COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO-04

Sub: Theory of Computer Science

COURSE: T.E. Year: 2020-2021 Semester: V
DEPT: Computer Engineering
SUBJECT CODE: CSC504 DUE DATE: 22/11/2020

Roll No. 50 Name: Amey Thakur

Class: TE-Comps B Date of Submission: 20/11/2020

Tutorial 4

1. Write a regular expression for the language in which the first character is a or c followed by any string in b over Σ = {a, b, c}.

2. Write a regular expression for strings containing no double letter over Σ = {a, b}.

3. Give the regular expression for the following
   i] Set of all string over {0,1}that ends with 1 and has no substring 00.
   ii] Set of all string over {0,1}with even no of 1's followed by odd no of 0's.
   iii] for the language which ends in either 01 or 101 over ∑ = { 0,1 }

4. Construct the DFA that accepts the language represented by
   A. 0*1*2*
   B. (11+01)*

5. Give the application of Regular Expression and Finite Automata.

6. State and explain pumping lemma for regular languages.

7. Prove that the following languages are not regular
   i) L= {ww | w∈ (0,1)*}
   ii) $L= \{0^i 1^i \mid i>=1\}$
   iii) $L= \{0^i 1^j \mid i>j\}$

Q.1 Write a regular expression for the language in which the first character is a or c followed by any string in b over $\Sigma = \{a, b, c\}$.

Ans:

$L = \{a, c, ab, cb, abb, cbb, \ldots\ldots\}$

$R = (a+c) \cdot b^*$

Q.2 Write a regular expression for strings containing no double letter over $\Sigma = \{a, b\}$.

Ans:

$L = \{a, b, ab, ba, aba, bab, abab, baba, \ldots\ldots\}$

$R = a \cdot (ba)^* b + b \cdot (ab)^* a$

Q.3. Give the regular expression for the following.

(i) Set of all strings over $\{0, 1\}$ that ends with 1 and has no substring 00.

Ans:

$L = \{1, 01, 101, 011, 1011, 0101, 1111, \ldots\ldots\}$

$R = (10+1)^* \cdot 1 + (01+1)^*$

(ii) Set of all strings over $\{0, 1\}$ with even number of 1's followed by odd number of 0's.

Ans:

$L = \{0, 110, 11000, \ldots\ldots\}$

$R = (11)^* \cdot (00)^* 0$

(iii) For the language which ends in either 01 or 101 over $\Sigma = \{0, 1\}$

Ans:

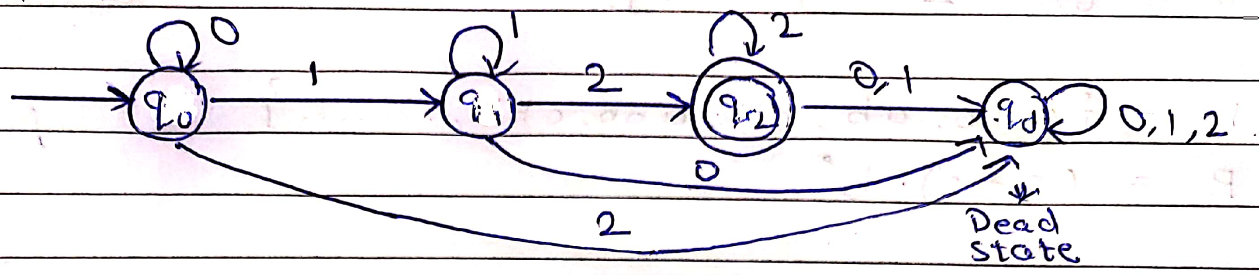
$L = \{01, 101, \ldots\}$

$R = (0+1)^* \cdot 01$

Q.4. Construct the DFA that accepts the language represented by.

(A) $0^* 1^* 2^*$

Ans:



- The finite automation M_1

We can define M_1 formally by writing

$M_1 = (Q, \Sigma, \delta, q_0, F)$

1. $Q = \{q_0, q_1, q_2\}$
2. $\Sigma = \{0, 1, 2\}$
3. δ is described as

	0	1	2
$\rightarrow q_0$	q_0	q_1	q_d
q_1	q_d	q_1	q_2
q_2	q_d	q_d	q_2

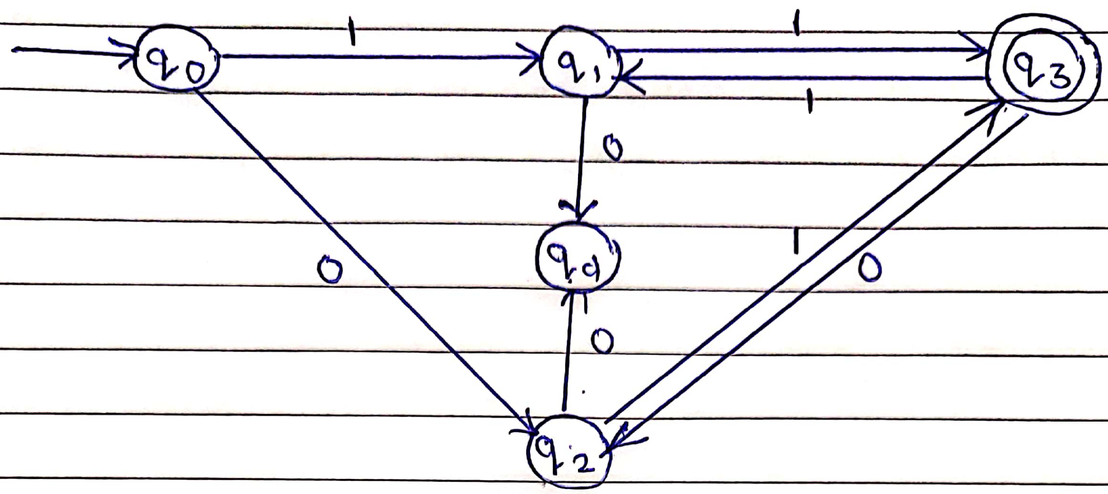
Note:

$q_d \rightarrow$ Dead state

4. q_0 is the initial state
5. F is the final/Acceptance state = { c }

(B) $(11+01)^*$

Ans:



- The finite automation M_1

We can define M_1 formally by writing
$M_1 = (Q, \Sigma, \delta, q_0, F)$

(1) $Q = \{q_0, q_1, q_2, q_3\}$
(2) $\Sigma = \{0, 1\}$
(3) δ is described as

	0	1
$\rightarrow q_0$	q_2	q_1
q_1	q_d	q_3
q_2	q_d	q_3
q_3	q_1	q_2

Note:
q_d = Dead state

(4) q_0 is the initial state
(5) F is the final / Acceptance state = $\{q_3\}$

Q.5 Give the application of Regular Expression and Finite Automata.

Ans:

Application of Regular Expression

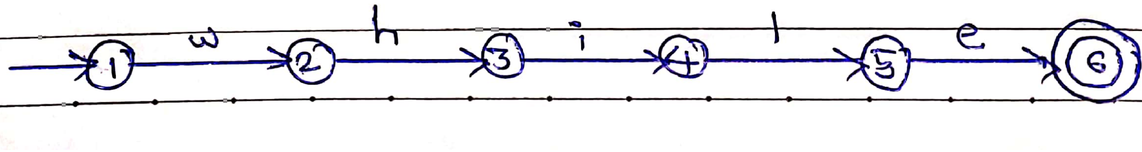
- Regular expressions are useful in a wide variety of text processing tasks and more generally string processing where data need not be textual. Common applications include data validation, data scraping (especially web scrapping), data wrangling, simple parsing, the production of syntax highlighting systems and many other tasks.
- While regexps would be useful on Internet search engines, processing them across the entire database could consume excessive computer resources depending on the complexity and design of the regex.

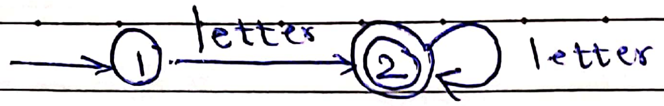
Application of Finite Automata

- Finite automata has several applications in many areas such as compiler design, special purpose hardware design, protocol specification, etc.

① Compiler Design.

- Lexical analysis or scanning is an important phase of a compiler.
- In lexical analysis, a program such as a C program is scanned and the different tokens (constructs such as variables, keywords, numbers) in the program are identified.
- A DFA is used for this operations.
- For example, Finite automation to recognize the tokens, 'while' keyword and variables are shown below





The well known lexical analyser tool, Lex uses DFA to perform lexical analysis.

② Hardware Design

- In the design of computers, finite automation is used to design control unit of a computer.
- A typical sequence of operations of a computer consists of a repetition of instructions and every instruction involves the actions fetch, decode, fetch the operand and execute.
- Every state of finite automation represents a specific stage of instruction cycle.

③ Protocol Specification.

- A system is composed of an interconnected set of subsystems.
- A protocol is a set of rules for proper coordination of activities of a system.
- Such a protocol can be described using finite automations.

Q.6 State and explain pumping lemma for regular languages.

Ans:

Pumping Lemma for Regular Languages.

Theorem

Let L be a regular language.
Then there exists a constant 'c' such that for every string w in L - $|w| \geq c$

We can break w into three strings, $w = xyz$, such that -

- $|y| > 0$
- $|xy| \leq c$
- For all $k \geq 0$, the string xy^kz is also in L.

Applications of Pumping Lemma

- Pumping lemma is to be applied to show that certain languages are not regular. It should never be used to show a language is regular.
  - If L is regular, it satisfies Pumping Lemma.
  - If L does not satisfy Pumping Lemma, it is non-regular.

Method to prove that a language L is not regular.

- At first, we have to assume that L is regular.
- So, the pumping lemma should hold for L.
- Use the pumping lemma to obtain a contradiction. -
  - Select w such that $|w| \geq c$
  - Select y such that $|y| \geq 1$
  - Select x such that $|xy| \leq c$
  - Assign the remaining string to z.
  - Select k such that the resulting string is not in L.

Hence L is not regular.

- 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.
- The opposite of this may not always be true. That is, if Pumping Lemma holds, it does not mean that the language is regular.

Problem

Prove that $L = \{a^i b^i \mid i \geq 0\}$ is not regular.

Solution

- At first, we assume that L is regular and n is the number of states.
- Let $w = a^n b^n$. Thus $|w| = 2n \geq n$.
- By pumping lemma, let $w = xyz$, where $|xy| \leq n$.
- Let $x = a^p$, $y = a^q$ and $z = a^r b^n$, where $p+q+r = n$, $p \neq 0$, $q \neq 0$, $r \neq 0$. Thus $|y| \neq 0$.
- Let $k = 2$. Then $xy^2z = a^p a^{2q} a^r b^n$.
- Number of as $= (p+2q+r) = (p+q+r)+q = n+q$
- Hence, $xy^2z = a^{n+q} b^n$. Since $q \neq 0$, xy^2z is not of the form $a^n b^n$.
- Thus, xy^2z is not in L. Hence L is not regular.

Q.7 Prove that the following languages are not regular.

(i) $L = \{ww \mid w \in (0,1)^*\}$

Ans:

Proof:

- Assume that L is Regular.
- Then it must have a pumping length = P

$S = 0^P 1 0^P 1$

x y z

$xy^iz \Rightarrow xy^2z$

00000000000100000001

$\notin L$

$P = 7$

$\underbrace{00}_{x}\underbrace{00000}_{y}\underbrace{10000000 1}_{z}$

$|y| > 0$

$|xy| \leq P$

∵ We got the contradiction.

∴ Language L is not Regular.

(ii) $L = \{0^i 1^i \mid i >= 1\}$

Ans:

Proof:

- Assume that L is Regular.
- Then it must have a pumping length = P

$S = a^P b^P$

$\Rightarrow S = 0000000 1111111$

x y z

$P = 7$

- case 1: The 'y' is in the 'zero' part

$\underbrace{000}_{x}\underbrace{0000}_{y}\underbrace{1111111}_{z}$

$xy^iz \Rightarrow xy^2z$

000000000001111111

$11 \neq 7$

$\notin L$

- Case 2: The 'y' is in the 'one' part

$xy^iz \Rightarrow xy^2z$

$\underbrace{0000000111}_{x}\ \underbrace{1111}_{y}\ \underbrace{1}_{z}$

000000011111111111

$7 \neq 11$

$\notin L$

- Case 3: The 'y' is in the '0' and '1' part

$xy^iz \Rightarrow xy^2z$

$\underbrace{00000}_{x}\ \underbrace{00\ 11}_{y}\ \underbrace{11111}_{z}$

0000000110011111111

$\notin L$

$|xy| \leq P$ $\quad$ $P=7$

∵ We got the contradiction

∴ Language L is not regular.

(iii) $L = \{0^i 1^j \mid i > j\}$

Ans:

Proof:

- Assume that L is Regular.
- Then it must have a pumping length = P

$S = 0^{P+1} 1^P$

x y z

$\Rightarrow S = 000000001111111$

P = 7

$xy^iz \Rightarrow xy^2z$

$\underbrace{0000000011}_{x}\ \underbrace{1111}_{y}\ \underbrace{1}_{z}$

00000000111111111111

$8 \ngtr 12$

no. of 0's $\ngtr$ no. of 1's

∵ We got the contradiction.

∴ Language L is not Regular.

$|y| > 0$

$|xy| \leq P$ $\quad$ $P=7$