Assignment on (Module-II)

Course: B. Tech (CSE) **Year/Semester: 3rd , V** **Session: 2022-23**

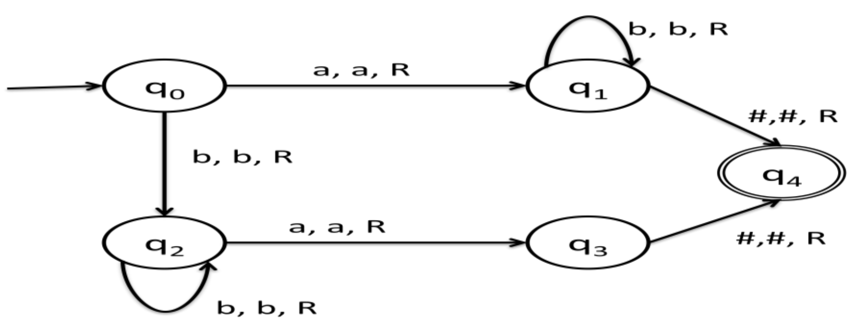
Subject Name & Code: Theory of Automata & Formal Languages (BCSC0011)

Faculty: Dr. Sandeep Rathor

1. Write the comparative study of DFA, PDA and TM.
2. Design a TM to compute the function f(n) = 2n.
3. Design a TM to recognize the strings $(ab)^{+}$.
4. Design a TM for L={$a^n b^m$/m>n}.
5. Design a TM for L={$a^n b^n c^n$/n≥0}.
6. Design a TM for L={w#w / w∈$(0,1)^*$}.
7. Design a TM for L={ww^R/ w∈$(0,1)^*$}.
8. Design a TM for calculate 2's complement of an input binary number.
9. Design a TM for subtraction of two unary numbers i.e f(m,n)=m-n.
10. Design a TM to compute the function f(x, y) = x + y where x, y > 0.Write the sequence of states that your TM will visit when started on the following Tape.

…	1	1	0	1	1	1	…

11. Consider the following Turing Machine M. The reject state qrej is not shown, and all "missing" transitions are assumed to go to the reject state.



- What is $\delta(q_2, a)$?
- Describe the computation of M on input ab.
- What is the language recognized by M?

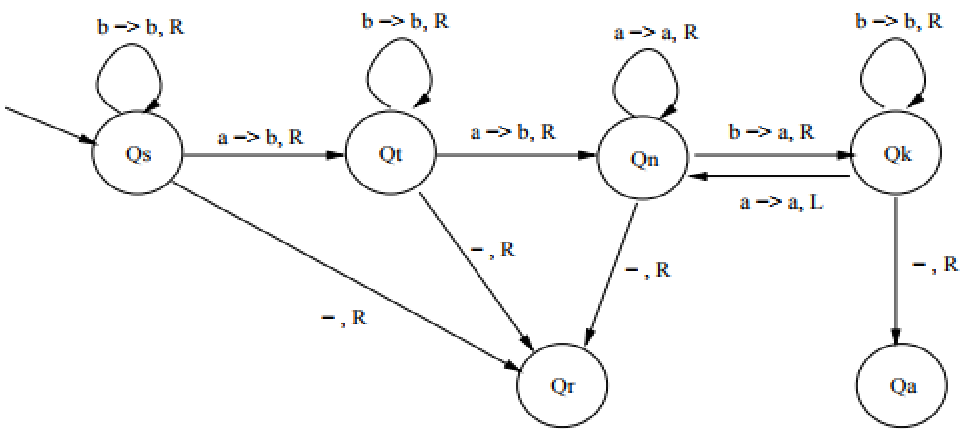
12. Discuss variants of Turing machine with transition function and block diagram of each.

13. Design a TM to compute multiplication of two unary numbers i.e f(m,n)=mxn.

14. Discuss the following:

    - Halting problem of TM.
    - Church thesis.
    - Recursive and recursive enumerable languages
    - Different complexity classes.
    - Turing machines with Stay options
    - Non deterministic Turing Machine

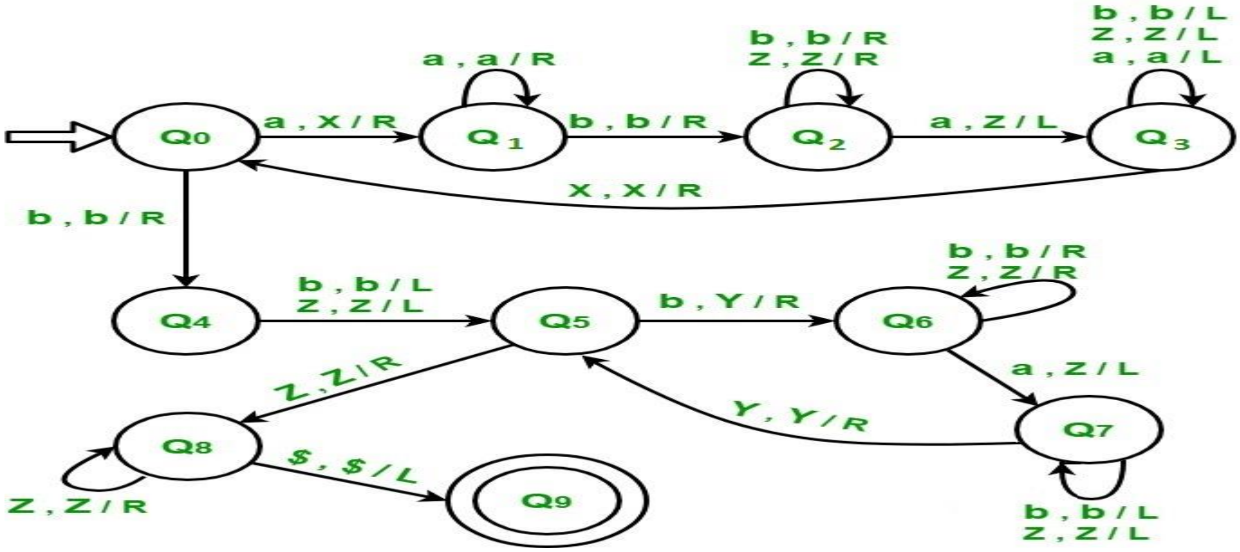
15. Does the following TM accept the string abba?



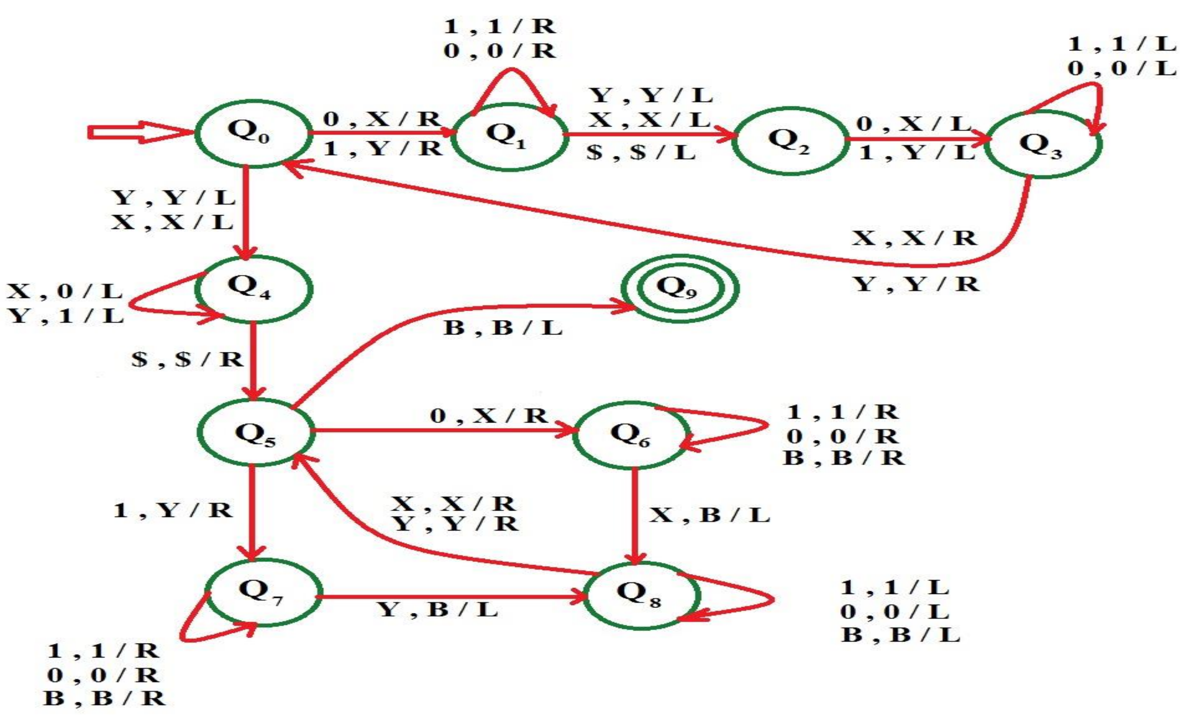
16. Give the complete description of the Turing Machine M (i.e. define all the 7 tuples) that accepts the language 10* + 01*.

Some solved problem for Revision purpose

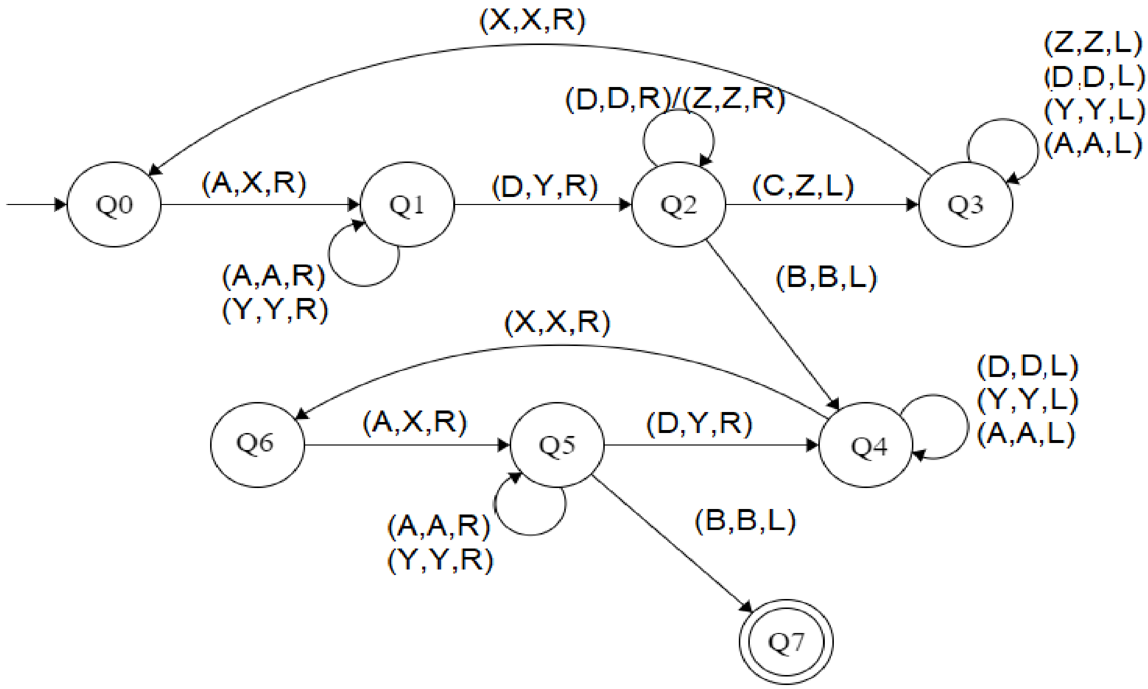
1. Construct Turing machine for L = $\{a^n b^m a^{(n+m)} \mid n, m \geq 1\}$



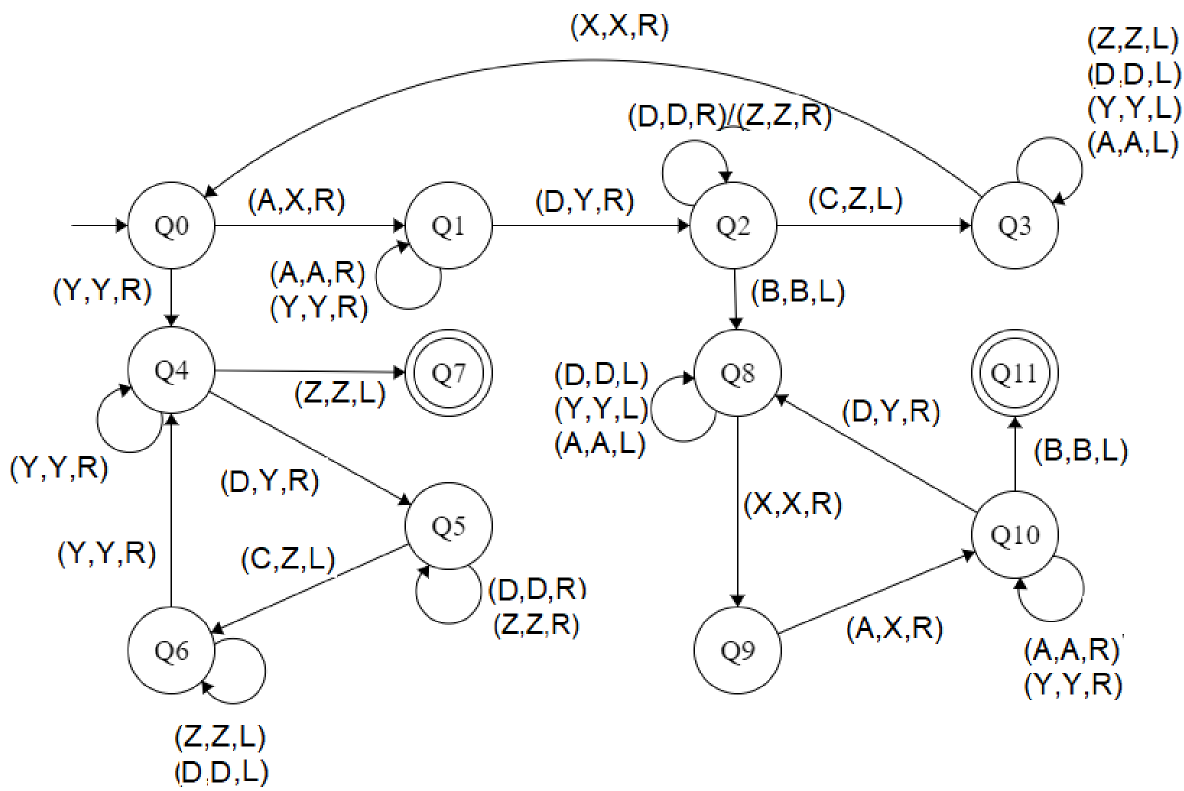
2. Construct a Turing Machine for language L = $\{ww \mid w \in \{0,1\}\}$



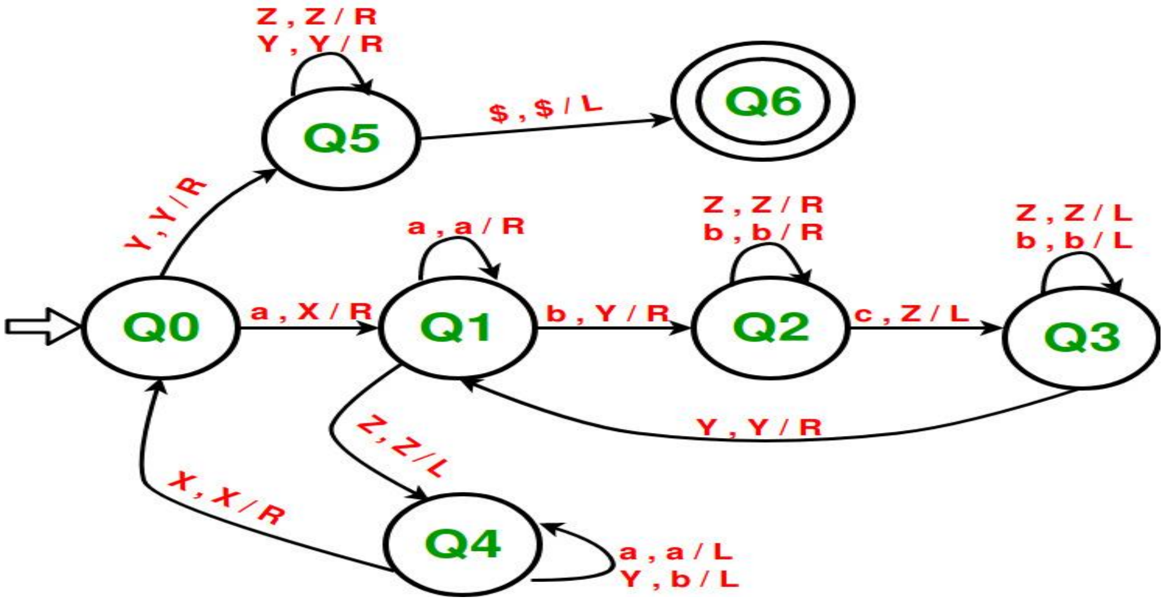
3. Construct a Turing machine for $L = \{a^i b^j c^k \mid i>j>k;\ k \geq 1\}$



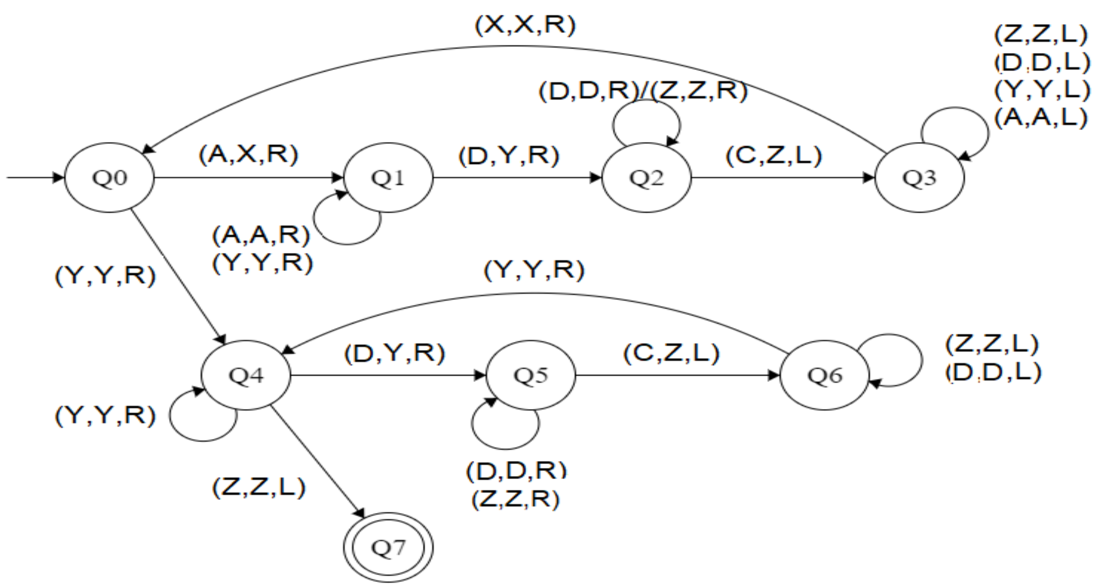
4. Construct a Turing machine for $L = \{a^i b^j c^k \mid i < j < k \text{ or } i > j > k\}$



5. Construct a Turing machine for $L = \{a^i b^j c^k \mid i*j = k;\ i, j, k \geq 1\}$



6. Construct a Turing machine for $L = \{a^i b^j c^k \mid i< j< k; i \geq 1\}$



7. **Decidable and Undecidable problems** : A problem is said to be Decidable if we can always construct a corresponding algorithm that can answer the problem correctly. We can intuitively understand Decidable problems by considering a simple example. Suppose we are asked to compute all the prime numbers in the range of 1000 to 2000. To find the

solution of this problem, we can easily devise an algorithm that can enumerate all the prime numbers in this range.

Now talking about Decidability in terms of a Turing machine, a problem is said to be a Decidable problem if there exists a corresponding Turing machine which halts on every input with an answer- yes or no. It is also important to know that these problems are termed as Turing Decidable since a Turing machine always halts on every input, accepting or rejecting it.

Undecidable Problems

The problems for which we can't construct an algorithm that can answer the problem correctly in finite time are termed as Undecidable Problems. These problems may be partially decidable but they will never be decidable. That is there will always be a condition that will lead the Turing Machine into an infinite loop without providing an answer at all.

We can understand Undecidable Problems intuitively by considering Fermat's Theorem, a popular Undecidable Problem which states that no three positive integers a, b and c for any n>2 can ever satisfy the equation: a^n + b^n = c^n.

If we feed this problem to a Turing machine to find such a solution which gives a contradiction then a Turing Machine might run forever, to find the suitable values of n, a, b and c. But we are always unsure whether a contradiction exists or not and hence we term this problem as an Undecidable Problem.

Examples – These are few important Undecidable Problems:

Whether a CFG generates all the strings or not?

As a CFG generates infinite strings, we can't ever reach up to the last string and hence it is Undecidable.

Whether two CFG L and M equal?

Since we cannot determine all the strings of any CFG, we can predict that two CFG are equal or not.

Some more Undecidable Problems related to Turing machine:

- Finiteness of a Turing Machine?
- Emptiness of a Turing Machine?
- Whether the language accepted by Turing Machine is regular or CFL?