Page No.:

Tut 6 Solutions

1 (a) Reverse

DFA

Given L & $A = (Q, \Sigma, \delta, q_0, F)$

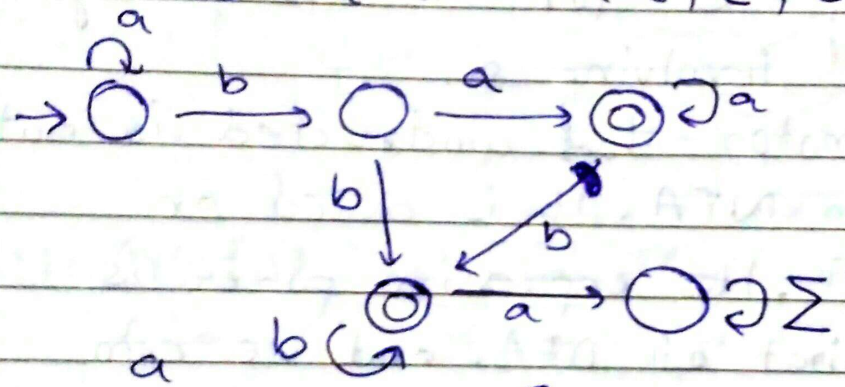
let us construct an NFA B for

$L' = \{ rev(w) \mid w \in L \}$

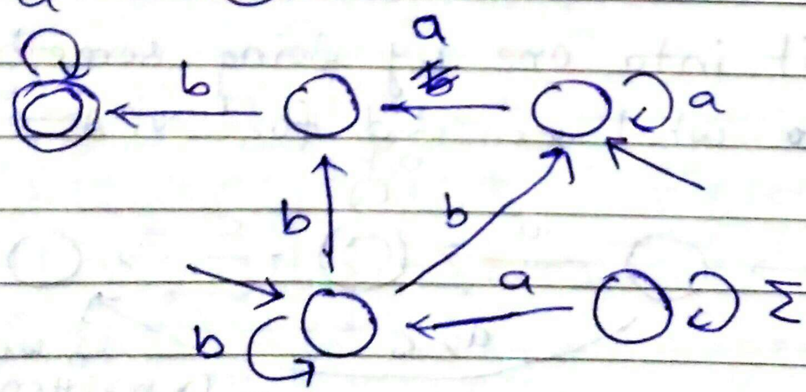
Let $B = (Q, \Sigma, \delta', Q_0, F')$

Eg

A:



B:



→ Reverse each transition

→ Initial states $\xleftrightarrow{\text{swap}}$ Final states

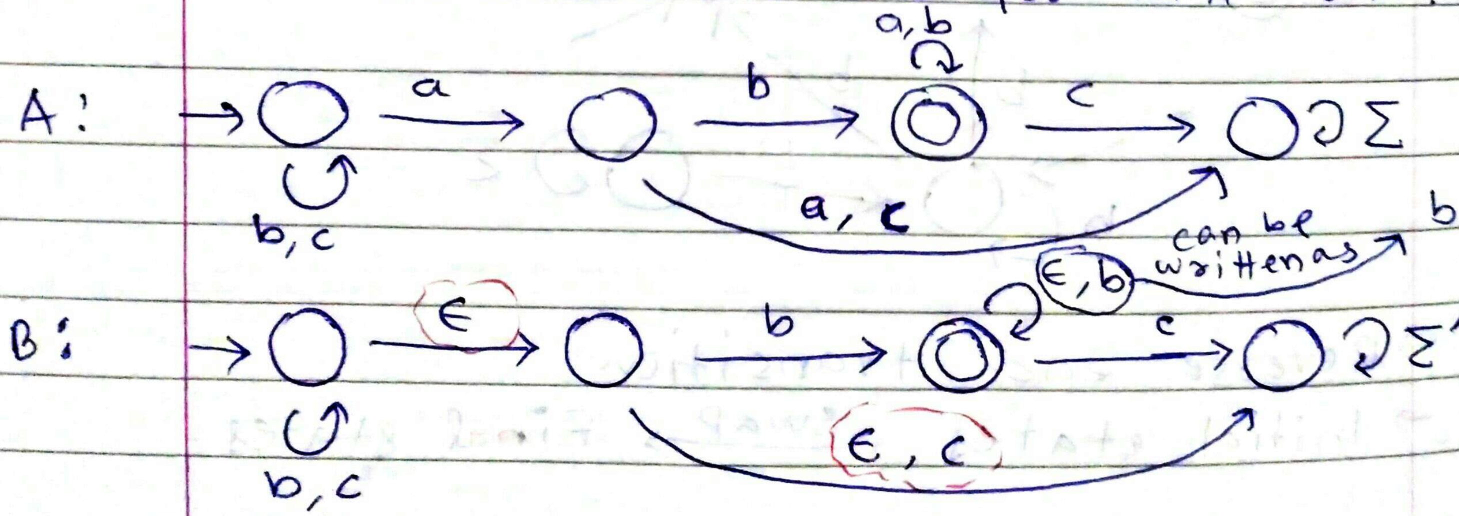
$F' = \{q_0\}$; $Q_0 = F$

$\delta'(q, a) = \{ q' \in Q \mid \delta(q', a) = q \}$

(b) Projection

Let $A = (Q, \Sigma, \delta, q_0, F)$ be the DFA for L, where $\Sigma = \{a, b, c\}$

- We will construct an automaton by substituting ϵ for a on every transition involving a.
- The automaton thus constructed is not a DFA nor an NFA. It is called an ϵ-NFA. Its expressive power is equal to that of DFA, and we can convert it into one by doing something similar to what we did for NFA → DFA.



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

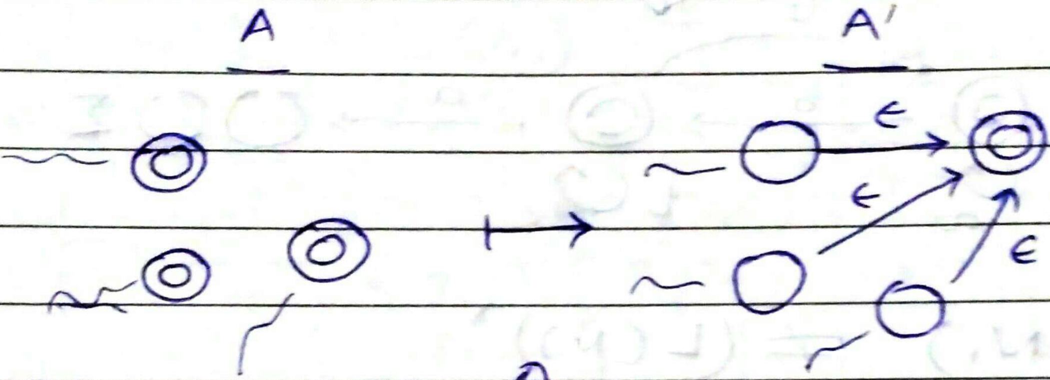
$\Sigma' = \{b, c\} = \Sigma \setminus \{a\}$

$\delta'(q, x) = \delta(q, x)$ for $x \in \Sigma'$

$\delta'(q, \epsilon) = \delta(q, a)$

c) Single Accepting State

Add an extra state, send all final states to that state via ϵ-transition.



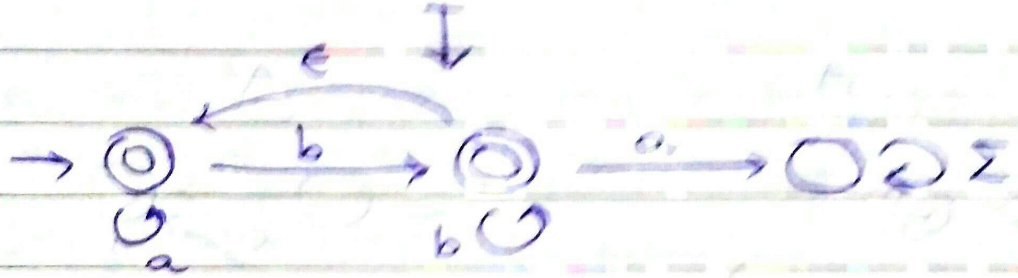
If $A = (Q, \Sigma, \delta, Q_0, F)$
$A' = (Q', \Sigma, \delta', Q_0, F')$

$Q' = Q \cup \{q_F\}$
$\delta'(q, x) = \delta(q, x)$ for every q, x
$F' = \{q_F\}$
$\delta'(q, \epsilon) = \{q_F\}$ for every $q \in F$

(d) $(L(N))^*$

N :

N_1 :



$L(N_1) \neq (L(N))^*$

$\because$ $a \notin L(N) \Rightarrow a \notin (L(N))^*$

but $a \in L(N_1)$

(e) half Language

Let $A = (Q, \Sigma, q_0, \delta, F)$

We will construct an NFA B for $L_{1/2}$

A state in B will be of the form (q, S)

- q is where A takes the input
- S is THE set of states of A which take you to a final state of A using words of length of input

P(Q) is the Power set of Q. that is, the set of all subsets of Q.

Eg. $Q = \{a, b\}$ $P(Q) = \{\phi, \{a\}, \{b\}, \{a, b\}\}$
$\phi \to \{\}$

The idea is to traverse in both directions & "meet in the middle".

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

$Q' = Q \times P(Q)$

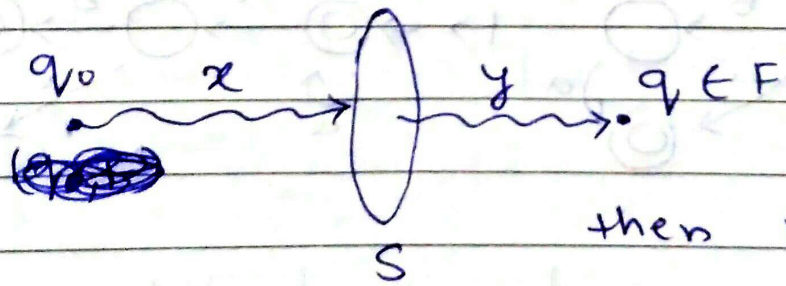
$q_0' = (q_0, F)$

$\delta'((q, S), a) = (\delta(q, a), T)$

where $T = \{p \in Q \mid \exists p' \in S \; \exists b \in \Sigma \; \delta(p, b) = p'\}$

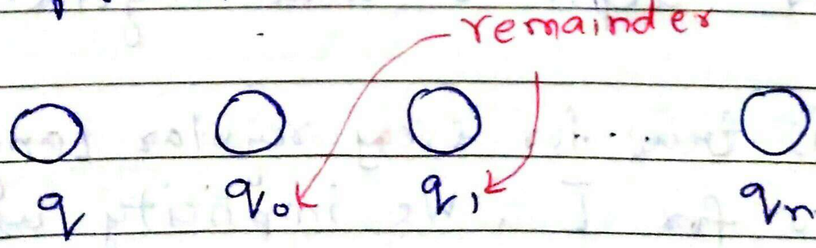
imp   a step back

$F' = \{(q, S) \mid q \in S\}$



$|x| = |y|$
$xy \in L(A)$
then $x \in L(B)$

2. Multiple of n

states:


$A = (Q, \Sigma, \delta, q_s, F)$ ← to avoid clash

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

$Q = \{q, q_0, q_1, \dots q_{n-1}\}$; $n \geq 1$

$q_s = q$ $\qquad$ $\delta(q, 0) = q_0$

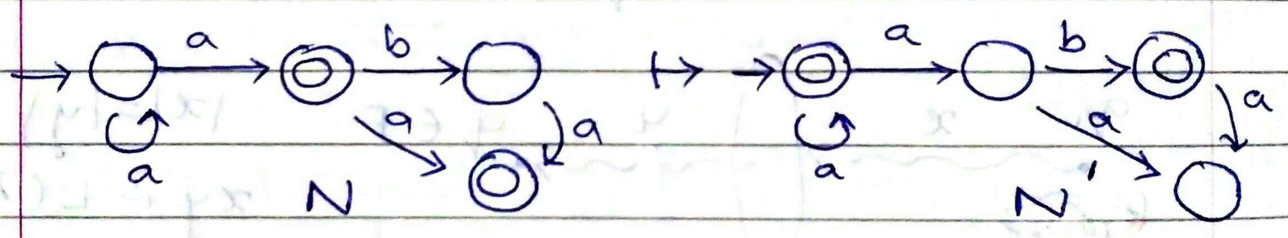
$F = \{q_0\}$ $\qquad$ $\delta(q, 1) = q_{(1 \% n)}$

$\delta(q_i, 0) = q_{(2i) \% n}$

$\delta(q_i, 1) = q_{(2i+1) \% n}$

3 Devilish Acceptance

idea : De Morgan's Identity



Let N accept L with devilish acceptance.
Then N' accepts $\overline{L}$ with angelic acceptance.

This is true for every regular language L, so also for $\overline{L}$. We implicity use the fact that NFA are closed under complementation

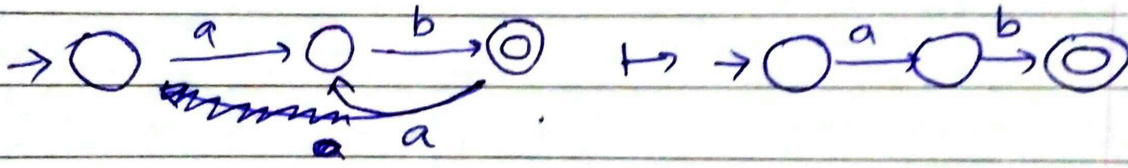
as in L14

Alt: Construct DFA with accepting states decided based on devilish condition.

4 Say no to prefixes

Chop of all transitions going out of final states.
Or send them to a sink (rejecting) state if you want to preserve determinism.

Eg $L = (ab)^+$ $L' = \{ab\}$



OR