# Assignment 2

Saturday, 31 August 2019 5:34 PM

Tuhin Sarkar
AP17110010068
CSE B

## ASSIGNMENT 02 (FLAT)

① Let numbers between 0 and 1 are countable

$$f : \mathbb{N} \to (0,1]$$

$f(1) = 0.a_{11}, 0.a_{12}, 0.a_{13} \dots$
$f(2) = 0.a_{21}, 0.a_{22}, 0.a_{23} \dots$
$\vdots$

$f(k) = 0.(a_{11}+1), 0.(a_{22}+1) \dots 0.(a_{kk}+1)$

where $a_{ii} \in \{1,2,3\dots\}$

But $f(k) = 0.a_{kk}$ as the diagonal trend shows.

This shows that our assumption is flawed and hence numbers between 0 and 1 are uncountable
$|(0,1]| = |\mathbb{R}|$

② Assume $P(\Sigma^*)$ is countable

$\therefore$ $L_1 = S_{11}\ S_{12}\ S_{13} \dots$
$L_2 = S_{21}\ S_{22}\ S_{23} \dots$
$L_3 = S_{31}\ S_{32}\ S_{33} \dots$
$\left\{ \begin{matrix} S \in \Sigma^* \\ L \in P(\Sigma^*) \end{matrix} \right\}$

Now, for some $w = 1100101$
$\overline{w} = 0011010$ (reversal)

Now let us take some
$L_K = \{ \overline{S_{11}}\ \overline{S_{22}} \dots \}$ [as $P(\Sigma^*)$ is countable]

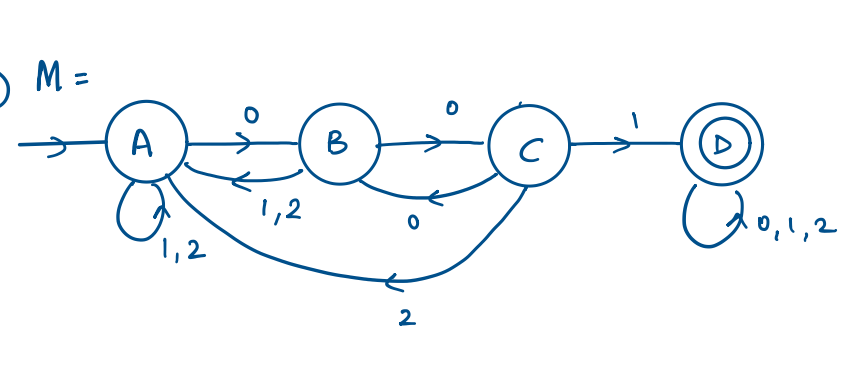
so, at a point,

$S_{kk} = \overline{S_{kk}}$

but a string cannot be reversal of itself. This ambiguity / contradiction occurs due to our wrong assumption ie $P(\Sigma^*)$ is countable

$\therefore P(\Sigma^*)$ is uncountable

③ M =



$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{A, B, C, D\}$
$\Sigma = \{0,1,2\}$
$q_0 = A$
$F = \{D\}$

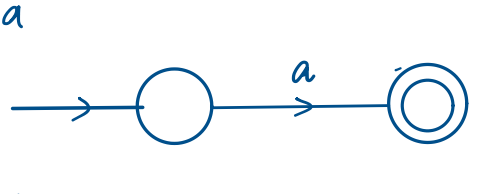
④ Let there be 2 machines that recognize $L_1$ and $L_2$ respectively.

$M(L_1) = (Q_1, \Sigma, \delta_1, q_1, F_1)$
$M(L_2) = (Q_2, \Sigma, \delta_2, q_2, F_2)$

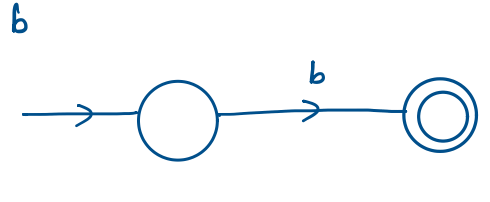
Let us device a machine M' such that it recognizes a machine
$M = (Q, \Sigma, \delta, q_0, F)$
$Q = Q_1 \times Q_2$
$\delta : (Q_1 \times Q_2) \times \Sigma \to (Q_1 \times Q_2)$
$\delta(r_1, r_2, \sigma) = (\delta_1(r_1, \sigma), \delta_2(r_2, \sigma))$
$q_0 = (q_1, q_2)$
$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ and } r_2 \in F_2\}$
$= (F_1 \times Q_1) \cup (Q \times F_2)$

The machine now obtained:
$M = (Q, \Sigma, \delta, q_0, F)$ will recognize $L_1 \cap L_2$

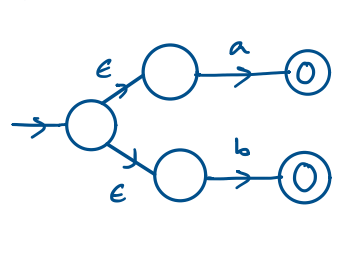
⑤ a



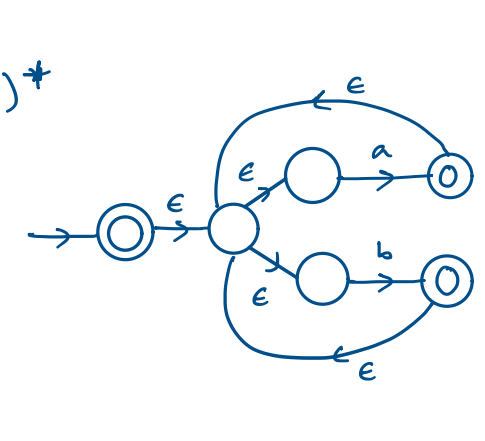
b



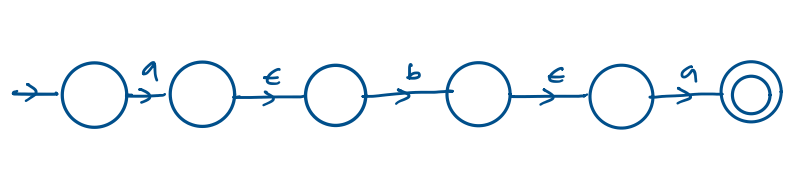
a ∪ b



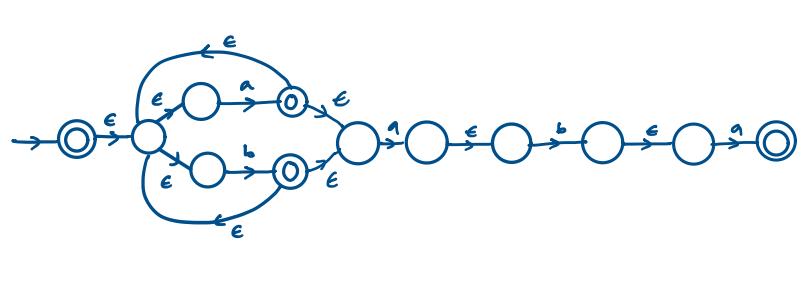
(a ∪ b)*



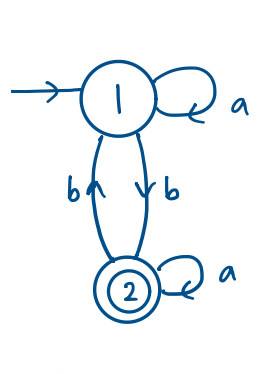
aba



(a ∪ b)* aba



⑥ a) M:

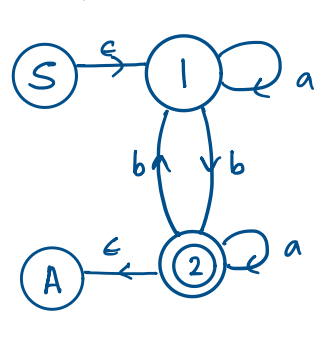


$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{1,2\}$
$\Sigma = \{a,b\}$
$q_0 = 1$
$F = \{2\}$

| δ | a | b |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 2 | 1 |

Equivalent GNFA:

G =



| δ' | 1 | 2 | A |
|---|---|---|---|
| S | ε | ∅ | ∅ |
| 1 | a | b | ∅ |
| 2 | b | a | ε |

$q_{rip} = 1$

$\delta(q_i, q_j) = \delta(q_i, q_{rip}) . \delta(q_{rip}, q_{rip})^* . \delta(q_j, q_{rip}) \cup \delta(q_i, q_j)$

$\delta(S,2) = \delta(S,1) . \delta(1,1)^* \delta(1,2) \cup \delta(S,2)$
$= \varepsilon a^* b \cup \emptyset$
$= a^* b$

$\delta(S,A) = \delta(S,1) . \delta(1,1)^* \delta(1,A) \cup \delta(S,A)$
$= \varepsilon a^* \emptyset \cup \emptyset$
$= \emptyset$

$\delta(2,2) = \delta(2,1) . \delta(1,1)^* \delta(1,2) \cup \delta(2,2)$
$= b a^* b \cup a$

$\delta(2,A) = \delta(2,1) . \delta(1,1)^* \delta(1,A) \cup \delta(2,A)$
$= b a^* \emptyset \cup \varepsilon$
$= \varepsilon$

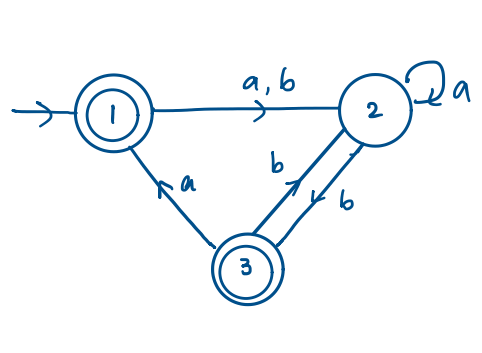
| δ' | 2 | A |
|---|---|---|
| S | $a^*b$ | ∅ |
| 2 | $ba^*b \cup a$ | ε |

$q_{rip} = 2$

$\delta(S,A) = \delta(S,2) . \delta(2,2)^* \delta(2,A) \cup \delta(S,A)$
$= a^* b (b a^* b \cup a)^* \varepsilon \cup \emptyset$
$= a^* b (b a^* b \cup a)^*$

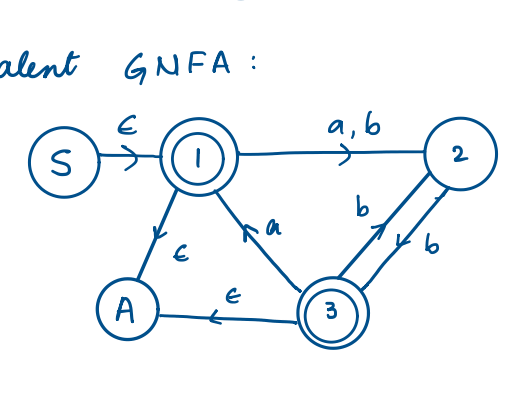
Equivalent regular expression:
$a^* b (b a^* b \cup a)^*$

b)



Equivalent GNFA:

G :



$G = (Q, \Sigma, \delta', q_0', F)$

| δ | 1 | 2 | 3 | A |
|---|---|---|---|---|
| S | ε | ∅ | ∅ | ∅ |
| 1 | ∅ | a∪b | ∅ | ε |
| 2 | ∅ | a | b | ∅ |
| 3 | a | b | ∅ | ε |

$\delta(q_i, q_j) = \delta(q_i, q_{rip}) . \delta(q_{rip}, q_{rip})^* . \delta(q_j, q_{rip}) \cup \delta(q_i, q_j)$

$q_{rip} = 1$

$\delta(S,2) = \delta(S,1) \delta(1,1)^* \delta(1,2) \cup \delta(S,2)$
$= \varepsilon \emptyset^* a \cup b \cup \emptyset$
$= a \cup b$

$\delta(S,3) = \delta(S,1) \delta(1,1)^* \delta(1,3) \cup \delta(S,3)$
$= \varepsilon \emptyset^* \emptyset \cup \emptyset$
$= \emptyset$

$\delta(S,A) = \delta(S,1) \delta(1,1)^* \delta(1,A) \cup \delta(S,A)$
$= \varepsilon \emptyset^* \varepsilon \cup \emptyset$
$= \varepsilon$

$\delta(2,2) = \delta(2,1) \delta(1,1)^* \delta(1,2) \cup \delta(2,2)$
$= \emptyset \cup a$
$= a$

$\delta(2,3) = \delta(2,1) \delta(1,1)^* \delta(1,3) \cup \delta(2,3)$
$= \emptyset \cup b$
$= b$

$\delta(2,A) = \delta(2,1) \delta(1,1)^* \delta(1,A) \cup \delta(2,A)$
$= \emptyset \cup \emptyset$
$= \emptyset$

$\delta(3,2) = \delta(3,1) \delta(1,1)^* \delta(1,2) \cup \delta(3,2)$
$= a \emptyset^* (a \cup b) \cup b$
$= a(a \cup b) \cup b$

$\delta(3,3) = \delta(3,1) \delta(1,1)^* \delta(1,3) \cup \delta(3,3)$
$= a \emptyset^* \emptyset \cup \emptyset$
$= \emptyset$

$\delta(3,A) = \delta(3,1) \delta(1,1)^* \delta(1,A) \cup \delta(3,A)$
$= a \emptyset^* \varepsilon \cup \varepsilon$
$= a \cup \varepsilon$

| δ' | 2 | 3 | A |
|---|---|---|---|
| S | a∪b | ∅ | ε |
| 2 | a | b | ∅ |
| 3 | a(a∪b)∪b | ∅ | a∪ε |

$q_{rip} = 2$

$\delta(S,3) = \delta(S,2) \delta(2,2)^* \delta(2,3) \cup \delta(S,3)$
$= (a \cup b) a^* b \cup \emptyset$
$= (a \cup b) a^* b$

$\delta(S,A) = \delta(S,2) \delta(2,2)^* \delta(2,A) \cup \delta(S,A)$
$= (a \cup b) a^* \emptyset \cup \varepsilon$
$= \varepsilon$

$\delta(3,3) = \delta(3,2) \delta(2,2)^* \delta(2,3) \cup \delta(3,3)$
$= a(a \cup b) \cup b\ a^* \emptyset \cup \emptyset$
$= \emptyset$

$\delta(3,A) = \delta(3,2) \delta(2,2)^* \delta(2,A) \cup \delta(3,A)$
$= a(a \cup b) \cup b\ a^* \emptyset \cup (a \cup \varepsilon)$
$= a \cup \varepsilon$

| δ | 3 | A |
|---|---|---|
| S | $(a\cup b)a^*b$ | ε |
| 3 | ∅ | a∪ε |

$q_{rip} = 3$

$\delta(S,A) = \delta(S,3) \delta(3,3)^* \delta(3,A) \cup \delta(S,A)$
$= (a \cup b) a^* b \emptyset^* (a \cup \varepsilon) \cup \varepsilon$
$= (a \cup b) a^* b (a \cup \varepsilon) \cup \varepsilon$

The corresponding reg. ex. is:
$= (a \cup b) a^* b (a \cup \varepsilon) \cup \varepsilon$

⑦ b) $L = \{0^n 1^n \mid n \geq 0\}$

Assume $L \in REG$
also assume $M(L)$ has $K$-states

Now, choose $x \in L$ st $|x| \geq K$

$x = 0^p 1^p$ st $2p \geq K$

$x = 0^p 1^p = uvw$

Case 1 : v only has 0's :
$x = uvw$
$x' = uvvw = uv^2w \notin L$ as $\#1's \neq \#0's$

Case 2 : v only has 1's :
$x = uvw$
$x' = uvvw = uv^2w \notin L$ as $\#1's \neq \#0's$

Case 3 : v has 0's and 1's
$v = 0^m 1^n$
$x = 0^p 1^p = 0^{p-m} 0^m 1^n 1^{p-n}$
$x' = uvvw = 0^{p-m} 0^m 1^n 0^m 1^n 1^{p-n} \notin L$

Therefore, L is not a regular language

c) $L = \{ww \mid w \text{ is a string } \{0,1\}\}$

P.T. $L \notin REG$

Assume $L \in REG$
Assume $M(L)$ has $K$ states

choose $x \in L$ st $|x| \geq K$
$x = 0^p 1 0^p 1$ st $2p + 2 \geq K$
$x = 0^p 1 0^p 1 = uvw$
$uv^i w \in L \ \forall i \geq 0$

Case 1 : v has only 0s
$v = 0^m$
$x = 0^p 1 0^p 1 = 0^p 1 0^{p-m} 0^m 1$
$x' = uvvw = 0^p 1 0^{p-m} 0^m 0^m 1$
$= 0^p 1 0^{p+m} 1$

Not symmetric, $\therefore x' \notin L$

similarly for Case 2 : v has only 1's

Case 3 : $v = 0^m 1 0^n$

$x = 0^p 1 0^p 1 = uvw = 0^{p-m} 0^m 1 0^n 0^{p-n} 1$
$x' = uvvw = 0^{p-m} 0^m 1 0^n 0^m 1 0^n 0^{p-n} 1$
$= 0^p 1 0^{m+n} 1 0^p 1$

Not symmetric, $\therefore x' \notin L$

Therefore, L is not regular

e) $L = \{a^{n^2} \mid n \geq 0\} \Rightarrow L_4 \notin Reg$

Assume $L \in REG$
also assume $M(L)$ has $K$-states

Now, choose $x \in L$ st $|x| \geq K$

$x = uvw$ st $|uv| \leq K$

$v \neq \varepsilon$

$\Rightarrow x' = uv^iw \in L_4 \ \forall i \geq 0$

$|x'| = |uvvw| = k^2 + |v| \leq k^2 + k < k^2 + 2k < k^2 + 2k + 1 = (k+1)^2$

$|x'| = |uvvw| < (k+1)^2$
$|x| = |uvw| = k^2$

There are no perfect squares betn 2 perfect squares. Hence, not regular.

⑧ PT $|\Sigma^*|$ is countable

Let us consider $\Sigma = \{0,1\}$

To prove that $\Sigma^*$ is countable, we have to define an onto fn

$$f : \mathbb{N} \to \Sigma^*$$

The strings in $\Sigma^*$ will be:

0, 1, 00, 01, 10, 11, 000 …

We can write them as:

$f(1) = 0$
$f(2) = 1$
$\vdots$
$f(n) = w \mid w \in \Sigma^*$

For a fixed numbers of digits, there are finite number of strings; this means it follows a one-one and onto relationship.

$\Rightarrow |\Sigma^*| = |\mathbb{N}|$
$\therefore |\Sigma^*|$ is countable

⑨ In case of a finite state machine, the memory (the number of states of FSA) is finite and hence not variable. ∴ For k states, if a string of k+1 length arrives, it will have no memory of the visit by the progression of the string in the FSA. That is it will atleast visit one state atleast twice (by pigeonhole principle), and the FSA will never remember its first visit. This becomes a problem that requires strings to be variable, as to accept it properly while remembering its progression will require states of FSA to be greater or equal to the length of string. Since, the string is variable, the states have to variable too (variable memory). Hence such problems cannot be solved by finite automata.
eg: $L = \{0^n 1^n \mid n > 0\}$

⑩ A grammar G is defined as a 4-tuple $(\Sigma, V, S, P)$

Σ : a finite set of symbols, the members of Σ are called terminal symbols

V : a finite set of symbols distinct from Σ the members of V are called variables (or) non terminal symbols

S : $S \in V$, start variable

P : the set of production values

A production rule is a mapping of
$(\Sigma \cup V)^* V (\Sigma \cup V)^* \to (\Sigma \cup V)^*$

$G_1 = (\{0,1\}, \{S, T, O, I\}, S, P)$

with P = $S \to OT$
$S \to OI$
$T \to SI$
$O \to 0$ (zero)
$I \to 1$

What is the language generated by $G_1$?

$S \to OT \to 0T \to 0SI \to 0S1 \to 0OI1 \to 0011$

$S \to OI \to 01$

$S \to OT \to 0SI \to 0OTI \to 0OSI1 \to 00S11 \to 00OI11 \to 000111$

$0011 \in L(G_1)$
$000111 \in L(G_1)$
$01 \in L(G_1)$

$L(G_1) = \{0^n 1^n \mid n \geq 1\}$

⑪ Types of grammar:

Type - 0) Grammar
No assumptions on the prod. rules in P. Also, called unrestricted grammar.

(Type - 1) Grammar
$|\alpha| \leq |\beta|$, this assumption is made. Also called length increasing Grammar, Context - sensitive Grammar.

(Type - 2) Grammar
Type 1 grammars with the assumption $|\alpha| = 1$. Also called Context - free grammar

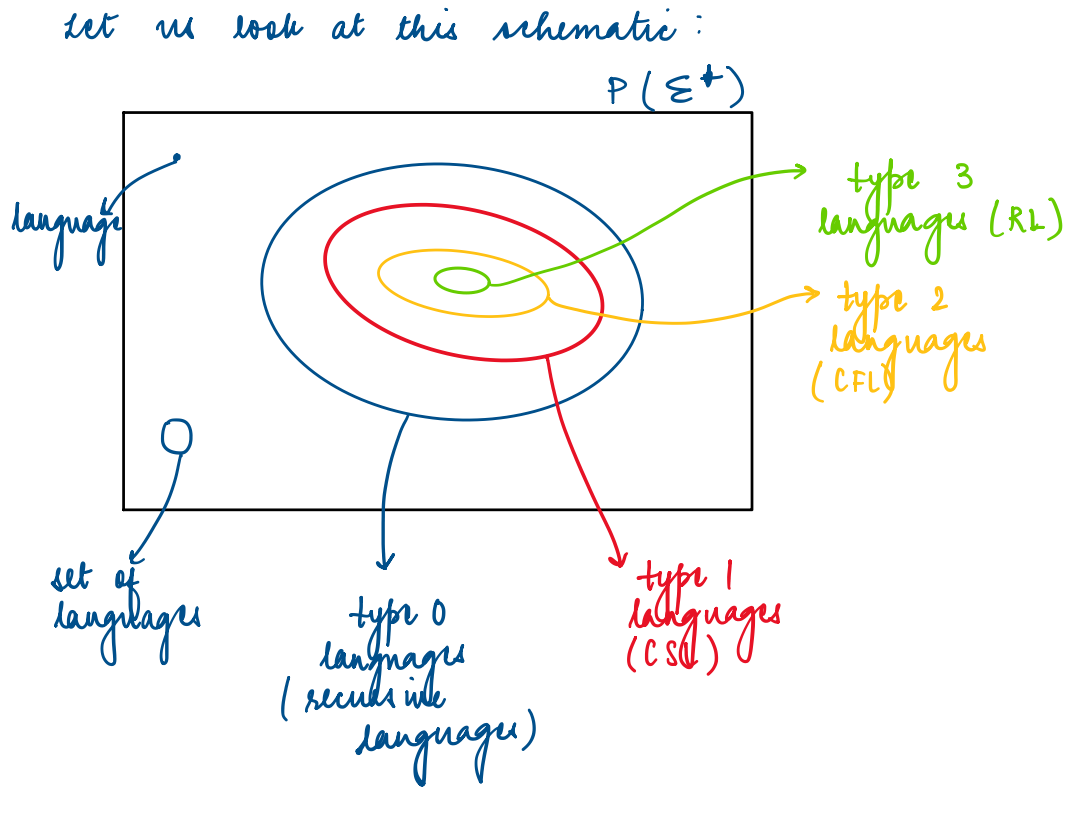
(Type - 3) Grammar
$A \to cB$
$A \to c$ where $c \in \Sigma$ [Therefore called left linear grammar]
$A \to \varepsilon$

or

$A \to Bc$
$A \to c$ where $c \in \Sigma$ [Therefore called right linear grammar]
$A \to \varepsilon$

These are called Regular Grammar

Let us look at this schematic:



* This hierarchy is called the Chomsky hierarchy.

⑫

$S \to ASB$
$A \to aAS \mid a \mid \varepsilon$
$B \to SbS \mid A \mid bb$

Add a state $S_0$, s.t $S_0 \to S$

$S \to ASB$
$A \to aAS \mid a \mid \varepsilon$
$B \to SbS \mid A \mid bb$
$S_0 \to S$

Now, null removal for
(i) $A \to \varepsilon$

$S \to ASB \mid SB$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid \varepsilon \mid bb \mid A$
$S_0 \to S$

(ii) $B \to \varepsilon$

$S \to ASB \mid SB \mid AS \mid S$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid bb \mid A$
$S_0 \to S$

Now, unit removal for
(i) $S \to S$

$S \to ASB \mid SB \mid AS$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid bb \mid A$
$S_0 \to S$

(ii) $S_0 \to S$

$S \to ASB \mid SB \mid AS$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid bb \mid A$
$S_0 \to ASB \mid SB \mid AS$

(iii) $B \to A$

$S \to ASB \mid SB \mid AS$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid bb \mid aS \mid aAS \mid a$
$S_0 \to ASB \mid SB \mid AS$

Now, we remove more than 2 variables

$X \to SB$
$S \to AX \mid SB \mid AS$
$A \to aS \mid aAS \mid a$
$B \to SbS \mid bb \mid aS \mid aAS \mid a$
$S_0 \to AX \mid SB \mid AS$

Now, we remove combinations of terminating and non terminating states

$Y \to a$
$Z \to b$
$S \to AX \mid SB \mid AS$
$A \to YS \mid YAS \mid a$
$B \to SZS \mid ZZ \mid YS \mid YAS \mid a$
$S_0 \to AX \mid SB \mid AS$

Again we remove more than 2 variables:

$P \to YA$
$Q \to SZ$
$Y \to a$
$Z \to b$
$S \to AX \mid SB \mid AS$
$A \to YS \mid PS \mid a$
$B \to QS \mid ZZ \mid YS \mid PS \mid a$
$S_0 \to AX \mid SB \mid AS$

This is the Chomsky Normal Form for this grammar.

⑬ When a string can be produced from grammar in multiple or more than one way is called an ambiguous string. A grammar that produces such strings is called an ambiguous grammar.

Eg : $G : (\{a,b\}, \{A,B\}, S, P)$

P : $S \to a \mid aAb \mid abSb$
$A \to aAAb \mid bS$

$S \to abSb \to abab$ or $S \to aAb \to abSb \to abab$

⑭ a) R, S, T, X are variables here.

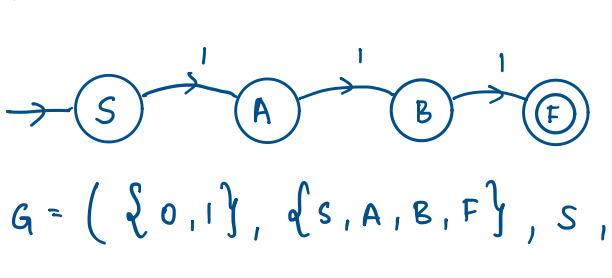
f) True

$T \to XTX \to aTX \to aXX \to abX \to aba$

m) True

$T \to XTX \to XXX$

⑮ a) NFA :

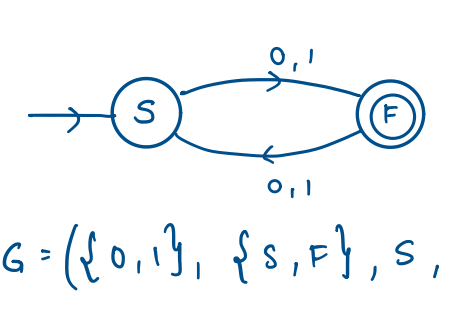


$G = (\{0,1\}, \{S, A, B, F\}, S, P)$

Therefore grammar,

$S \to 1A$
$A \to 1B$
$B \to 1F$
$F \to \varepsilon$

c) NFA :

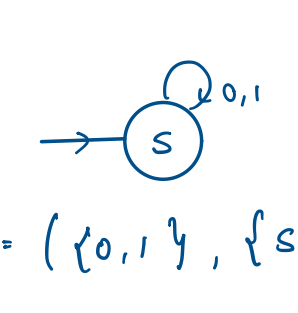


$G = (\{0,1\}, \{S, F\}, S, P)$

Therefore grammar:

$S \to 0F \mid 1F$
$F \to 0S \mid 1S \mid \varepsilon$

f)



$G = (\{0,1\}, \{S\}, S, P)$

Therefore grammar.

$S \to 0S \mid 1S$