Lecture #05

Date: Sept 2nd, 25 Day: Tuesday

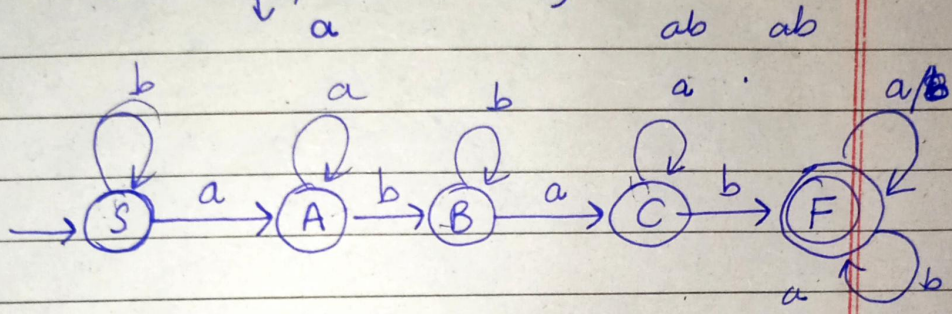
Construct a DFA:

$\mathcal{L} = \{ x \mid x \in \Sigma^* ;$ x contains atleast two occurces of 'ab' $\}$

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

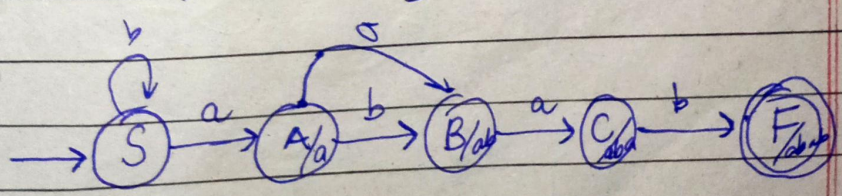
babab

$\mathcal{L} = \{$ abab, aabab, ababa, abbab, ababa, ababb, baabab, abbab, abaabab, aababa, aaababab $\}$



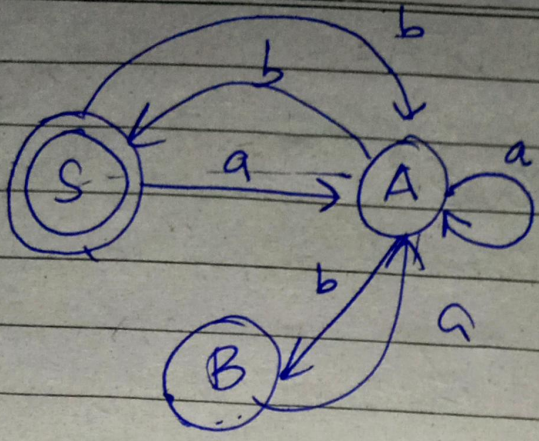
$\mathcal{L} = \{ x \mid x \in \Sigma^* ;$ x contains atleast 2 occurances of 'ab' and $|x| \leq 6 \}$

ab

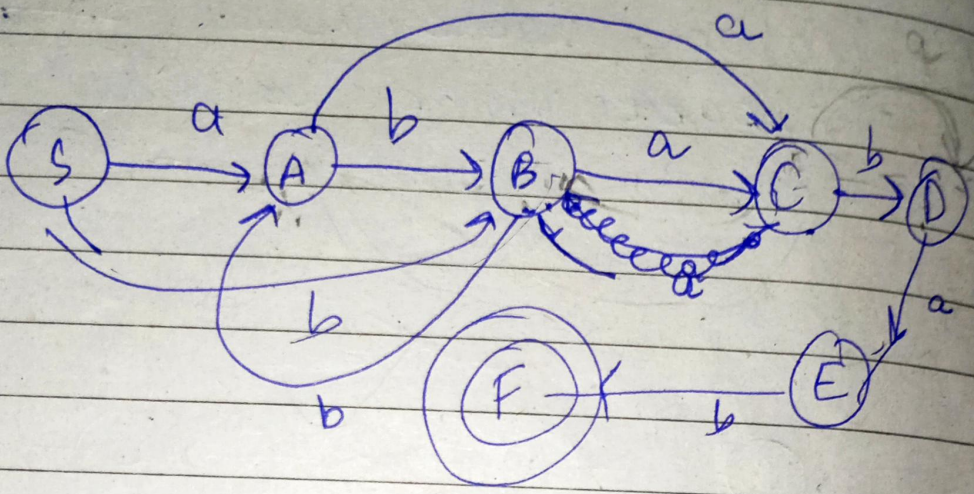


$\mathcal{L} = \{$ abab, aabab, babab, abaab, abbab, ababa, ababb, aaabab, ababab, bbabab, ababaa, ababbb $\}$

aba



aaaab

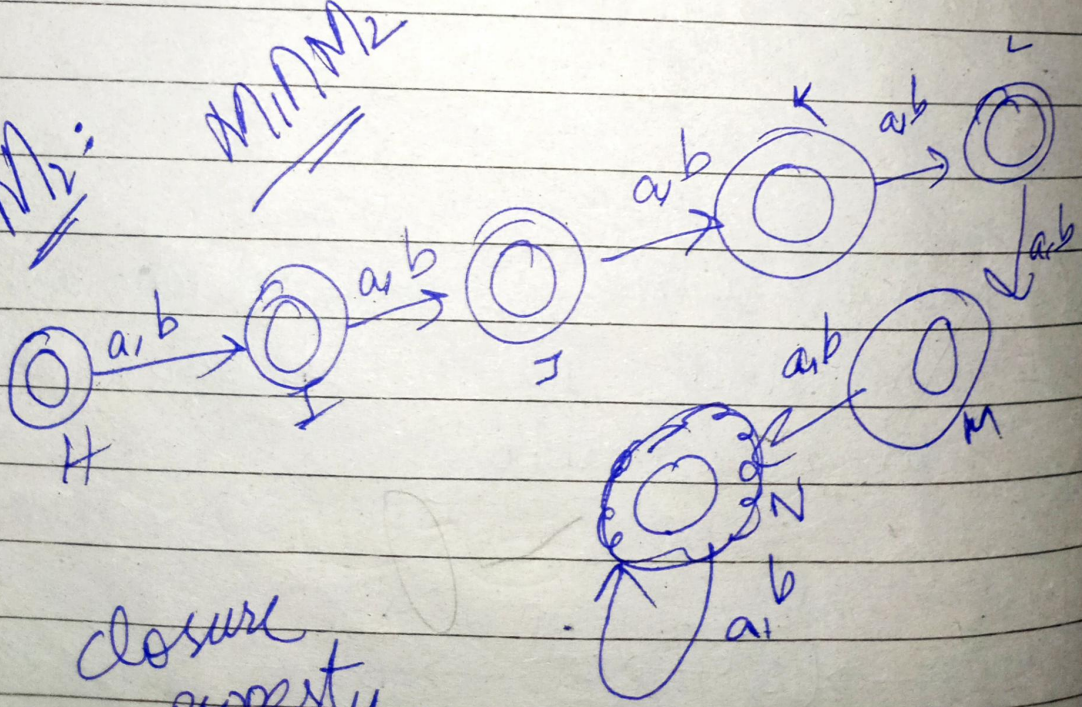


aabab

Dead State :-

M_2:

$M_1 \cap M_2$



closure property.

Date: Day:

$L_4 = \{ abab \}$

$L_5 = \{ aabab, babab, \}$

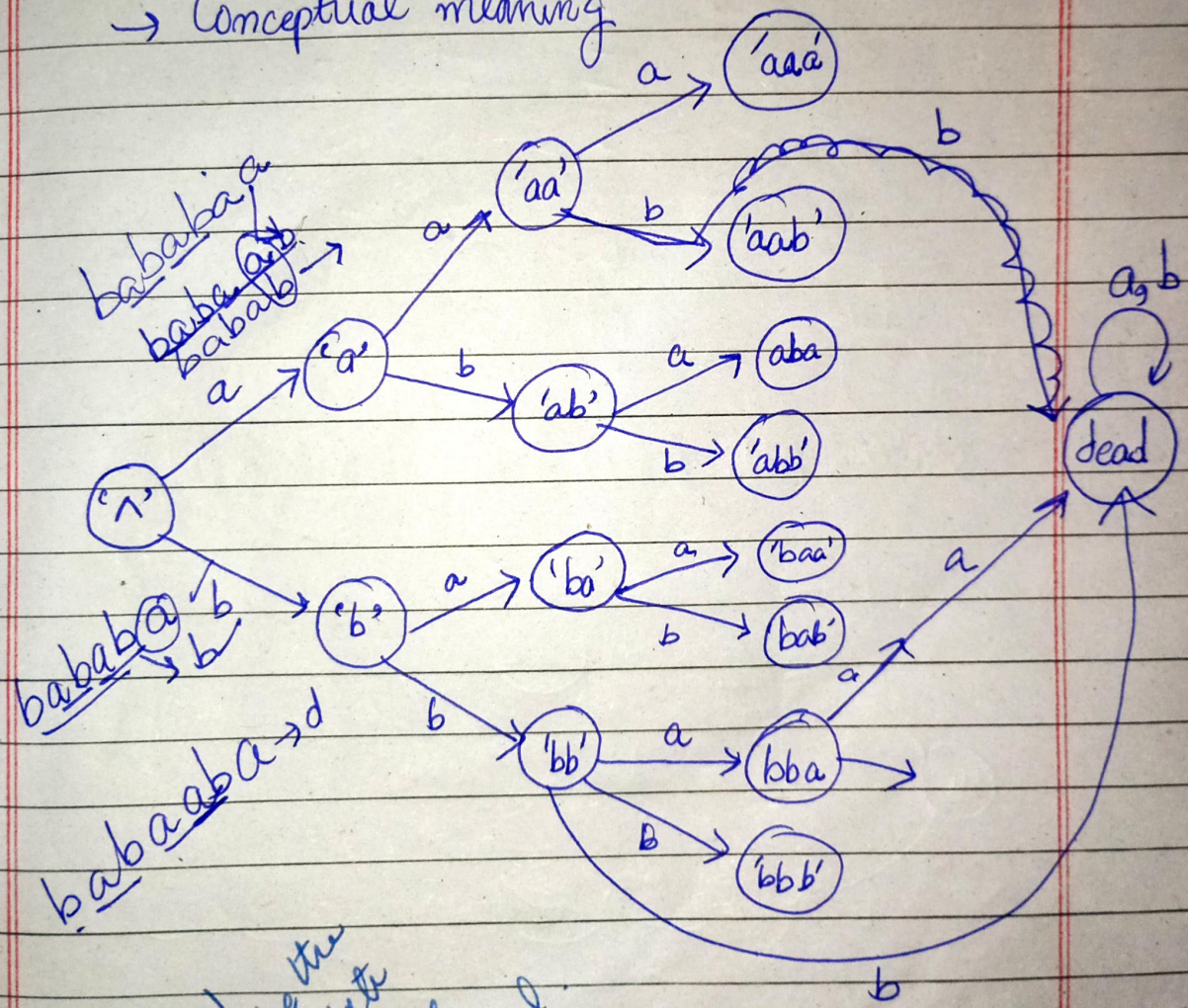
$L_6 = \{ aaabab, bbabab, ababab, baabab$
$abaaabab, abbbab, ababab, abbaab \}$

$\rightarrow \Sigma^2 abab$

$ab\Sigma^2 ab$

$abab\Sigma^2$

→ Conceptual meaning



N-1 will be the final state only one is nonfinal

Lecture # 06

Date: Sept 4, 25 Day: Thurs

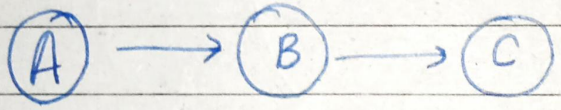
Finite Automata

DFA NFA

For NFA:

Transition function is diff.

$$\delta_{NFA} = Q * \Sigma \Rightarrow P(Q) / 2^{Q}$$



$Q = \{A, B, C\}$

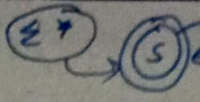
Transition can be skipped (mapped to dead state)

$P(Q) = \{ \emptyset, \{A\}, \{B\}, \{C\}, \{A,B\}, \{A,C\}, \{B,C\}, \{A,B,C\} \}$.

can be transited to multiple states.

* DFA is a subset of NFA.

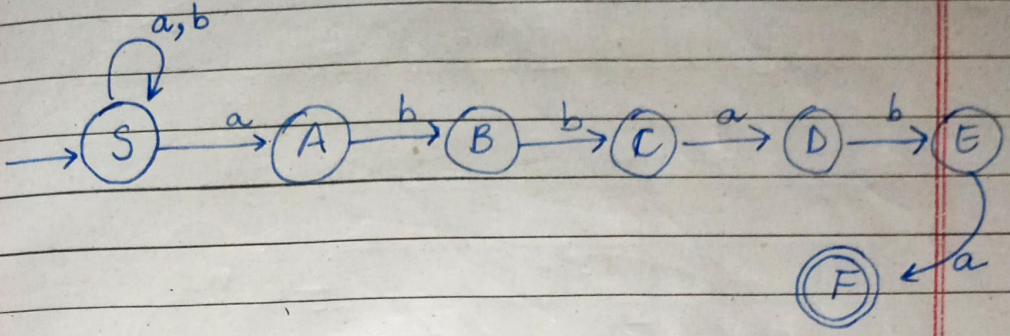
abbaba.

ex accept  a,b.

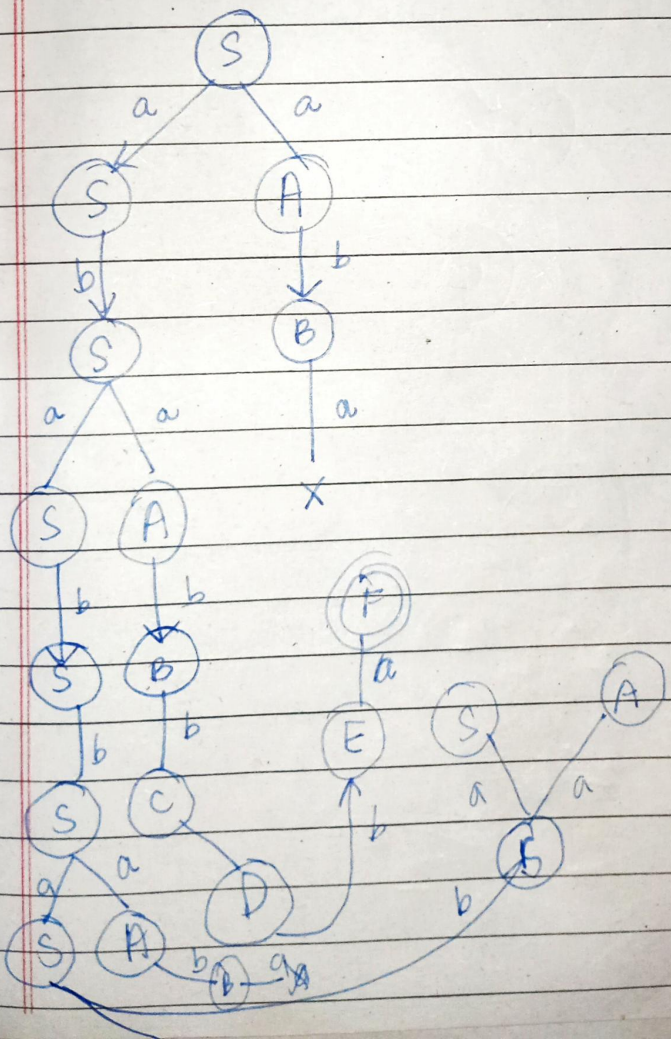
Date: Day:

strings ends with 'abbaba'

$L = \{$ abbaba, aabbaba, babbaba, ...

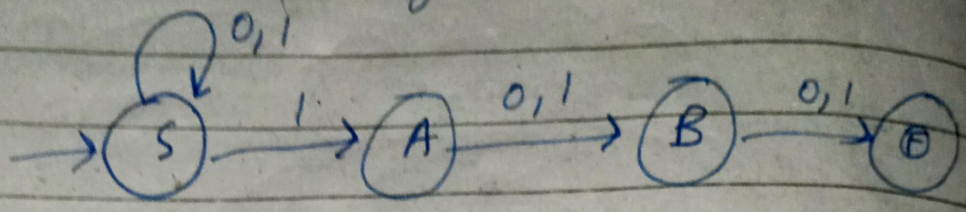


Trace Tree

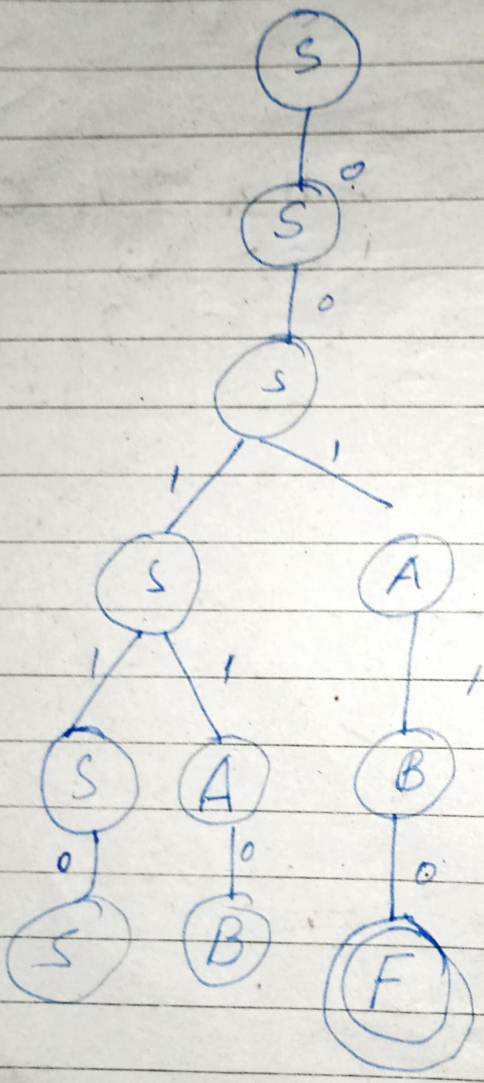


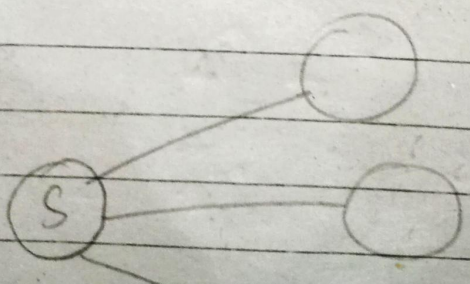
Date: Day:

Third last digit is 1



00110





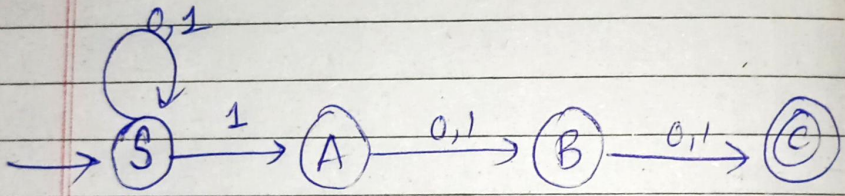
Lecture # 07

Date: Sept 9th, 25 "ToA" Tuesday

$L = \{ x \mid x \in \Sigma^* ;$ 3rd last digit in 'x' is '1' $\}$

$L = \{100, 101, 110, 111, 0100, 0101, 0110, 0111, 1100, 1101, 1110, 1111, - - - \}$

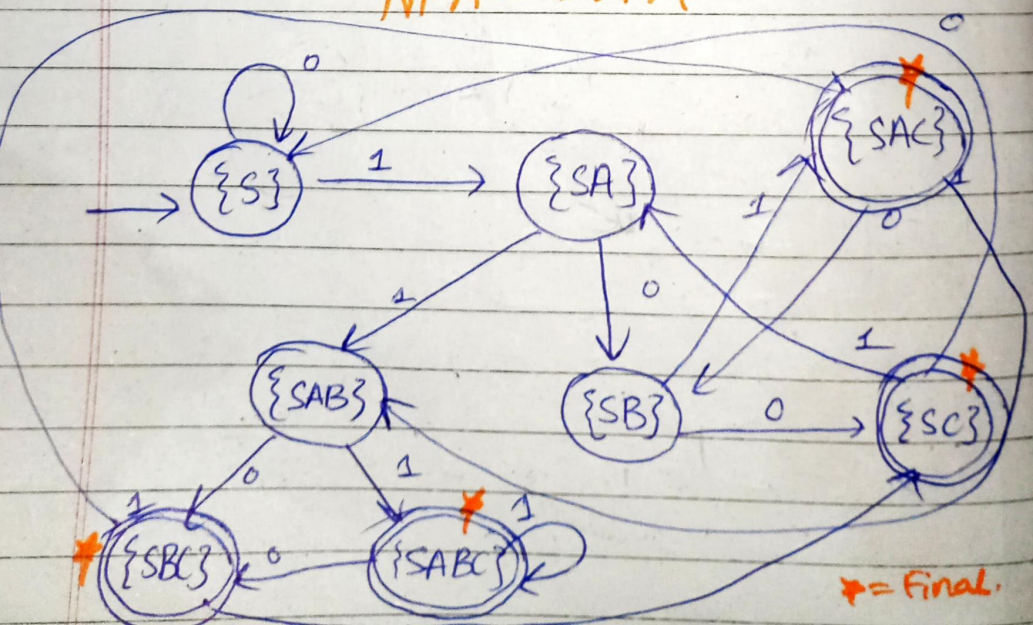
$= \Sigma^* \{ 100 \,||\, 101 \,||\, 110 \,||\, 111 \}$



(NFA)

Subset Construction Method:

NFA ⟶ DFA



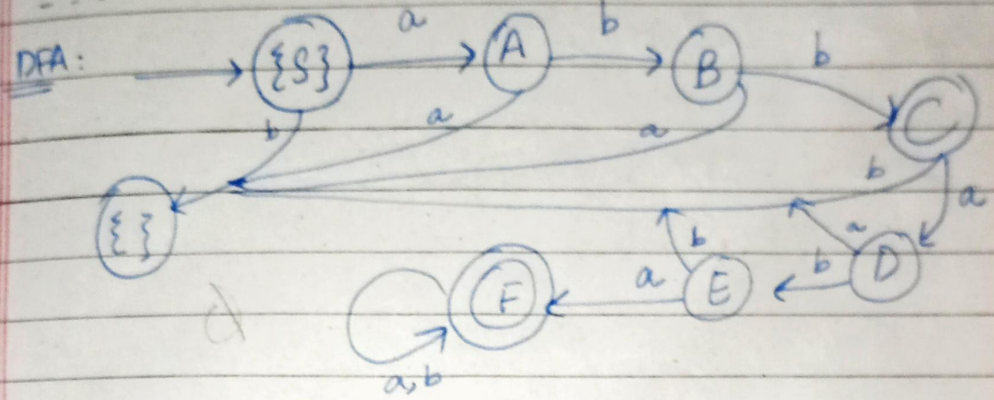
$\{a, b\}$

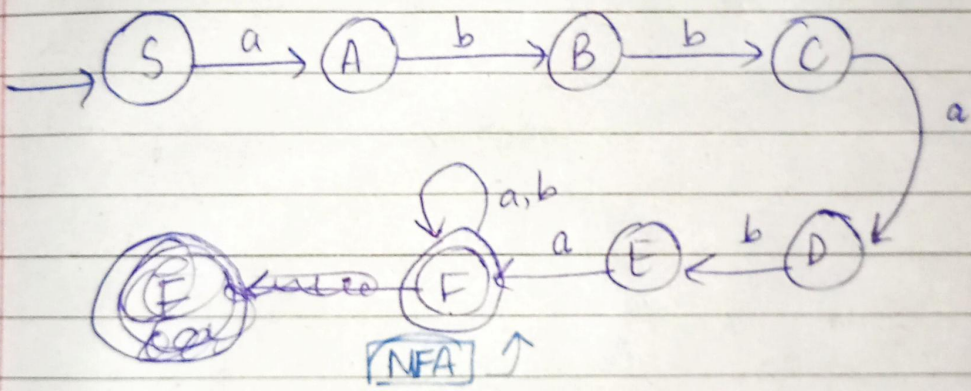
Date: Day:

Starts with 'abbaba'

abbaba Σ^*

$L = \{$ abbaba, abbabaa, abbabab, ... $\}$





we cannot extract pattern for negation ones.... only

SO we will first make NFA for the patterned one ↪ make DFA and thennn apply NEGATION ∨.

Date: Day:

abbbbc

abc

NFA-$\wedge$; NFA-NULL

$\delta_{NFA-\wedge} : Q * (\Sigma \cup \wedge) \to P(Q)$

$a^1 b^2 c^3$ abbccc

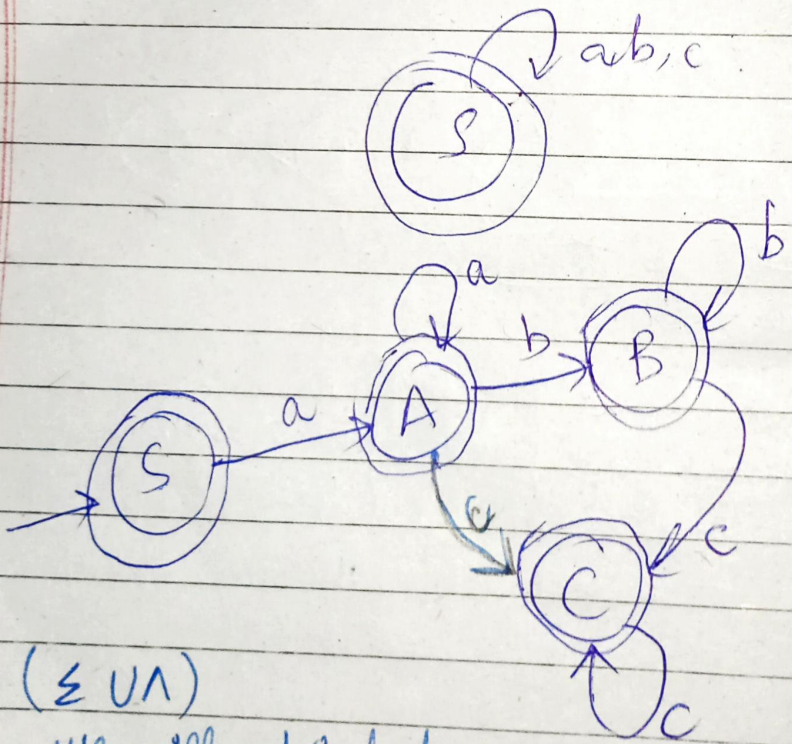
$a^0 b^0 c^0$ ($\wedge$)

$L = \{ a^* b^* c^* \}$ aa, bb, cc, bc,

$\wedge$, a, b, c, ab, aa, ac, bb,

$L = \{ abc, aabc, \}$

$\wedge$, a, b, c, ab, bc, ac, aa, bb, cc,
abc, aabc, abbc, abcc,

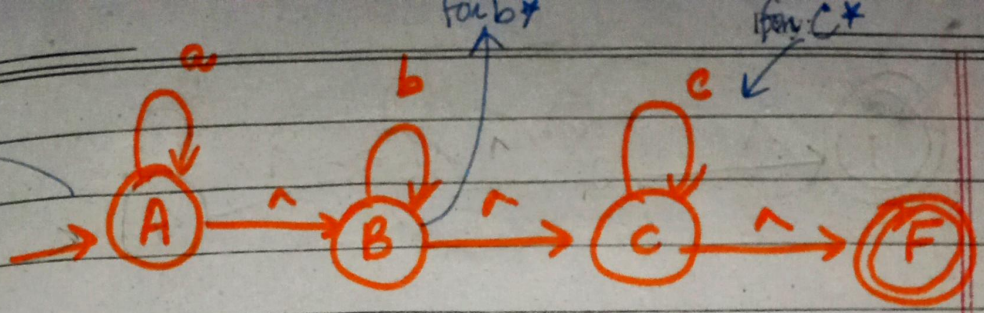


$(\Sigma \cup \wedge)$

we will not include '$\wedge$' in 'Σ' as 'Σ' it is the set of alphabets having not real character

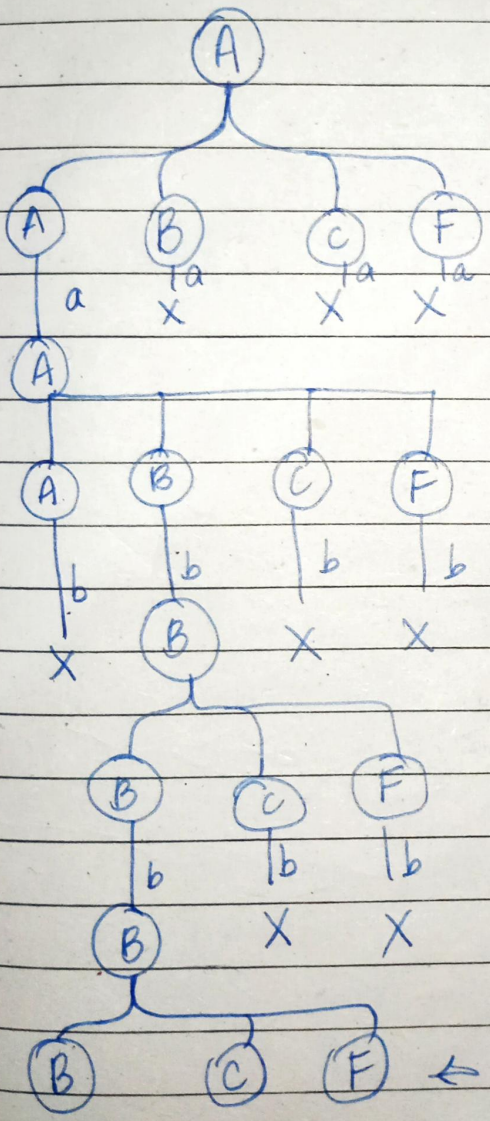
'$\wedge$' used to concatenate Two diff machines.

Date:



Here : ^ : without reading any charactes we can move to other / next state.

Trace Tree / Computational Tree



^-closure then
Σ then
^-closure

string 'abb'

First take ^-closure then check alphabet then again closure and so-on... If the leaf of the tree has any state belongs to f_n its accepted.....
F belong to f_n

Date: Day:

Extended Transition Funct

$$\delta^*(A, abb) = ?$$

$$\delta^*(A, \wedge) = \wedge(A) = \{A, B, C, F\}$$

$$\delta^*(A, a) = \wedge\Big(\bigcup \big(\delta(k, a) \mid k \in \{A, B, C, F\}\big) \delta^*(A, \wedge)\Big)$$

$$= \wedge\big(\delta(A, a) \cup \delta(B, a) \cup \delta(C, a) \cup \delta(F, a)\big)$$

$$= \wedge(\{A\} \cup \{\} \cup \{\} \cup \{\})$$

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

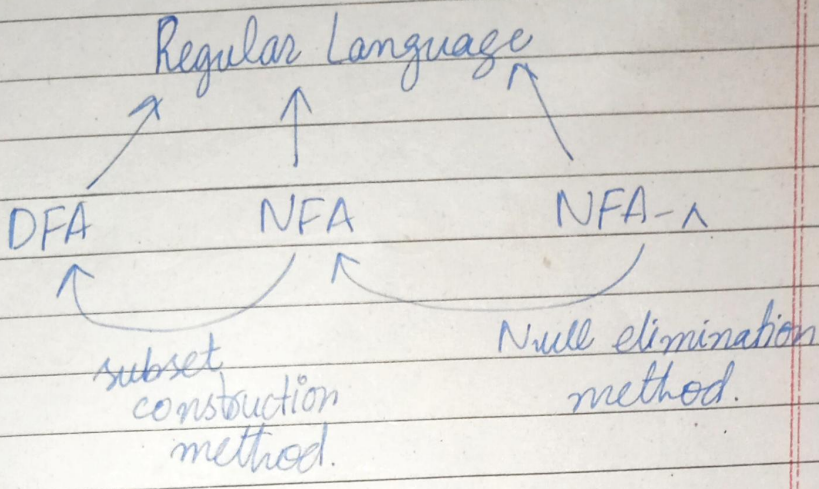
$$\delta(a, b) = \{A, B, C, F\}$$

Lecture #08

Day:

Date:

Topic : Null Elimination Method
NFA-∧ to NFA conversion

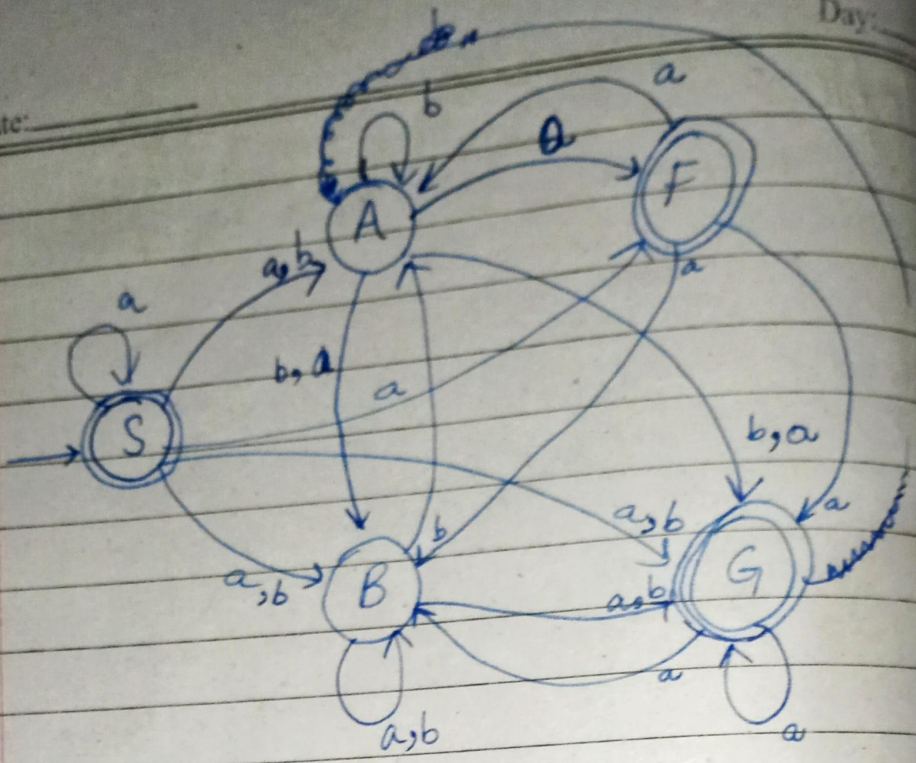


No. of state doesn't change ....

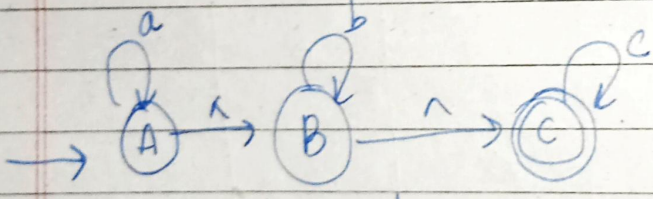
	a	b	∧
→ S	{S}	{B}	{A}
A	{F}	{-}	{B}
B	{ }	{A}	{G}
G	{B}	{ }	{ }
+ F	{F, G}	{ }	{ }

null transition

∧(S) = {S, A, B, G}
∧(A) = {A, B, G}
∧(B) = {B, G}.
∧(F) = {F}



if we can back track to 'S' from any of the final state using 'λ' then state will also be final.



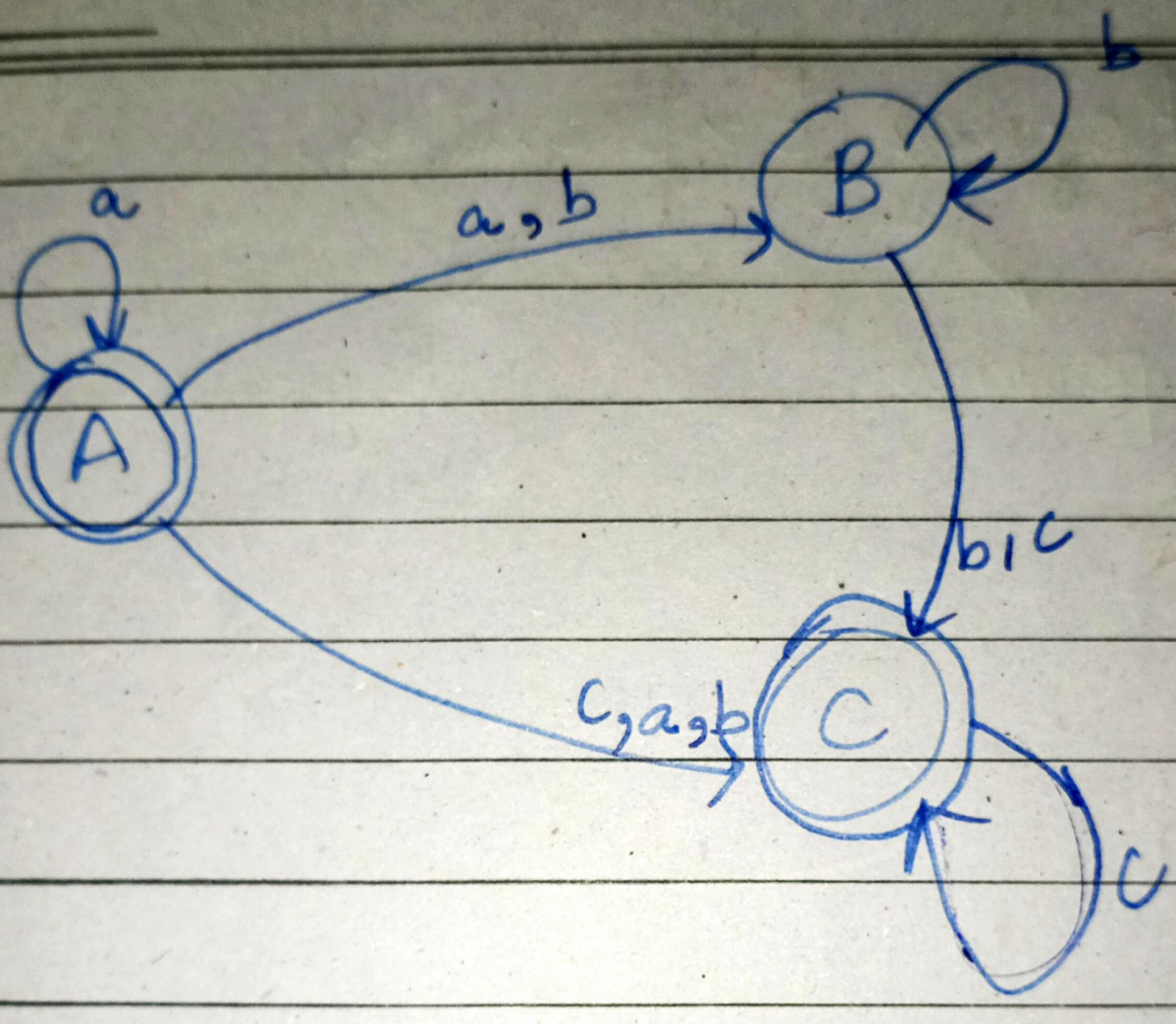
	a	b	c	λ
A	{A}	{ }	{B, C}	{B, C}
B	{ }	{B}	{C}	{C}
C	{ }	{ }	{C}	{ }

$\lambda(A) = \{A, B, C\}$

$\lambda(B) = \{B, C\}$

$\lambda(C) = \{C\}$

Day:



Lecture # 09

Date: 11 Sept 2025 Day: Thursday

Topic: Regular Expressions

- Regular operants: 'Σ'
- Regular operators: Union (+), Contenation, Klene Star (*), +ive star (+), paranthesis

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

union = $a + b$

contatenation = $a \cdot b$

Klene star = Σ^* (min = 0)

positive star = Σ^+ (min = 1)

paranthesis = $(a + b)$

Smallest regular expression = minimal

$L_1 = \{a\} \Rightarrow R.E = a$

$L_2 = \{b\} \Rightarrow R.E = b$

$L = L_1 \cup L_2 \Rightarrow a + b$

$L = L_1 \cdot L_2 \Rightarrow ab$

$L = L_1^* \Rightarrow a^*$

$L = L_2^* L_1 \Rightarrow b^* a$

$L = L_1 \cdot L_2^+ + L_1$

Precedence:
1. Stars
2. Concatenation
3. Union

Date: Day:

$L = ((L_1 (L_2^* + L_1)$

$L = ab^+ + a$

$L = \{a, ab, abb, abbb, abbbb, \ldots\ldots\}$

$L = \{aa, ab, ba, bb\}$

$R.E = \wedge \cdot (a+b) \cdot (a+b)$

$= (a+b)^2$

$L = \{a, b, aa, ab, ba, bb\}$

$= (a+b)(a+b+\wedge)$

$L = \{\wedge, a, b, aa, ab, ba, bb\}$ $a\wedge = a$

$= (a+b+\wedge)(a+b+\wedge)$

$L = \{x \mid x \in \Sigma^*$; a is followed by exactly 3 b's$\}$

~~$L = \{abbb, abbbbb, abbbbbbb$~~

$L = \{\wedge, b, bb, bbb, abbb, bbbb, \cdots\}$

$= b^* \cdot (abbb)^*$

→ make the

Date: Day

$L = \{x \mid x \in \Sigma^* \text{ . } x \text{ is not divisible by } 2\}$ $\Sigma = \{0,1\}$

State Elimination method

DFA → RFA

→ Arden Theorem
→ Elimination method.

$L = \{1, 01, 11, 001, 011, 111, \ldots\}$