TOC Class
Test

Subject: Theory of Computation [CS 2204]

Date: 12/04/2021

Name: Abhiroop Mukherjee

Exam Roll. No.: 510519109

G Suite ID: 510519109.abhirup@students.iiests.ac.in

No. of Pages Uploaded: 9

Q1) a) False, ~~Con~~ A context-free language over some Σ can't be a subset of some Regular Language over the same ~~alph~~ alphabet Σ.

→ We know that every Regular Language is not context free, but vice versa is true

Eg: A PDA can accept language produced by ~~FA~~ RG, but A language produced by ~~PDA~~ CFG can't be accepted by FA.

b) Given a Non-Deterministic Finite Automata

$$M = (K, \Sigma, \Delta, s, F), \quad \Delta \subseteq K \times \Sigma^* \times K$$

we create following ~~PDA~~ Push-down Automata.

$$M' = (K', \Sigma', \Gamma', \Delta', s', F')$$

$$\Delta' \subseteq K \times \Sigma^* \times \Gamma'^* \times K \times \Gamma'^*$$

with $K' = K$

$\Sigma' = \Sigma$

$\Gamma' = \{e\}$

$\Delta' = \{(p, \alpha, e, q, e) \mid (p, \alpha, q) \in \Delta\}$

$s' = s$

$F' = F$

i.e we follow same procedure as NDFA M follows, but at each step we push and pop nothing (empty string)

→ As this construction just straightaway copies the given NDFA M, we can say that

$$L(M') = L(M)$$

∴ The given statement is <u>True</u>

②

c) The given statement is true

→ given Σ_1 and Σ_2 two alphabets, $\Sigma_1 \cap \Sigma_2 = \phi$

There can't be common strings between L_1, a language over Σ_1 and L_2, language over Σ_2

proof by contradiction

→ Let's assume that there ~~are some~~ is a common strings ~~w~~ w ~~in~~ between L_1 and L_2

→ This means w has ~~some~~ the symbols common to L_1 and L_2

→ that mean

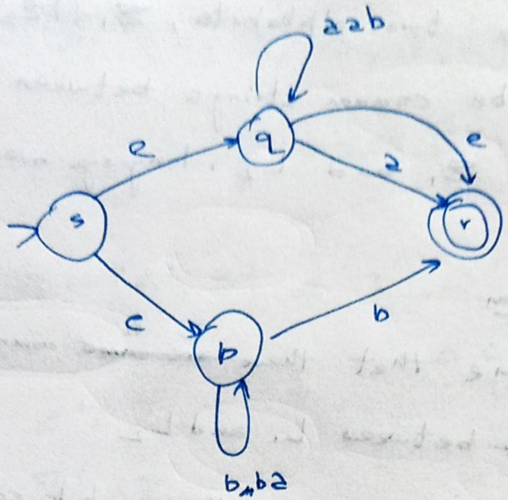
$$L_1 \cap L_2 \neq \phi$$

→ which contradicts the given statement that $L_1 \cap L_2 = \phi$

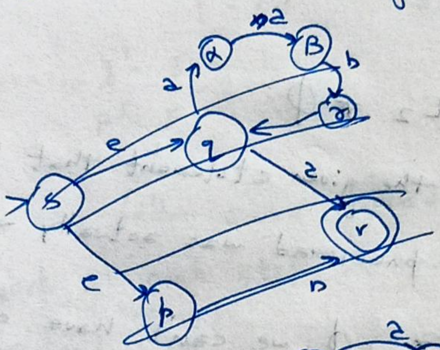
→ This means what we presumed was actually wrong.

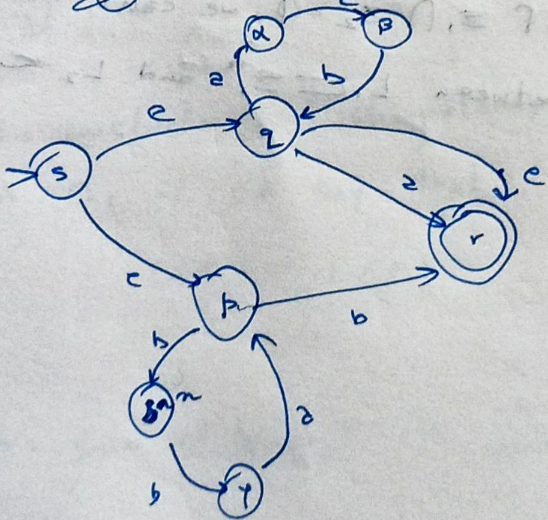
→ Therefore if $\Sigma_1 \cap \Sigma_2 = \phi$, we can't have common strings between $L_1 \subset \Sigma_1^*$ and $L_2 \subset \Sigma_2^*$

2) given Following NDFA

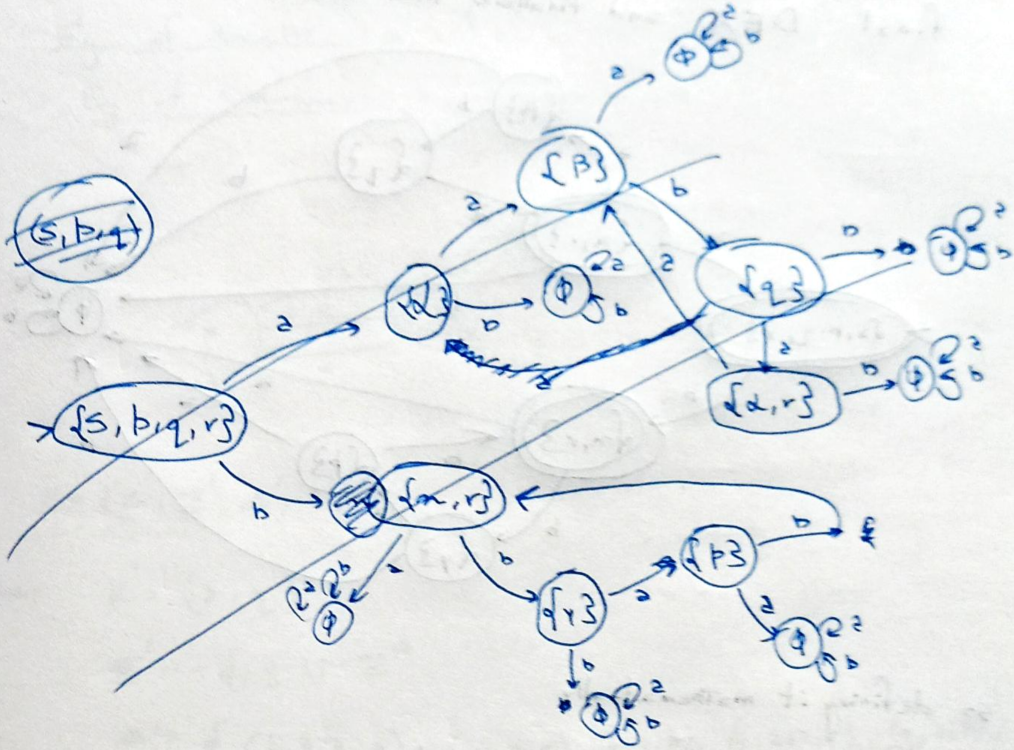


→ first we convert strings to single symbol transitions

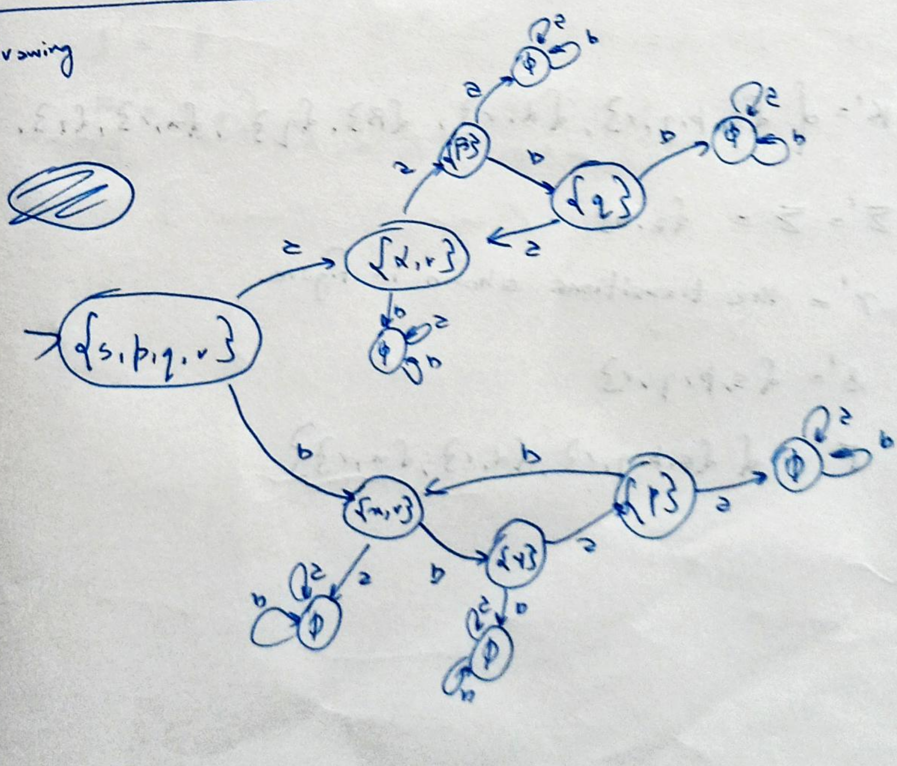




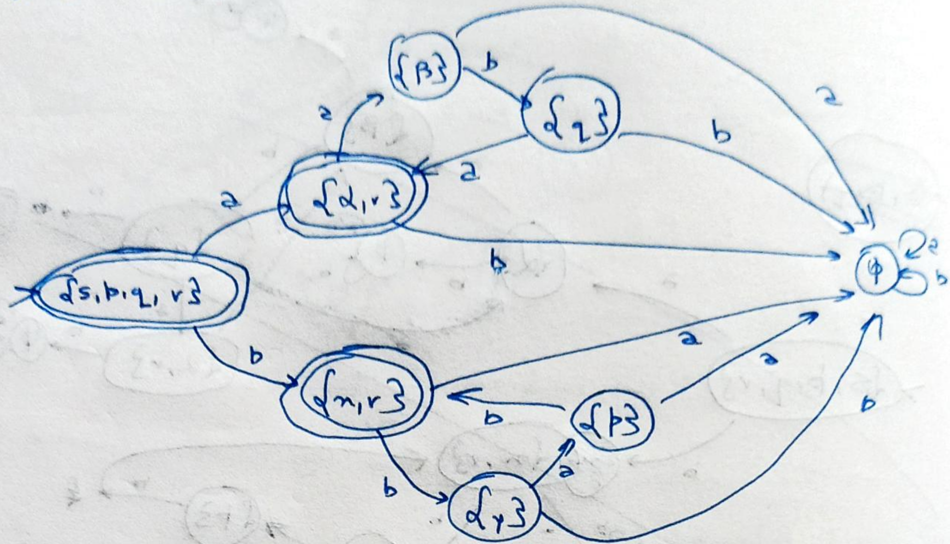
→ then we merge the empty clousures



redrawing



→ now we merge the ϕ state together to create final DFA and finalizing final states



so defining it mathematically

$$M' = (K', \Sigma', \delta', s', F')$$

$K' = \{ \{s,p,q,r\}, \{\alpha,r\}, \{\beta\}, \{q\}, \{n,r\}, \{\gamma\}, \{p\}, \phi\}$

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

δ' = the transitions shown in figure

$s' = \{s,p,q,r\}$

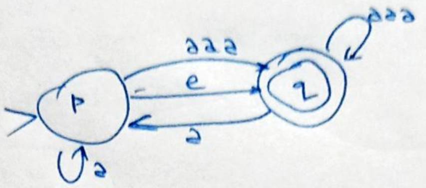
$F' = \{ \{s,p,q,r\}, \{\alpha,r\}, \{n,r\} \}$

Q3) a) $\{w \in \{a,b\}^* \mid$ The no. of a's is divisible by 3$\}$

[FA]

Eg of accept : a^0, a^3, a^6 ....

Eg of Non-accept $a, a^2, \rightarrow a^4, a^5, \ldots$



$M = (K', \Sigma', \Delta', s', F')$

where $K' = \{p, q\}$

$\Sigma' = \{e\} \cup \Sigma^*$

$\Delta' = \{(p, aaa, q), (p, e, q), (p, a, p), (q, aaa, q), (q, a, p)\}$

$s' = p$

$F' = \{q\}$

Q3) b) $L = \{ a^m b^n c^l \mid l, m, n \geq 0 \text{ and } m + l+n \}$

[CFG]

Rules

1) $S \to ABC$

2) $A \to aA$

3) $A \to e$

4) $B \to bB$

5) $B \to e$

6) $C \to cC$

7) $C \to e$

$G = \{ V, \Sigma, R, S \}$

with $V = \{ a, b, c, S, A, B, C \}$

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

$R = \{ S \to ABC, A \to aA, A \to e, B \to bB,$
$B \to e, C \to cC, C \to e \}$

$S = S$

c) $L = \{a^m b^n \mid m,n \geq 0 \text{ and } m \geq n\}$

[PDA]

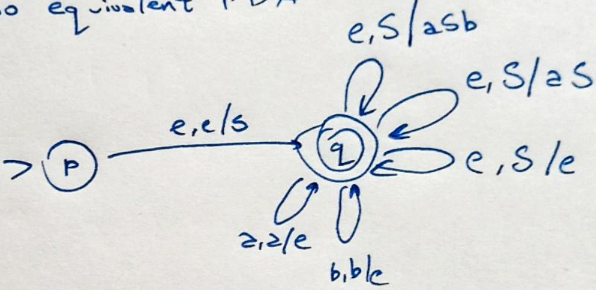
trying to make ~~RG~~ CFG

$S \to aSb$

$S \to aS$

$S \to e$

so equivalent PDA



$M = (K, \Sigma, \Gamma, \Delta, s, F\}$

$K = \{p, q\}$

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

$\Gamma = \{S, a, b\}$

$\Delta = \{(p,e,e,q,S), (q,e,S,q,aSb), (q,e,S,q,aS),$
$(q,e,S,q,e), (q,b,b,q,e), (q,a,a,q,e)\}$

$s = \{p\}$

$F = \{q\}$