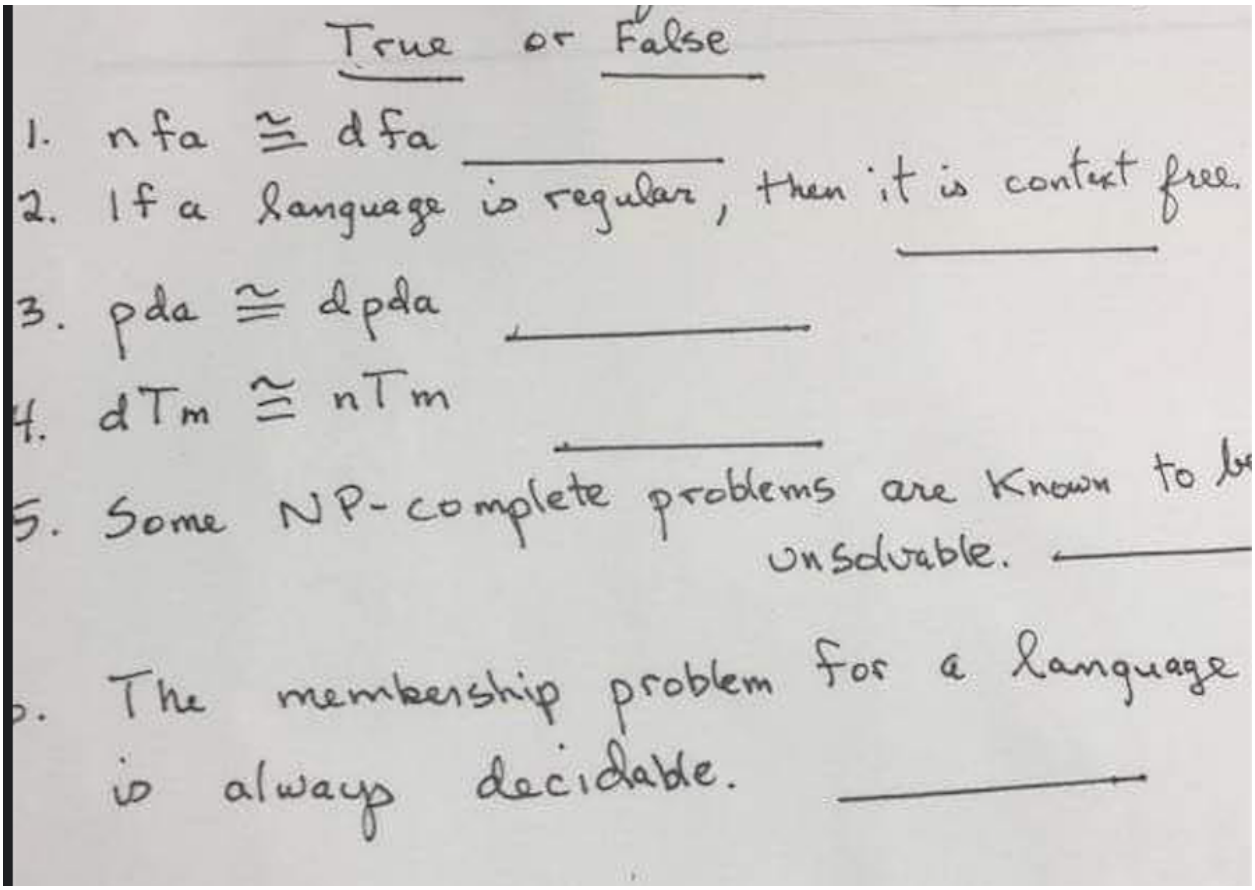
Question1



1)TRUE- A language is accepted by a DFA if and only if it is accepted by NFA.

DFA is an special case of NFA.

2)TRUE- Any language that can be generated using regular expressions can be generated by a context-free grammar, but not all context-free languages are regular languages.

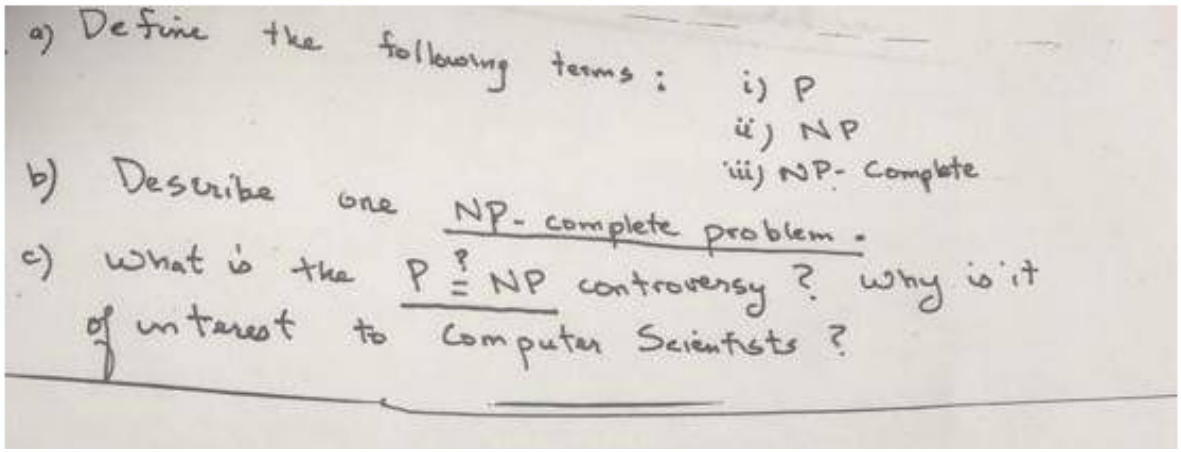
3)FALSE- A deterministic pushdown automaton has at most one legal transition for the same combination of input symbol, state, and top stack symbol. This is where it differs from the nondeterministic pushdown automaton.

4)TRUE- every non deterministic TM has an equivalent deterministic TM.

5)FALSE:- NP-**complete** problems are exponential in number of operations,they are verified in polynomial time.

6)TRUE

where the answer is yes or no can be phrased as a *membership problem* in a *language*. The *language* is the set of strings for which the answer is yes.
the set of strings are always finite.



P:-
The Problems which are solved with an algorithm runs in a polynomial time are comes under the class P.

NP:-
More number of programs don

- **Describe P,NP, and NP-Complete Problems and show one example for each.**

NP-complete is a class of problems which are in NP and are NP-hard. NP-complete problems transform to each-other by polynomial-time many-one reductions so if a polynomial-time algorithm exists for any one of them, then polynomial algorithms exist for all of them. However, no polynomial algorithm for any NP-complete problem has yet been found.
For example, the clique problem discussed above is NP-complete

Question 4?

PCP
A- Describe post's correspondence problem.
B- Is post's correspondence problem solvable? Explain.
C- Given the following:
A = (a , abaaa , ab) and B = (aaa , ab , b)
As two strings, provide a solution to PCP.

A) post's correspondence problem:

It is an undecidable decision problem.PCP is a NP- complete.

In PCP, ->consists of two lists of strings of alphabets.

Example:

A=a1,a2,a3,...an;

B=b1,b2,b3,..bn;

for n>0;

B)The instance of PCP has a solution, if the sequence of A and B are in equal length and yields,

same string.

C)

solution:

A=(a,abaaa,ab) B=(aaa, ab,b)

x1 x2 x3

A a abaaa ab

B aaa ab b

Here,

length of A= 8

length of B = 6

lx2x1x3l ? ly2y1y3l , length is not same.

SO, this Post Correspondence problem is undecidable

//for any calrification, please do comments

a) Describe briefly the concept of a Universal Turing machine (UTM).

b. State the Halting problem and explain briefly why the problem is undecidable.

c. Discuss the Turing - Church Thesis and comment on its significance.

d. Why are undecidability results of interest to Computer Scientists?

Universal Turing Machine
or
General Turing Machine

Basicaly this is just like a child, who will do whatever we do in front of that child.

Eg IF you laugh child laugh

Similarly UTM is a turing machine lets say M.

we gave it two things

1. A turing machine m_1
2. An input to m_1

how UTM i.e. M will behave.

⇒ If m_1 Halt for w, M will also halt

⇒ If m_1 loop for w, M will also loop

NOTE: All we get to know is that universal turing machine is an immitating machine.

Halting Problem

IT ASKS:

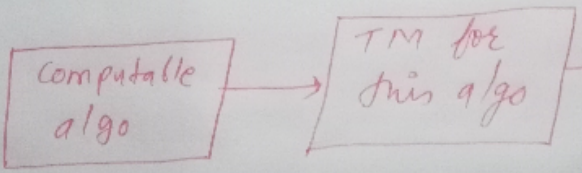
Is it possible to tell whether a given machine will Halt for some given input.

Halting means that the program on certain input will accept it and halt or reject it and halt, and it would never go into an infinite loop.

This is an undecidable problem because we can not design an algorithm which will tell us whether a given program will halt or not.

Church-Turing Thesis

This states about relation in computation and turing machine.



This basically says that if there exist any algorithm which do certain computation then there also exist a turing machine to do the same computation.

Significance:-

- Difficult computation calculation made easy
- less time required to compute.
- Multi computation at a time possibility
- Man work reduced

UNIVERSAL TIMING MACHINE-

The concept of a time machine typically conjures up images of an implausible plot device used in a few too many science-fiction storylines. But according to Albert Einstein's general theory of relativity, which explains how gravity operates in the universe, real-life time travel isn't just a vague fantasy.

Traveling forward in time is an uncontroversial possibility, according to Einstein's theory. In fact, physicists have been able to send tiny particles called muons, which are similar to electrons, forward in time by manipulating the gravity around them. That's not to say the technology for sending humans 100 years into the future will be available anytime soon, though.

Time travel to the past, however, is even less understood. Still, astrophysicist Eric W. Davis, of the EarthTech International Institute for Advanced Studies at Austin, argues that it's possible. All you need, he says, is a wormhole, which is a theoretical passageway through space-time that is predicted by relativity.

HALTING PROBLEM-

In computability theory, the halting problem is a decision problem which can be stated as follows: Given a description of a program, decide whether the program finishes running or continues to run, and will thereby run forever. This is equivalent to the problem of deciding, given a program and an input, whether the program will eventually halt when run with that input, or will run forever.The halting problem is a decision problem about properties of computer programs on a fixed Turing-complete model of computation. The problem is to determine, given a program and an input to the program, whether the program will eventually halt when run with that input. In this abstract framework, there are no resource limitations of memory or time on the program's execution; it can take arbitrarily long, and use arbitrarily much storage space, before halting. The question is simply whether the given program will ever halt on a particular input.

For example, in pseudocode, the program does not halt; rather, it goes on forever in an infinite loop. On the other hand, the program

halts very soon.

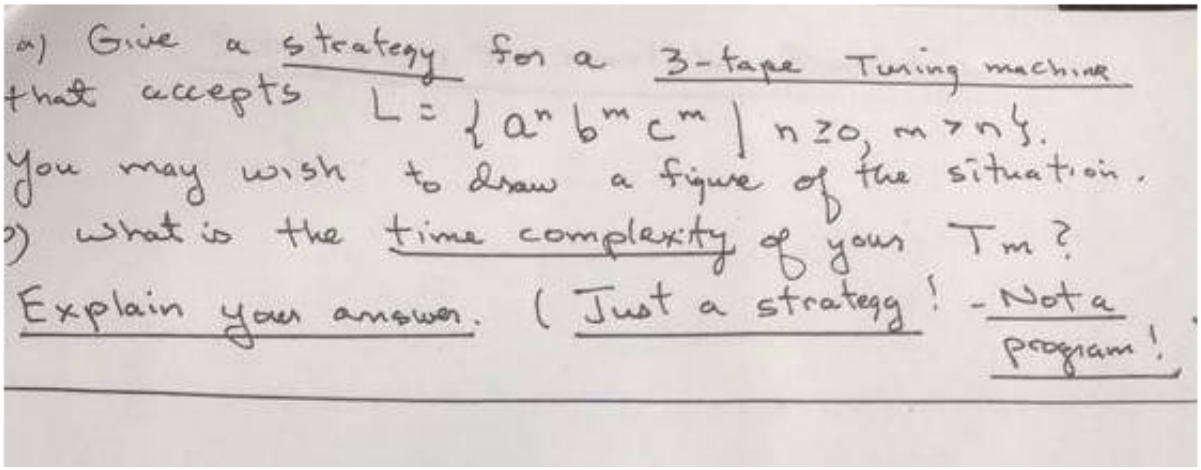
The halting problem is famous because it was one of the first problems proven algorithmically undecidable. This means there is no algorithm which can be applied to any arbitrary program and input to decide whether the program stops when run with that input.

TIMING CHURCH THEORY-

It is commonly believed that the Catholic Church persecuted Galileo for abandoning the geocentric (earth-at-the-center) view of the solar system for the heliocentric (sun-at-the-center) view.

The Galileo case, for many anti-Catholics, is thought to prove that the Church abhors science, refuses to abandon outdated teachings, and is not infallible. For Catholics, the episode is often an embarrassment. It shouldn

Question 6?



**************Part a****************

Let 's' be the input string.
This problem can be divided into two sub problems.

1. We will check whether $|s|_a < |s|_b$
2. We will check whether $|s|_b = |s|_c$

First tape is the input string.
Initialize second and third tape with symbol X_o to symbolize the left end.

Sub-problem 1 : to check whether $|s|_a < |s|_b$

1. For each 'a' in s:
   - move first tape head to right
   - change symbol to x under second tape head and move right
   - nothing for third tape
2. When b is encountered on first tape state is changed
3. In next state
   - For each 'b' in s:
     - move first tape head to right
     - change symbol to B(blank) under second tape head and move left
     - change symbol to y under third tape head and move right
     - if there are no 'b' left in 's' and second tape has reached it's left end **halt and reject** (as this means that $|s|_a = |s|_b$)
       - else change state in which
       - For each 'b' in s:
         - change symbol to y under third tape head and move right
         - move first tape head to right

Now third tape consist of as many y as there are b in input string 's'

Sub-problem 2 : to check whether $|s|_b = |s|_c$

1. For each 'c' in s:
   - move first tape head to right
   - change symbol to B(blank) under third tape head and move left
   - nothing for second tape
   - if there are no 'y' left in 's' and third tape has reached it's left end **halt and accept**
     - else in all other cases **halt and reject**

This is just the strategy which can be used to solve this problem, proper states need to be chosen to perform the proper operations.

***************Part b****************

The time complexity of this turing machine is linear.

There is no extra movement for any of the input symbol, all the movements are corresponding to the input symbols.

So, **time complexity = O(n)**

Question7?
Question8?
Question9?
Question10?