctly describes the language accepted by A?




**MCQ**




A. 1(0*11)*
B. 0(0 + 1)*
C. 1(0 + 11)* Ans
D. 1(110*)*



5. Which of the following regular expressions represent(s) the set of all binary numbers that are divisible by three? Assume that the string ϵ is divisible by three.

**MSQ**

A. $(0+1(01^*0)^*1)^*$ ✓

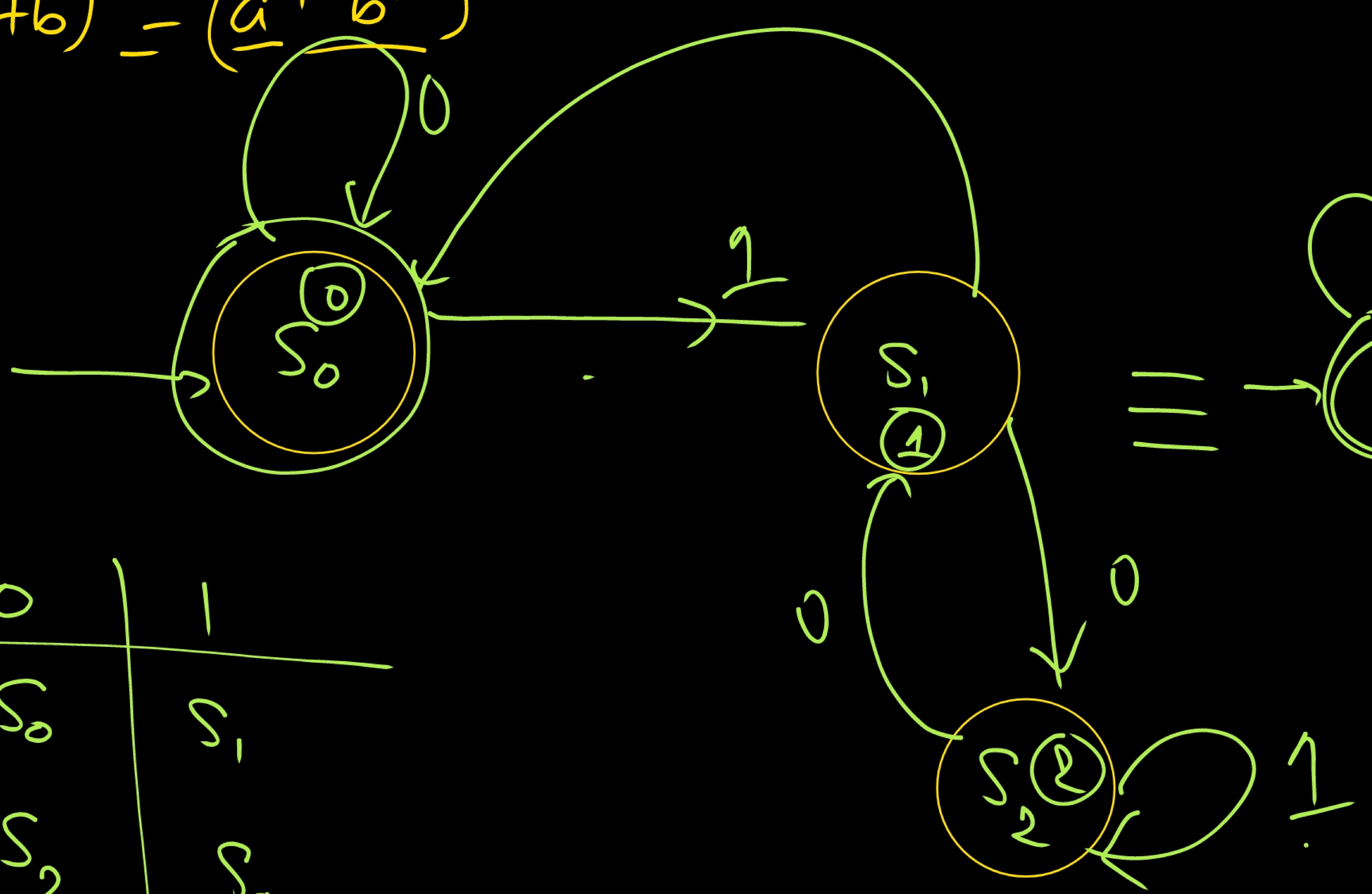
$\underbrace{0}_{a}+\underbrace{1(01^*0)^*1}_{b}$

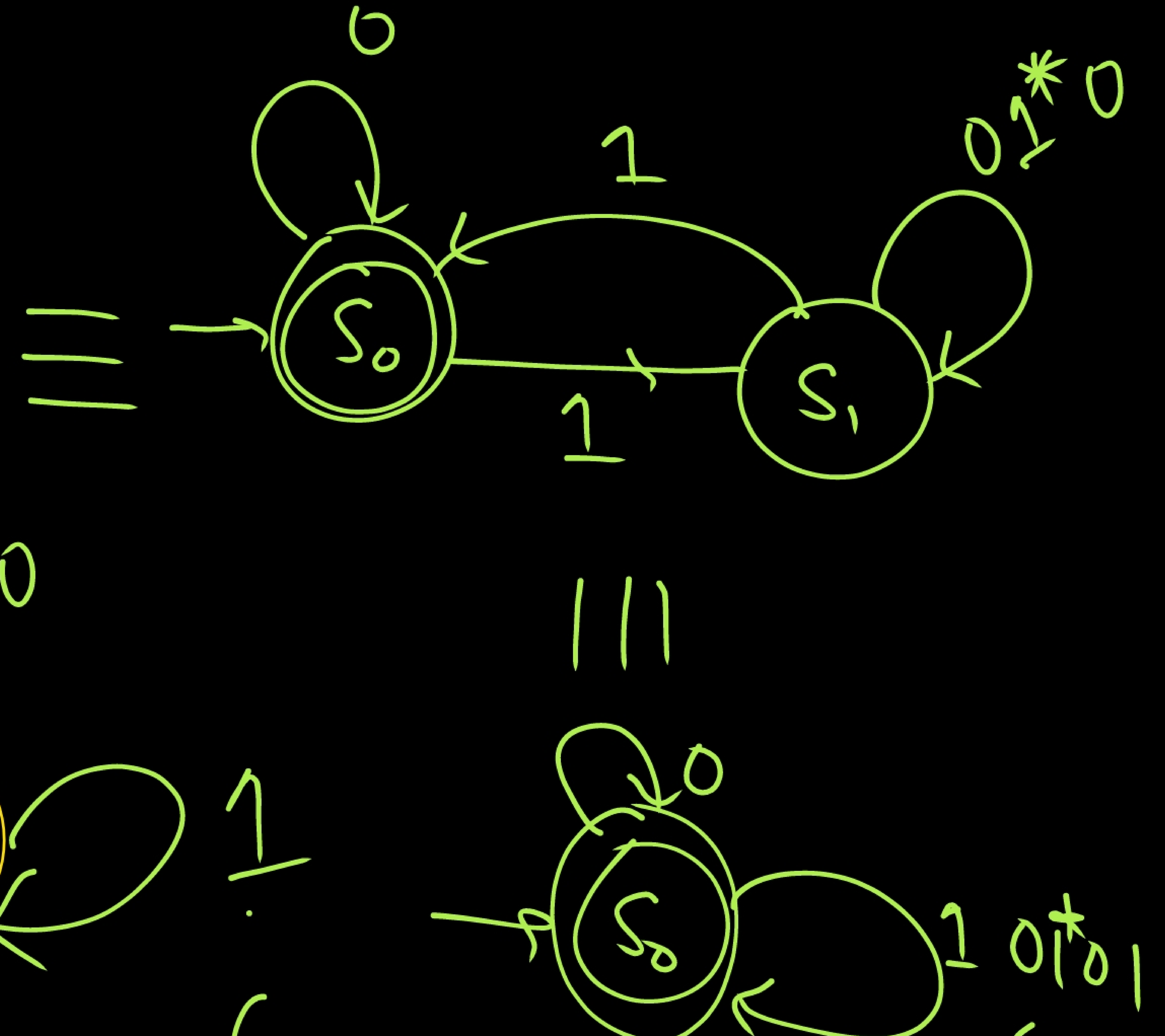
$(a+b)^* = (a^* b^*)^*$

B. $(0+11+10(1+00)^*01)^*$ ✓

C. $(0^*(1(01^*0)^*1)^*)^*$

D. $(0+11+11(1+00)^*00)^*$





$(0+101^*01)^* \rightarrow A$

| | 0 | 1 |
|---|---|---|
| $S_0$ | $S_0$ | $S_1$ |
| $S_1$ | $S_2$ | $S_0$ |
| $S_2$ | $S_1$ | $S_2$ |

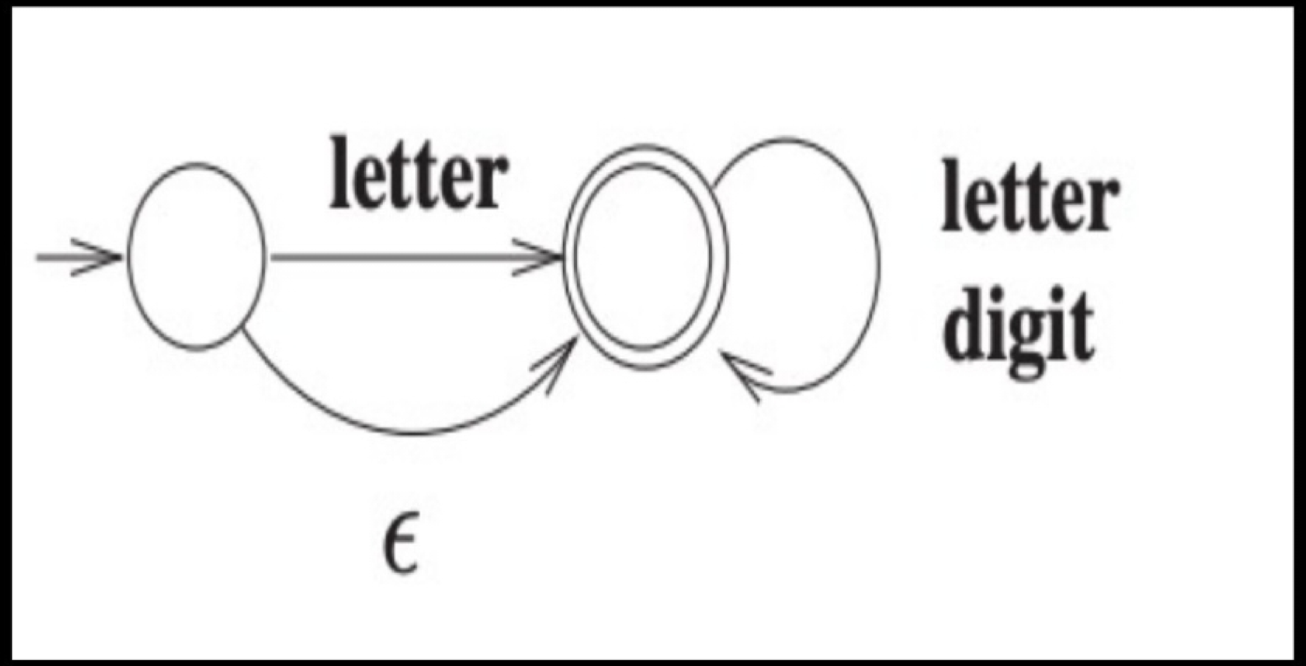
6. Consider the following definition of a lexical token id for an identifier in a programming language, using extended regular expressions:

letter → [A.......Z a........z]
digit → [0....9]
id → letter(letter | digit)*

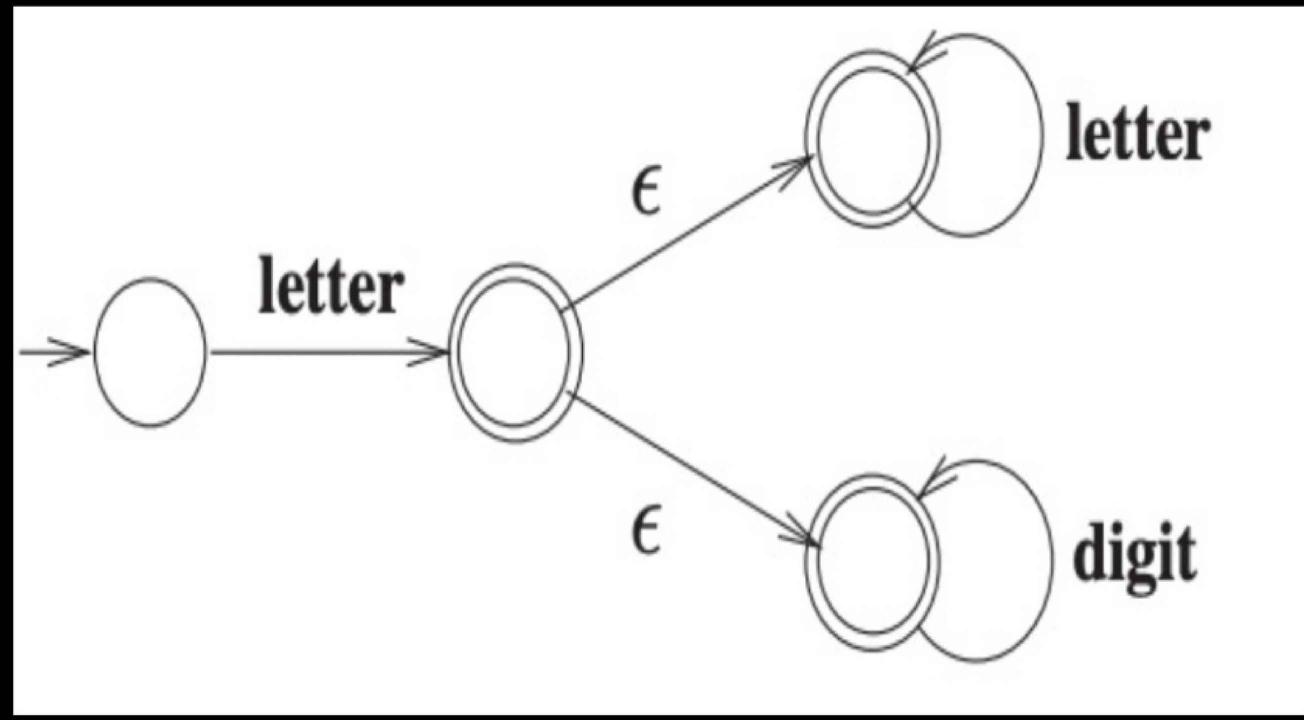
Which one of the following Non-deterministic Finite-state Automata with - transitions accepts the set of valid identifiers? (A double-circle denotes a final state)

**MCQ**

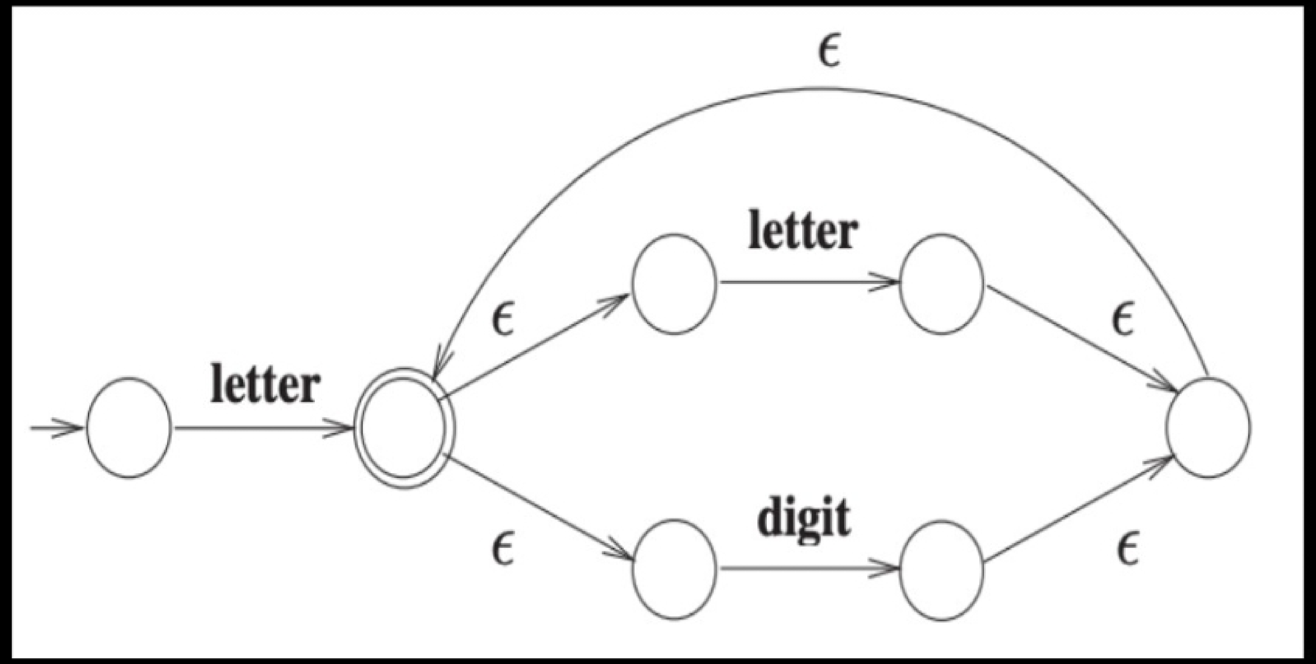
A.



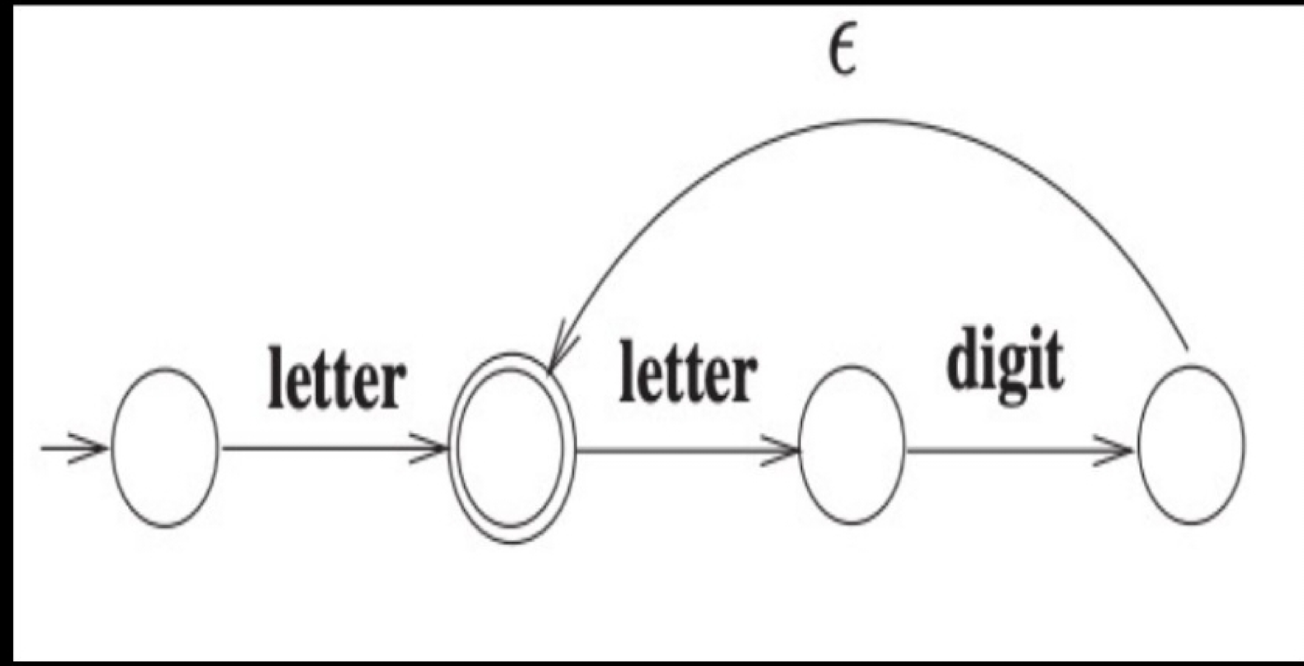

B.




C.




D.



7. Which one of the following regular expressions represents the set of all binary strings with an odd number of 1’s?

MCQ

A. ((0+1)*1(0+1)*1)*10*

B. (0*10*10*)*0*1

C. 10*(0*10*10*)*

D. (0*10*10*)*10*

# Thank You