# UNIT-II

Subject-Theory of Computation

Prof. Shweta Tiwaskar
Shweta.tiwaskar@viit.ac.in
**Department of Computer Engineering**

**BRACT'S, Vishwakarma Institute of Information Technology, Pune-48**

**(An Autonomous Institute affiliated to Savitribai Phule Pune University)**
**(NBA and NAAC accredited, ISO 9001:2015 certified)**

## 5.5 CLOSURE PROPERTIES OF REGULAR SETS

In this section we discuss the closure properties of regular sets under (i) set union, (ii) concatenation, (iii) closure (iteration), (iv) transpose, (v) set intersection, and (vi) complementation.

***Property 1: The union of two regular set is regular.***

$L_1$ = {a, aaa, aaaaa,.....} (Strings of odd length excluding Null)
and $L_2$ ={ ε, aa, aaaa, aaaaaa,.......} (Strings of even length including Null)
$L_1 \cup L_2$ = { ε, a, aa, aaa, aaaa, aaaaa, aaaaaa,.......}
(Strings of all possible lengths including Null)
RE ($L_1 \cup L_2$) = a* (which is a regular expression itself)

***Property 2: T****he concatenation of two regular sets is regular.*

$L_1$ = {0, 00, 10, 000, 010, ......} (Set of strings ending in 0)
and $L_2$ = {1,10, 11,.....} (Set of strings beginning with 1)
$L_1 L_2$ = {01,010,0010,0001,00010,00011,............}
RE($L_1 L_2$ ) =1(0+1)*0 which is a regular expression itself.

**Property 3.** *The closure of a regular set is regular.*

f L = {a, aaa, aaaaa, .......} (Strings of odd length excluding Null)
i.e., RE (L) = a (aa)*
L* = {a, aa, aaa, aaaa , aaaaa,……………} (Strings of all lengths excluding Null)
RE (L*) = a (a)*

- **Property 4.** *The transpose of a regular set is regular.*

We have to prove $\mathbf{L^T}$ is also regular if **L** is a regular set.

Let, L = {01, 10, 11, 10}

RE (L) = 01 + 10 + 11 + 10

$\mathbf{L^T}$ = {10, 01, 11, 01}

RE ($\mathbf{L^T}$)= 01 + 10 + 11 + 10 which is regular

**Property 5.** *The complement of a regular set is regular.*

So, L = {ε, aa, aaaa, aaaaaa, .......} (Strings of even length including Null)

Complement of **L** is all the strings that is not in **L**.

So, L’ = {a, aaa, aaaaa, .....} (Strings of odd length excluding Null)

RE (L’) = a(aa)* which is a regular expression itself.

- **Property 6.** *The intersection of two regular set is regular.*

Let us take two regular expressions

$RE_1 = a(a^*)$ and $RE_2 = (aa)^*$

So, $L_1$ = { a,aa, aaa, aaaa, ....} (Strings of all possible lengths excluding Null)

$L_2$ = { ε, aa, aaaa, aaaaaa,.......} (Strings of even length including Null)

$L_1 \cap L_2$ = { aa, aaaa, aaaaaa,.......} (Strings of even length excluding Null)

RE ($L_1 \cap L_2$) = aa(aa)* which is a regular expression itself.

# Pumping Lemma

- In the theory of formal languages, the **pumping lemma for regular languages** is a lemma that describes an essential property of all regular languages
- It says that all sufficiently long words in a regular language may be *pumped*—that is, have a middle section of the word repeated an arbitrary number of times—to produce a new word that also lies within the same language

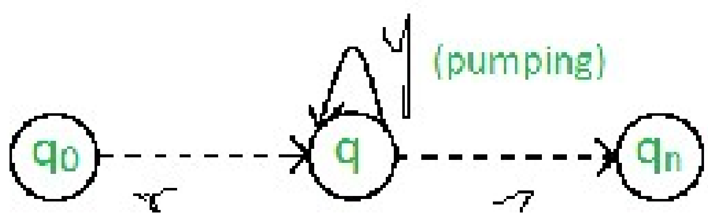
Let L be a regular language. Then there exists an integer  P >=1, Such that every string w in L of length at least p can be written as w=xyz, w can be divided into three substrings , satisfying the following conditions

- $|y| \geq 1$
- $|xy| \leq p$
- $(\forall n \geq 0)(xy^n z \in L)$

In simple terms, this means that if a string y is 'pumped', i.e., if y is inserted any number of times, the resultant string still remains in L.

Pumping Lemma is used as a proof for irregularity of a language. Thus, if a language is regular, it always satisfies pumping lemma

If there exists at least one string made from pumping which is not in L, then L is surely not regular.



# Application of Pumping Lemma

- Pumping Lemma is to be applied to show that certain languages are not regular

This theorem can be used to prove that certain sets are not regular. We now give the steps needed for proving that a given set is not regular.

**Step 1** Assume that $L$ is regular. Let $n$ be the number of states in the corresponding finite automaton.

**Step 2** Choose a string $w$ such that $|w| \geq n$. Use pumping lemma to write $w = xyz$, with $|xy| \leq n$ and $|y| > 0$.

**Step 3** Find a suitable integer $i$ such that $xy^iz \notin L$. This contradicts our assumption. Hence $L$ is not regular.

***Note:*** The crucial part of the procedure is to find $i$ such that $xy^iz \notin L$. In some cases we prove $xy^iz \notin L$ by considering $|xy^iz|$. In some cases we may have to use the 'structure' of strings in $L$.

Show that $L = \{0^i1^i \mid i \geq 1\}$ is not regular.

**Step 1** Suppose $L$ is regular. Let $n$ be the number of states in the finite automaton accepting $L$.

**Step 2** Let $w = 0^n1^n$. Then $|w| = 2n > n$. By pumping lemma, we write $w = xyz$ with $|xy| \leq n$ and $|y| \neq 0$.

**Step 3** We want to find $i$ so that $xy^iz \notin L$ for getting a contradiction. The string $y$ can be in any of the following forms:

*Case 1* $y$ has 0's, i.e. $y = 0^k$ for some $k \geq 1$.

*Case 2* $y$ has only 1's, i.e. $y = 1^l$ for some $l \geq 1$.

*Case 3* $y$ has both 0's and 1's, i.e. $y = 0^k1^j$ for some $k, j \geq 1$.

In Case 1, we can take $i = 0$. As $xyz = 0^n1^n$, $xz = 0^{n-k}1^n$. As $k \geq 1$, $n - k \neq n$. So, $xz \notin L$.

In Case 2, take $i = 0$. As before, $xz$ is $0^n1^{n-l}$ and $n \neq n - l$. So, $xz \notin L$.

In Case 3, take $i = 2$. As $xyz = 0^{n-k}0^k1^j1^{n-j}$, $xy^2z = 0^{n-k}0^k1^j0^k1^j1^{n-j}$. As $xy^2z$ is not of the form $0^i1^i$, $xy^2z \notin L$.

Thus in all the cases we get a contradiction. Therefore, $L$ is not regular.

- Case1: Y contains only 0's
- e.g String :0000011111
- X=00,y=00, Z=011111
- i=2, $XY^2Z$=000000011111 does not belong to L($0^n1^n$|n>=1}
- As in the generated string $XY^2Z$=000000011111, no of 0's & 1'S are not equal
- Case2: Y contains only 1's
- e.g String :0000011111
- X=00000,y=11, Z=111
- i=2, $XY^2Z$=000001111111 does not belong to L($0^n1^n$|n>=1}
- As in the generated string $XY^2Z$=000000011111, no of 0's & 1'S are not equal

- Case3: Y contains 0's & 1's
- e.g String :0000011111
- X=0000,y=01, Z=1111
- i=2, $XY^2Z$=000001011111 does not belong to L=($0^n1^n$|n>=1}
- As in the generated string $XY^2Z$=000001011111, even though no of 0's & 1'S are equal but 1' is appearing in between 0's and 1's.

# Example 2

Show that $L = \{ww \mid w \in \{a, b\}^*\}$ is not regular.

***Solution***

**Step 1** Suppose $L$ is regular. Let $n$ be the number of states in the automaton $M$ accepting $L$.

$xyz \in R$

$xy^2z \in R$

W=ab p=5

$a^Pba^Pb$ aaaaabaaaaab

Divide it INTO X Y and Z

X=aa

Y=aaa

Z=baaaaab

$xy^2z$ = aaaaaaaabaaaaab

$XY^iZ$, i as 2, $XY^2Z$

$xy^2z \notin R$

No. of b a's are not equal

$XY^2Z$ does not belong to L=(ww|w€{a,b}*}

As the generated string $XY^2Z$=baaaaab is not in the form of ww.

- Prove that the given language is not regular. Using Pumping lemma.

$$\{a^i b^j \mid j>i,\ i,j>0\}$$

**String=abb**
X=a, y=b, Z=b
i=2, $XY^iZ = XY^2Z$ = abbb
**String=aabbb**
X=a, Y=a, Z=bbb
i=2, $XY^2Z$= aaabbb does not belong to language L
So we have proved that above language is not regular.

- Prove that the given language is not regular. Using Pumping lemma.

$$\{a^i b^j \mid j<i,\ i,j>0\}$$

**String=aab**
X=a, y=a, Z=b
i=2, $XY^iZ= XY^2Z=$ aaab belongs to L
**String=aabbb**
X=a, Y=a, Z=bbb
i=2, $XY^2Z=$ aaabbb does not belong to language L
For i=2 aaabbb does not belong to language L
So we have proved that above language is not regular.

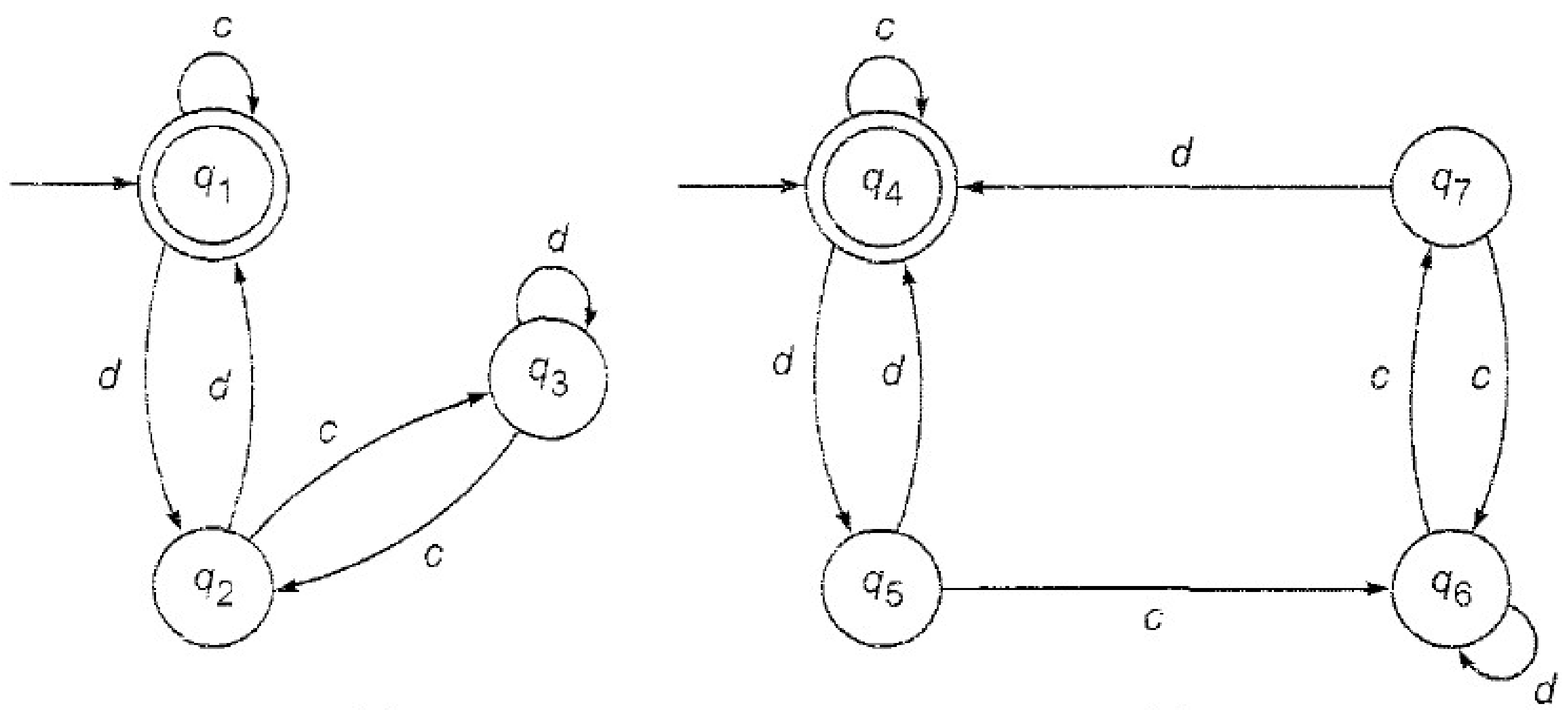
# Equivalence of Two Finite Automata

- When two FA are not equivalent one FA reaches a final state on application of ‘w’ whereas other FA reaches a non final state.
- Comparison Method:

***Case 1*** If we reach a pair $(q, q')$ such that $q$ is a final state of $M$, and $q'$ is a nonfinal state of $M'$ or vice versa, we terminate the construction and conclude that $M$ and $M'$ are not equivalent.

***Case 2*** Here the construction is terminated when no new element appears in the second and subsequent columns which are not in the first column (i.e. when all the elements in the second and subsequent columns appear in the first column). In this case we conclude that $M$ and $M'$ are equivalent.

Consider the following two DFAs $M$ and $M'$ over $\{0, 1\}$ given in Fig. : .
Determine whether $M$ and $M'$ are equivalent.



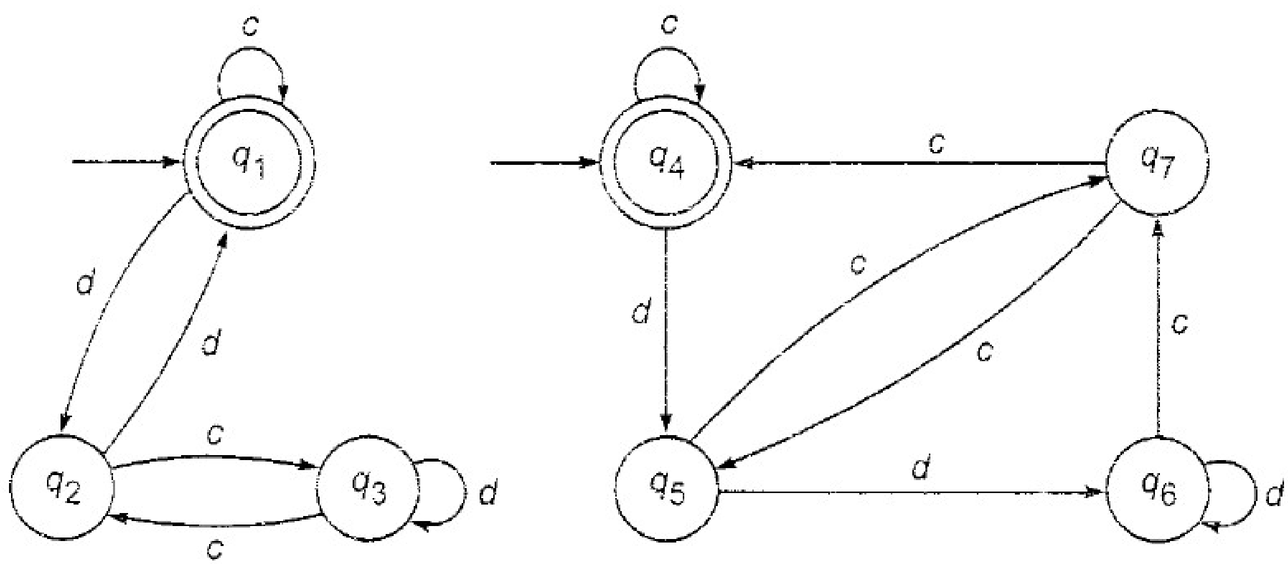
| (q,q’) | (qc,qc’) | (qd,qd’) |
|---|---|---|
| (q1,q4) | (q1,q4) | (q2,q5) |
| (q2,q5) | (q3,q6) | (q1,q4) |
| (q3,q6) | (q2,q7) | (q3,q6) |
| (q2,q7) | (q3,q6) | (q1,q4) |

**We can conclude that FA M & M’ are equivalent**

| $(q, q')$ | $(q_c, q'_c)$ | $(q_d, q'_d)$ |
|---|---|---|
| $(q_1, q_4)$ | $(q_1, q_4)$ | $(q_2, q_5)$ |
| $(q_2, q_5)$ | $(q_3, q_6)$ | $(q_1, q_4)$ |
| $(q_3, q_6)$ | $(q_2, q_7)$ | $(q_3, q_6)$ |
| $(q_2, q_7)$ | $(q_3, q_6)$ | $(q_1, q_4)$ |

As we do not get any pair (q,q') where q is final state & q' is the non final state, these 2 FA's are equivalent.

Show that the automata $M_1$ and $M_2$ defined by Fig. . ˙ are not equivalent.



| (q.q’) | (qc,qc’) | (qd,qd’) |
|---|---|---|
| (q1,q4) | (q1,q4) | (q2,q5) |
| (q2,q5) | (q3,q7) | **(q1,q6)** |
| | | |

**(q1,q6) in this pair q1 ia final state and q6 is non final state so we will terminate the proof and conclude that Two FA’s are not equivalent.**