6- Properties of Regular Languages

6.1 Pumping Lemma for Regular Languages:

This theorem is used to prove that certain languages are non-regular.

Statement:

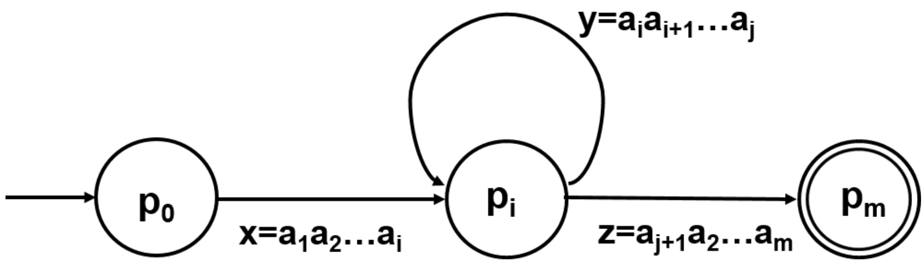
Let L be a Regular Language.

There exists a constant $n > 0$, such that for every string $w \in L$ such that $|w| \geq n$, we can break w into three strings x, y and z, such that:

a. $y \neq \varepsilon$.
b. $|xy| \leq n$.
c. For all $k \geq 0$, the string $xy^kz \in L$.

PROOF:

- Let the Language L be a Regular Language.
- Then $L = L\ (A)$ for some DFA A.
- Consider any string w such that $|w| \geq n$ i.e., $w = a_1a_2a_3 \ldots a_m$ where $m \geq n$.
- Let $p_i = \hat{\delta}(q_0, a_1a_2a_3 \ldots a_i)$. i.e., p_i is the state after reading the first i symbols.
- $p_0 = q_0$.
- By the pigeonhole principle all p_i's cannot be distinct.
- $\therefore$ We find two integers i and j , with $0 \leq i < j \leq n$, such that $p_i = p_j$.
- Now we break w into x, y and z i.e., $w = xyz$, such that
  1. $x = a_1a_2a_3 \ldots a_i$.
  2. $y = a_{i+1}a_{i+2}a_{i+3} \ldots a_j$.
  3. $z = a_{j+1}a_{j+2}a_{j+3} \ldots a_m$.



- The string **xy^kz (For $k >= 0$)** is processed by the above DFA as follows
    - For **$k = 0$**
        - The DFA goes from **p_0** to **p_i** on **x**.
        - The DFA goes from **p_i** to **p_m** on **z**.
        - Hence the string **xz** is accepted.
    - For **$k > 0$**
        - The DFA goes from **p_0** to **p_i** on **x.**
        - The DFA goes from **p_i** to **p_i** on **y k** times (Loop) i.e., y repeated k times.
        - The DFA goes from **p_i** to **p_m** on **z.**
        - Hence the string **xy^kz (For $k > 0$)** is accepted.

Proving Language is Non Regular (Pumping Lemma)

1. Assume the language L is Regular.
2. Let $w \in L$ and $|w| \geq n$.
3. Split w into x, y and z such that
    - $y \neq \varepsilon$.
    - $|xy| \leq n$ and
4. Show that for some $k \geq 0$, $xy^kz \notin L$. Hence conclude the language L is Non Regular.

Ex: Prove that the language

$L = \{a^n b^n \mid n \geq 0\}$ is non regular.

1. Let the language L be regular.
2. Let $w = a^n b^n$ and $|w| = 2n >= n$.
3. Split w into x, y, z such that $y \neq \varepsilon$ and $|xy| \leq n$.

   $X = a^i \quad 0 <= i < n$

   $Y = a^j \quad j >= 1$

   $Z = a^{n-i-j} b^n$
4. $xy^k z = a^i (a^j)^k \; a^{n-i-j} b^n$

   $= a^{i+jk+n-i-j} b^n$

   $= a^{n+jk-j} b^n$

   $= a^{n+j(k-1)} b^n$

   For k = 0

   $xy^k z = a^{n-j} b^n$

   Since j >=1 the number of a's in the above string will be less than the number of b's i.e., number of a's ≠ number of b's.

   Hence the language L is not a regular language.

Ex: Prove that the language

$L = \{a^m b^n \mid m > n\}$ is non regular.

1. Let the language L be regular.
2. Let $w = a^n b^{n-1}$ and $|w| = 2n-1 >= n$.
3. Split w into x, y, z such that $y \neq \varepsilon$ and $|xy| \leq n$.

   $X = a^i \quad 0 <= i < n$

   $Y = a^j \quad j >= 1$

   $Z = a^{n-i-j} b^{n-1}$
4. $xy^k z = a^i (a^j)^k \; a^{n-i-j} b^{n-1}$

   $= a^{i+jk+n-i-j} b^{n-1}$

   $= a^{n+jk-j} b^{n-1}$

   $= a^{n+j(k-1)} b^{n-1}$

   For k = 0

$xy^kz = a^{n-j}b^{n-1}$

Since j >=1 the number of a's in the above string will be less than or equal to the number of b's.

Hence the language L is not a regular language.

Ex: Prove that the language

$L = \{a^m b^n \mid m < n\}$ is non regular.

1. Let the language L be regular.
2. Let $w = a^n b^{n+1}$ and $|w| = 2n-1 >= n$.
3. Split w into x, y, z such that $y \neq \varepsilon$ and $|xy| \leq n$.

   $X = a^i$ $0 <= i < n$

   $Y = a^j$ $j >= 1$

   $Z = a^{n-i-j}b^{n+1}$
4. $xy^kz = a^i (a^j)^k \ a^{n-i-j}b^{n+1}$

   $= a^{i+jk+n-i-j}b^{n+1}$

   $= a^{n+jk-j}b^{n+1}$

   $= a^{n+j(k-1)}b^{n+1}$

   For k = 2

   $xy^kz = a^{n+j}b^{n+1}$

   Since j >=1 the number of a's in the above string will be greater than or equal to the number of b's.

   Hence the language L is not a regular language.

6.2 Closure Properties of Regular Languages.

- Union of Regular Languages is a Regular Language.
- Concatenation of Regular Languages is a Regular Language.
- Closure of a Regular Language is a Regular Language.
- Intersection of Regular Languages is a Regular Language.
- Complement of a Regular Language is a Regular Language.
- Difference of two Regular Languages is a Regular Language.
- Reversal of a Regular Language is a Regular Language.
- Homomorphism of a Regular Language is a Regular Language.
- Inverse Homomorphism of a Regular Language is a Regular Language.

Theorem: If L and M are regular languages then L U M is a regular language.

PROOF: *(Union of Regular Languages is a Regular Language)*

∵ L is a RL, L = L(R) for some RE R.

∵ M is a RL, M = L(S) for some RE S.

∵ R and S are REs, R + S is a RE.

L(R + S) = L(R) U L(S) = L U M .

Theorem: If L and M are regular languages then LM is a regular language.

PROOF: *(Concatenation of Regular Languages is a Regular Language)*

∵ L is a RL, L = L(R) for some RE R.

∵ M is a RL, M = L(S) for some RE S.

∵ R and S are REs, RS is a RE.

L(RS) = L(R)L(S) = LM .

Theorem: If L is a regular language then L* is a regular language.

PROOF: *(Closure of Regular Language is a Regular Language)*

∵ L is a RL, L = L(R) for some RE R.

∵ R is a RE, R* is a RE.

L(R*) = (L(R))* = (L)* = L* .

Theorem: If L is a regular language then $\overline{L}$ is a regular language.

PROOF: *(Complement of a Regular Language is a Regular Language).*

Idea: to complement the accepting and non-accepting states of the machine.

Let L = L(A) for some DFA A=(Q, Σ, δ, q_0, F)

Then $\overline{L}$ = L(B) for some DFA B=(Q, Σ, δ, q_0, Q - F)

The string w is in L(B) iff $\hat{\delta}$(q0,w) in Q – F.

Theorem: If L and M are regular languages then L ∩ M is a regular language.

PROOF: *(Intersection of Regular Languages is a Regular Language)*

L ∩ M = $\overline{\overline{L} \cup \overline{M}}$ (De Morgan's Theorem).

L and M are regular languages. (Given)

∴ $\overline{L}$ and $\overline{M}$ are regular languages. (Proven)

∴ $\overline{L}$ U $\overline{M}$ is a regular language and $\overline{\overline{L} \cup \overline{M}}$ is a Regular Language. (Proven)

Theorem: If L and M are regular languages then L − M is a regular language.

PROOF: *(Difference of Regular Languages is a Regular Language)*

$L - M = L \cap \overline{M}$.

$\because$ M is regular, $\overline{M}$ is Regular.

$\because$ L and $\overline{M}$ are Regular, $L \cap \overline{M}$ is regular.

Theorem: Reversal of a Regular Language is a Regular Language.

- If $w = a_1a_2a_3 \ldots a_{n-1}a_n$, then the reversal $w^R = a_na_{n-1} \ldots a_3a_2a_1$.
- If L is a Language then L^R is the language consisting of reversals of strings of L.
- Ex: L = {001, 10, 111} L^R = {100, 01, 111}

PROOF: *(Reversal of Regular Language is a Regular Language)*

Given a Language L for some DFA A, we may construct a DFA for L^R as follows.

- Reverse all edges in A.
- Make the start state of A the only final state.
- Create a new start state p_0, with ε transitions to all the final states of A.