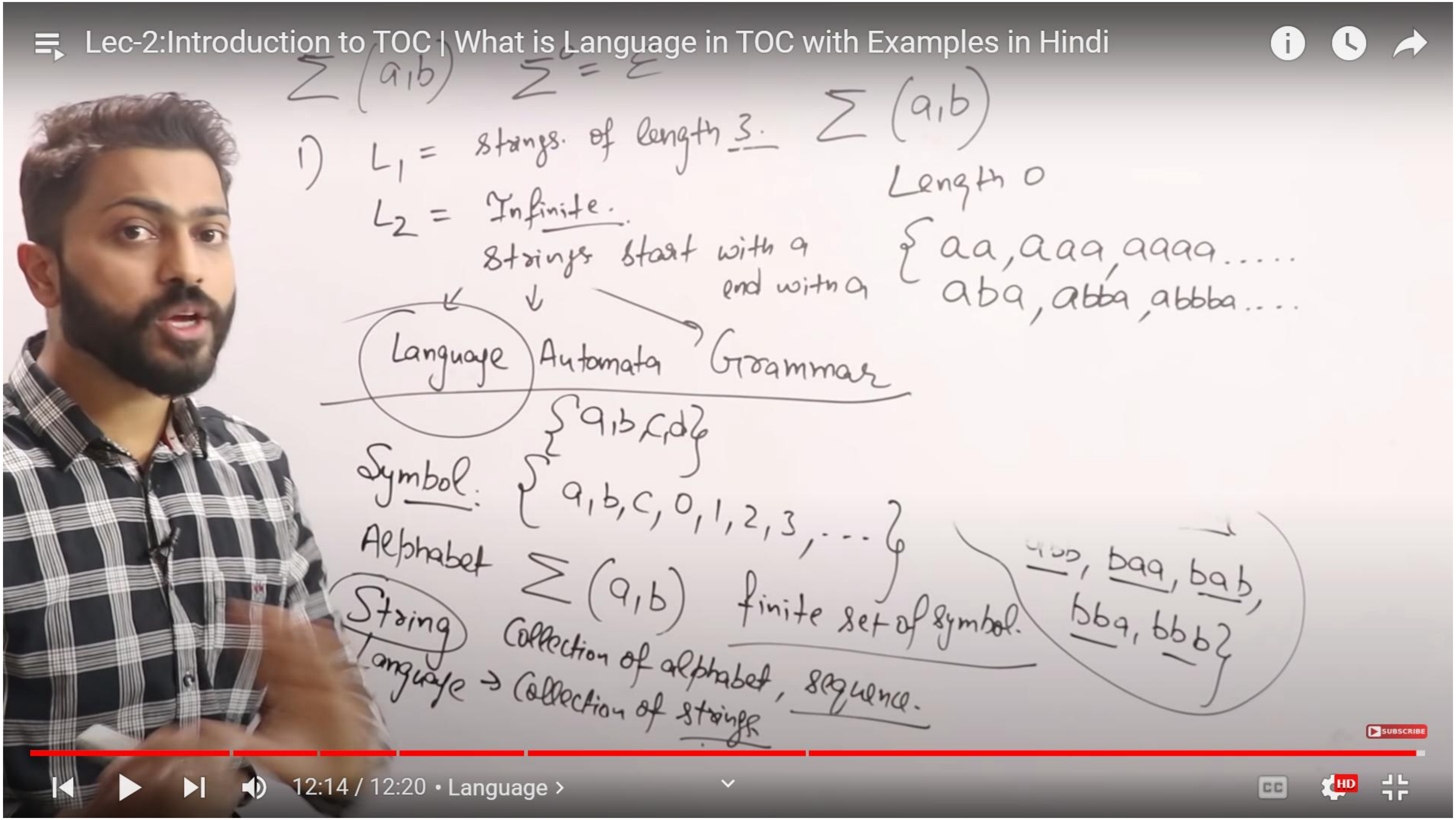
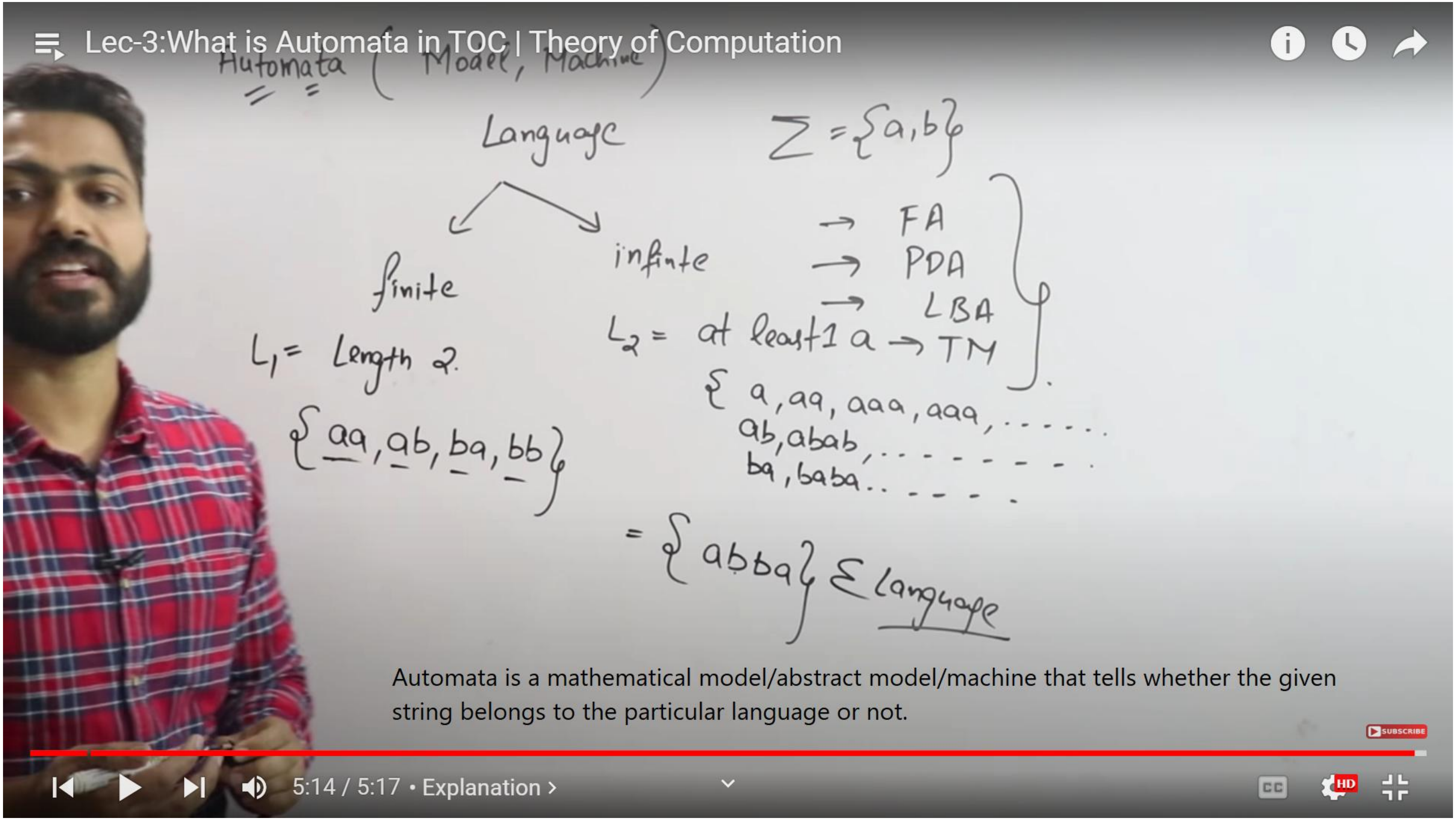

Lec-2:Introduction to TOC | What is Language in TOC with Examples in Hindi
1) L1 = strings of length 3.
L2 = Infinite.
Strings start with a end with a
Length 0
{aa, aaa, aaaa.....
aba, abba, abbba....
Language Automata Grammar
{a,b,c,d}
Symbol: {a,b,c,0,1,2,3,...}
Alphabet Σ(a,b) finite set of symbol.
String Collection of alphabet, sequence.
Language → Collection of strings
baa, bab, bba, bbb}
12:14 / 12:20 • Language ›


Lec-3:What is Automata in TOC | Theory of Computation
Automata (Model, Machine)
Language
$\Sigma = \{a, b\}$
finite
infinite
FA
PDA
LBA
TM
$L_1 =$ Length 2.
$\{aa, ab, ba, bb\}$
$L_2 =$ at least 1 a
$\{a, aa, aaa, aaa, \ldots$
$ab, abab, \ldots$
$ba, baba \ldots$
$= \{abba\} \in$ language
Automata is a mathematical model/abstract model/machine that tells whether the given string belongs to the particular language or not.
SUBSCRIBE
5:14 / 5:17 • Explanation ›
HD

Power of Σ $\quad\quad$ $\Sigma = \{a, b\}$

Σ^0 = Set of all strings with length '0' = λ, ε $\quad 2^0$

Σ^1 = " " " " " " 1 $\{a, b\}$ $\quad 2^1$

Σ^2 = " " " " " " 2 $\quad \Sigma^* = \Sigma^+ + \underset{\varepsilon}{\Sigma^0}$

$\Sigma^* - \varepsilon = \Sigma^+$

$\Sigma . \Sigma \quad \{a,b\}\{a,b\} = \{\ldots, bb\}$ $\quad 2^2$

Σ^3 = " " " " " 3

$\Sigma.\Sigma.\Sigma = \{a,b\}\{a,b\}\{a,b\}$ $\quad \{aa, ab, ba, bb\}\{a,b\}$ $\quad 2^3$

$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

$(a+b)^*$ $\quad 2^n$

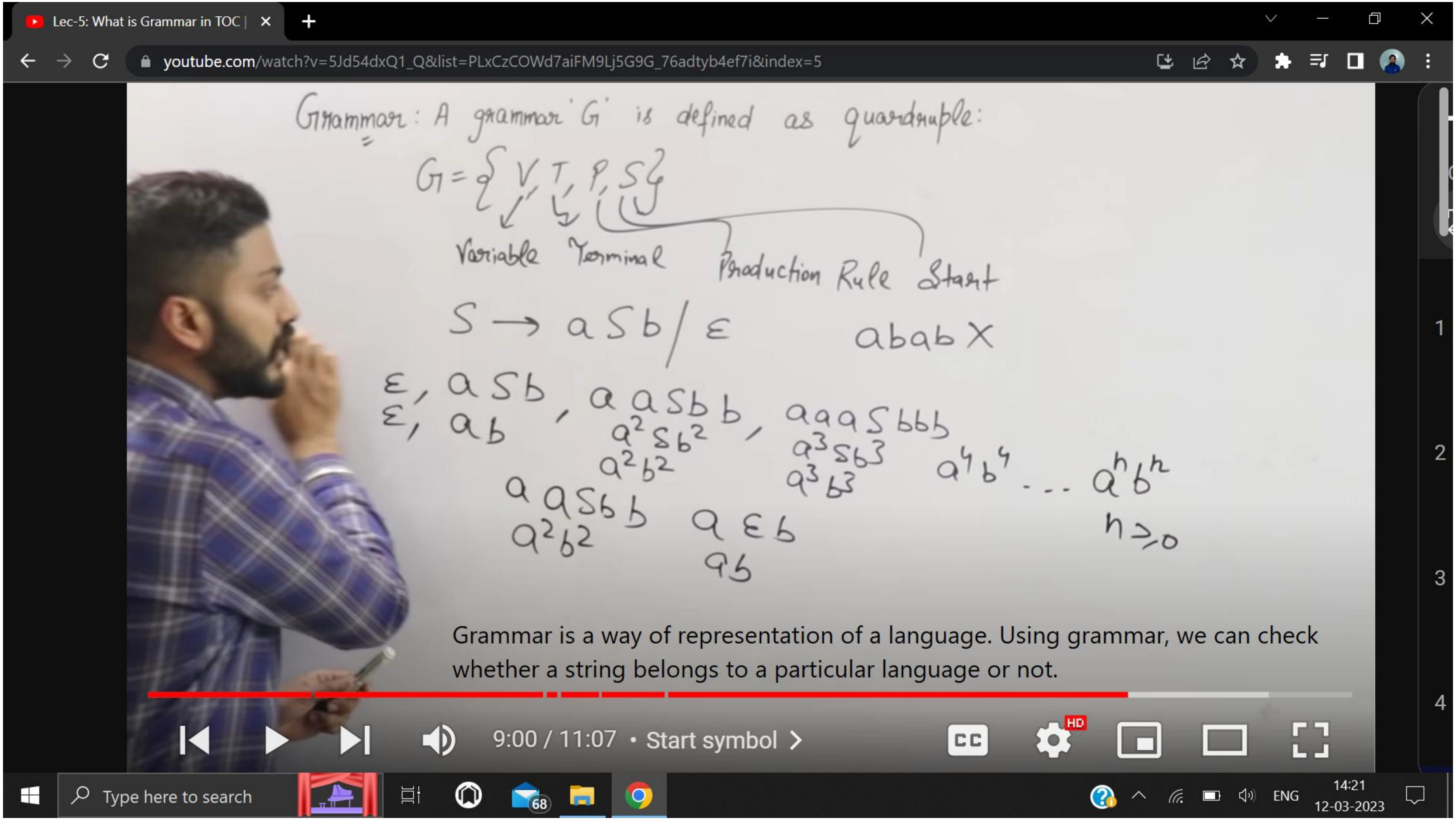
Σ^4

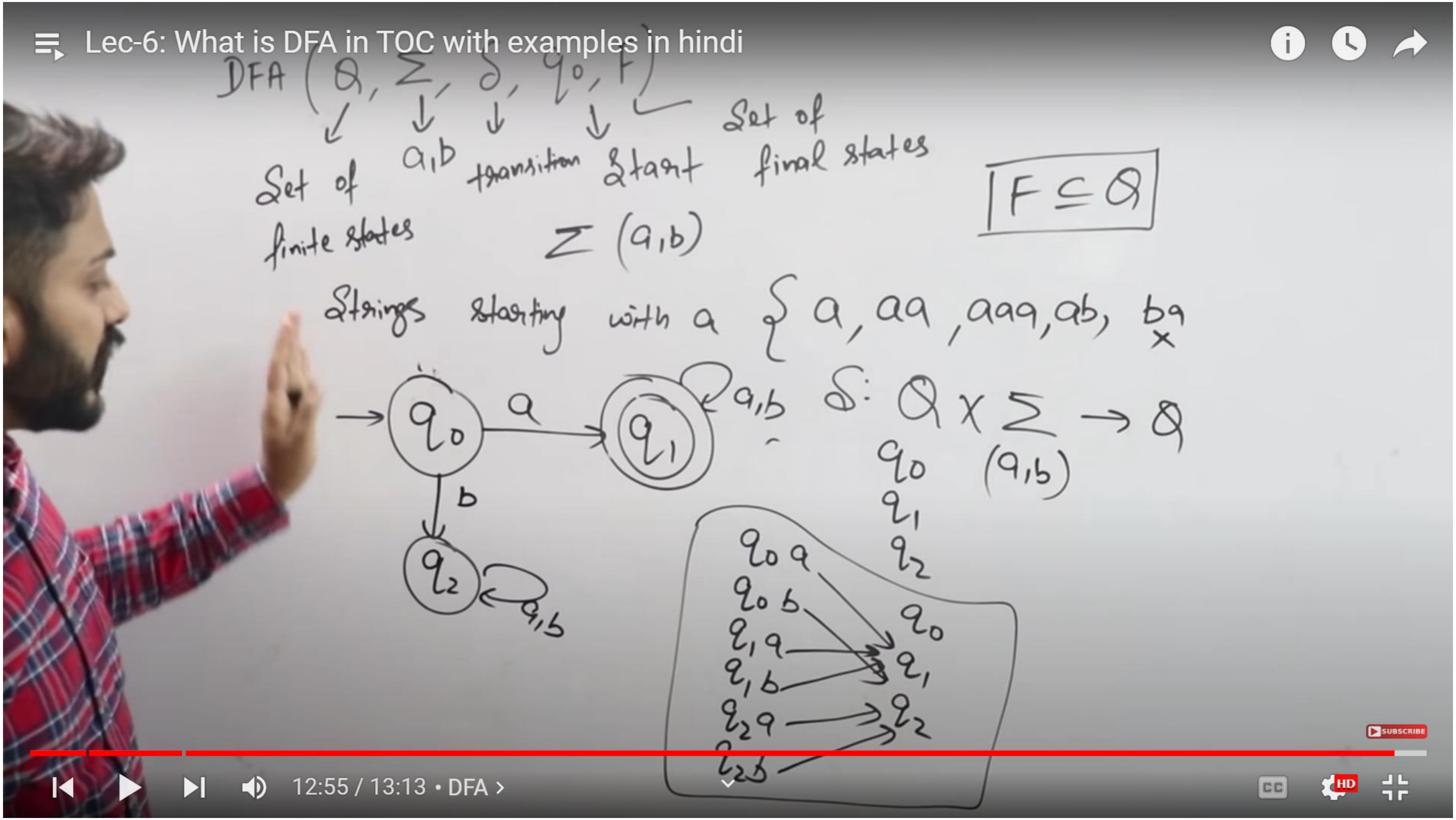
$\vdots$

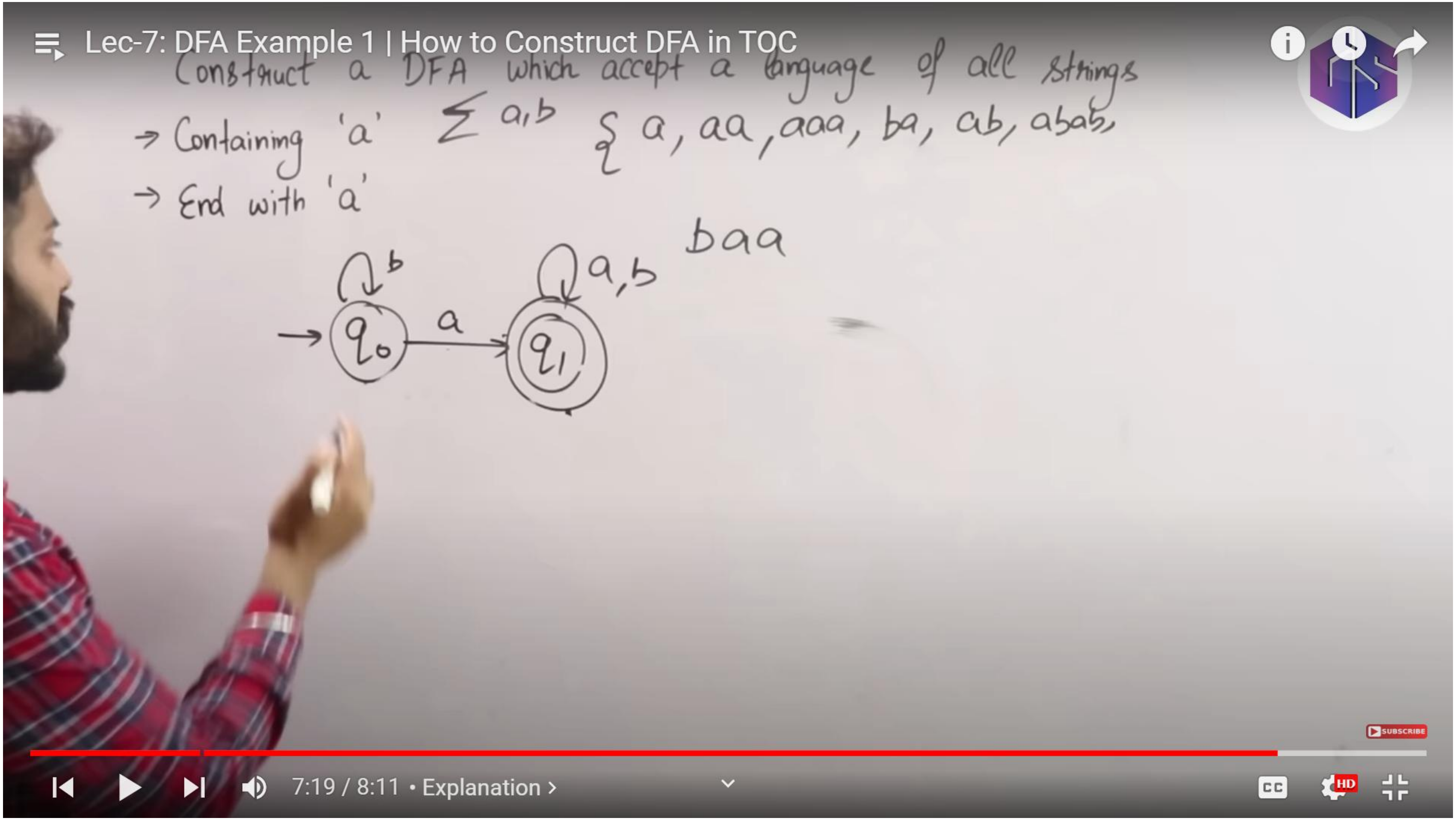
Σ^* (Kleene closure) = Infinite language.

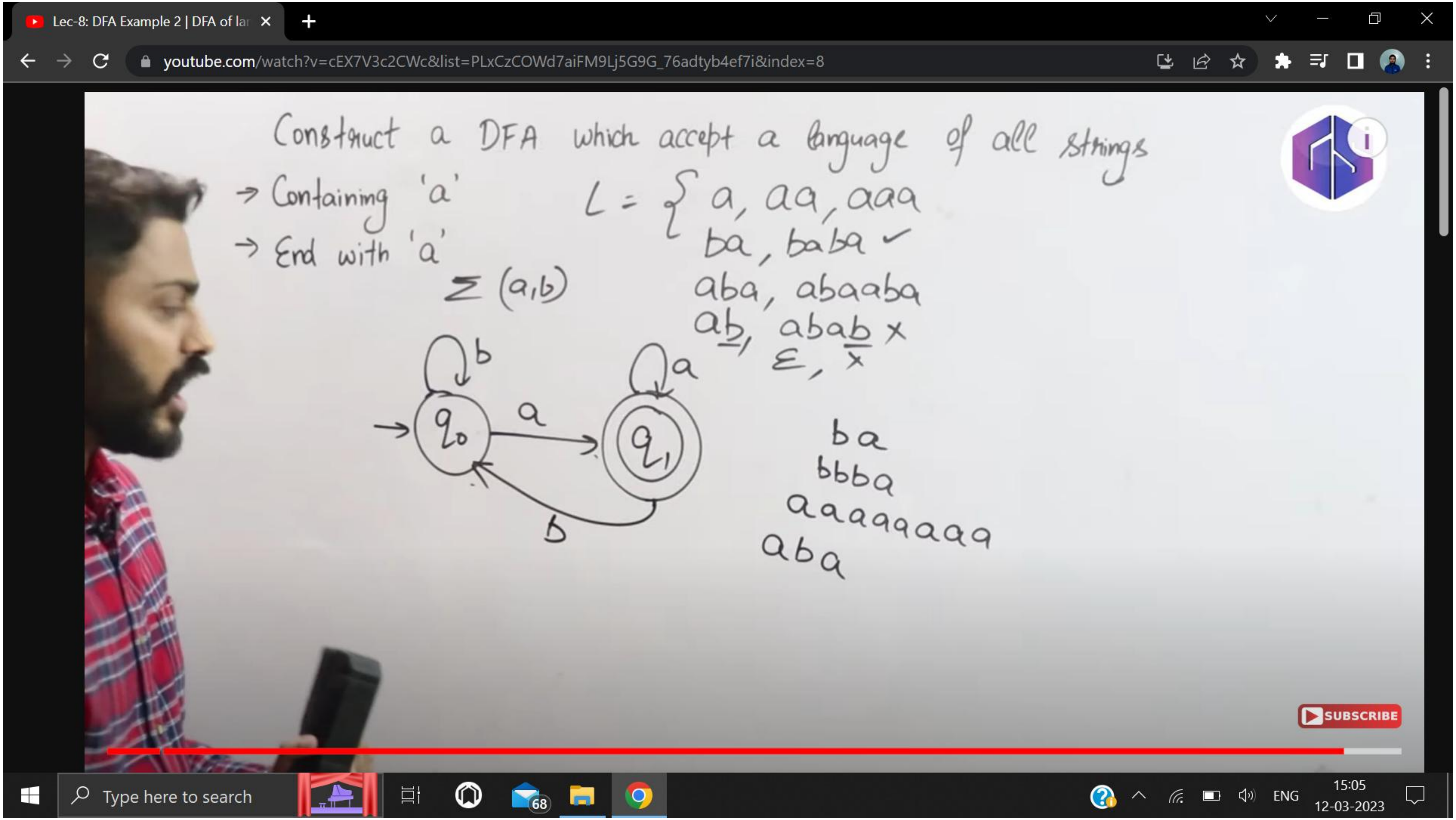
Σ^+ Positive closure

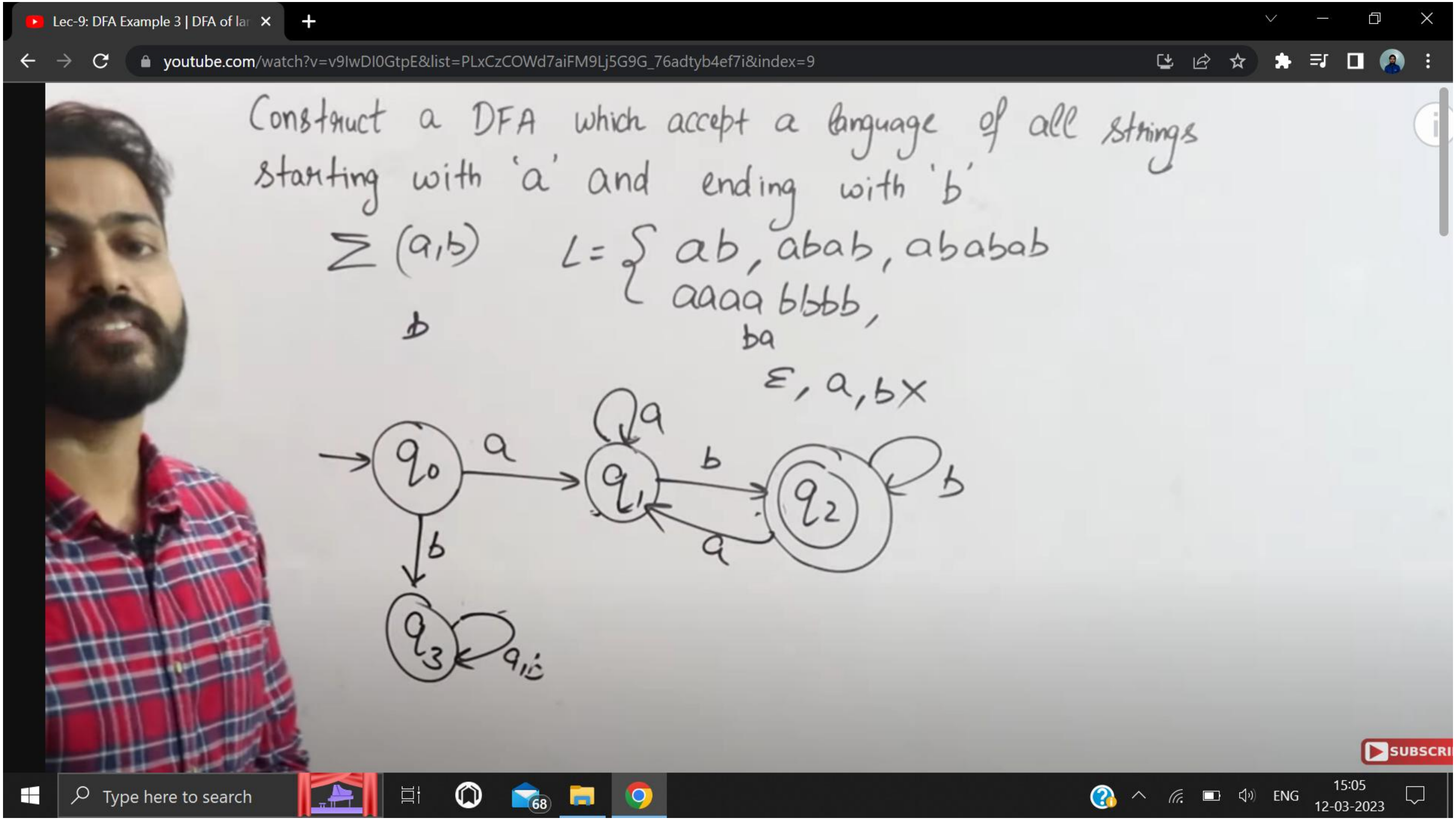
Identity element (E) exists in E*(kleene closure) and not exists in E+(positive closure)

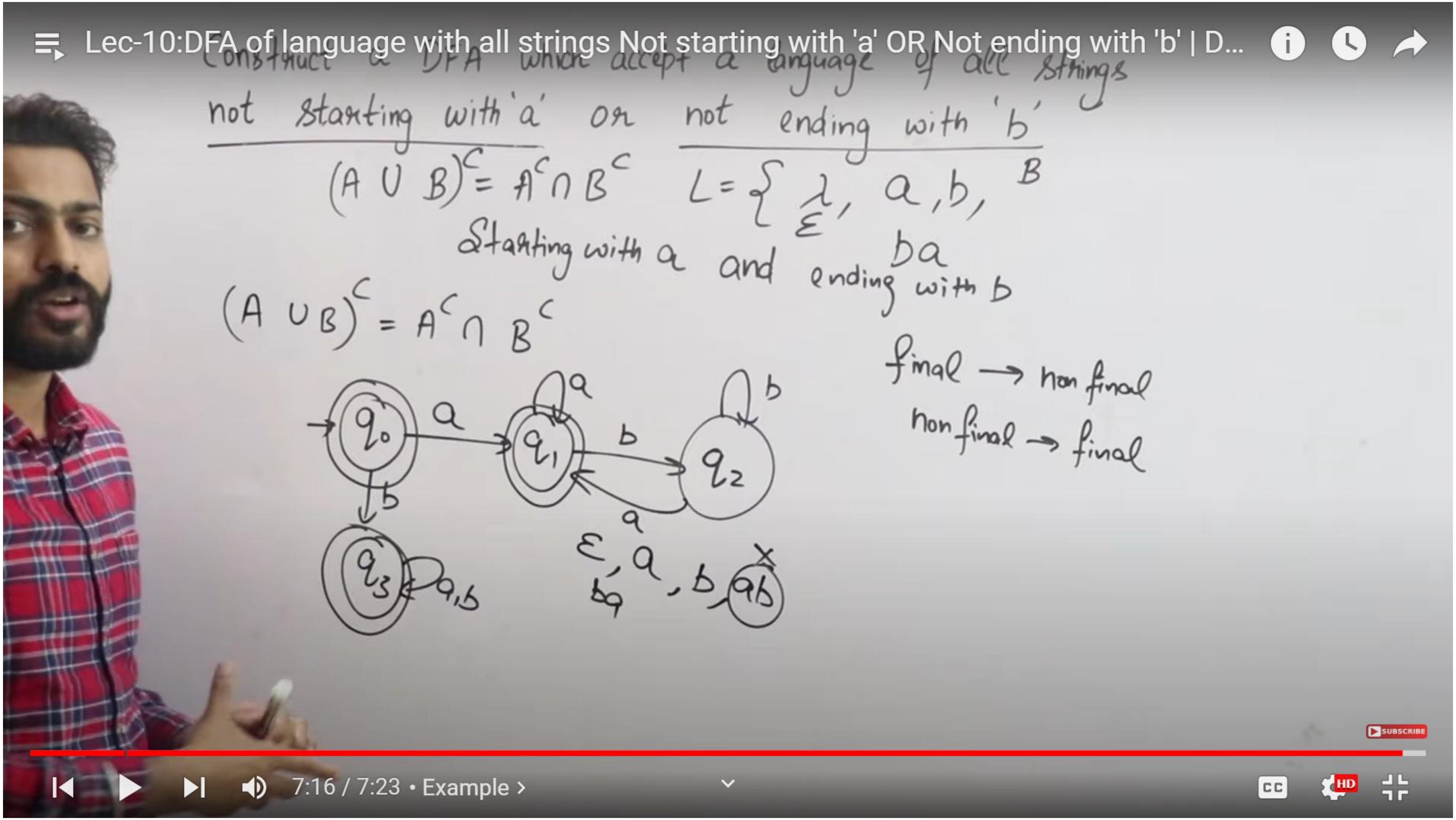

Lec-5: What is Grammar in TOC |
youtube.com/watch?v=5Jd54dxQ1_Q&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=5
Grammar: A grammar 'G' is defined as quardruple:
G = {V, T, P, S}
Variable
Terminal
Production Rule
Start
S → aSb / ε
abab X
ε, aSb, aaSbb, aaaSbbb
ε, ab, a²Sb², a³Sb³
a²b²
a³b³
a⁴b⁴ ... aⁿbⁿ
n ≥ 0
aaSbb
a²b²
aεb
ab
Grammar is a way of representation of a language. Using grammar, we can check whether a string belongs to a particular language or not.
9:00 / 11:07 • Start symbol
HD
Type here to search
ENG
14:21
12-03-2023

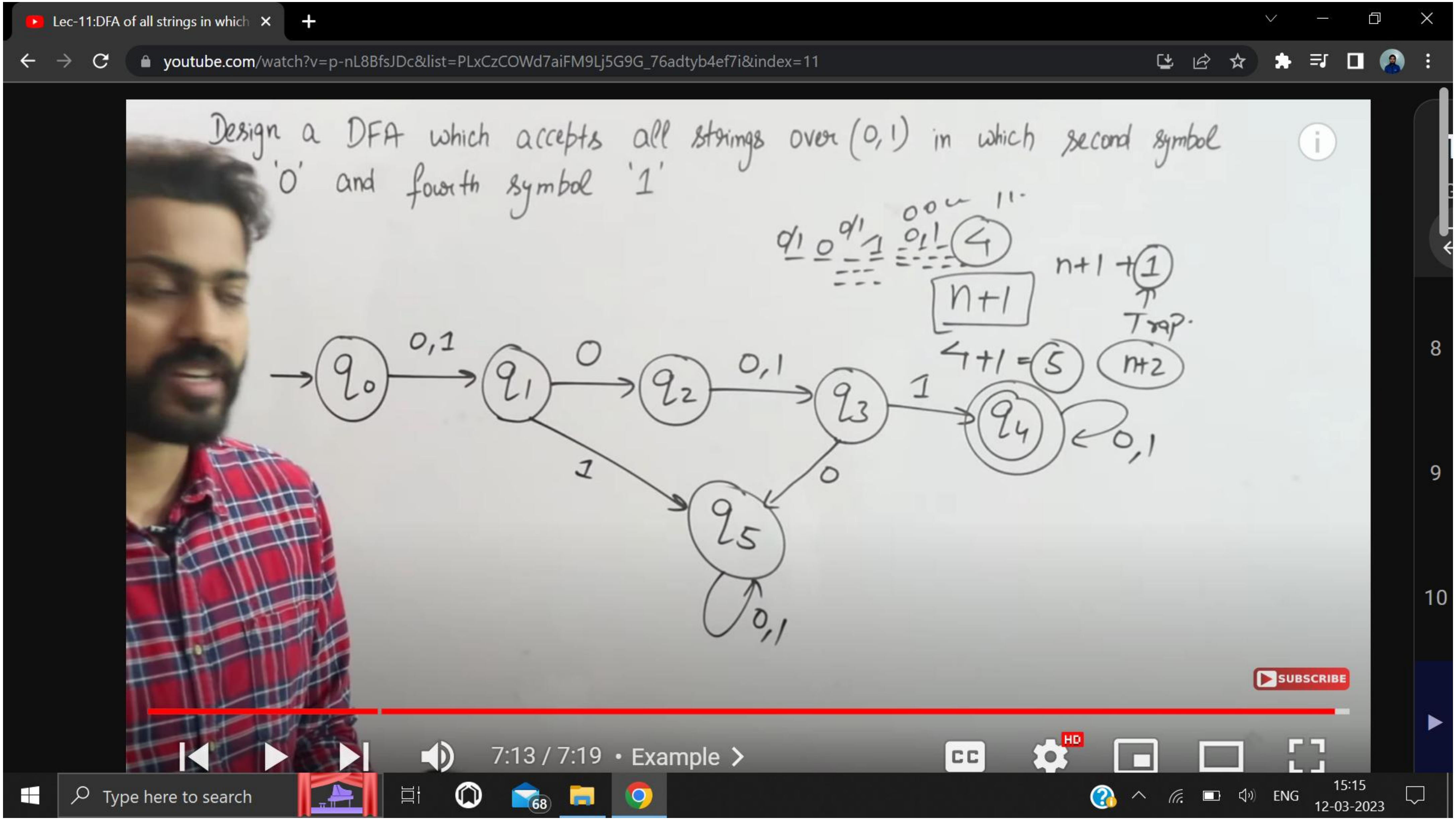

Lec-6: What is DFA in TOC with examples in hindi
DFA (Q, Σ, δ, q0, F)
Set of finite states
a, b
transition
Start
Set of final states
F ⊆ Q
Σ (a, b)
Strings starting with a {a, aa, aaa, ab, ba
a
b
a, b
a, b
q0
q1
q2
δ: Q x Σ → Q
q0 (a, b)
q1
q2
q0 a
q0 b
q1 a
q1 b
q2 a
q2 b
q0
q1
q2
12:55 / 13:13 • DFA
SUBSCRIBE
HD

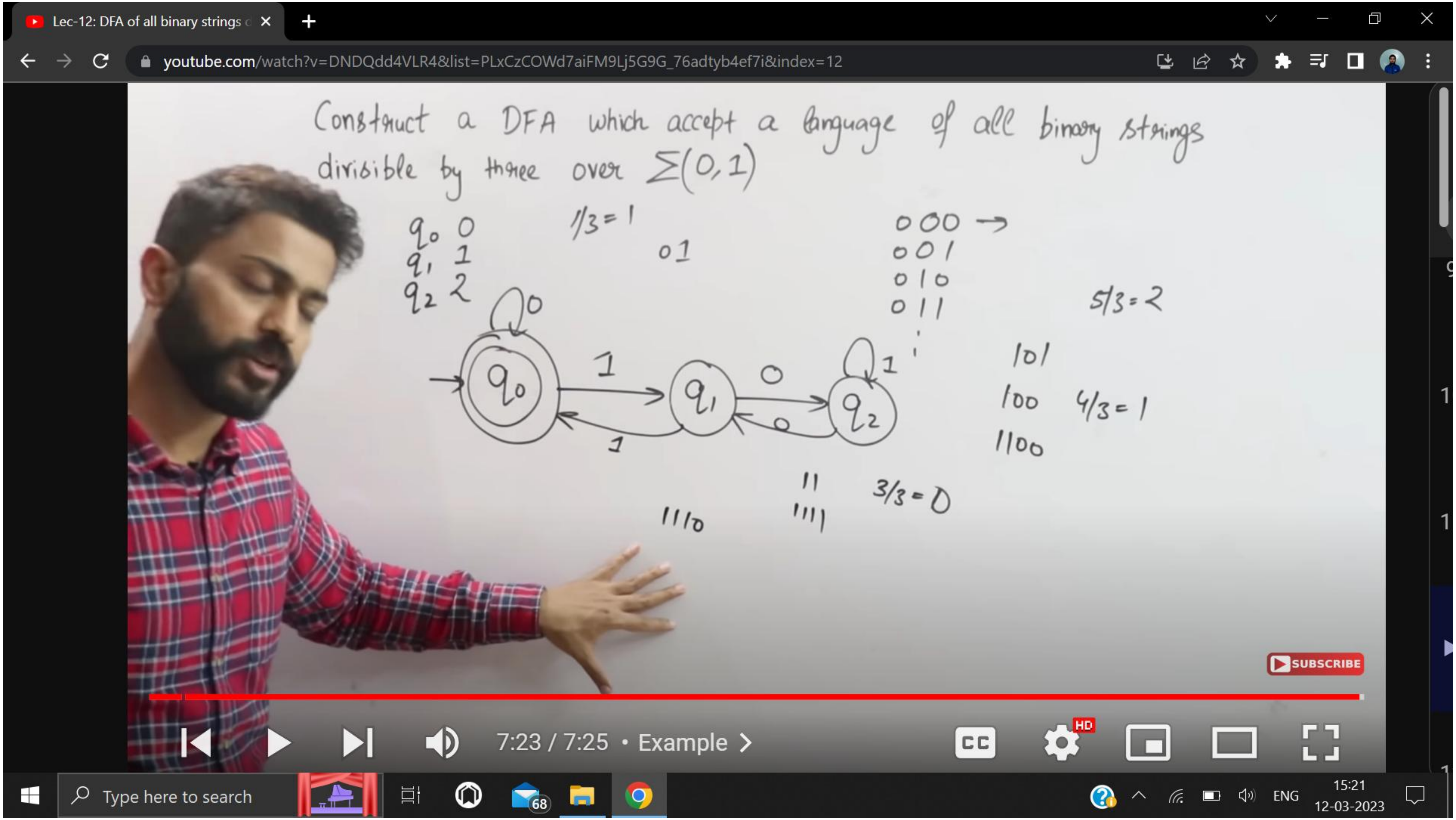

Lec-7: DFA Example 1 | How to Construct DFA in TOC
Construct a DFA which accept a language of all strings
→ Containing 'a' Σ a,b { a, aa, aaa, ba, ab, abab,
→ End with 'a'
baa
b
a,b
a
q0
q1
SUBSCRIBE
7:19 / 8:11 • Explanation ›
HD

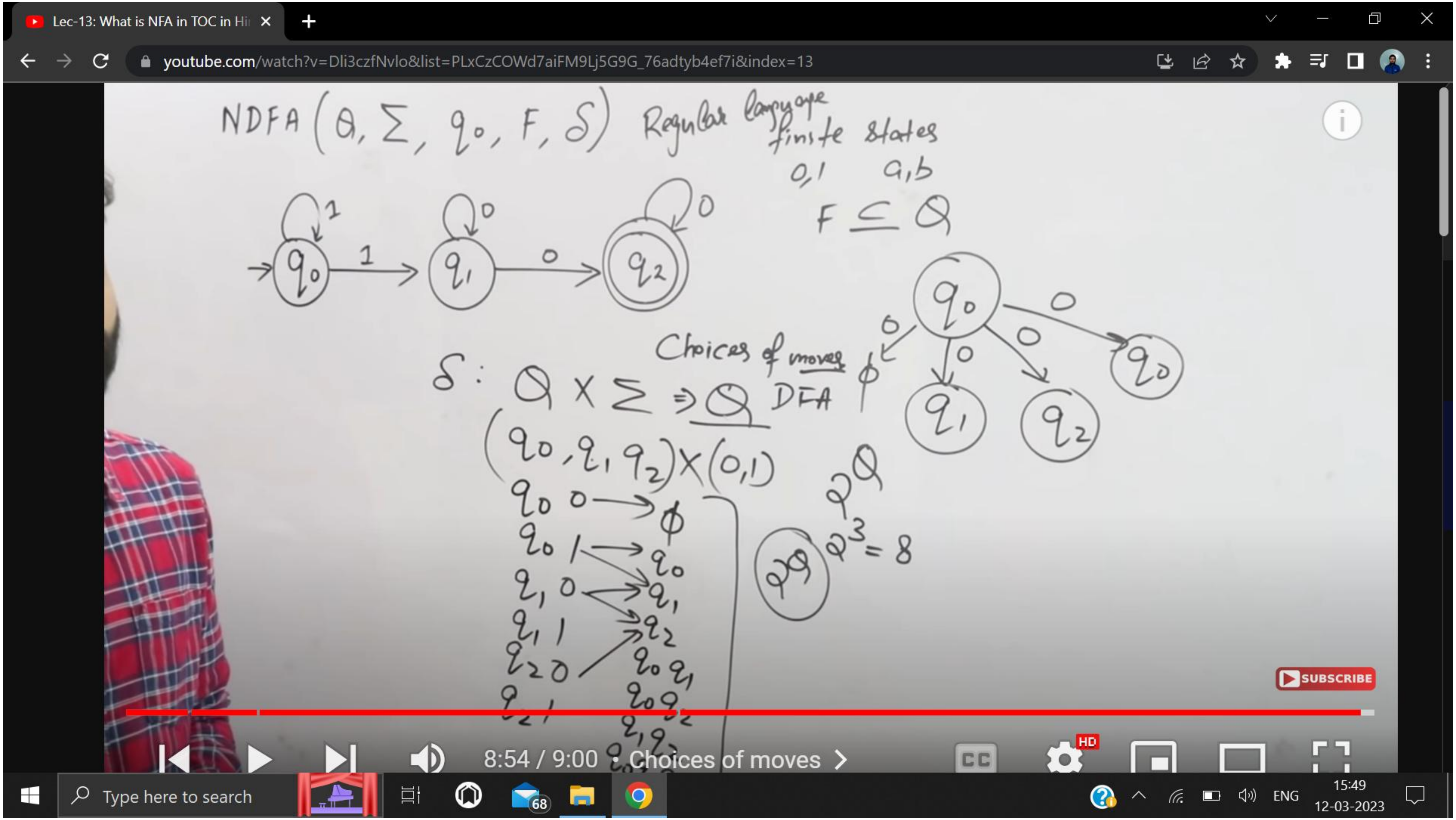

Lec-8: DFA Example 2 | DFA of lar
youtube.com/watch?v=cEX7V3c2CWc&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=8
Construct a DFA which accept a language of all strings
→ Containing 'a'
→ End with 'a'
Σ (a,b)
L = { a, aa, aaa
ba, baba ✓
aba, abaaba
ab, abab ×
ε,
b
a
a
b
q0
q1
ba
bbba
aaaaaaaa
aba
SUBSCRIBE
Type here to search
ENG
15:05
12-03-2023

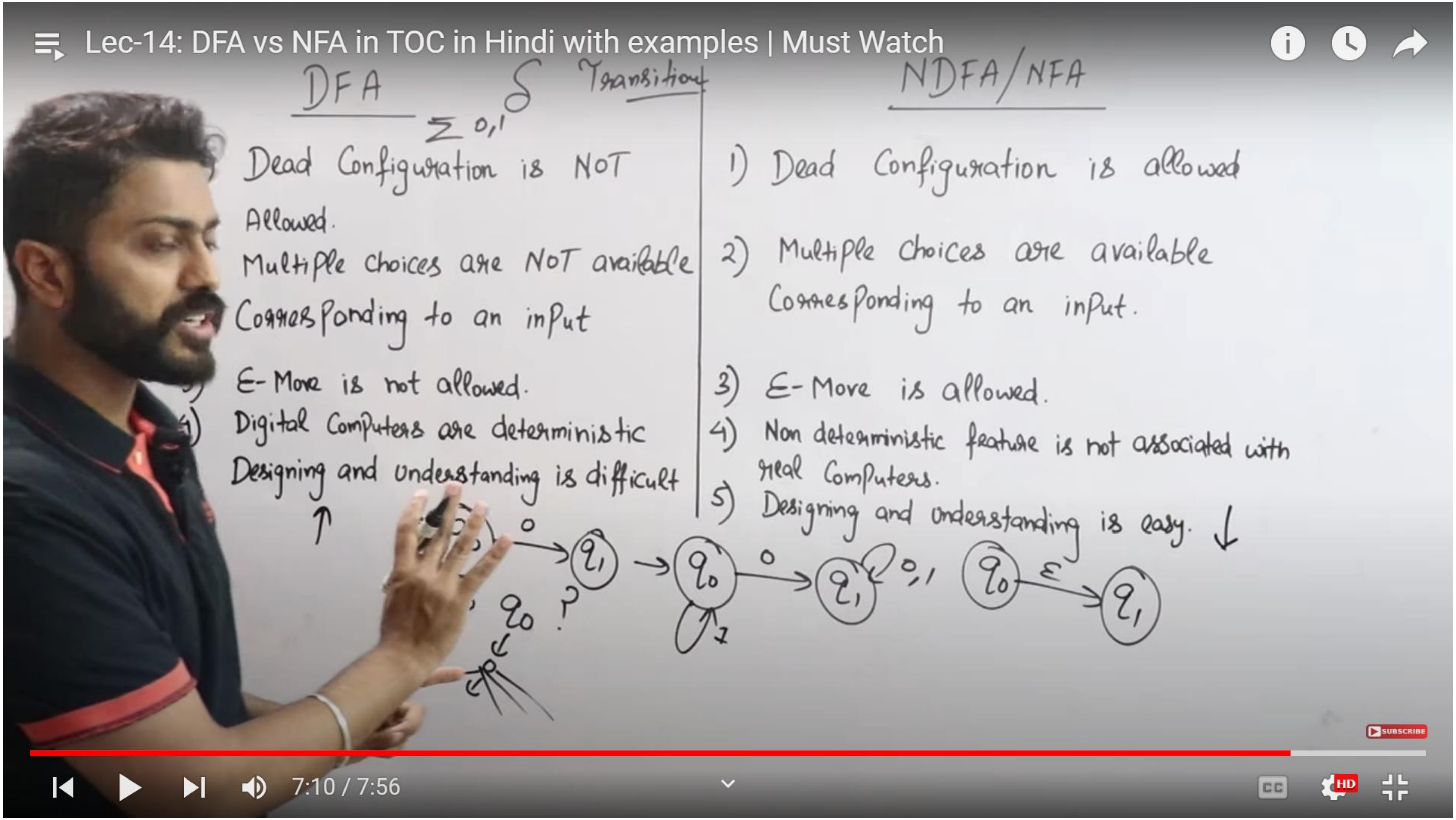

Lec-9: DFA Example 3 | DFA of lar
youtube.com/watch?v=v9lwDI0GtpE&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=9
Construct a DFA which accept a language of all strings starting with 'a' and ending with 'b'
Σ (a,b)
L = { ab, abab, ababab
aaaa bbbb,
ba
ε, a, b X
q0
q1
q2
q3
a
b
a, b
SUBSCRI
Type here to search
68
ENG
15:05
12-03-2023

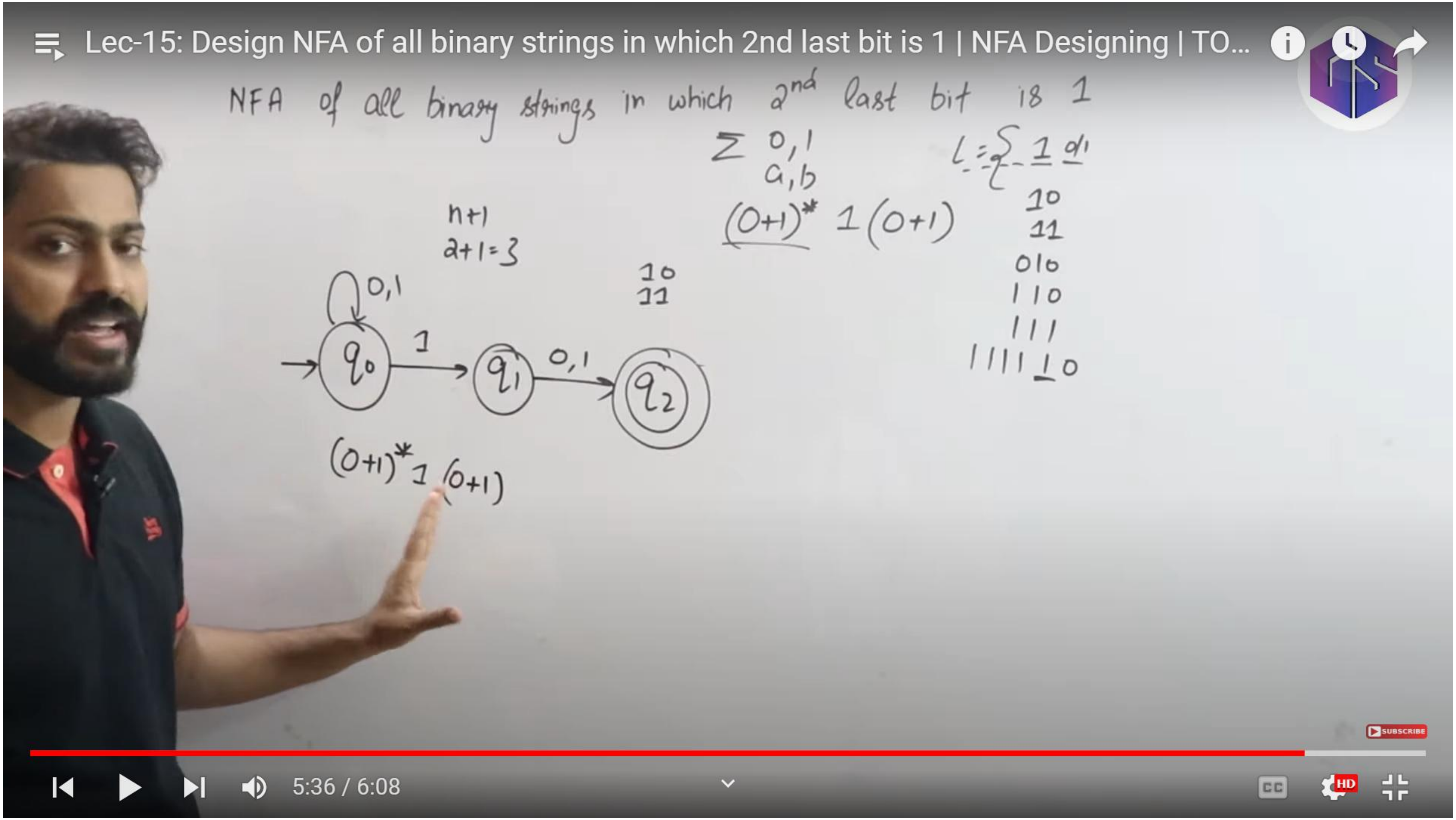

Lec-10:DFA of language with all strings Not starting with 'a' OR Not ending with 'b' | D...
Construct a DFA which accept a language of all strings
not starting with 'a' or not ending with 'b'
(A ∪ B)^c = A^c ∩ B^c
L = { λ, ε, a, b, ba
B
Starting with a and ending with b
(A ∪ B)^c = A^c ∩ B^c
final → non final
non final → final
q0
q1
q2
q3
a
b
a
b
a
a, b
ε, a, b, ab
ba
7:16 / 7:23 • Example ›
SUBSCRIBE
HD


Lec-11:DFA of all strings in which
youtube.com/watch?v=p-nL8BfsJDc&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=11
Design a DFA which accepts all strings over (0,1) in which second symbol '0' and fourth symbol '1'
n+1
n+1 + 1
Trap.
4+1 = 5
n+2
q0
0,1
q1
0
q2
0,1
q3
1
q4
0,1
1
0
q5
0,1
SUBSCRIBE
7:13 / 7:19 • Example
8
9
10
Type here to search
ENG
15:15
12-03-2023

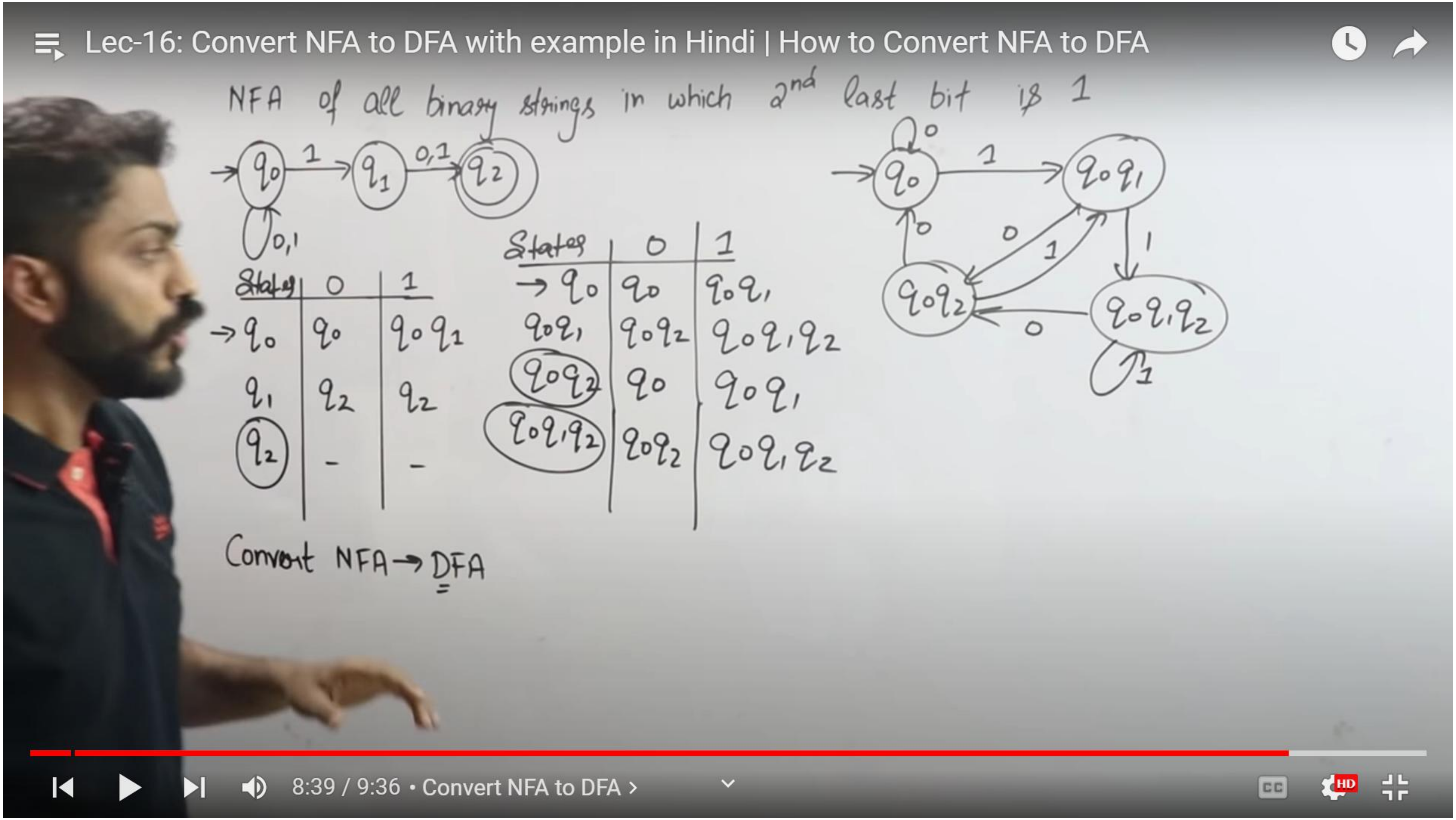

Lec-12: DFA of all binary strings
youtube.com/watch?v=DNDQdd4VLR4&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=12
Construct a DFA which accept a language of all binary strings divisible by three over Σ(0,1)
q0 0
q1 1
q2 2
1/3=1
01
000 →
001
010
011
5/3=2
101
100
4/3=1
1100
11
3/3=0
1110
1111
7:23 / 7:25 • Example
SUBSCRIBE
Type here to search
ENG
15:21
12-03-2023

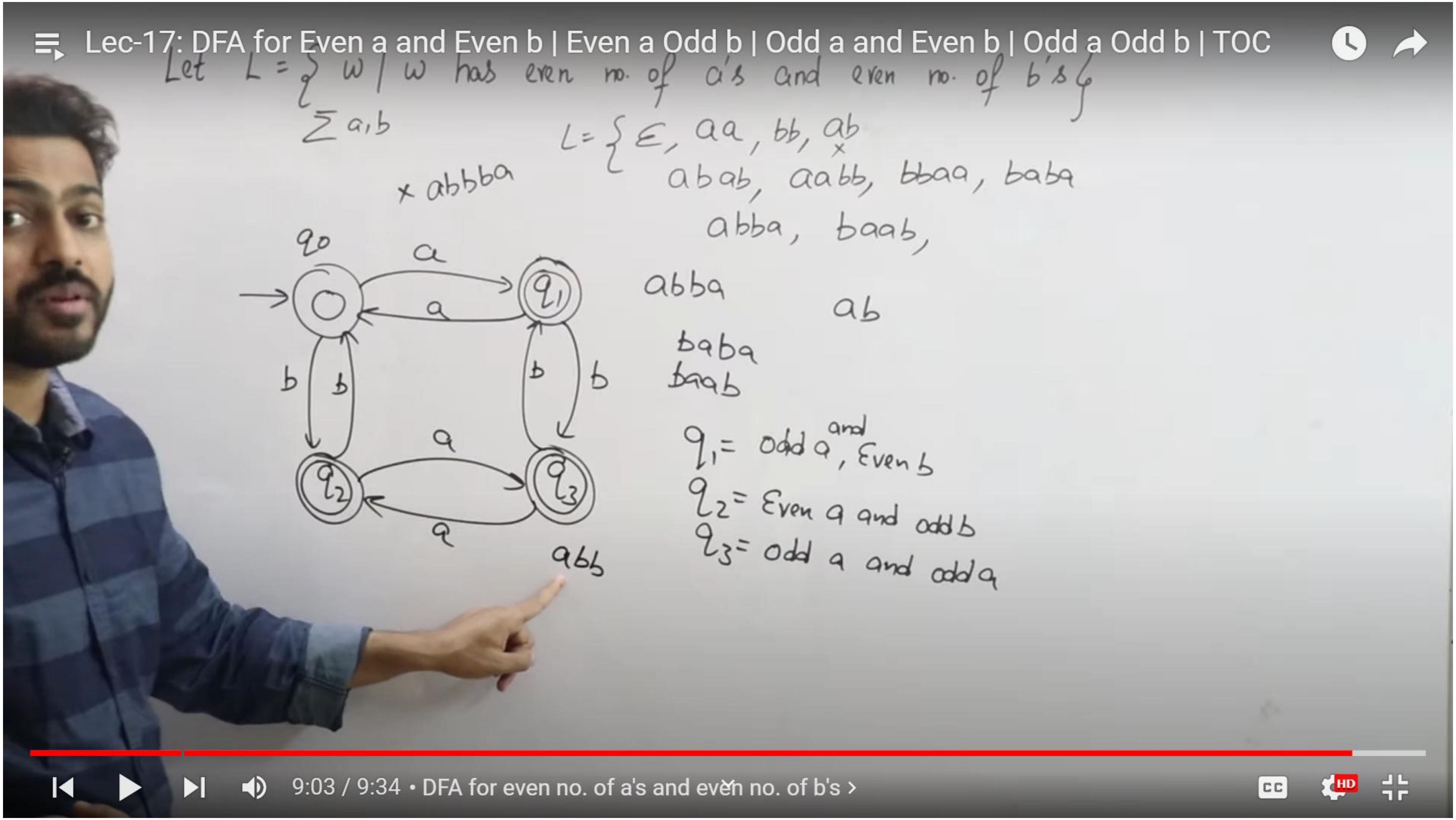

Lec-13: What is NFA in TOC in Hi
youtube.com/watch?v=Dli3czfNvlo&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=13
NDFA (Q, Σ, q0, F, δ) Regular language
finite states
0,1 a,b
F ⊆ Q
Choices of moves
δ : Q X Σ ⇒ Q DFA
(q0, q1, q2) X (0,1)
2^Q
2^3 = 8
SUBSCRIBE
8:54 / 9:00 • Choices of moves
Type here to search
ENG
15:49
12-03-2023

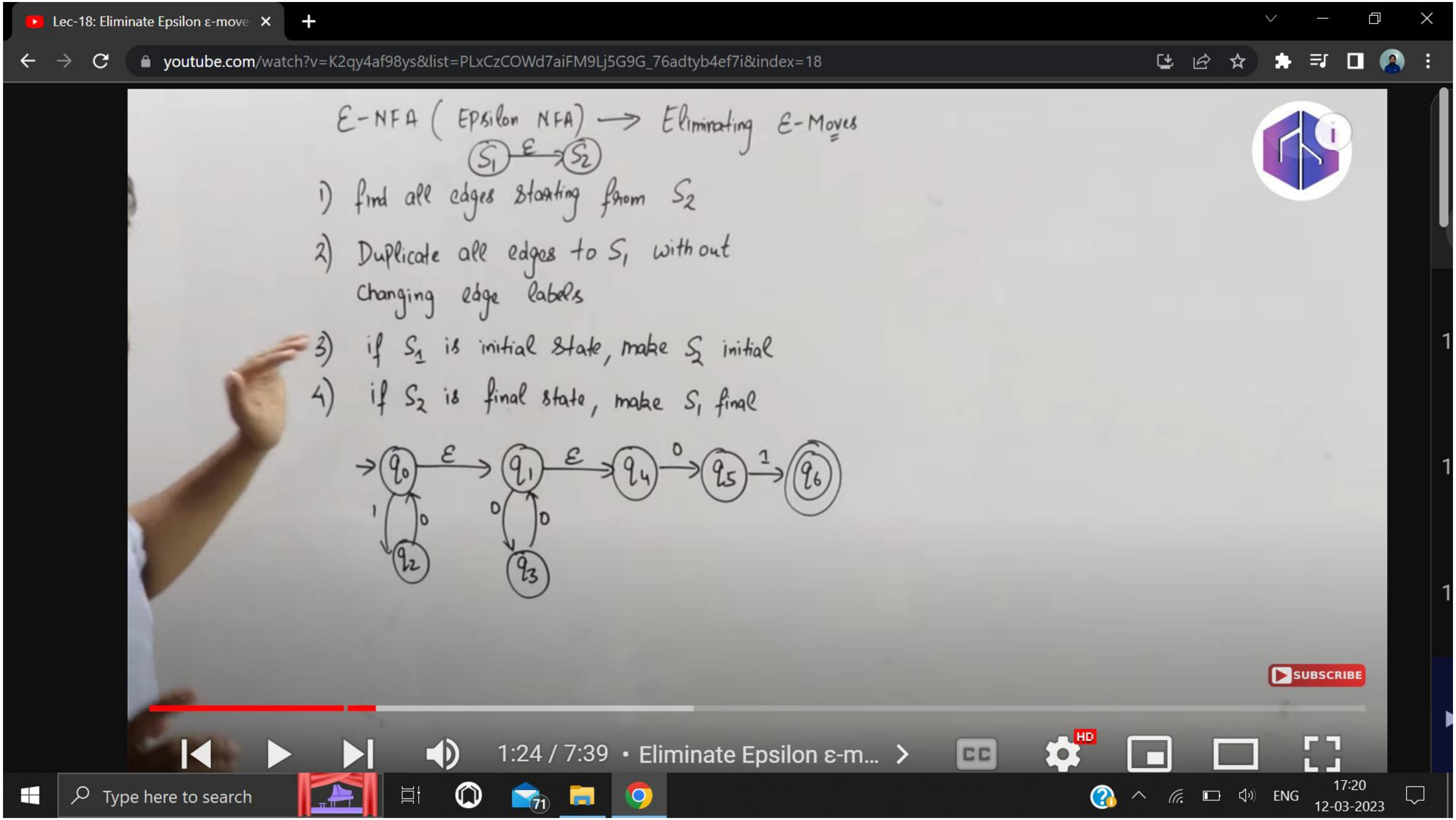

Lec-14: DFA vs NFA in TOC in Hindi with examples | Must Watch
DFA
δ Transition
Σ 0,1
Dead Configuration is NOT Allowed.
Multiple choices are NOT available Corresponding to an input
ε-Move is not allowed.
Digital Computers are deterministic
Designing and understanding is difficult
NDFA/NFA
1) Dead Configuration is allowed
2) Multiple choices are available Corresponding to an input.
3) ε-Move is allowed.
4) Non deterministic feature is not associated with real Computers.
5) Designing and understanding is easy.
7:10 / 7:56
SUBSCRIBE
HD

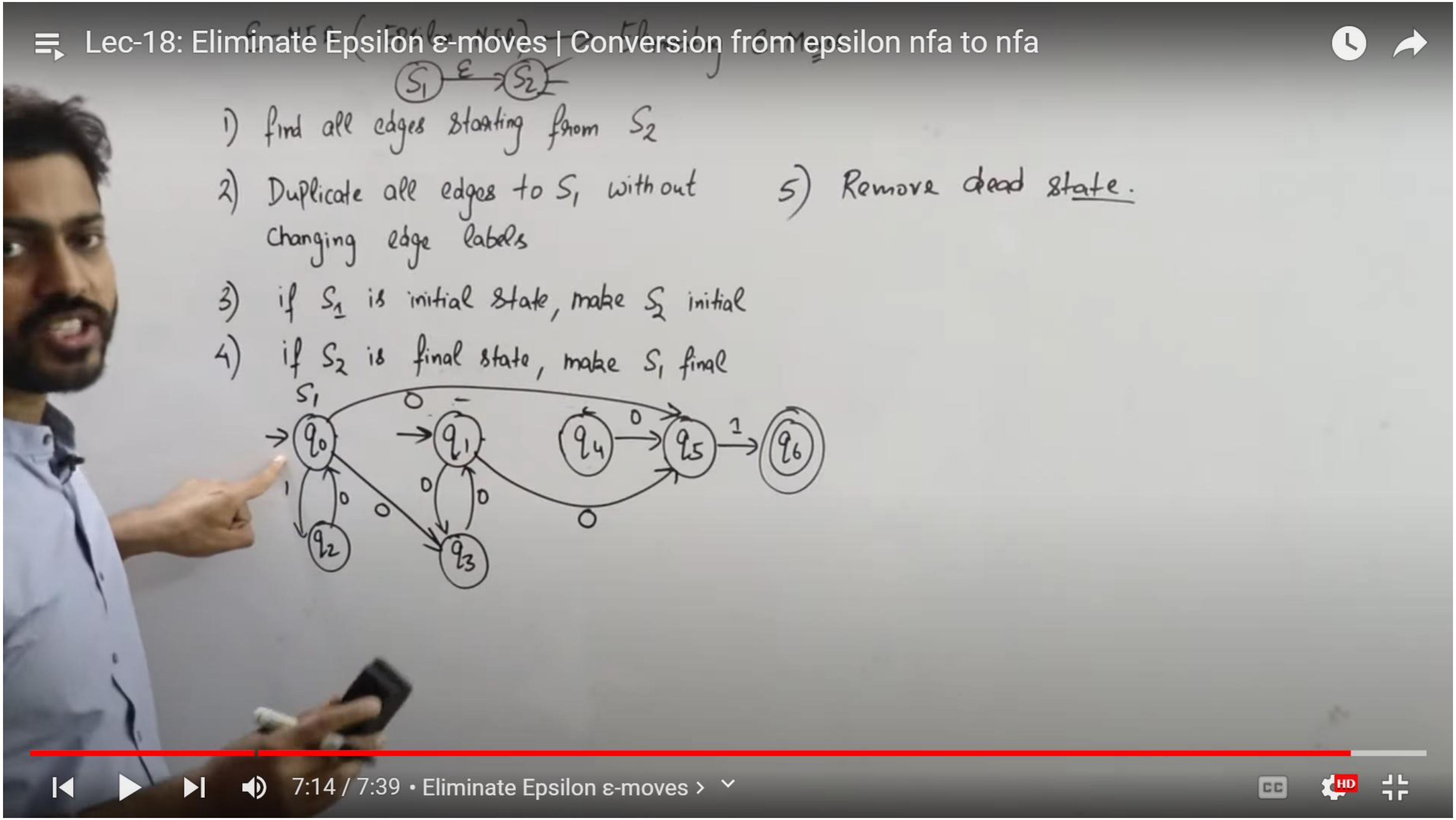


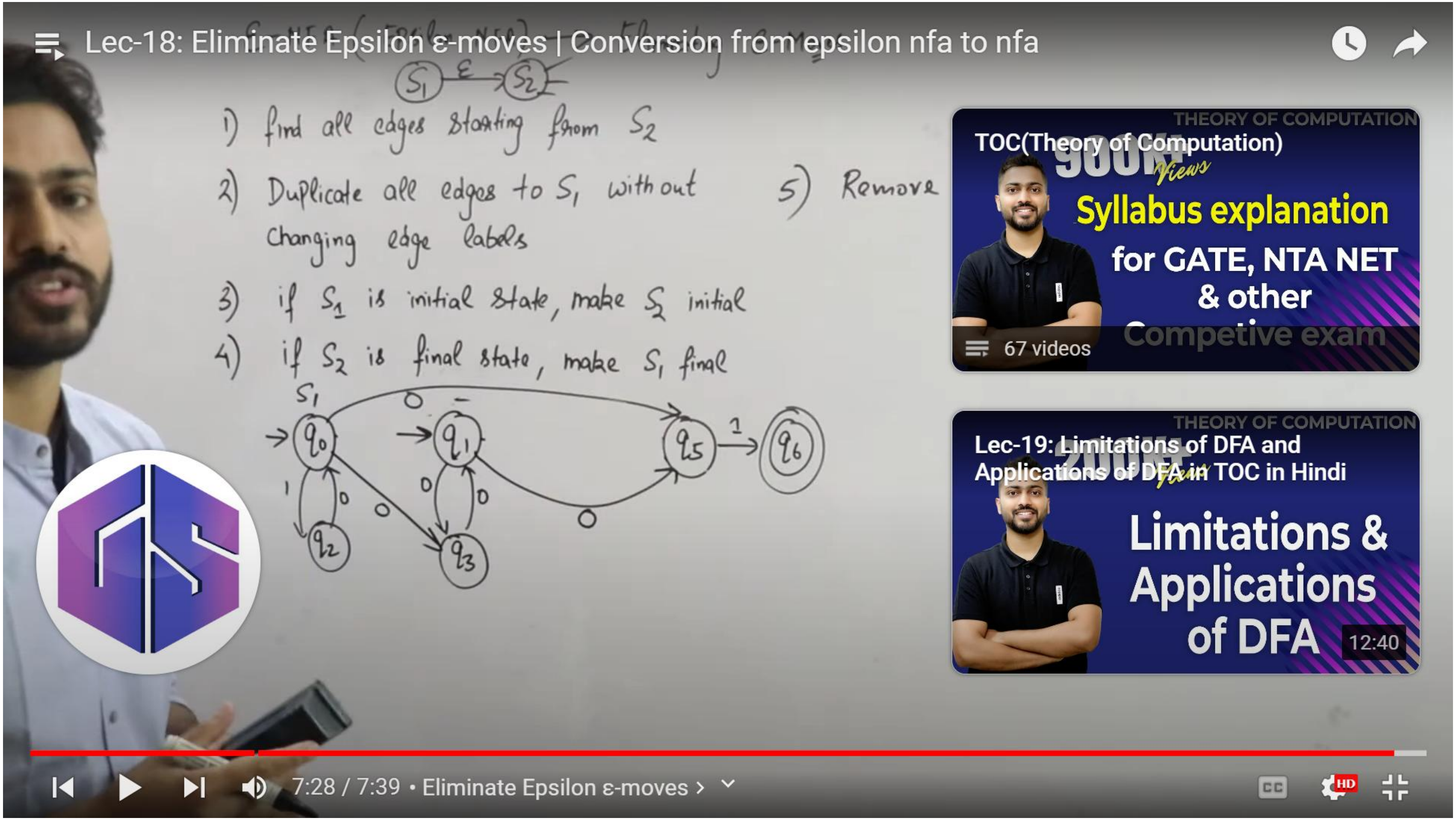
Lec-15: Design NFA of all binary strings in which 2nd last bit is 1 | NFA Designing | TO...
NFA of all binary strings in which 2nd last bit is 1
Σ 0,1
a,b
(0+1)* 1 (0+1)
L = { 1 0/1
10
11
010
110
111
1111 10
n+1
2+1=3
0,1
1
0,1
q0
q1
q2
10
11
(0+1)* 1 (0+1)
SUBSCRIBE
5:36 / 6:08

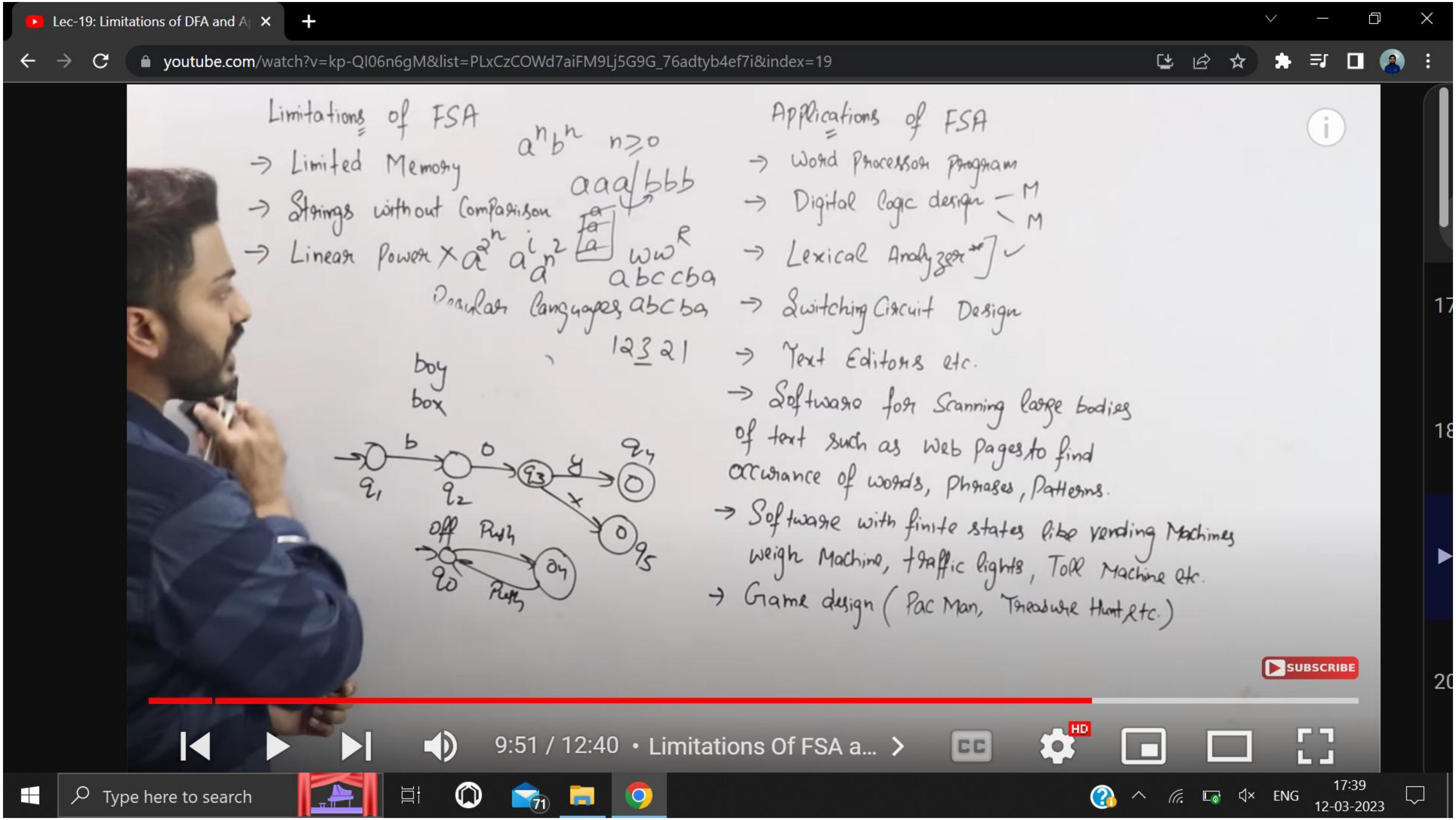

Lec-16: Convert NFA to DFA with example in Hindi | How to Convert NFA to DFA
NFA of all binary strings in which 2nd last bit is 1
States | 0 | 1
→q0 | q0 | q0q1
q1 | q2 | q2
q2 | - | -
States | 0 | 1
→q0 | q0 | q0q1
q0q1 | q0q2 | q0q1q2
q0q2 | q0 | q0q1
q0q1q2 | q0q2 | q0q1q2
Convert NFA → DFA
8:39 / 9:36 • Convert NFA to DFA ›

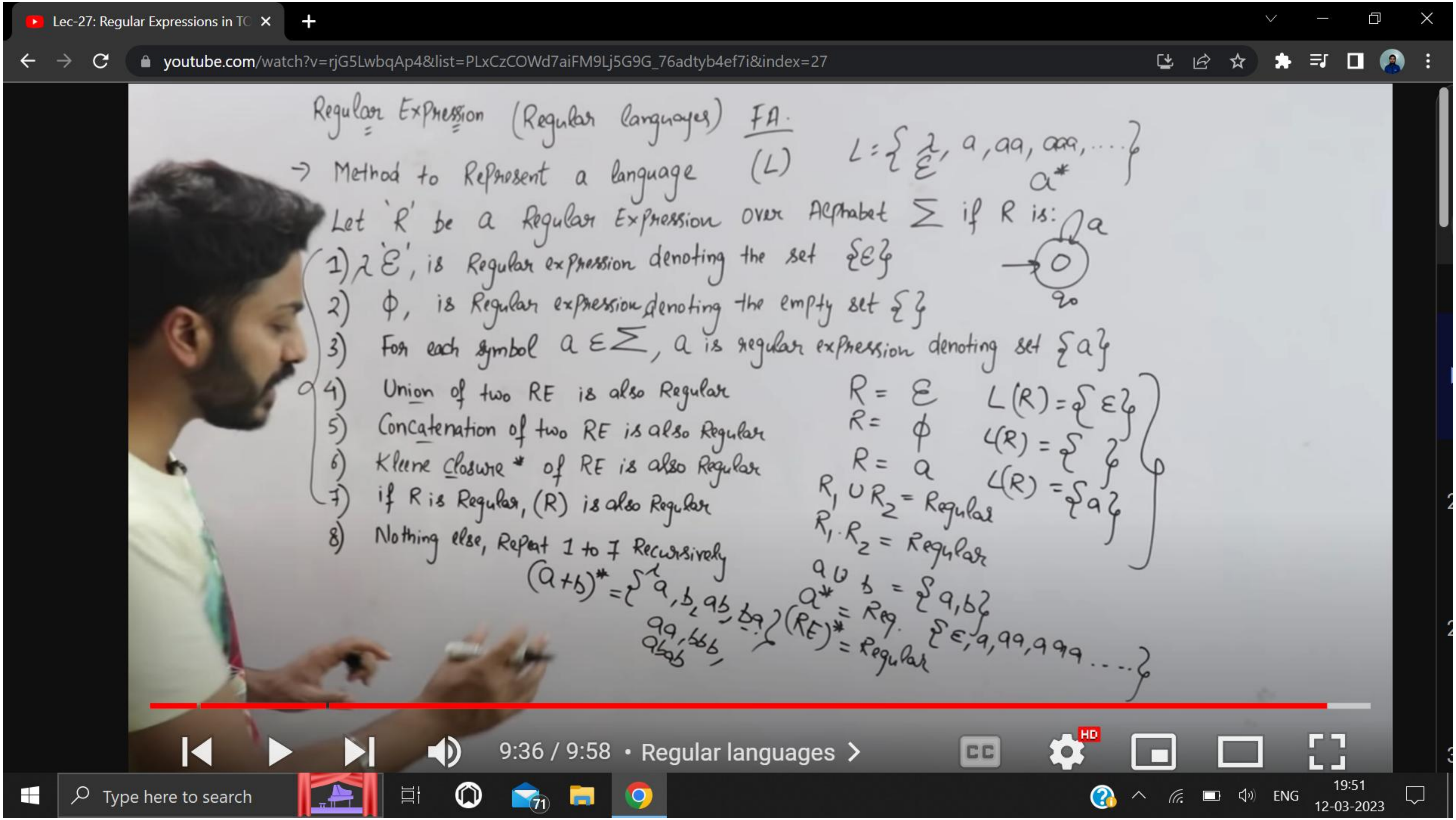

Lec-17: DFA for Even a and Even b | Even a Odd b | Odd a and Even b | Odd a Odd b | TOC
Let L = { w | w has even no. of a's and even no. of b's }
Σ a,b
L = { ε, aa, bb, ab
abab, aabb, bbaa, baba
abba, baab,
× abbba
q0
q1
q2
q3
a
a
b
b
b
b
a
a
abba
ab
baba
baab
q1 = odd a and Even b
q2 = Even a and odd b
q3 = odd a and odd a
abb
9:03 / 9:34 • DFA for even no. of a's and even no. of b's ›
HD


Lec-18: Eliminate Epsilon ε-move
youtube.com/watch?v=K2qy4af98ys&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=18
ε-NFA (Epsilon NFA) → Eliminating ε-Moves
S1 —ε→ S2
1) find all edges starting from S2
2) Duplicate all edges to S1 without changing edge labels
3) if S1 is initial state, make S2 initial
4) if S2 is final state, make S1 final
q0 —ε→ q1 —ε→ q4 —0→ q5 —1→ q6
1
0
0
0
q2
q3
SUBSCRIBE
1:24 / 7:39 • Eliminate Epsilon ε-m...
CC
HD
Type here to search
71
ENG
17:20
12-03-2023


Lec-18: Eliminate Epsilon ε-moves | Conversion from epsilon nfa to nfa
$S_1 \xrightarrow{\varepsilon} S_2$
1) find all edges starting from S_2
2) Duplicate all edges to S_1 without changing edge labels
3) if S_1 is initial state, make S_2 initial
4) if S_2 is final state, make S_1 final
5) Remove dead state.
S_1
q_0 q_1 q_2 q_3 q_4 q_5 q_6
0 1
7:14 / 7:39 • Eliminate Epsilon ε-moves ›
HD


Lec-18: Eliminate Epsilon ε-moves | Conversion from epsilon nfa to nfa
S1 ε S2
1) find all edges starting from S2
2) Duplicate all edges to S1 without changing edge labels
3) if S1 is initial state, make S2 initial
4) if S2 is final state, make S1 final
5) Remove
S1
q0 q1 q2 q3 q5 q6
0 1 0 0 0 0 0 1
THEORY OF COMPUTATION
TOC(Theory of Computation)
Syllabus explanation
for GATE, NTA NET
& other
Competive exam
67 videos
THEORY OF COMPUTATION
Lec-19: Limitations of DFA and Applications of DFA in TOC in Hindi
Limitations &
Applications
of DFA
12:40
7:28 / 7:39 • Eliminate Epsilon ε-moves ›
HD


Lec-19: Limitations of DFA and A
youtube.com/watch?v=kp-QI06n6gM&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=19
Limitations of FSA
→ Limited Memory
→ Strings without Comparison
→ Linear Power
$a^n b^n$ $n \geq 0$
aaa bbb
ww^R
abccba
abcba
12321
Popular Languages
boy
box
Applications of FSA
→ Word Processor Program
→ Digital logic design
→ Lexical Analyzer
→ Switching Circuit Design
→ Text Editors etc.
→ Software for Scanning large bodies of text such as web pages to find occurance of words, Phrases, Patterns.
→ Software with finite states like vending Machines weigh Machine, traffic lights, Toll Machine etc.
→ Game design (Pac Man, Treasure Hunt etc.)
9:51 / 12:40 • Limitations Of FSA a...
SUBSCRIBE
Type here to search
ENG
17:39
12-03-2023


Lec-27: Regular Expressions in TC
youtube.com/watch?v=rjG5LwbqAp4&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=27
Regular Expression (Regular languages) FA
→ Method to Represent a language (L)
L = { λ/ε, a, aa, aaa, ... } a*
Let 'R' be a Regular Expression over Alphabet Σ if R is:
1) λ 'ε', is Regular expression denoting the set {ε}
2) φ, is Regular expression denoting the empty set { }
3) For each symbol a ∈ Σ, a is regular expression denoting set {a}
4) Union of two RE is also Regular
5) Concatenation of two RE is also Regular
6) Kleene Closure * of RE is also Regular
7) if R is Regular, (R) is also Regular
8) Nothing else, Repeat 1 to 7 Recursively
a
q0
R = ε L(R) = {ε}
R = φ L(R) = { }
R = a L(R) = {a}
R1 ∪ R2 = Regular
R1 · R2 = Regular
a ∪ b = {a, b}
a* = Reg. {ε, a, aa, aaa ....}
(RE)* = Regular
(a+b)* = {λ, a, b, ab, ba, aa, bb, abab, ...}
9:36 / 9:58 • Regular languages
Type here to search
ENG
19:51
12-03-2023

Lec-28: Regular Expressions for F... ×

youtube.com/watch?v=QddGS_Revb4&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=28

Regular Expression Example 1

$\Sigma (a, b)$

Reg ✓
FA ✓
RE ✓

* Language → Finite, Infinite

1) No string $\{\ \}$ ϕ
2) Length 0 $\{\varepsilon\}$ $\in, \lambda$
3) Length 1 $\{a, b\}$ $(a+b)$
4) Length 2 $\{aa, ab, ba, bb\}$ $(aa+ab+ba+bb)$
5) Length 3 $\boxed{(a+b)(a+b)(a+b)}$ aaa, aab, aba, abb, baa . . . .
6) At most 1 0, 1 $\{\varepsilon, a, b\}$ $(\varepsilon+a+b)$
7) At most 2 $(\varepsilon+a+b)(\varepsilon+a+b)$

⋮

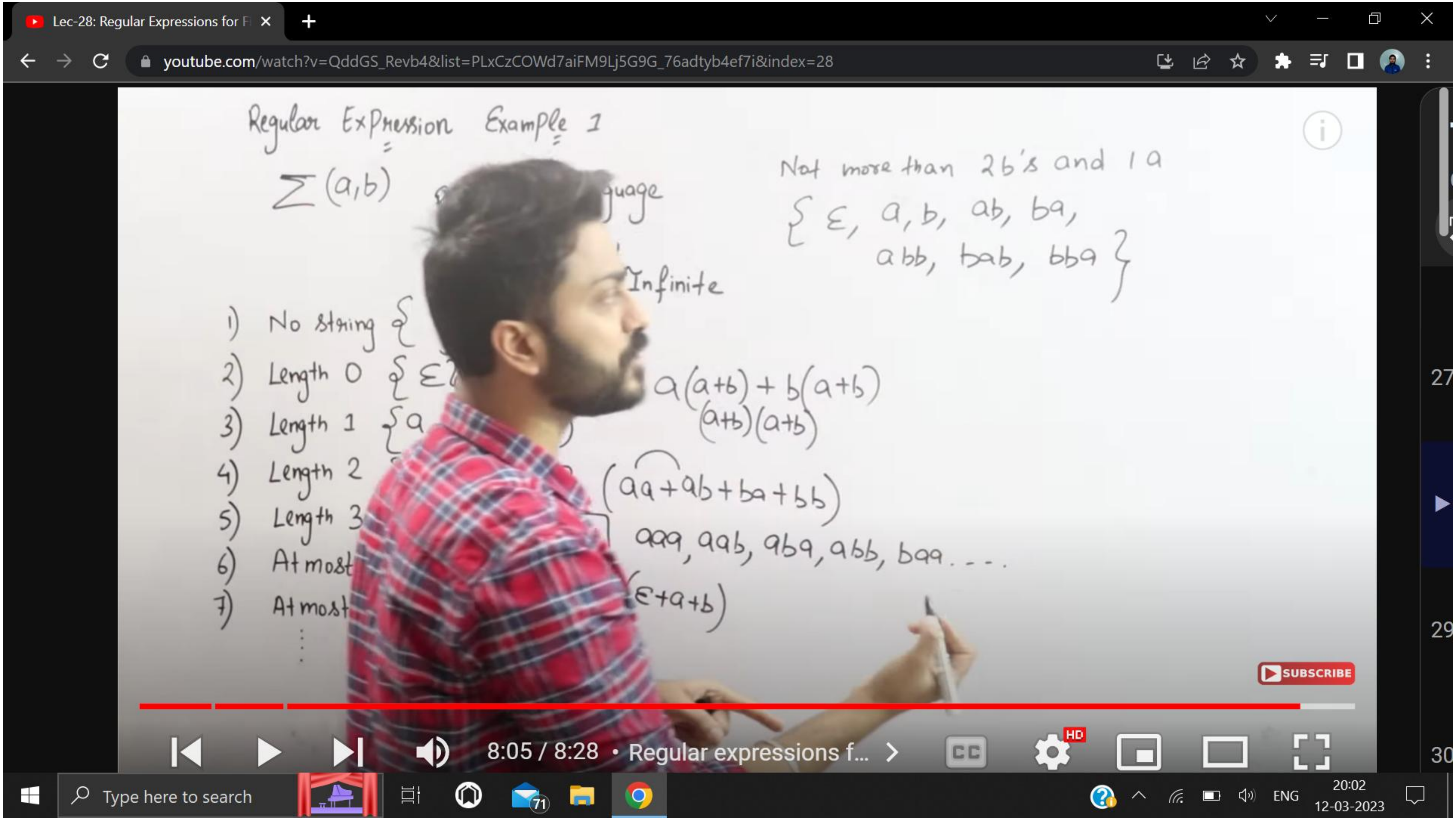
$a(a+b) + b(a+b)$
$(a+b)(a+b)$

SUBSCRIBE

6:51 / 8:28 • Regular expressions f...

Type here to search

ENG 20:01 12-03-2023


Lec-28: Regular Expressions for F
youtube.com/watch?v=QddGS_Revb4&list=PLxCzCOWd7aiFM9Lj5G9G_76adtyb4ef7i&index=28
Regular Expression Example 1
Σ(a,b)
uage
Infinite
1) No string {
2) Length 0 {ε
3) Length 1 {a
4) Length 2
5) Length 3
6) At most
7) At most
Not more than 2b's and 1a
{ε, a, b, ab, ba, abb, bab, bba}
a(a+b) + b(a+b)
(a+b)(a+b)
(aa+ab+ba+bb)
aaa, aab, aba, abb, baa . . .
(ε+a+b)
SUBSCRIBE
8:05 / 8:28 • Regular expressions f...
Type here to search
ENG
20:02
12-03-2023

Lec-30: Important Question on Regular Expressions for all Competitive Exams | TOC

Which two of the following four regular expressions are equivalent?

(i) $(00)^*(\varepsilon+0)$

(ii) $(00)^*$

(iii) 0^*

(iv) $0(00)^*$

(a) (i) and (ii)

(b) (ii) and (iii)

(c) (i) and (iii)

(d) (iii) and (iv)

Any No. of zero $0^* = \varepsilon, 0, 00, 000, 0000, \ldots$

Even no of zero $(00)^* = \varepsilon, 00, 0000, 000000, \ldots$

$(00)^*0 = 0(00)^*$

Odd no of zero $0, 000, 00000$

$(00)^*(\varepsilon+0)$

$\underbrace{(00)^*\varepsilon}_{\text{Even no.}} + (00)^*0$

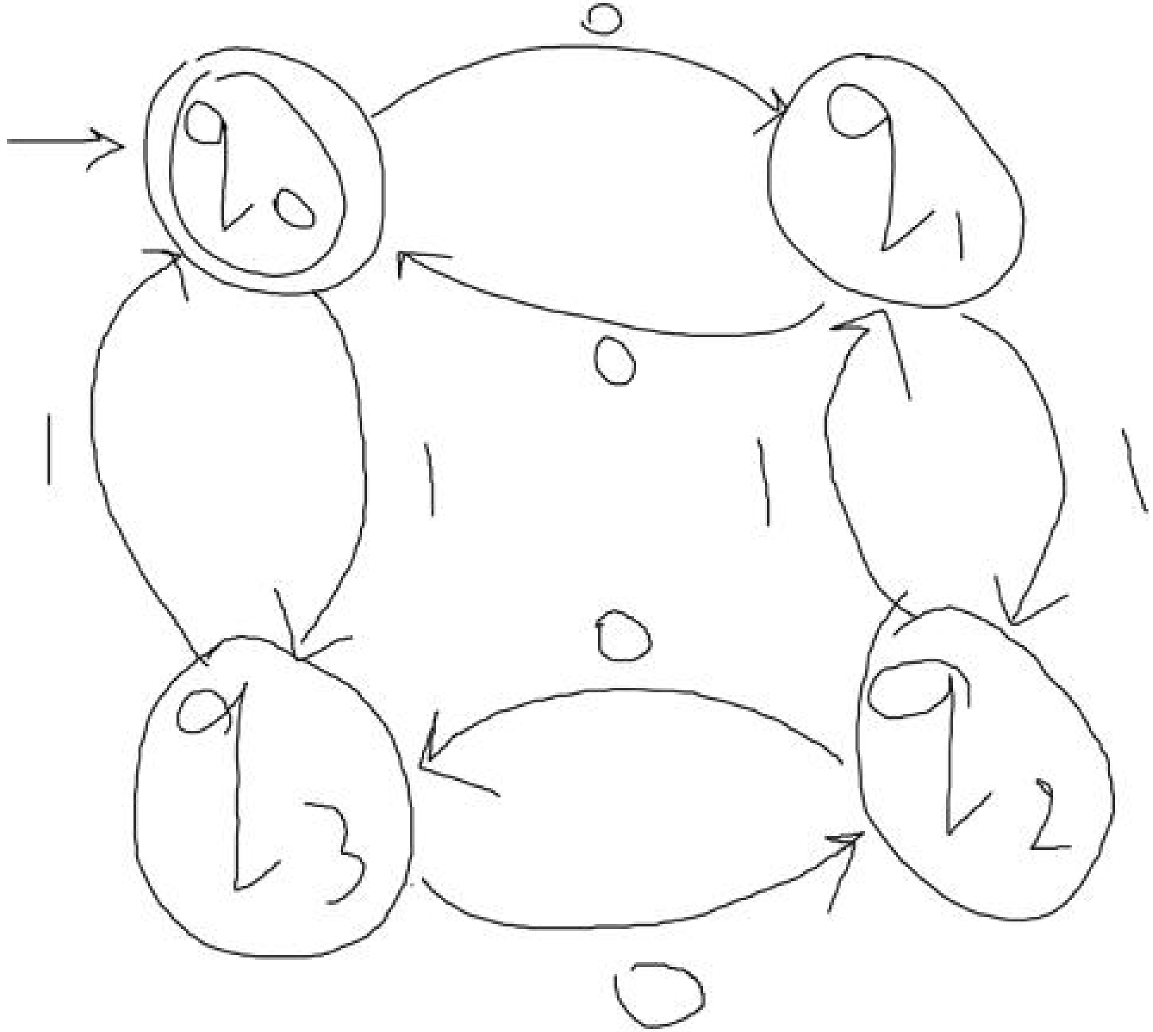
Even no. + Odd

0^*

5:26 / 5:50 • 00)*0=0(00 ›

HD

1. Design FA which accepts even number of 0's and even no of 1's.



2. Prove that for every integer n≥0 the number 42n+1+ 3 n+2 is multiple of 13

Prove $4^{2n+1} + 3^{n+2}$ is a multiple of 13 for all $n \geq 0$. Base Case ($n = 0$): $4^1 + 3^2 = 13$, which is divisible by 13.

Induction Hypothesis: Assume true for k, i.e. $4^{2k+1} + 3^{k+2}$ is a multiple of 13.

Prove for $k+1$, i.e. show $4^{2k+3} + 3^{k+3}$ is a multiple of 13

$4^{2k+3} + 3^{k+3} = 16 \cdot 4^{2k+1} + 3 \cdot 3^{k+2} = 13 \cdot 4^{2k+1} + 3\left[4^{2k+1} + 3^{k+2}\right]$

The first term in that final sum $(13 \cdot 4^{2k+1})$ is clearly divisible by 13. The second term in the sum $(3\left[4^{2k+1} + 3^{k+2}\right])$ is divisible by 13 by I.H.

3. Prove that 6n ≡ 0 (mod 9) for all integers n ≥ 2.

Prove that $6n \equiv 0 \pmod{9}$ for all integers $n \geq 2$

To prove that $6n \equiv 0 \pmod{9}$ for all integers $n \geq 2$, we need to show that 9 divides 6n, or equivalently, that n is divisible by 3.

We can prove this by mathematical induction.

Base case: When $n = 2$, we have $6n = 6(2) = 12$, which is divisible by 3. Thus, the base case is true.

Inductive step: Assume that $6k \equiv 0 \pmod{9}$ for some integer $k \geq 2$. We need to show that $6(k+1) \equiv 0 \pmod{9}$.

We have:

$6(k+1) = 6k + 6$

By the induction hypothesis, we know that 6k is divisible by 9, so we can write:

$6k = 9m$

where m is some integer. Substituting this into the above equation, we get:

$6(k+1) = 9m + 6$

Factoring out a 3, we get:

$6(k+1) = 3(3m + 2)$

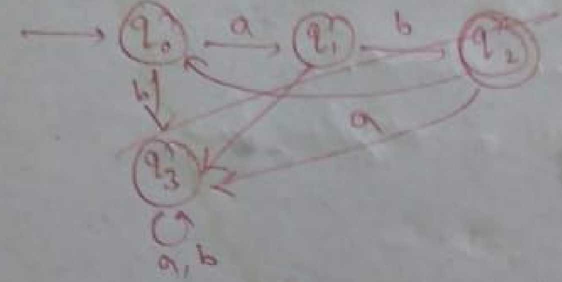
Since $3m + 2$ is an integer, we see that $6(k+1)$ is divisible by 3. Therefore, $6(k+1)$ is divisible by 9, which means that $6(k+1) \equiv 0 \pmod 9$.

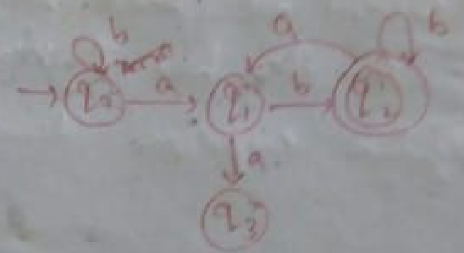
By mathematical induction, we have shown that $6n \equiv 0 \pmod 9$ for all integers $n \geq 2$.

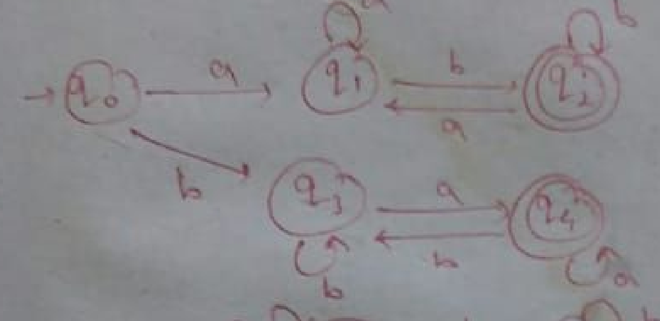
A lang. L is accepted by some ε-NFA if and only if L is accepted by some DFA.

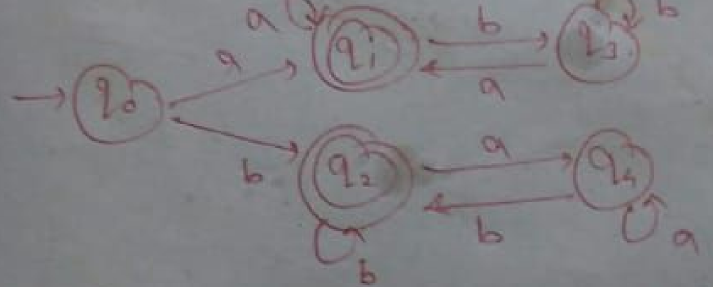
DFA ≡ NFA ≡ ε-NFA
(All accept regular languages)

(1) $L = \{ab, ababab, ababab, \dots\}$ · $\{ab, bab, bbabab, \dots\}$









Adv. IOT	Dis. IOT
(i) Efficiency	(i) Security risk
(ii) Cost saving	(ii) Interoperability issue
(iii) Improved customer exp.	(iii) Data privacy concerns
(iv) Enhanced safety & security	(iv) Reliability issues
(v) real-time data insights	(v) Complexity

(2) Arden's Theorem: Use to find regular expression of Finite Automaton

Statement: Let P & Q be 2 regular ".

If P does not contain null string, then $R = Q + RP$ has a unique sol^n that is $R = QP^*$

Proof: $R = Q + RP$

$= Q + (Q+RP)P$ [After putting the value $R = Q + RP$]

$= Q + QP + RP^2$

$= Q + QP + (Q+RP)P^2$

$= Q + QP + QP^2 + RP^3$

When we put value of R recursively again & again, we get

$R = Q + QP + QP^2 + QP^3 + \dots = Q(\varepsilon + P + P^2 + P^3 + \dots)$

$R = QP^*$ [As P^* represents $(\varepsilon + P + P^2 + P^3 + \dots)$]

(3) Kleen's Theorem !

1. Any regular lang. is accepted by a finite automaton.
2. If L is accepted by a FA, then L is regular.

(4) "Regular Lang" is defined as smallest class of lang. which contains all finite languages & closed with respect to union, concatenation & Kleene closure.

Every regular expression denotes a regular lang.