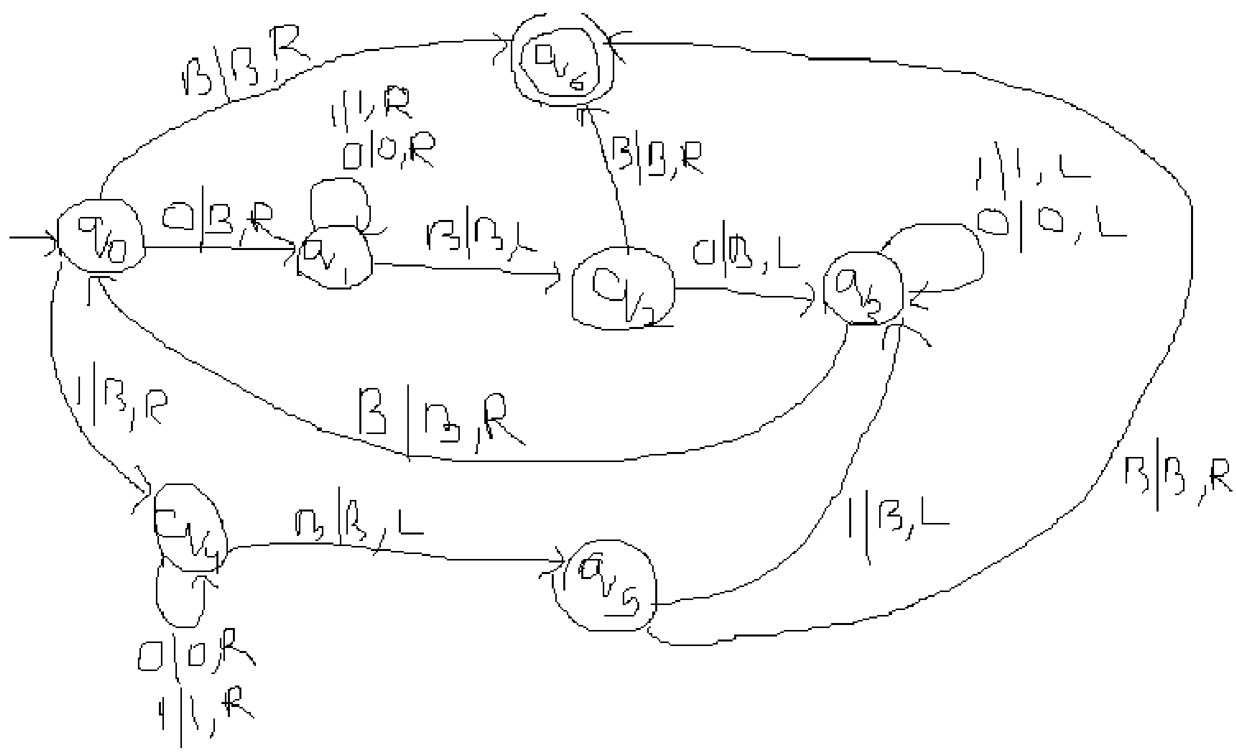
IAT5 Solution

Sub: Automata Theory and Computability (18CS54)

Department of Computer Sc. and Engg.

Q.1. Design a Turing Machine for L= {W is a Palindrome over ∑ ={0,1}}. Write the transition function for the same and also indicate the moves made by TM for input string W=10101

Ans:



ID for **W=10101**

q_0, 10101B|-- B$q_4$0101B |--B0 $q_4$101B |--B01 $q_4$01B |--B010 $q_4$1B |--B0101 q_4B |--B010 $q_5$1B |--B01 $q_3$0BB |-- B0 $q_3$10BB|-- B $q_3$0 10BB|-- q_3B010BB|-- B $q_0$010BB|-- B B $q_1$10BB

|-- B B 1 $q_1$0BB|-- B B 10 q_1BB|-- B B 1 $q_2$0BB|-- B B $q_3$1BBB |-- B q_3B 1BBB
|-- B B q_0 1BBB |-- B BB q_4BBB |-- B B q_5B BBB|-- B BB q_6BBB **<u>Accepted</u>**

Q.2.What is NULL production, useless symbol and Unit production? Explain with an example. Eliminate NULL production, useless symbol and Unit production from the following grammar.

S → ABC
A → BC | a
B → bAC | ϵ
C → cAB | ϵ

Ans:

- Any production rule in the form A → B where A, B ∈ Non-terminal is called **unit production.**
- In a CFG, a non-terminal symbol **'A'** is a nullable variable if there is a production **A** → **ε** or there is a derivation that starts at **A** and finally ends up with ε: A → ……… → ε
- **Useless productions** – The productions that can never take part in derivation of any string, are called useless productions. Similarly, a variable that can never take part in derivation of any string is called a useless variable or **useless symbol.**

Step1: Remove NULL productions

Null set={S,A,B,C}

S → ABC | AB |AC | BC | A | B | C
A → BC | a | B | C

B → bAC | bA | bC | b
C → cAB | cA | cB | c

Step1: Remove useless and unit productions

No useless symbols.

After removing Unit production.

S → ABC | AB |AC | BC | bAC | bA | bC | b | cAB | cA | cB | c|a
A → BC | a | bAC | bA | bC | b | cAB | cA | cB | c

B → bAC | bA | bC | b
C → cAB | cA | cB | c

Q.3. Write Short notes on:

(a) Decidability and Undecidability

(b) Order of growth

Decidability and Undecidability

Decidable language -A decision problem P is said to be decidable (i.e., have an algorithm) if the language L of all yes instances to P is decidable.
Example- (I) (Acceptance problem for DFA) Given a DFA does it accept a given word?

(II) (Emptiness problem for DFA) Given a DFA does it accept any word?

(III) (Equivalence problem for DFA) Given two DFAs, do they accept the same language?

Undecidable language -– A decision problem P is said to be undecidable if the language L of all yes instances to P is not decidable or a language is undecidable if it is not decidable. An undecidable language maybe a partially decidable language or something else but not decidable. If a language is not even partially decidable, then there exists no Turing machine for that language.
Partially decidable or Semi-Decidable Language -– A decision problem P is said to be semi-decidable (i.e., have a semi-algorithm) if the language L of all yes instances to P is RE. A language 'L' is partially decidable if 'L' is a RE but not REC language.

Order of growth:

$$g(n) = (3)^2$$

$$g(n) = 9$$

i.e. $f(n) < g(n)$ is true.

Hence we can conclude that for $n > 2$, we obtain

$$f(n) < g(n)$$

Thus always upper bound of existing time is obtained by big oh notation.

7.6.1 Order of Growth

Measuring the performance of an algorithm in relation with the input size n is called order of growth. For example, the order of growth for varying input size of n is as given below.

n	log n	n log n	n^2	2^n
1	0	0	1	2
2	1	2	4	4
4	2	8	16	16
8	3	24	64	256
16	4	64	256	65,536
32	5	160	1024	4,294,967,296

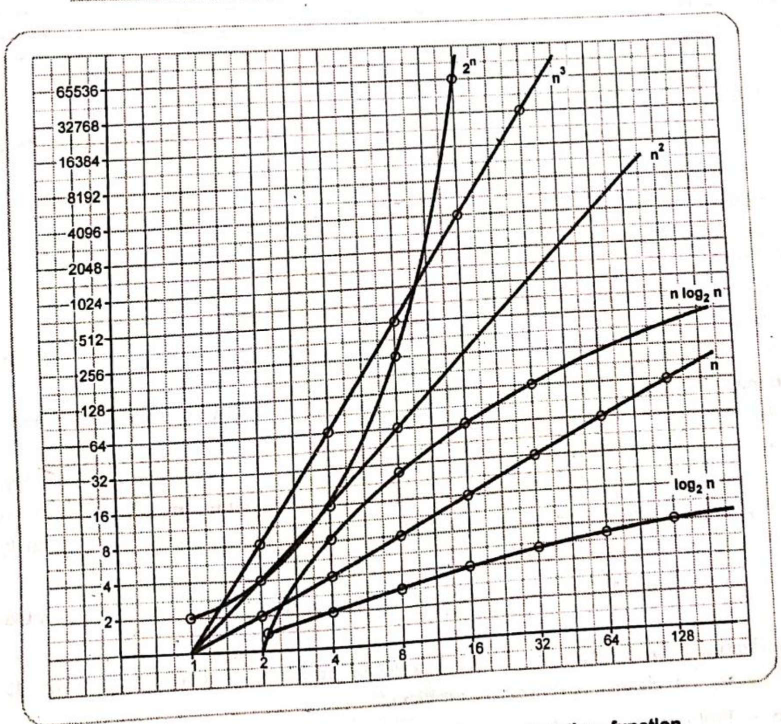


Fig. 7.6.2 Rate of growth of common computing time function

Q.4. Define Chomsky Normal Form (CNF). Convert the following CFG to CNF.

S→ ASB | ϵ
A→ aAS | a
B→ SbS | A | bb

Ans: CNF stands for Chomsky normal form. A CFG(context free grammar) is in CNF(Chomsky normal form) if all production rules satisfy one of the following conditions:

- A non-terminal generating two non-terminals. For example, S → AB.
- A non-terminal generating a terminal. For example, S → a.

Removing NULL productions:

Null set ={S}

S→ ASB | AB
A→ aAS | a | aA
B→ SbS | A | bb | Sb | bS | b