

RULE #1

If R.E is x then FA will be

if RE is null



RULE 2 (UNION OF TWO FAs):

We will do it in three steps

1: name of state

2: make table

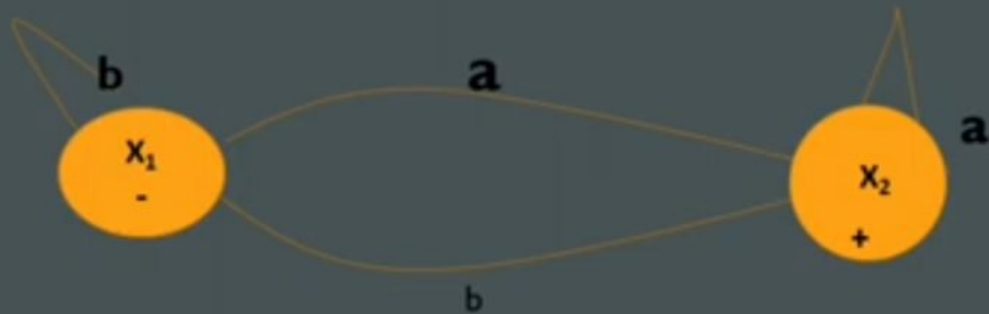
3: draw union diagram of both DFA

Q# how we find union of two DFA

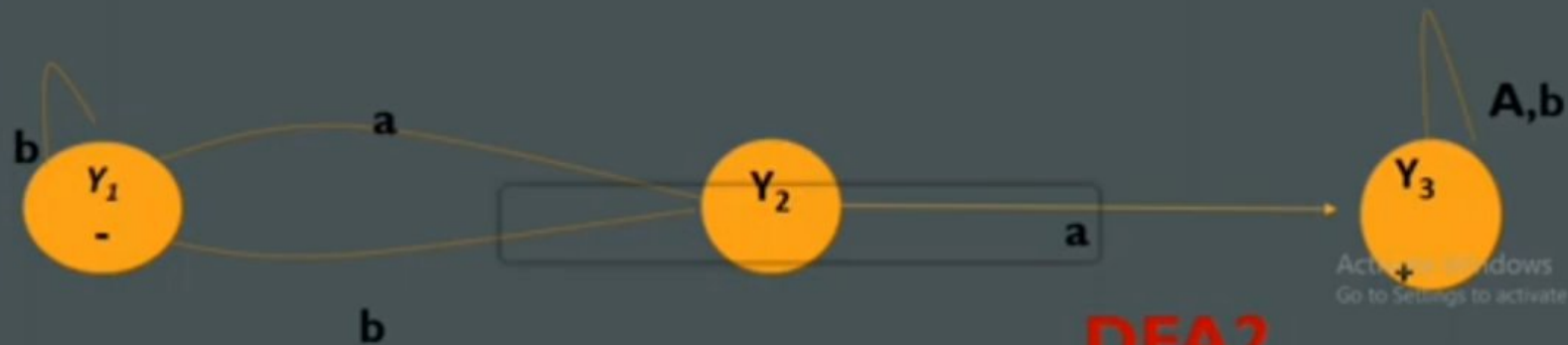
Answer:

Q# how we find union of two DFA

Answer:



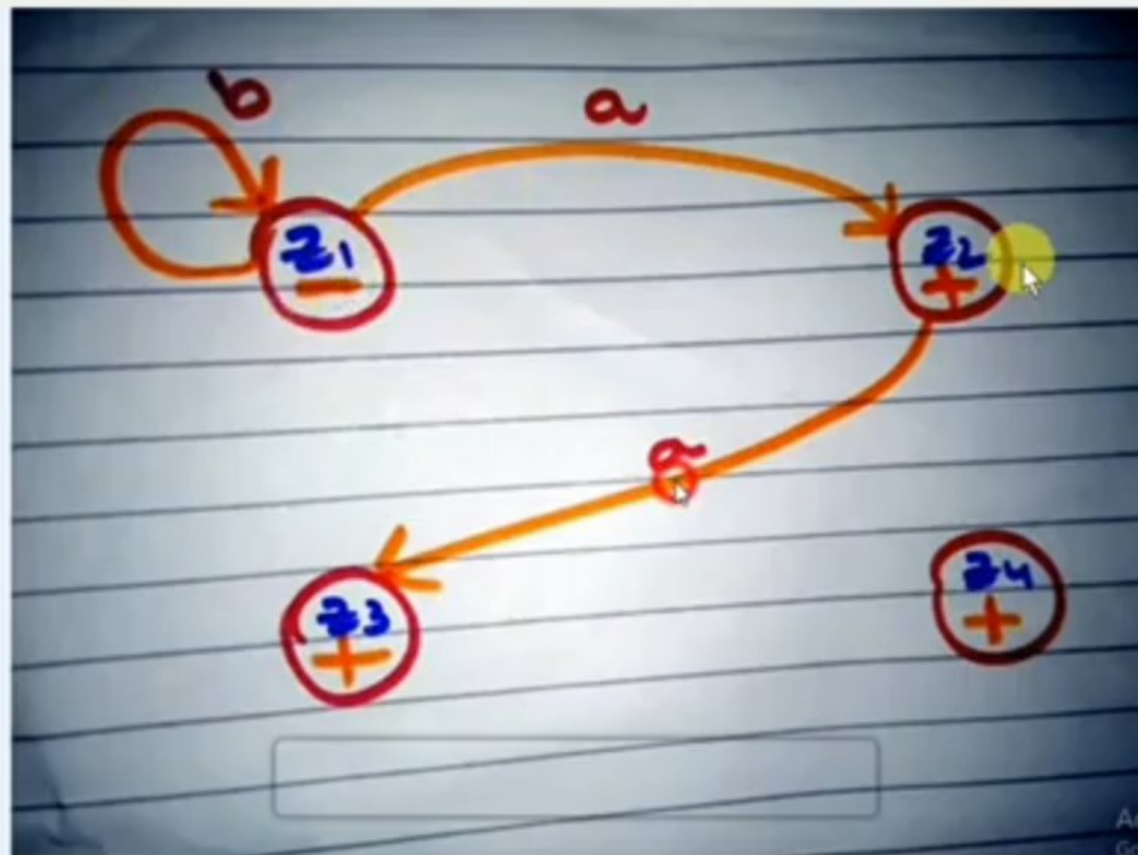
DFA 1



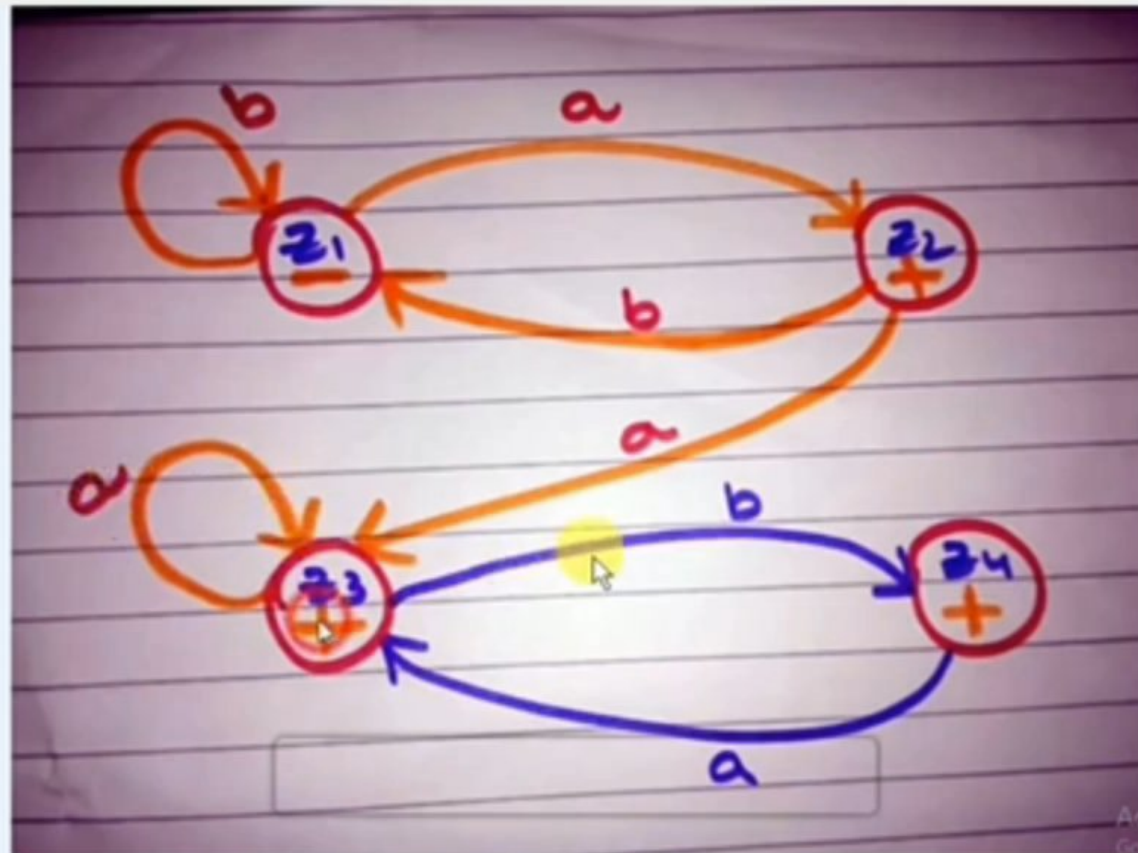
DFA 2

Activate Windows
Go to Settings to activate Windows.

Old States	a	b
- $x_1 y_1 = z_1$	$x_2 y_2 = z_2$	$x_1 y_1 = z_1$
+ $x_2 y_2 = z_2$	$x_2 y_3 = z_3$	$x_1 y_1 = z_1$
+ $x_2 y_3 = z_3$	$x_2 y_3 = z_3$	$x_1 y_3 = z_4$
+ $x_1 y_3 = z_4$	$x_2 y_3 = z_3$	$x_1 y_3 = z_4$



Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

Q#1 Check the given statements or correct or not if not then correct it.

1. String in regular language can not be infinite **True**
2. Concatenation of finite letters from alphabets called sigma **False**
3. There cannot be more than one FA,s for same language. **False**

Q.Is Kleen's star & Kleen's closure are different?

Answer:- (Page 7)

there is no difference between both

Activate Windows
Go to Settings to activate Windows.

Read the given statement Yes/No justify your answer why?

Is RE infinite?

Answer:- (Page 11)

No, All finite languages are regular.

Q: An alphabet is concatenation of letters and is also called sigma?

Answer:- (Page 3)

No, Concatenation of finite number of letters from the alphabet is called a string

Activate Windows
Go to Settings to activate Windows.

Rule# 1

Rule# 2 :union of two FA

Rule#3 concatenation of two FA

Rule#4 kleens cloure of FA
kleens star of an FA

Activate Windows
Go to Settings to activate Windows.

HOW TO TAKE CONCATINATIONOF TWO NFA

Example:

$R_1 = \text{aba}$

$R_2 = \text{bb}$

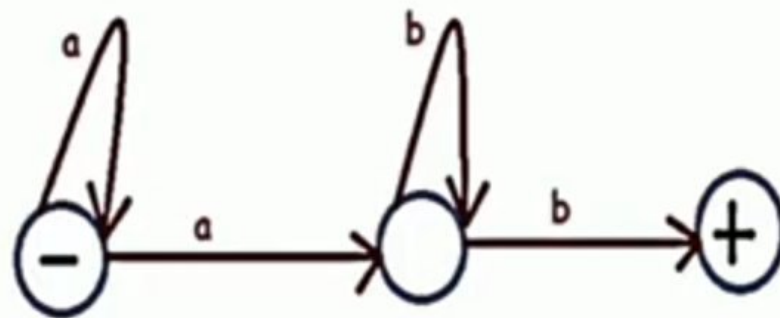
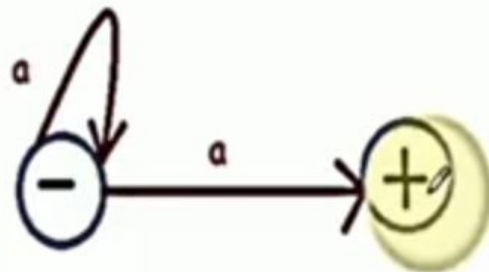
$R_1 R_2$

$= \text{ababb}$

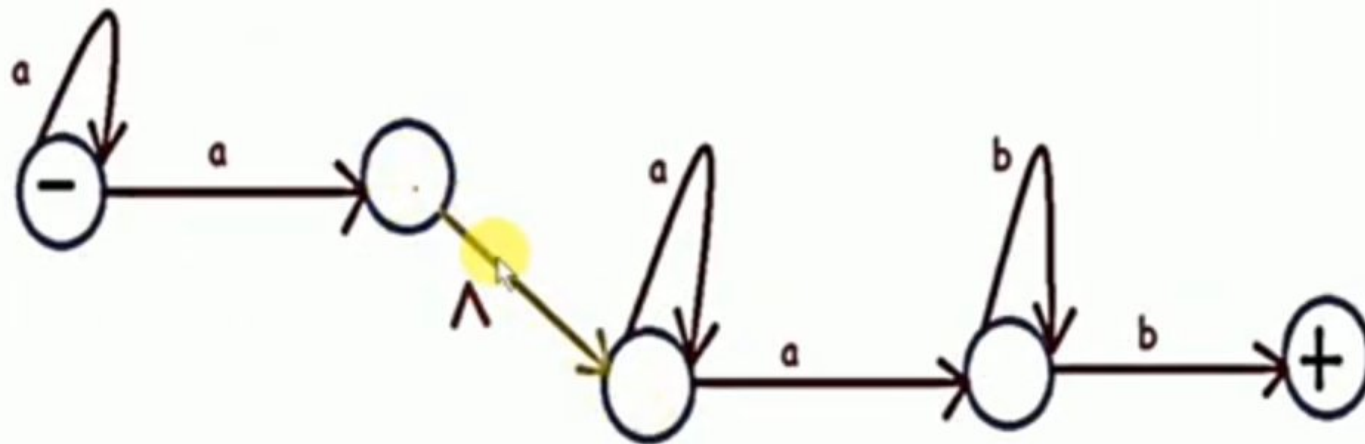
$R_2 R_1 = \text{bbaba}$

r_1 and r_2 are not equal

Question: Concatenation of two NFA's



Question: Concatenation of two NFA's



Activate Windows
Go to Settings to activate Windows.

Q. What is the difference b/w Mealy & Moore machine?

Answer:-

Moore machines

Safer to use because outputs change at clock edge

May take additional logic to decode state into outputs

For Moore machine, output is valid after state transition. Output associated with stable present state

Mealy machines

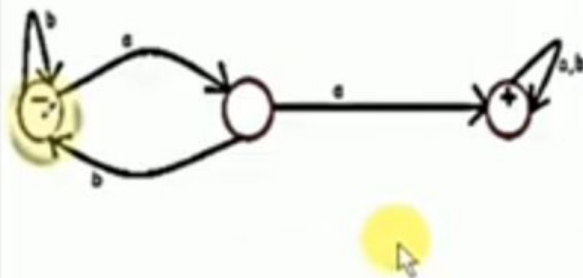
Typically have fewer states

React faster to inputs. don't wait for clock

Asynchronous outputs can be dangerous

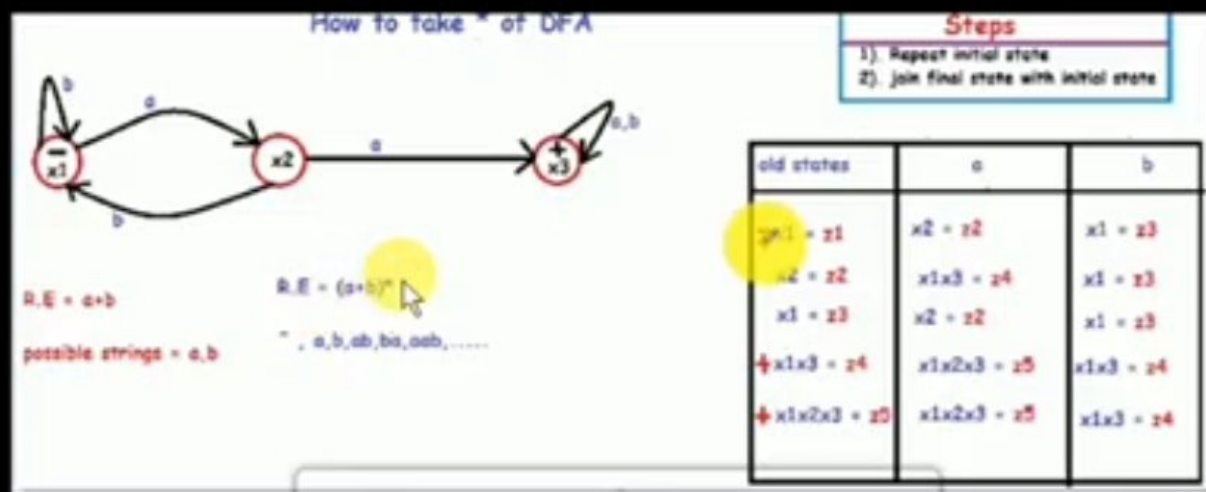
For Mealy machine, output is valid on occurrence of active clock edge. Output associated with transition from present state to next state. Output in Mealy machine occurs one clock period before output in equivalent Moore machine.

How to take * of DFA



Steps

- 1). Repeat initial state
- 2). Join final state with initial state



Activate Windows
Go to Settings to activate Windows.

Q.How can we show transition table of NFA to FA?

Answer:- (Page 45)

there may be more than one transition for a certain letter and there may not be any transition for certain letter, so starting from the initial state corresponding to the initial state of given NFA, the transition table along with new labels of states, of the corresponding FA, can be built introducing an empty state for a letter having no transition at certain state and a state corresponding to the combination of states, for a letter having more than one transitions.

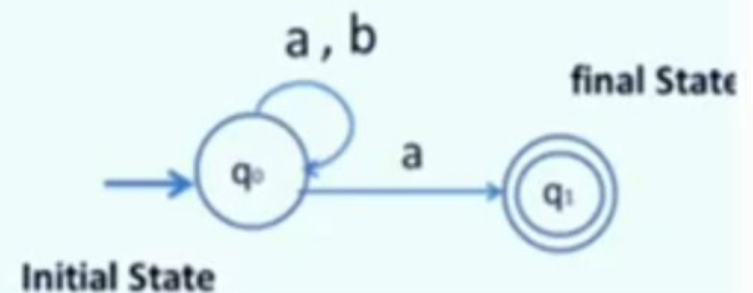
Non-Deterministic Finite Automata

NFA is actually DFA With different rules for more easiness!

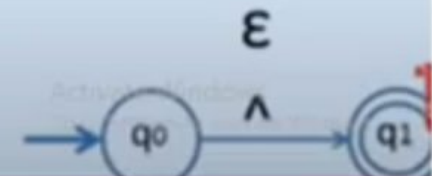
NFA Rules:

1. One letter can move to more than one state, next state is not determined
2. Dead End State is not required in NFA
3. Can move to other state without any input
(i.e., it allows null / epsilon transition)

Accept all words ending with a
 $R = (a+b)^*a$



ϵ - NFA



NFA can be consider a **TG** as well then
make it R.E then make its **FA**
NFA and TG as same

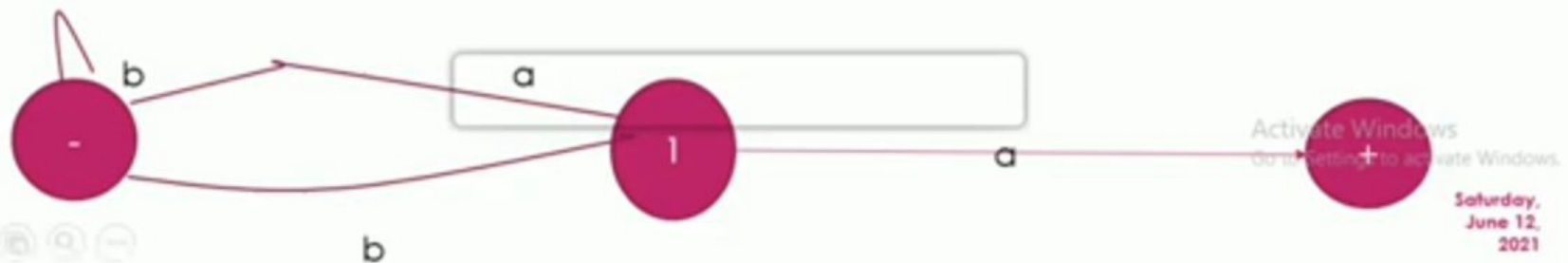
According to kleens theorem if a
language can be accept by a R. E hen it
can be accept by FA

NFA to FA

Example:



R.E = $(a+b)^*aa(a+b)^*$



Finite Automata With Outputs

MEALY MACHINE

$$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

where

Q = Finite Set of States

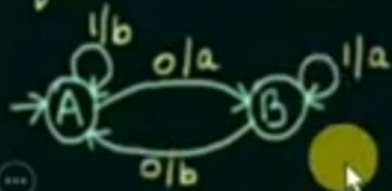
Σ = Finite non-empty set of Input Alphabets

Δ = The set of Output Alphabets

δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $\Sigma \times Q \rightarrow \Delta$

q_0 = Initial State / Start State



MOORE MACHINE

$$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

where

Q = Finite Set of States

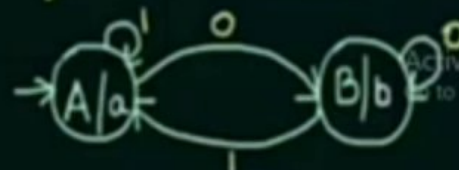
Σ = Finite non-empty set of Input Alphabets

Δ = The set of Output Alphabets

δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $Q \rightarrow \Delta$

q_0 = Initial State / Start State



HUB

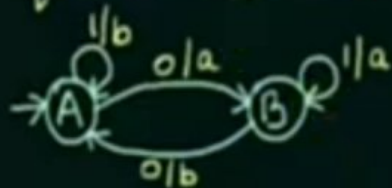
Alphabets

Δ = The set of Output Alphabets

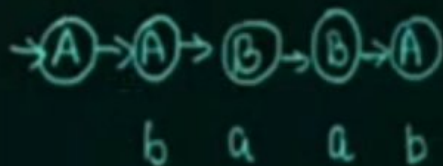
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $\Sigma \times Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



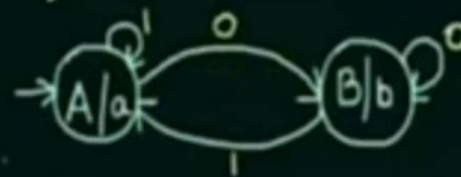
Alphabets

Δ = The set of Output Alphabets

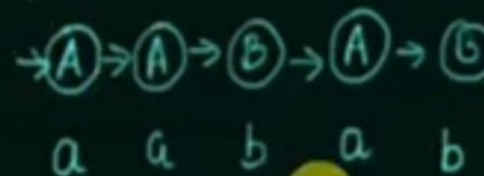
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



Activate Windows
Go to Settings to activate Windows.

HUB

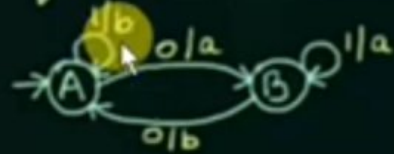
Alphabets

Δ = The set of Output Alphabets

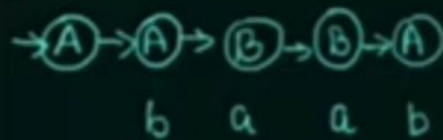
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $\Sigma \times Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



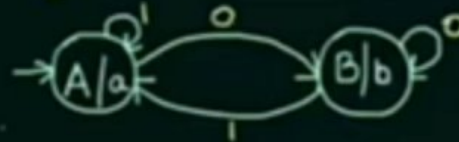
Alphabets

Δ = The set of Output Alphabets

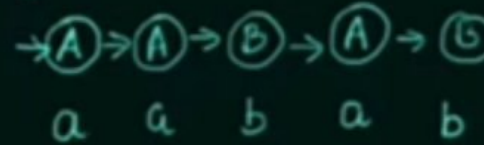
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



N=n

n=n+1

Activate Windows
Go to Settings to activate Windows.



HUB

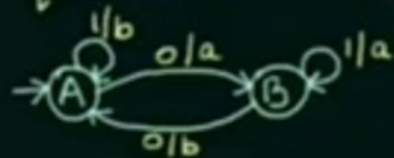
Alphabets

Δ = The set of Output Alphabets

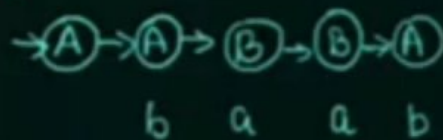
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $\Sigma \times Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



Alphabets

Δ = The set of Output Alphabets

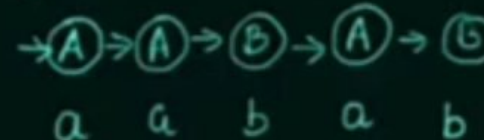
δ = Transition function: $Q \times \Sigma \rightarrow Q$

λ = Output function: $Q \rightarrow \Delta$

q_0 = Initial State / Start State



Eg. 1010



N-n

n=n+1

Activate Windows
Go to Settings to activate Windows.

HUB

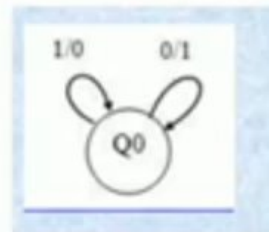
Comparison of Moore and Mealy machines

- ✦ Mealy machines usually have less states
 - ✧ React faster to inputs – don't need to wait for clock
 - ✧ Asynchronous outputs and asynchronous feedback are dangerous
- ✦ Moore machines are safer to use
 - ✧ Outputs change at clock edge (always one cycle later)
 - ✧ May need more logic to decode state into outputs



Write at least one similarity and one difference between DFA & NFA

What is Σ and Γ for this machine



Answer

$\Sigma = \{0, 1\}$ $\Gamma = \{0, 1\}$

Activate Windows
Go to Settings to activate Windows.

Thursday,
June 17,
2021

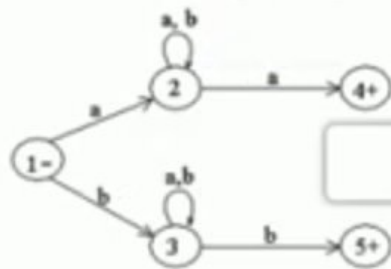
Question:

In transition diagram of an FA, how can we represent initial and final states?

Answer:- we represent the initial state with minus sign (-) and final with plus sign (+)

Question

Describe the language for the following TG



Answer:- (Page 22) Beginning and ending in same letters. The language L may be expressed by RE $a(a + b)^*a + b(a + b)^*b$

Design Moore machine to generate 1's complement of a given binary number

Solution:

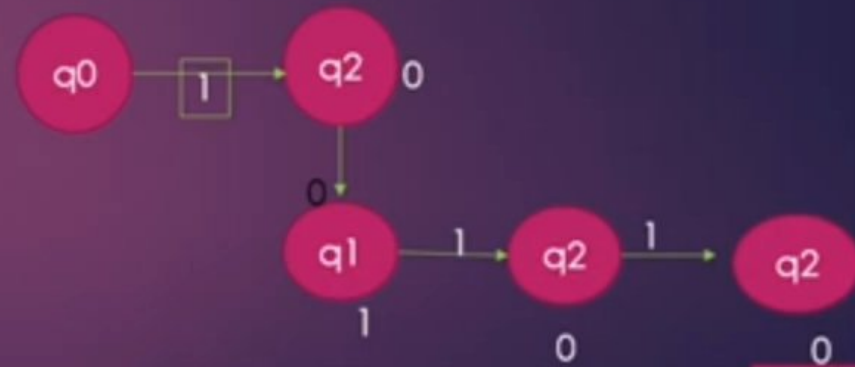
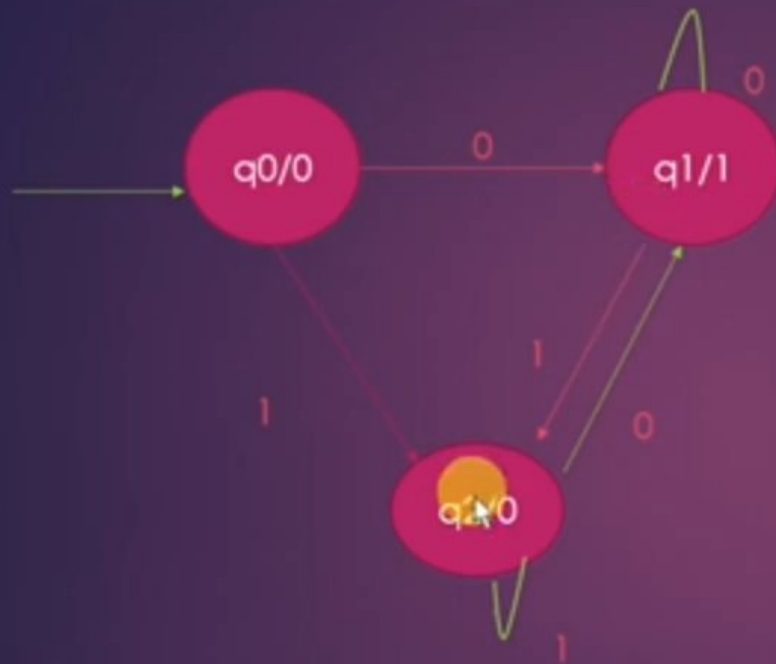
If input 0 _____ output 1
 1 _____ 0

Three states required

1. Start state
2. Is for taking 0 as input produce and 1 as output state
3. Is taking 1 as input and 0 as output

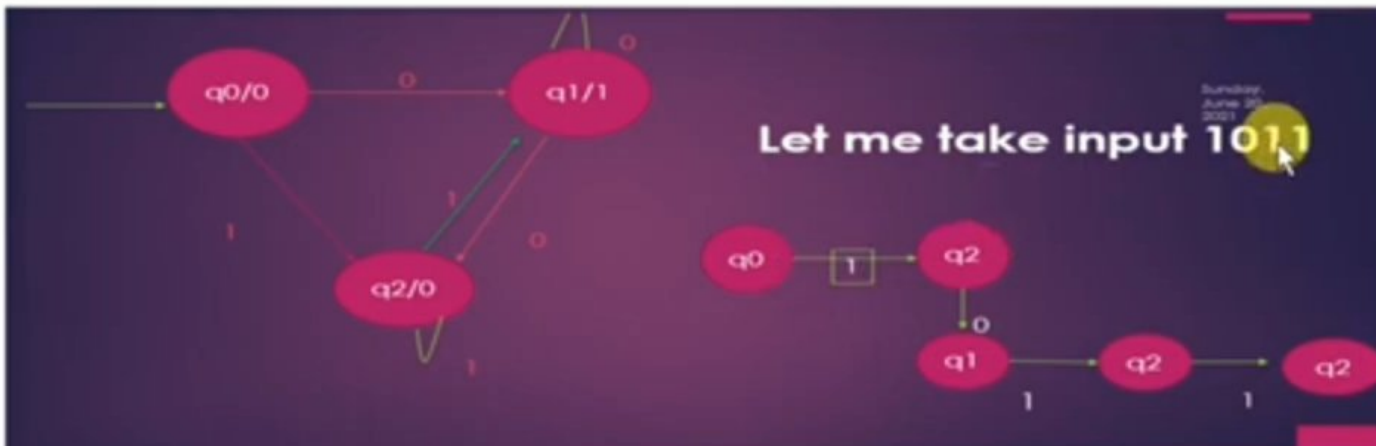
Sunday,
June 20,
2021

Let me take input 1011



Activate Windows
Go to Settings to activate Windows.

HUB



Current state	New state		o/p output
	0	1	
q0	q1	q2	0
q1	q1	q2	1
q2	q1	q2	0

Output
0100

Activate Windows
Go to Settings to activate Windows.

HUB

Q:FINITE AUTOMATA *NFA* EXAMPLES

WHAT IS NFA IN FINITE AUTOMATON???

Note

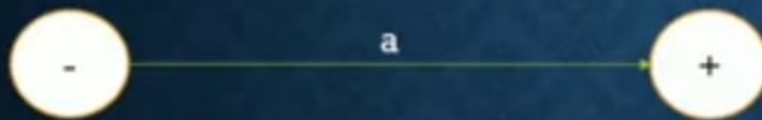
This concept will help you to make *NFA* in finite automaton definition

Q:1

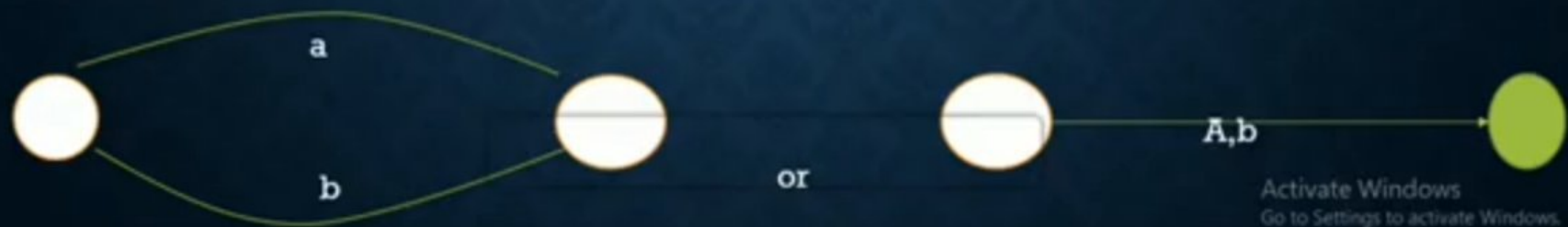
R.E = a

or may be question **b,c**, empty

To make NFA initial state and final state



Q:2 R.E=a+b



Question 3:

R.E = **ab**



Activate Windows
Go to Settings to activate Windows.

Question:

All strings end with a

If $\Sigma = \{a, b\}$

R.E = $(a, b)^* a$

NFA



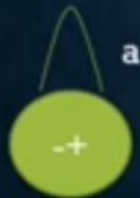
Activate Windows
Go to Settings to activate Windows.

Question

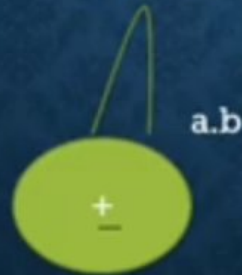
If $\Sigma = \{a\}$, **or** $\{a, b\}$ all strings

$\Sigma = a^*$

NFA



$\Sigma = (a, b)^*$



note :

a^*

b^*

Activate Windows
Go to Settings to activate Windows.

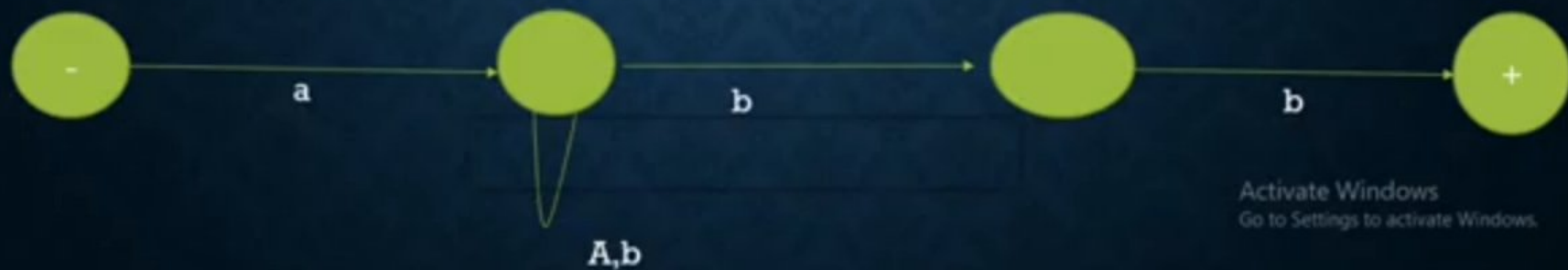
Q:

if $\Sigma = \{a, b\}$ for all strings strt with a end with bb

R.E = $a(a+b)^*bb$

sb sy pely constant ko draw kary gy means without star

NFA



Activate Windows
Go to Settings to activate Windows.

$\Sigma = \{a, b\}$

Q:

All possible strings that start and end with same letter

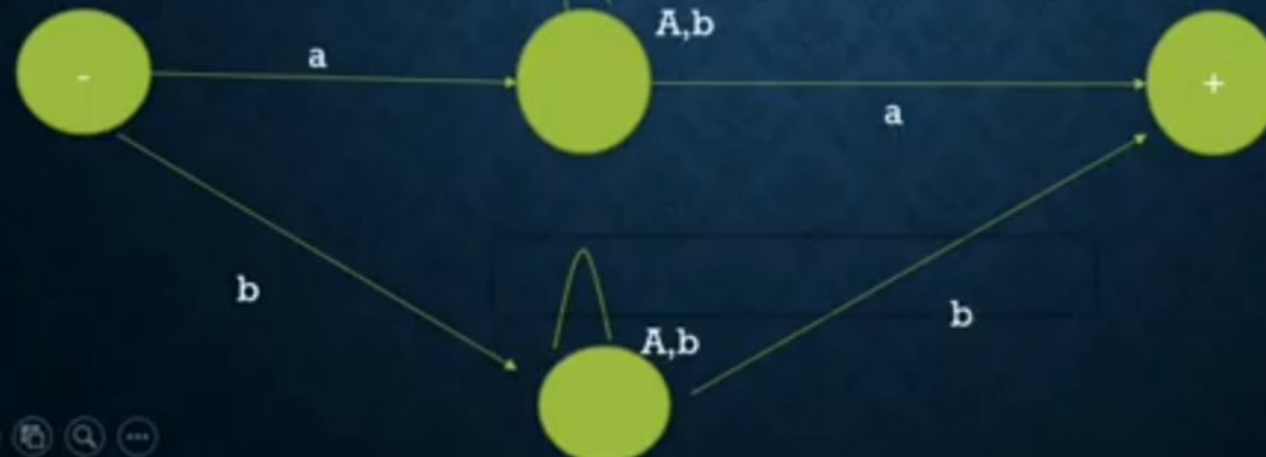
R.E

$a(a+b)^* a$

$b(a+b)^* b$

$a(a+b)^* a + b(a+b)^* b$ one at a time

NFA



Activate Windows
Go to Settings to activate Windows.

Q all strings that contains aa

R.E

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

NFA



QUESTIONS GIVE ASWERS IN COMMENTS PLZITS UPTO U

All strings of even length if $\Sigma = \{a, b\}$

Hint R.E

$(aa)^*$

Strings

$\{\Lambda, aa, aaaa, aaaaaa, \dots\}$

NFA



Q:

if $\Sigma = \{a, b\}$

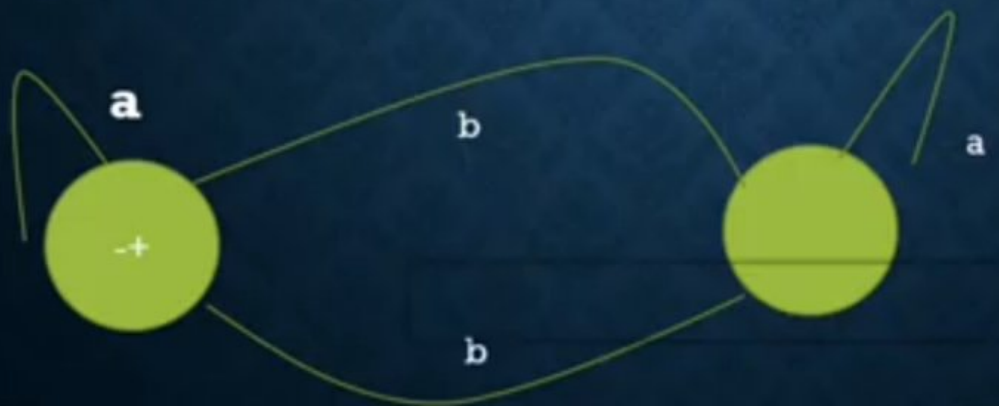
All strings of even no of b

R.E

$(ba\ b)^*$

NFA

also $a^*(b\ a^*b)a^*$



Activate Windows
Go to Settings to activate Windows.

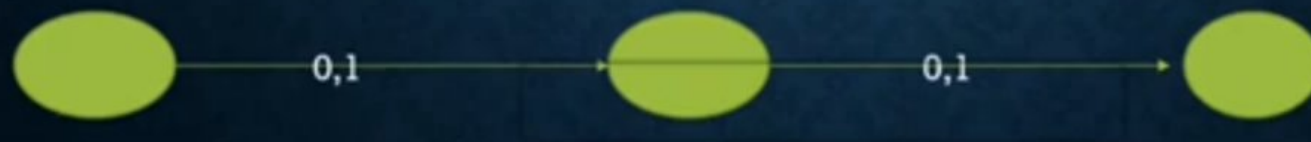
If $\Sigma = \{0,1\}$

All strings of length 2

R.E

$(0+1)(0+1)$

NFA



Activate Windows
Go to Settings to activate Windows.

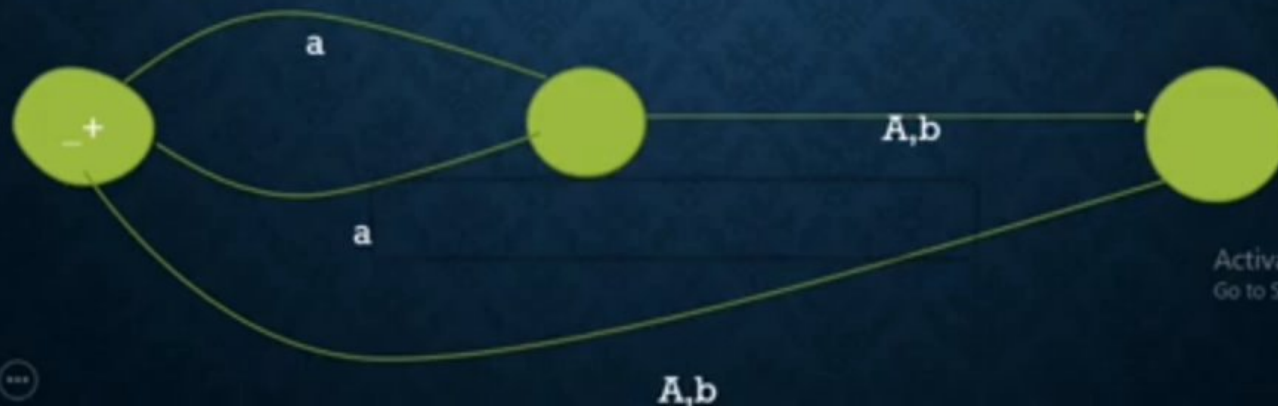
If $\Sigma = \{a, b\}$

All strings o multiplr of 3

R.E

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

NFA



Activate Windows
Go to Settings to activate Windows.

Q:

All strings that contains atleast one a and end with b

R.E

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

????????????????

Plz solve

Activate Windows
Go to Settings to activate Windows.

Theory of Automata (CS402)

Lecture 7

Some examples and their practice

TG

GTG

FA corresponding to **finite languages**(using both methods),
Transition graphs.

Rules of TG

1. Can have more than one initial state
2. Empty transaction is allowed
3. We allow the machine to read more than one character at a time

Question solution

If $\Sigma = \{a, b\}$ draw TG for all strings containing bbb or aaa

R.E

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

Diagramme:



Activate Windows
Go to Settings to activate Windows.



Above is Another solution of this question

Activate Windows
Go to Settings to activate Windows.

Remember please

NFA

FA

Each state has 2 arrow

DFA

TG

GTC

Aik arrow par aik 1 character likh skty

NFA also called type of TG

Bu every TG is not NFA

TG basically reduce the complexity

Activate Windows
Go to Settings to activate Windows.

Note:

It is to be noted that the states x and y are called **Dead States, Waste Baskets or Davey John Lockers**, as the moment **one** enters these states there is **no way** to leave it.

CS402 Mid Term Papers Fall 2019

Mostly mcqs past papers by moazz file

Long questions pahly 1 to 3 leacturz ma sy thy

Short questions

1,finte FA ki maximum length 6

2,TG ka structure tha uska table banan th

Cs402 today paper.

Mcqs from past and conceptual.

Fa create krna tha. Condition btai hui thi us k mtabik.

Aik fa bnana tha jis m strt or end m aa aye.

Recursive definition of a language ...expression jo recursively explorer krna tha.

TG ko eliminate krna tha .

2 TG thy un ki final states ko reduce kr k sirf aik final state m convert krna tha....overall paper was eas

Examples and practice

Question 1:

Construct **FA** and **TG** for the language which has strings

$\Sigma = \{a, b\}$

1. Now generate strings

$\Lambda, ab, ba, aba, bab$

2. Draw FA

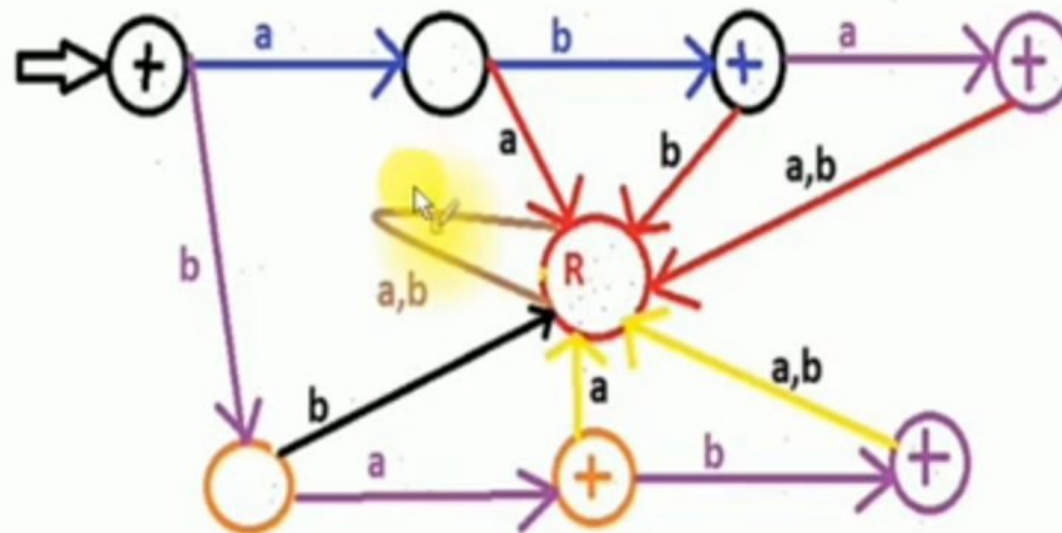
Activate Windows
Go to Settings to activate Windows.

Question#1: Construct FA and TG for the language which has strings:

if $\Sigma = \{a,b\}$

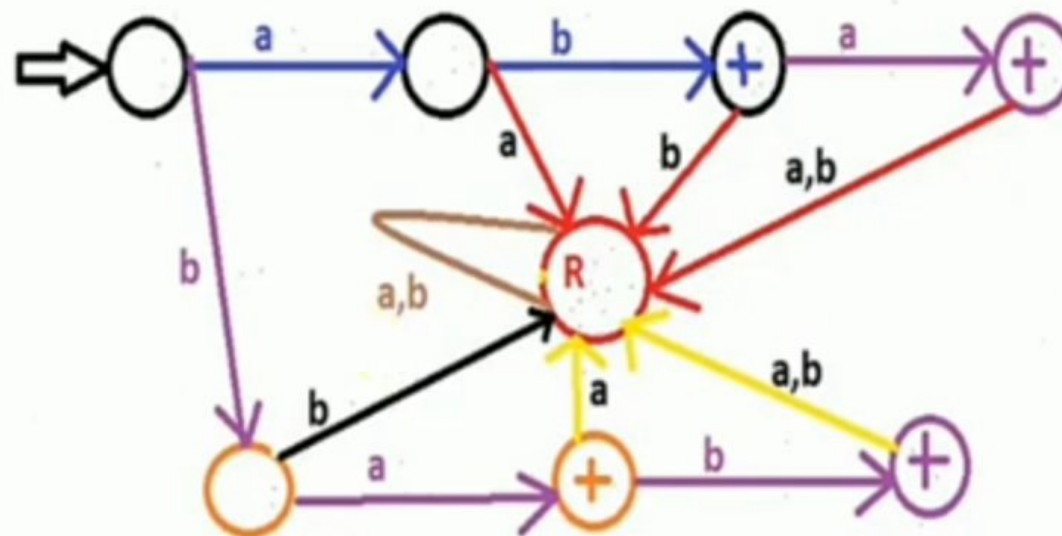
$\epsilon, ab, ba, aba, bab$

FA:



Activate Windows
Go to Settings to activate Windows.

FA:



ab , ba , aba, bab

TG:



Activate Windows
Go to Settings to activate Windows.

Question#2

Build FA & TG for language that must contain **aab** as substring.
if $\Sigma = \{a,b,c\}$

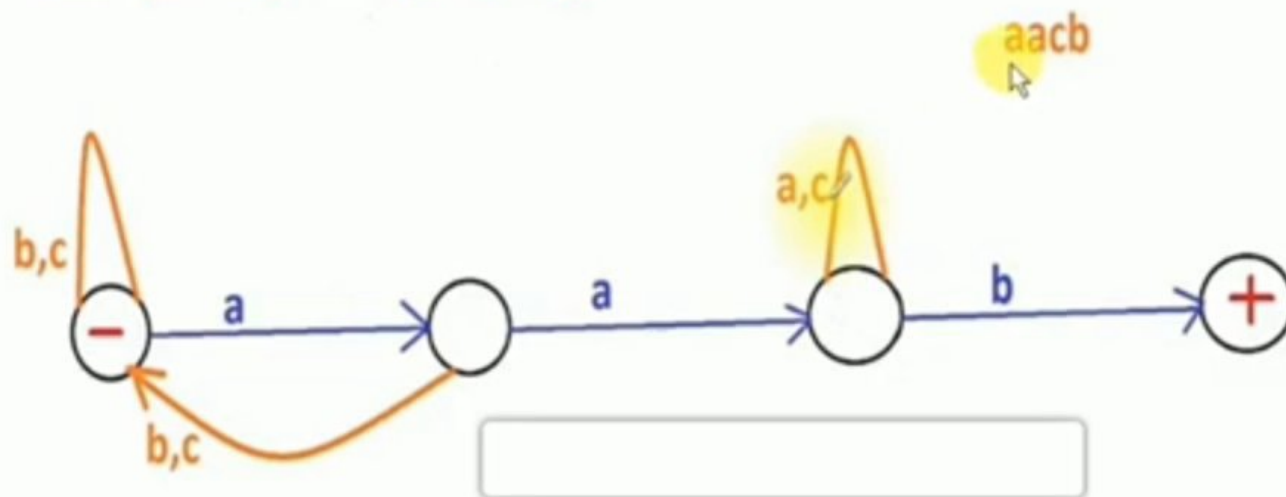
$$\text{R.E} = (a+b+c)^* \text{aab} (a+b+c)^*$$



Question#2

Build FA & TG for language that must contain **aab** as substring.
if $\Sigma = \{a,b,c\}$

$$\text{R.E} = (a+b+c)^* \text{aab} (a+b+c)^*$$

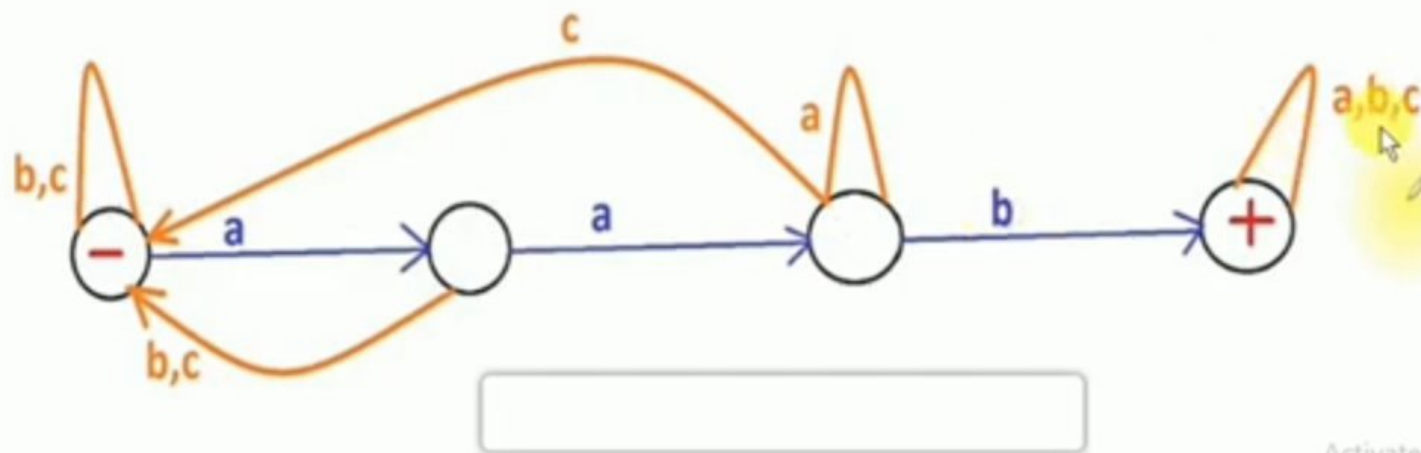


Activate Windows
Go to Settings to activate Windows.

Question#2

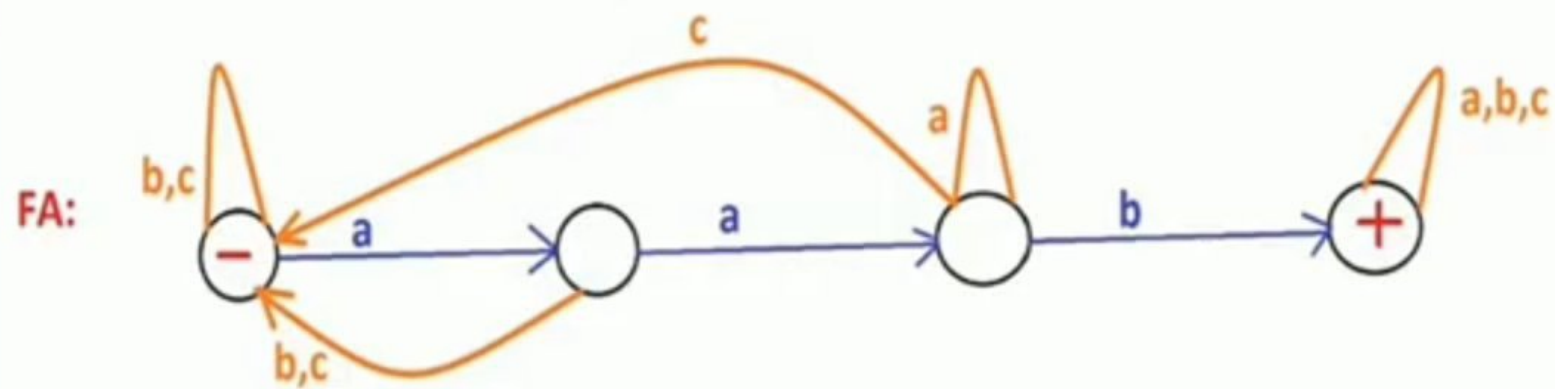
Build FA & TG for language that must contain **aab** as substring.
if $\Sigma = \{a,b,c\}$

$$\text{R.E} = (a+b+c)^* \text{aab} (a+b+c)^*$$



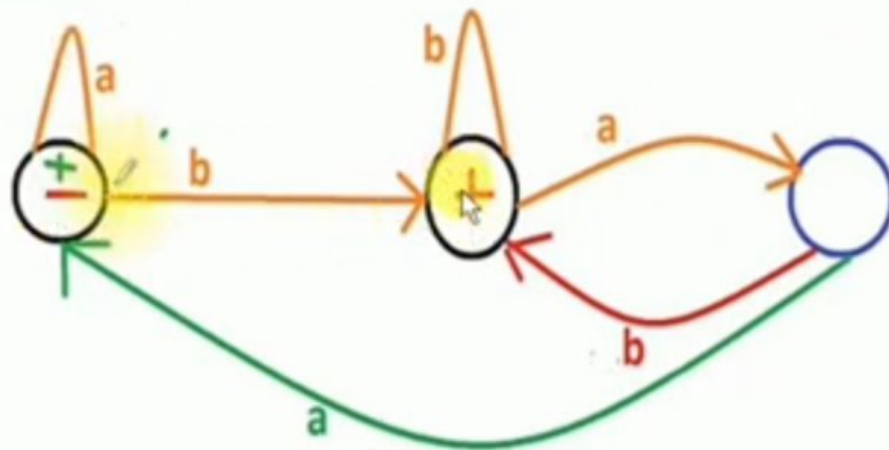
Activate Windows
Go to Settings to activate Windows.

$$R.E = (a+b+c)^* aab (a+b+c)^*$$



Activate Windows
Go to Settings to activate Windows.

Question#3: Build an FA for all strings not ending with ba.

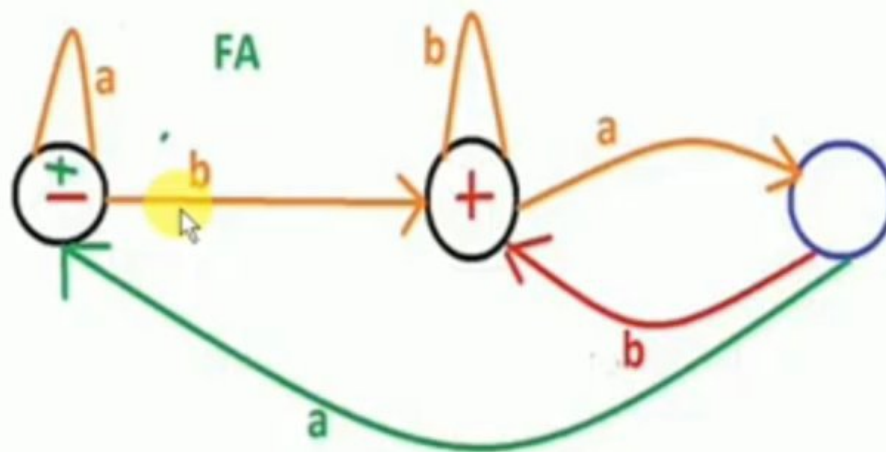


Activate Windows
Go to Settings to activate Windows.

Question#3: Build an FA for all strings not ending with ba.

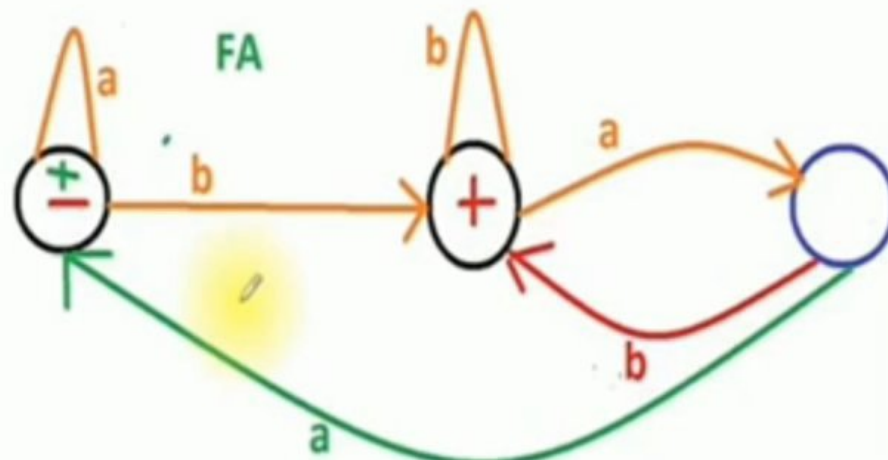
R.E =

$(ab+aa+bb)$

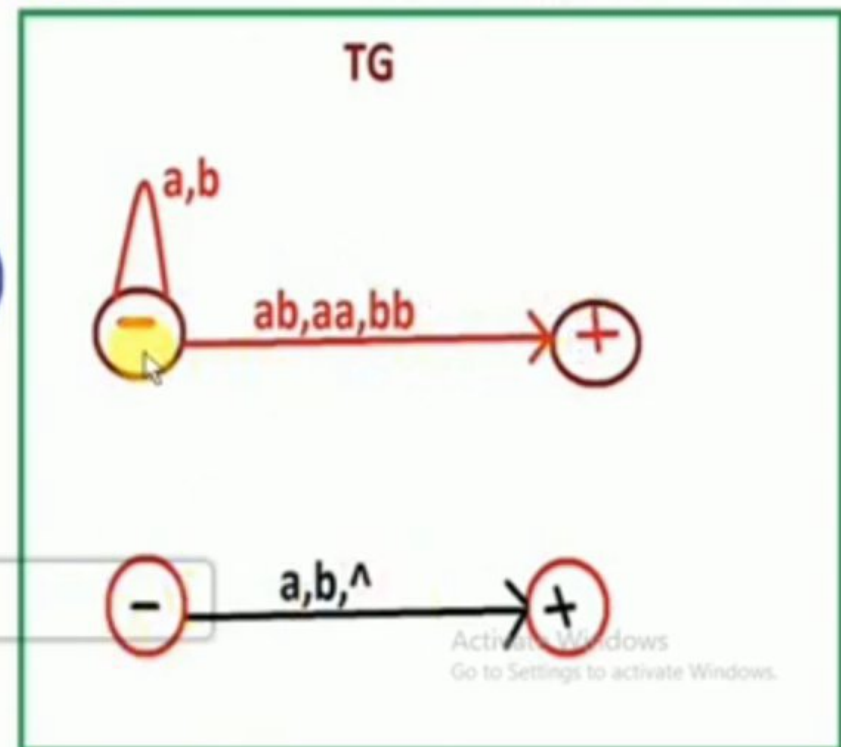


Activate Windows
Go to Settings to activate Windows.

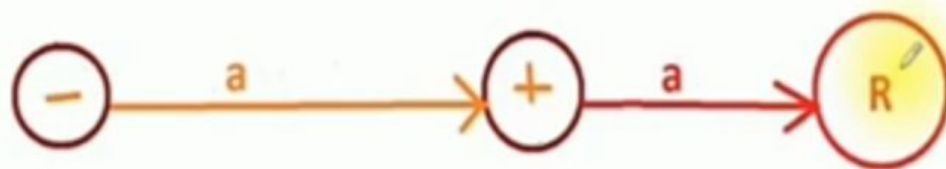
Question#3: Build an FA for all strings not ending with ba.



R.E = $(a+b)^*(ab+aa+bb)+a+b+^$



Question#4: Build FA without two consecutive a's.



Activate Windows
Go to Settings to activate Windows.

Q. Difference and common between NFA & DFA

Answer:- (Page 25)

Difference

1-In FA Finite number of states, having one initial and some (maybe none) final states. While in NFA Finite many states with one initial and some final state.

2-In FA for each state and for each input letter there is a transition showing how to move from one state to another while in NFA there may be more than one transition for certain letters and there may not be any transition for certain letters.

3-In FA



is valid while in NFA



is not valid.

Activate Windows
Go to Settings to activate Windows.

2) Write the first step to convert  TG to FA?

Answer:- (Page 26)

Step 1 If a TG has more than one start states, and then introduces a new start state connecting the new state to the old start states by the transitions labeled by Λ and make the old start states the non-start states

Activate Windows
Go to Settings to activate Windows.

Q#2. What is the difference between the strings and the words of a language?

Answer:- (Page 3)

Concatenation of finite number of letters from the alphabet is called a string. Words are strings belonging to some language.

Q#3. Is there any difference in PALINDROME and reverse of palindrome explain it?

Answer:- (Page 6)

PALINDROME is The language and words of **PALINDROME** are called reverse of palindromes



Q.Is Kleen's star & Kleen's closure are different?

Answer:- (Page 7)

there is no difference between both



Activate Windows
Go to Settings to activate Windows.

Q. What is the difference b/w Mealy & Moore machine?

Answer:- Click here for detail

Moore machines

- 1 .Safer do use because outputs change at clock edge
- 2 .May take additional logic to decode state into outputs
3. For Moore machine, output is valid after state transition. Output associated with stable present state

Mealy machines

Typically have fewer states

React faster to inputs. don't wait for clock

Asynchronous outputs can be dangerous

For Mealy machine, output is valid on occurrence of active clock edge.

Output associated with transition from present state to next state. Output in Mealy machine occurs one clock period before output in equivalent Moore machine.

Activate Windows
Go to Settings to activate Windows.

Theory of Automata (CS402)

Lecture 8

Concept /definition/rules

Some examples and their practice

GTG

GTG (generalized transition graph)

Properties of GTG

- 1- GTG consists finite number of states
- 2- At least one start state and some (maybe none) final states.
- 3- GTG consist finite set of input letters (Σ) from which input strings are formed
- 4- Only one arrow for all states.

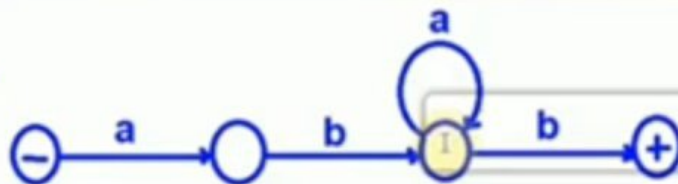
Question/example

R.E (regular expression) = ab^+a^+b $\Sigma = \{a,b\}$

NFA=



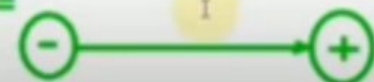
NFA=



TG =



GTG=

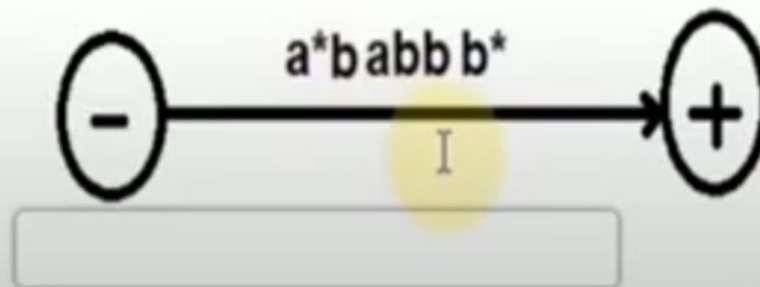


Activate Windows
Go to Settings to activate Windows.

EXAMPLE

R.E= $a^*babb\ b^*$

GTG=



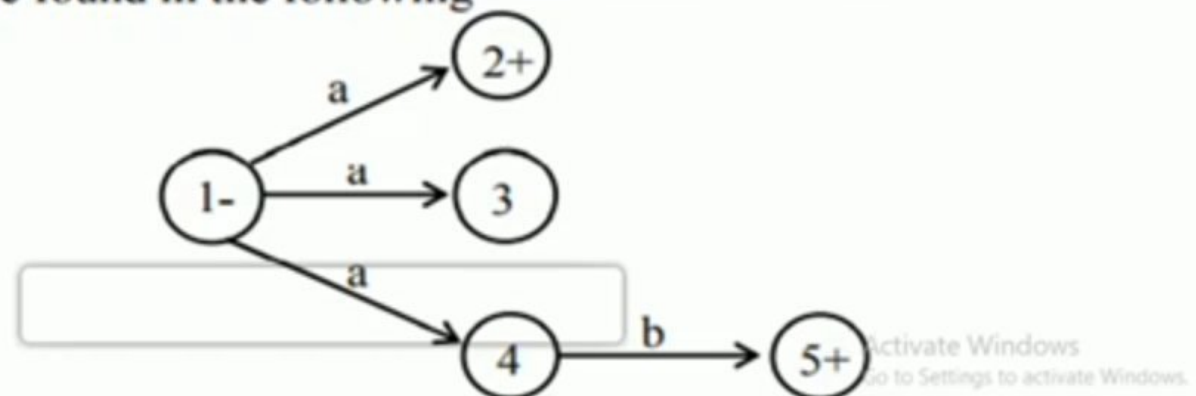
Activate Windows
Go to Settings to activate Windows.

NFA, that the string is supposed to be accepted, if there exists at least

**one successful path, otherwise rejected. /aik
arrow par aik alphabet ata ha**

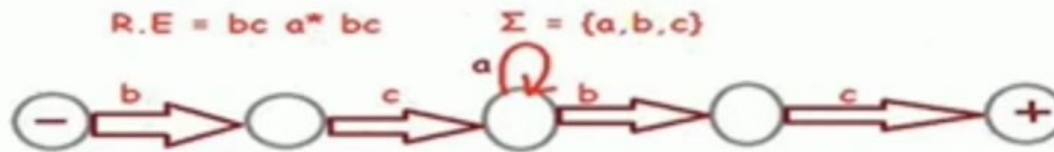
The examples of **NFA**s can be found in the following

Example

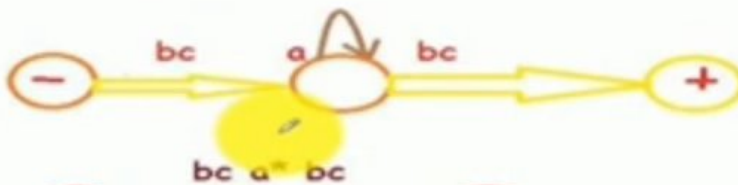


It is to be noted that the above **NFA** accepts the language consisting of a and ab.

NFA



TG



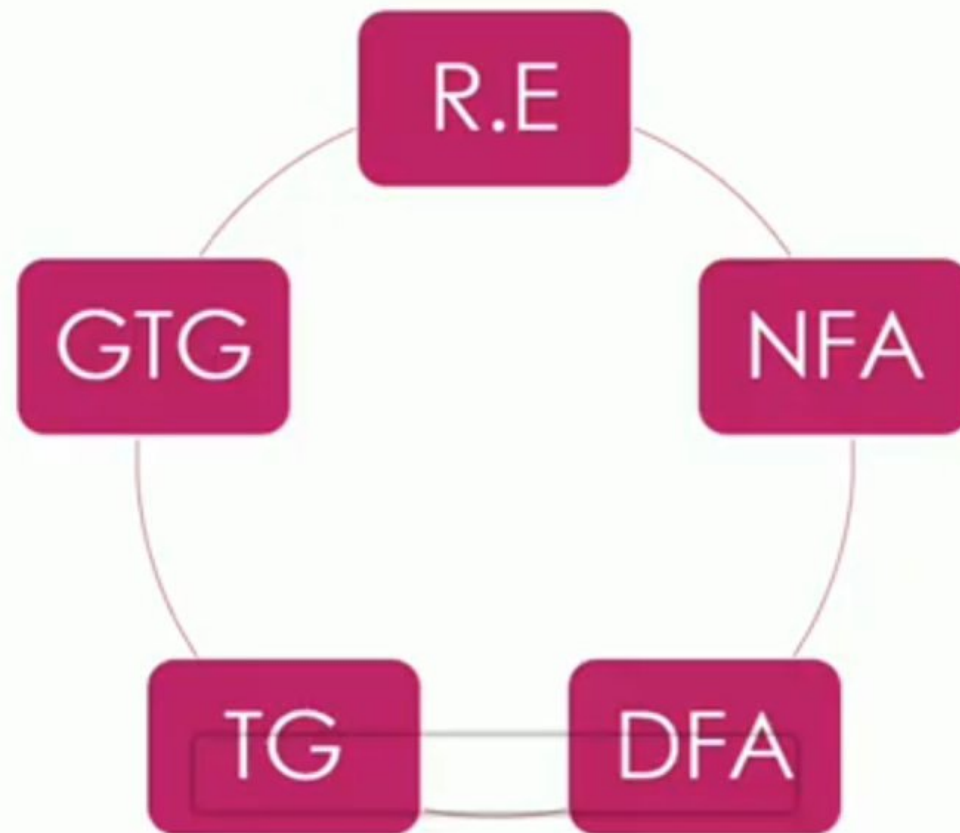
GTG



Transition Graphs (TGs) and Generalized Transition Graphs (GTGs)

Transition Graphs	Generalized Transition Graphs
Finite number of states	same
Finite set of input strings	same
Finite set of transitions including NULL string	Finite set of transitions including NULL string and transitions can represent Regular expressions

Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

Kleene's Theorem Part I

Statement:

If a language can be accepted by an **FA** then it can be accepted by a **TG** as well.

Proof :

Since every **F.A** is also a **TG** ,therefore every language that can be defined by an FA can also be defined by TG

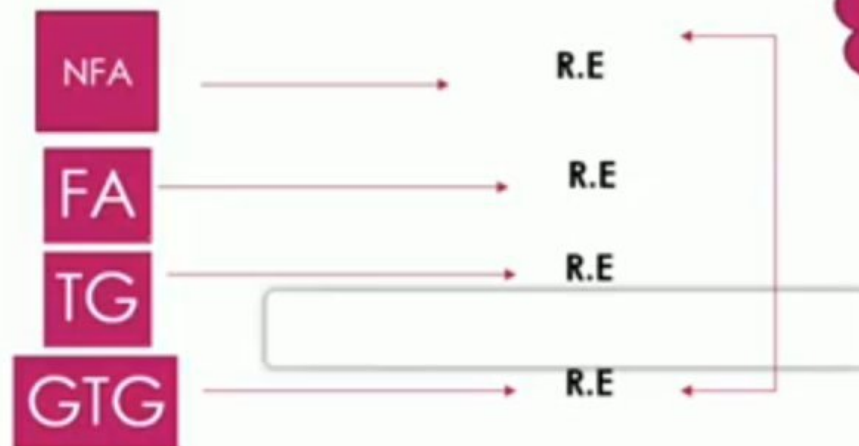
EXAMPLE:

every **FA** considered as **TG**
So ,there is nothing to prove

Kleene's Theorem Part II

Statement:

If a language can be expressed by a **TG**
then it can be accepted by an **R.E** as well



STATE
ELIMINATION
RULES

Activate Windows
Go to Settings to activate Windows.

Kleens theorem PART 3

STATEMENT:

If a language can be expressed by a RE then it can be accepted by an FA as well

EXAMPLE:

$\Sigma = \{a, b\}$

R.E = $(a+b)^*a(a+b)^*$

F.A =



RULES of part 3

Rule# 1

Rule# 2:union of two FA

Rule#3:concatenation of two FA

Rule#4 kleens cloure of FA
kleens star of an FA

Recursive definition of R.E

$$R.E_1 = a*b$$

$$R.E_2 = (a+b)^*$$

$$R.E_3 = RE_1 + RE_2 = a*b + (a+b)^*$$

$$RE_4 = RE_1 RE_3 = a*b(a+b)^*$$

$$RE_5 = (RE_1)^* = (a*b)^*$$

Regular Expression

“A regular expression
(shortened as regex or regexp)
also referred to as” **rational expression**)

Defination :

“The language accepted by finite automata
can easily describe by simple expression
is called” Regular Expression

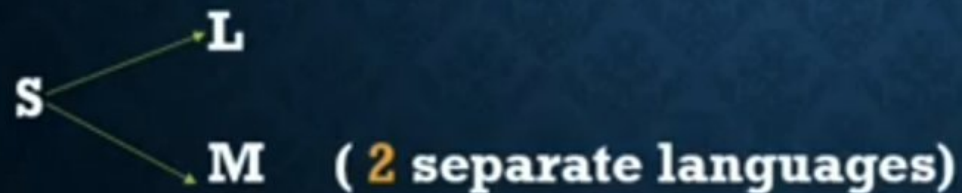
FOR INSTANCE

- In RE a^* “means zero” zero and more occurrence of “x”
- It can generate $L = \{\Lambda, x, xx, xxx, \dots\}$
- In RE a^+ means one and more occurrence of “x”
- $L = \{x, xx, xxx, \dots\}$

FOR UNION

If **L** & **M** are two regular languages their (**L U M**) is also called union

$$\mathbf{L \cup M} = \{s \mid s \text{ is in } \mathbf{L} \text{ and } s \text{ is in } \mathbf{M} \}$$



INTERSECTION

$L \cap M = \{ st \mid s \text{ is in } L \text{ and } T \text{ is in } M$

$L \longrightarrow s$

$M \longrightarrow t$

THEORY OF AUTOMATA

- LECTURE 4
- AND SOME IMPORTANT QUESTION AND COCEPTS

Difference of regular expressions

$r1 = a^* + b^*$

$r2 = (a + b)^*$

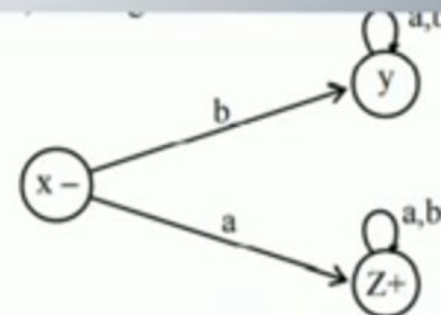
- Here $r1$ does not generate any string of concatenation of a and b ,
- While $r2$ generates such strings

Is $r1$ generate any string??

Describe difference of these two regular expression?

TRANSITION TABLE/DIAGRAM

Old States	New States	
	Reading a	Reading b
x -	z	y
y	y	y
z +	z	z

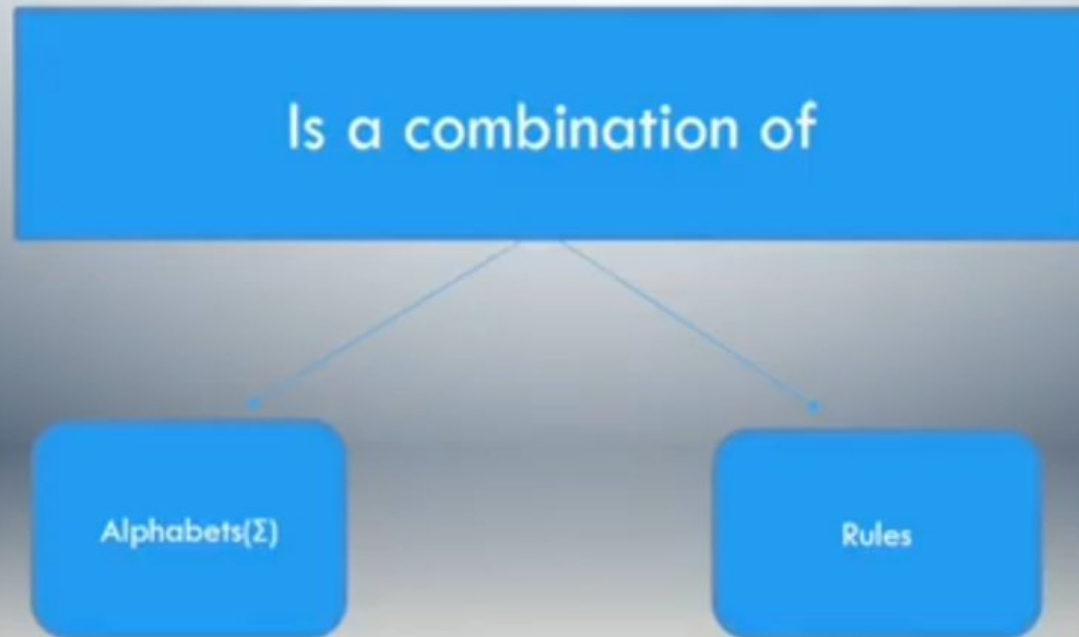


LANGUAGE

Is a combination of

Alphabets(Σ)

Rules



EXAMPLE

if $\Sigma = \{a, b\}$ alphabets

All strings which are ends with a (Rule)

Question :

What is string? Differenc b/w string and words

Language = English

Alphabets $= \Sigma = \{a, b, c, d, \dots, z\}$

Then

Strings

words

Aba

cat

Bkmm

walk

Bkts

fish

LENGTH OF STRING

Strings always b positive

Adda (length = 4)

abdafbas (length = 7)

Concatenation of string

When two or more strings combine to form a new string

$X = ab$

$Y = c$

$XY = abc$

&

$YX = cab$

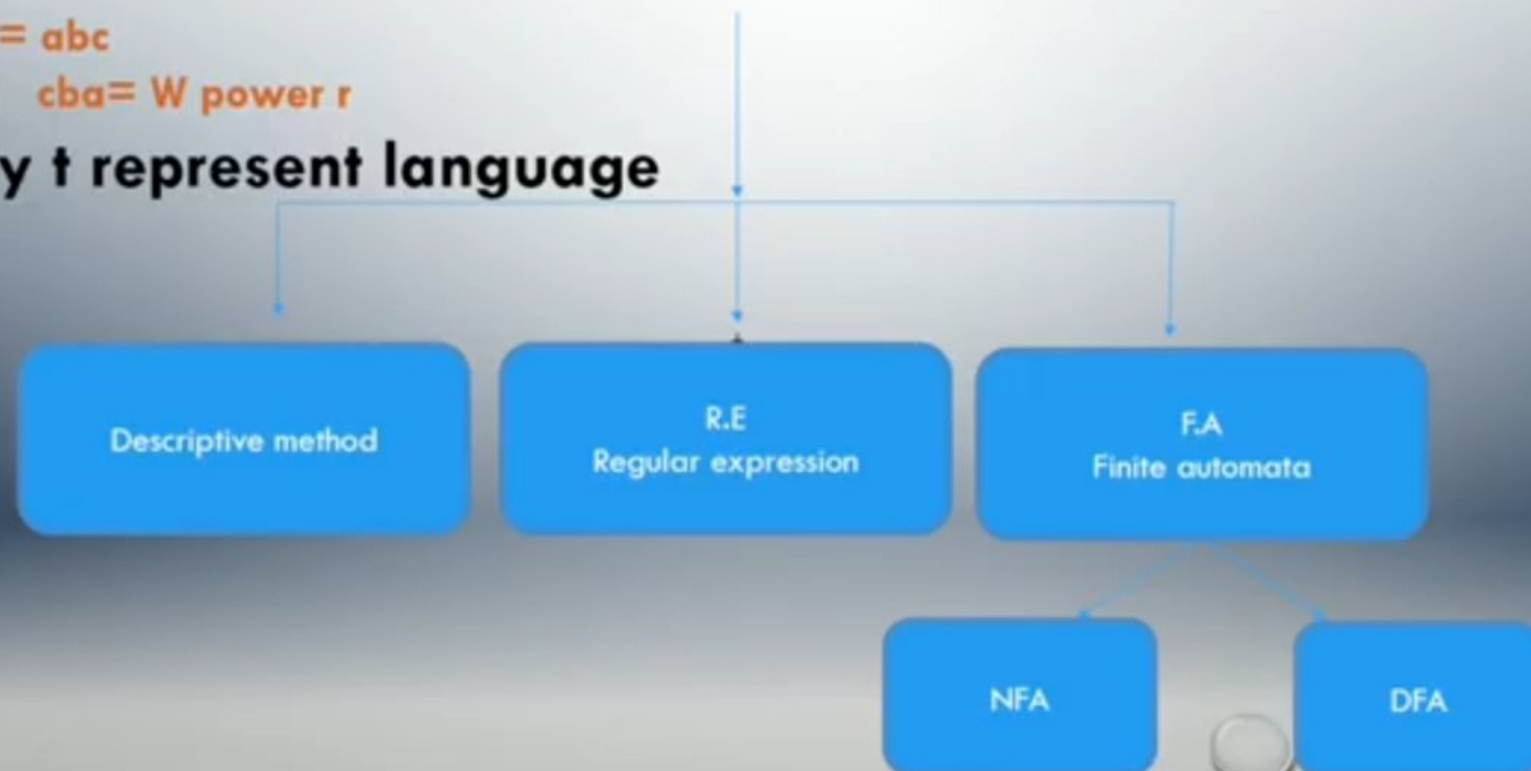
$xy \neq yx$

REVERSE OF STRINGS

If $W = abc$

Then $cba = W^r$

Way to represent language



DEScriptive method

example:

If $\Sigma = \{a, b\}$

All strings ending with a

$\{a, ba, aa, aa, baa, \dots\}$ order should be ascending

$\{a, aba, ba, aba, aa, \dots\}$ wrong length must be arranged

Example :

Write all strings which contain 00

$\{00, 100, 001, 000, 1001, \dots\}$

PALANDROME

if $\Sigma = \{aa, bab\}$ ind length of string = babaabab

Length = (bab)(aa)(bab) = 3 length

Length mean how many letr use in question

Reverse string: bab aa bab

Aesa string jisko revrse kr k likhny sy wo hi ban jy

Regular expression

Language

```
graph TD; Language --> SingleLetter[Single letter language]; Language --> Finit[Finit language]; Language --> Infinite[Infinite letter language];
```

Single letter
language

Finit language

Infinite letter language

Single letter language

Contain only one alphabet

examples

If $\Sigma = \{b\}$

$R.E = b$

$\Sigma = \{\text{null } \Lambda\}$

$R.E = \text{null } \Lambda$

$\Sigma = \{a\}$

$R.E = a$

finite language letter

Defination : strings are countable

Example :

$\Sigma = \{a,b\}$

All strings of length **2 two**

Rules:

1: generate strings

2. Add strings

Apply rule 1 = $\{aa,bb,ab,ba\}$

Pply rule 2 R.E = $aa+bb+ab+ba$

Infinite language

Definition : strings are not countable

Properties :

- 1. Kleene closure**
- 2. Positive closure**

Kleene closure

Represented by $*$

Contains null Λ

Example:

$\Sigma = \{a\}$

All strings including null Λ

$\{\Lambda, a, aa, aaa, \dots\}$

R.E = a^*

Positive closure

Represented by $+$

no null Λ

Example:

If $\Sigma = \{a\}$

Write all strings without a Λ

$= \{a, aa, aaa, \dots\}$

R.E = a^+

Examples

If $\Sigma = \{a\}$

Write regular expression for all strings

$\{\Lambda, a, aa, aaa, aaaa, aaaa, \dots\}$

Note : kici letter k sary strings bnany ho to * lgaty h

$R.E = a^*$

Example 2

If $\Sigma = \{a\}$

Write all strings without Λ

$\{a, aa, aaa, \dots\}$

Note :

Null na hony ki waja sy +

$R.E = a^+$

If $\Sigma = \{a, b\}$

For all strings

$\{\Lambda, a, b, ab, ba, aab, abb, aaa, \dots\}$

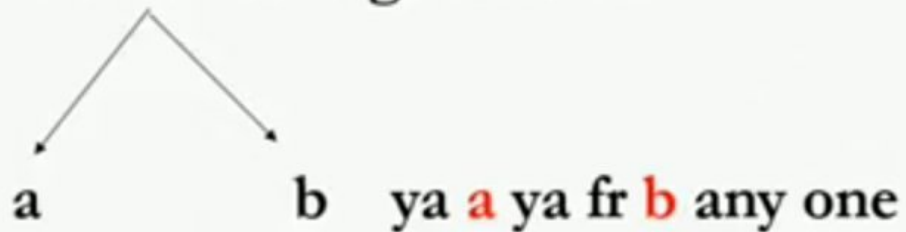
$R.E = (a+b)^*$

+ symbol in regular expression

A+b

Aik hi string bny ga

Possible strings of $a + b$



Example:

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

$(ab)^* = \{\Lambda, ab, abab, ababab, \dots\}$

$(a + b)^* = \{\Lambda, ab, baaaba\}$

- Mean loop
- Loop mean repeat krna

If **If $\Sigma = \{a,b\}$** countable strings are finite

Note:

Count kry or + ka sign kr dy

Write R.E for the strings ab,ba,aab
{ab,ba,aab}

R.E=ab+ba+aab

Example:

Difference b/w two R.E

A+b

Make only **2** two strings

A ya b

Ab ba aab bba nahi ban skty

(a+b)

All strings

Like

[Λ ,a,b,ab,ba,.....]

If $\Sigma = \{a, b\}$

All possible strings that starts and end with **b**

$\{bb, bab, bbb, baab, babb, \dots\}$

R.E = $b(a+b)^*b$

Example 2:

If $\Sigma = \{a, b\}$

All possible strings that contain exactly **2a**

$\{aa, baa, aba, aab, abab, babab, \dots\}$

R.E = $b^*a b^*a b^*$

Here **b^*** has any value describe below

$b^* = \Lambda, b, bb, bbb, bbbb$ upto so on

Example:

All possible strings that contains double **b**

{bb,abb,bba,bbb,ababb,bbaba,.....}

R.E $(a+b)^*bb(a+b)^*$

Example:

All possible strings that starts and end with same letter

two cases

$a(a+b)^*a$

$b(a+b)^*b$

So R.E

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

If $\Sigma = \{0,1\}$

All possible strings whose length is 3

R.E $(0+1)(0+1)(0+1)$
 ↓ ↓ ↓
 0 1 0

If $\Sigma = \{0,1\}$

All possible strings with 3rd letter is 1

$(0+1)(0+1) 1 (0+1)^*$
 ↓
 1