CS306 Quiz-1 Solutions

Q1) Which of the following sets are regular and which are not? Give Justification.

(a) $\{a^n b^m \mid n, m \geq 1 ;\ n \geq m ;\ m \leq 481\}$

Soln

Language is Regular.

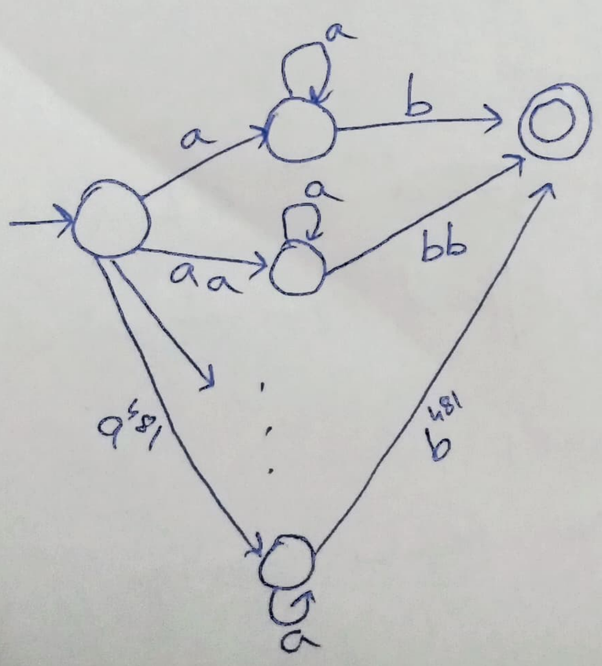
i.e. Say when $m = 1 \Rightarrow n \geq 1$

Illy Say when $m = 2 \Rightarrow n \geq 2$

$\vdots$

Therefore Language $= \{a^m \cdot a^* \cdot b^m \mid m \leq 481, m \geq 1\}$

Since Range of m is finite, we can construct an NFA as follows:



b) $\{a^n b^m \mid n,m \geq 1;\ n \geq m;\ m \geq 481\}$

Soln

Language is Non-Regular.

i) Proof by Myhill-Nerode:

"if for any $x, y \in \Sigma^*$ and for every $z \in \Sigma^*$,

$xz, yz \in L$ or $xz, yz \notin L$

then x, y are in-distinguishable over Language L"

+

" a Language is Regular if and only if it Partitions Σ^* into finitely many equivalence classes".

Assume strings $x = a^k$, $y = a^{k+1}$ (for some $k > 481$)

Now, for $z = b^{k+1}$

$\Rightarrow xz = a^k . b^{k+1} \notin L$, but

$yz = a^{k+1} . b^{k+1} \in L$

$\therefore$ Therefore should belong to different equivalence classes.

$\Rightarrow$ Infinite classes for every $k > 481$

$\Rightarrow$ Non-Regular Language //

(ii) Proof by Pumping-Lemma:

" for all $x \in L$ with $|x| \geq P$, $x = uvw$

Such that,

$$|uv| \leq P$$

$$|v| \geq 1$$

and $\forall i \geq 0 : uv^i w \in L$. "

Suppose $x = a^P b^P$

$$|uv| \leq P \Rightarrow v \in a's$$

Say $\underbrace{a^{P-k}}_{u} \cdot \underbrace{a^{k}}_{v} \cdot \underbrace{b^{P}}_{w}$

@ $i = 0 \Rightarrow a^{P-k} \cdot b^{P}$

here #a's < b's $\Rightarrow \notin L$

$\therefore$ Language is Non-Regular //

Q2) Give an example of L' such that L' is not regular but satisfies the pumping Lemma.

Soln

$$L = \{ a b^j c^j \mid j \geq 0 \} \cup \{ a^i b^j c^k \mid i, j, k \geq 0 ; i \neq 1 \}$$

Say pumping length $= 2$,

All strings in L with length ≥ 2 are as follows:

i, $x = a b^j c^j \mid j \geq 1$

Here, take $u = \epsilon$, $v = a$, $w = b^j c^j$

$\Rightarrow$ Pumping v^i for $i \geq 0$ $\Rightarrow$ string of form $a^i b^j c^k \in L$.

ii, $x = a a b^j c^k$

$u = \epsilon$, $v = aa$, $w = b^j c^k$

$\Rightarrow$ pumping v^i $\Rightarrow$ string of form $a^{2i} b^j c^k \in L$.

iii, $x = a^\lambda b^j c^k \mid \lambda > 2$

$u = \epsilon$, $v = a$, $w = a^{\lambda - 1} b^j c^k$

$\Rightarrow$ pumping v^i $\Rightarrow$ string of form $a^{\lambda - 1 + i} b^j c^k \in L$

(iv) $x = b^j c^k \mid j \geq 1,\ j+k \geq 2$

$u = \epsilon,\ v = b,\ w = b^{j-1} c^k$.

$\Rightarrow$ pumping $v^i \Rightarrow$ string of form $b^{j-1+i} c^k \in L$.

(v) $x = c^k \mid k \geq 2$

$u = \epsilon,\ y = c,\ w = c^{k-1}$

$\Rightarrow$ pumping $v^i \Rightarrow$ string of form $c^{k-1+i} \in L$.

$\therefore$ Therefore, L satisfies Pumping Lemma for Regular Language //

However L is not a Regular Language:

→ Regular Languages are closed under concatination.

Assume L is regular.

$\Rightarrow L \cap \{ab^*c^*\}$ shd be regular.

$= \{a \underbrace{b^j c^j}_{\substack{\downarrow \\ \text{can't be compared by a DFA!}}} \mid j \geq 0\}$

$\therefore$ Contradiction. //

Q3) Let A be a regular set. Consider the following sets, One is Regular & one is not. which is which? Give a proof and a counter-example.

(a) $S_1 = \{x \mid \exists n \geq 0, \exists y \in A, x = y^n\}$

This Language is not necessarily Regular.

Counter-Example:

Let $A = ab^*$

then $S_1 = \{ \underbrace{ab^k ab^k}_{} \ldots n \text{ times} \mid n \geq 0\}$

Can't exactly match with DFA!

(b) $S_2 = \{x \mid \exists n \geq 0, \exists y \in A, y = x^n\}$

This Language is Regular.

Proof by Constructing an NFA:

Let the DFA corresponding to set A be $\langle Q, q_0, F, \delta \rangle$
& $|Q| = m$
(Say $Q_1, Q_2, \ldots Q_m$)

Corresponding NFA for $S_2 = \langle Q', q_0', F', \delta' \rangle$
where $|Q'| = m^m$

i.e, each state $\in Q' = (q_1, q_2 \ldots q_m)$ where $q_i \in Q$

Start state:

$$q_0' = (Q_1, Q_2, \ldots Q_m)$$

Transition:

$$\delta'((q_1, q_2, \ldots q_m), a) =$$
$$(\delta(q_1, a), \delta(q_2, a), \ldots \delta(q_m, a))$$

(where $a \in \Sigma$)

Final states:

for every state $q = (q_1, q_2, \ldots q_m) \in Q'$:

check if $q_1 \in F$

↳ if Yes $\Rightarrow$ q is final in S_2

↳ if No: Say q_1 was Q_K

go-to q_K in q.

& repeat to check if $q_K \in F$

The set of all states one finds by this algo $= F'$

//