2.7.3 To Show that Language is Non-regular

This is a basic and important theorem used for checking whether given string is accepted by regular expression or not. In short, this lemma tells us whether given language is regular or not.

One key theme is that any language for which it is possible to **design the finite automata** is definitely the **regular language**.

Theorem : Let L be a regular set. Then there is a constant n such that if z is any word in L and $|z| \geq n$ we can write $z = u\,v\,w$ such that $|u\,v| \leq n$, $|v| \geq 1$ for all $i \geq 0$, $u\,v^i\,w$ is in L. The n should not be greater than the number of states.

Proof : If the language L is regular it is accepted by a DFA. $M = (Q, \Sigma, \delta, q_0, F)$. With some particular number c states say, n. Consider the input can be $a_1, a_2, a_3, \ldots a_m$, $m \geq n$. The mapping function δ could be written a $\delta(q_0, q_1, q_2, q_3, \ldots q_i) = q_i$.

The transition diagram is as shown in Fig. 2.7.1.

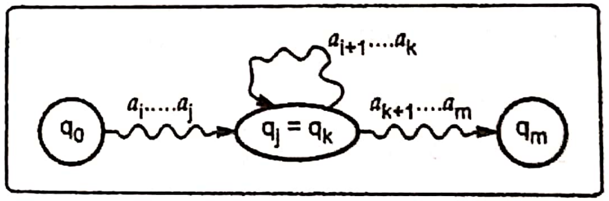


Fig. 2.7.1 Pumping lemma

If q_m is in F i.e. $q_1, q_2, q_3, \ldots q_m$ is in L(M) then $a_1, a_2, \ldots a_j\, a_{k+1}\, a_{k+2} \ldots a_m$ is also in L(M). Since there is path from q_0 to q_m that goes through q_j but not around the loop labelled $a_{j+1} \ldots a_k$. Thus

$$\delta\left(q_0, a_1, a_j\ a_{k+1} \ldots a_m\right) = \delta\left(\delta\left(q_0, q_1, \ldots q_j\right), a_{k+1} \ldots a_m\right)$$

$$= \delta\left(q_j, q_{k+1} \ldots q_m\right)$$

$$= \delta(q_k, q_{k+1} \ldots q_m)$$

$$= q_m$$

That what we have proved is that given any long string can be accepted by FA, we should be able to find a substring near the beginning of the string that may be pumped i.e. repeated as many times as we like and resulting string may be accepted by FA.

The pumping lemma is used to check whether given language is **regular or not.**

Example 2.7.2 *State and prove pumping lemma for regular languages. Apply pumping lemma for following ...*

1. *State and prove pumping lemma for regular languages.*

2.8 Closure Properties of Regular Languages

VTU : Feb.-09,11, Aug.-09,10, Dec.-11, Marks 6

If certain languages are regular and language L is formed from them by certain operations (such as union or concatenation) then L is also regular. These properties are called closure properties of regular languages. Such languages represent the class of regular laguages which is closed under the certain specific operations.

The closure properties express the idea that when one or many languages are regular then certain related languages are also regular. The closure properties of regular languages are as given below.

1. The union of two regular languages is regular.
2. The intersection of two regular languages is regular.
3. The complement of a regular languages is regular.
4. The difference of two regular languages is regular.
5. The reversal of a regular languages is regular.
6. The closure operation on a regular language is regular.
7. The concatenation of regular language is regular.
8. A homomorphism of regular languages is regular.
9. The inverse homomorphism of regular language is regular.

Theorem 1 : If L_1 and L_2 are two languages then $L_1 \cup L_2$ is regular.

Proof : If L_1 and L_2 are regular then they have regular expression $L_1 = L(R_1)$ and $L_2 = L(R_2)$. Tl $L_1 \cup L_2 = L(R_1 + R_2)$ thus we get $L_1 \cup L_2$ as regular language. (any language given by some regular expressio regular).

Theorem 2 : The complement of regular language is regular

Proof : Consider L_1 be regular language which is accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$. The complemen regular language is $\overline{L_1}$ which is accepted by $M' = (Q, \Sigma, \delta, q_0, Q - F)$. That means M is a DFA with final states and M' is a DFA in which all the non-final states of M become final. In other words, we can say that the strings that are accepted by M are rejected by M' similarly, the strings rejected by M are accepted by M'.

Thus as $\overline{L_1}$ is accepted by DFA M', it is regular.

Theorem 3 : If L_1 and L_2 are two regular languages then $L_1 \cap L_2$ is regular.

Proof : Consider that languages L_1 is regular. That means there exists some DFA M_1 that accepts L_1. We can write $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Similarly being L_2 regular there is another DFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Let, L be the language obtained from $L_1 \cap L_2$. We can the simulate $M = (Q, \Sigma, \delta, q, F)$.

where $Q = Q_1 \cap Q_2$

$\delta = \delta_1 \cap \delta_2$ a mapping function derived from both the DFAS.

$q \in Q$ which is initial state of machine M.

$F = F_1 \cap F_2$, the set of final states, which is common for M_1 and M_2 both.

It is clear that there exists some DFA which accepts $L_1 \cap L_2$ i.e. L. Hence L is a regular language. This proves that if L_1 and L_2 are two regular languages then $L_1 \cap L_2$ is regular. In other words the regular language is closed under intersection.

Theorem 4 : If L_1 and L_2 are two regular languages then $L_1 - L_2$ is regular.

Proof : The $L_1 - L_2$ can also be denoted as $L_1 \cup \overline{L_2}$

Consider L_1 be regular language which is accepted by DFA $M = (Q, \Sigma, \delta, q_0, F)$. The complement of regular language L_1 is $\overline{L_1}$ which is accepted by $M' = (Q, \Sigma, \delta, q_0, Q - F)$. That means M is a DFA with final state's set F and M' is a DFA in which all the non final states of M become final states and all the final states of M become non-final states. Thus L_1 and $\overline{L_2}$ are two regular languages. That also means : these languages are accepted by regular expressions. If $L_1 = L(R_1)$ and $\overline{L_2} = L(R_2')$. Then $L_1 \cup \overline{L_2} = L(R_1 + R_2')$. This ultimately shows that $L_1 \cup \overline{L_2}$ is regular. In other words $L_1 - L_2$ is regular. Thus regular languages are closed under difference.

Theorem 5 : The reversal of a regular languages is regular.

Proof : Reversal of a string means obtaining a string which is written from backward that is w^R is denoted as reversal of string w. That means $L(w^R) = (L(w))^R$. This proof can be done with basis of induction.

Basis : If $w = \varepsilon$ or ϕ then w^R is also ε or ϕ

i.e. $(\varepsilon)^R = \varepsilon$ and $(\phi)^R = \phi$

Hence $L(w^R)$ is also regular.

Induction :

Case 1 : If $w = w_1 + w_2$ then

$w = (w_1)^R + (w)^R$

As the regular language is closed under union. Then w is also regular.

Case 2 : If $w = w_1 w_2$

Consider $w_1 = (ab, bb)$

and $w_2 = (bbba, aa)$

The $w_1^R = (ba, aa)$

$w_2^R = (aaab, bb)$ then

$w = w_1^R \, w_2^R$

(ba, aa, aaab, bb) is also regular. Thus given language is regular one.

Theorem 6 : The closure operation on a regular language is regular.

Proof : If language L_1 is regular then it can be expressed as $L_1 = L(R_1^*)$. Thus for a closure operation a language can be expressed as a language of regular expressions. Hence L_1 is said to be a regular language.

Theorem 7 : If L_1 and L_2 are two languages then $L_1 \cdot L_2$ is regular. In other words regular languages are closed under concatenation.

Proof : If L_1 and L_2 are regular then they can be expressed as $L_1 = L(R_1)$ and $L_2 = L(R_2)$. Then $L_1 \cdot L_2 = L(R_1 \cdot R_2)$ thus we get a regular language. Hence it is proved that regular languages are closed under concatenation.

Theorem 8 : A homomorphism of regular languages is regular.

Proof : The term homomorphism means substitution of string by some other symbols. For instance the string "aabb" can be written as 0011 under homomorphism. Clearly here, a is replaced by 0 and b is replaced by 1. Let Σ is the set of input alphabets and Γ be the set of substitution symbols then $\Sigma^* \rightarrow \Gamma^*$ is homomorphism. The definition of homomorphism can be extended as

Let, $w = a_1 a_2 \ldots a_n$

$$h(w) = h(a_1)h(a_2)\ldots h(a_n)$$

If L is a language that belongs to set Σ, then the **homomorphic image** of L can be defined as :

$$h(L) = \{h(w) : w \in L\}$$

To prove that if L is regular h(L) is also regular consider following example -

Let, $\Sigma = \{a, b\}$ and $w = abab$

Let $h(a) = 00$

and $h(b) = 11$

Then we can write $h(w) = h(a)\ h(b)\ h(a)\ h(b)$

$= 00110011$

The homomorphism to language is applied by applying homomorphism on each string of language.

$\therefore$ If $L = ab^*b$ then,

$L = \{ab, abb, abbb, abbbb, \ldots\}$

Now $h(L) = \{0011, 001111, 00111111, 0011111111, \ldots\}$

$\therefore$ $h(L) = 00\,(11)^+$ As it can be represented by a regular expression, it is a regular language. Hence it is proved that if L is regular then h(L) is also regular. In other words, family of regular languages is closed under homomorphism.

Theorem 9 : The inverse homomorphism of regular language is regular.

Proof : Let $\Sigma^* \rightarrow \Gamma^*$ is homomorphism.

The Σ is the input set and Γ be the substitution symbols used by homomorphic function.

Let, L be the regular language where $L \in \Sigma$, then h(L) be homomorphic language.

The inverse homomorphic language can be represented as $h^{-1}(L)$

Let,

$$h^{-1}(L) = \{w \mid w \in L\}$$

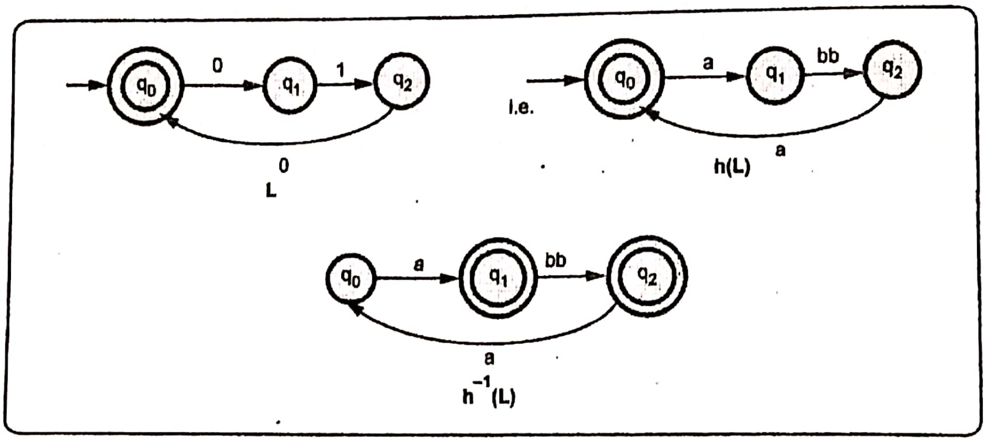
If L is regular then h(L) is also regular because regular language is closed under homomorphism. That if there exist a FA $M = (Q, \in, \delta, q_0, F)$ which accepts L then h(L) must also be accepted by FA M. For complement of L i.e. language L' the inverse homomorphic language is $h^{-1}(L)$. Let M' be the FA in which all the final states of M become non-final states and all the non-final states of M become the final states. Clearly the language L' can be accepted by M' Hence $h^{-1}(L)$ must also be accepted by FA M'.

For example - Let L = (010)* be a regular language. And L = {010, 010010,...}

Let h(0) = a and h(1) = bb be homomorphic function. Then

$$h(L) = (abba)^*$$

This L can be represented by following DFA.



Thus there exists a FA which accepts $h^{-1}(1)$. This shows that, inverse homomorphism of regular language is regular.

University Questions

1. *If L_1 and L_2 are regular languages then prove that family of regular language are closed under $L_1 - L_2$.* VTU : Feb.-09, Marks 6
2. *Prove that if L is a regular language over alphabet Σ - then $\overline{L}$ is also a regular language.* VTU : Aug.-09, Marks 4
3. *Prove that if L and M are regular languages, then so is $L \cap M$.* VTU : Aug.-10, Marks 6
4. *Show that if L is regular then complement of L denoted by L' is also regular.* VTU : Feb.-11, Dec.-11, Marks 4