Practice Exam

SUBJECT: Theoretical Foundation of Computer Science 300 / 552

Index No: 12334 / 302976

TIME ALLOWED:

Two (2) Hour examination preceded by a 10-MINUTE READING PERIOD during which time only notes can be made. The supervisor will indicate when answering may commence.

AIDS ALLOWED:

To be supplied by the Candidate: Nil

To be supplied by the University: Nil

Calculators are not allowed.

GENERAL INSTRUCTIONS

This paper consists of four (4) questions with a total of 100 marks.

QUESTION ONE: Classification by Computability

PROBLEMs 1 to 4 on the following page each describe a problem in English, set notation or in terms of strings. It is your task to do the following, where applicable, for each of the four problems:

1. Classify the problem into one the appropriate category; Regular, Context-Free, Turing Decidable, or Undecidable. **(2 marks)**
2. For a problem that is <u>Regular</u>, prove that this is the case by constructing either a DFA, a NFA, or a Regular Expression that accepts the language of the problem. **(2 marks)**
3. For a problem that is <u>Context-Free</u>, do both of the following:
    a. Prove that the problem is not Regular using the pumping lemma, and **(5 marks)**
    b. Prove that the problem is Context-Free by constructing either a PDA or CFG that accepts the language of the problem. **(5 marks)**
4. For a problem that is <u>Turing Decidable</u> do all of the following:
    a. Prove that it is not Context-Free using the pumping lemma for Context-Free grammars, and **(6 marks)**
    b. Prove that the problem is Turing Recognizable by constructing a Turing Machine, and **(6 marks)**
    c. Prove that the Turing Machine constructed is a decider. **(2 marks)**
5. For a problem that is Undecidable, prove this using reduction from A_{TM}. **(10 marks)**

If you are unable to do a pumping lemma proof, a small amount of marks may be awarded for a good explanation of why the problem is not Regular or Context Free.

You may choose to prove that something is not Regular (or Context-Free) using a method other than the pumping lemma, if you are sure that this form of proof is convincing. However, use of the pumping lemma is strongly recommended.

PROBLEM 1:
The language A ◙ B = $\{ab \mid a \in A, b \in B, |a| \leq |b|\}$ where A and B are Regular languages.

PROBLEM 2:
SPANNING = {<G>: Graph G = (V,E) has a spanning tree} where a spanning tree T is a subset of E such that for every vertex *v* in V, at least one edge in E has *v* as an endpoint and T contains no cycles (loops). For example, if V = {0,1,2} then a spanning tree might be {(0,1),(1,2)} or {(0,1),(2,0)} but not {(0,1),(1,2),(2,0)} since the latter contains a cycle ($0 \rightarrow 1 \rightarrow 2 \rightarrow 0$).

PROBLEM 3:
The controller for a traffic light receives input from a pressure pad on the minor road near the intersection that is connected to a timer. Normally, the timer is off and the signal is green for the major road. When the pressure pad is activated and the timer is off, the controller received the signal 01 and switches the signal on the major road to amber. Once the timer has expired it resets and sends the signal 10, at which point the major road changes to red and the timer starts again. When the timer completes for the second time, the signal 11 is sent and the main light remains red. At this point the timer resets again; once this completes the signal 00 is sent and the main light changes to green again. In an unexpected input is received, the traffic lights begin to flash and Main Roads is notified.

The controller for the traffic light controlling the minor road is managed by a separate controller which you do not need to consider.

PROBLEM 4:
For a ternary code ($\Sigma = \{a,b,c\}$), find all strings of the form $\{ a^n b^{2n} c^n \}$, $n>0$.

(**Total marks for this question:** ***48 marks***)

QUESTION THREE: Undecidability

For one of the two problems below, prove that it is undecidable by reduction from A_{TM}. Answer only one of the two problems. If answers to both problems are attempted, only the first will be marked.

PROBLEM 1:

$REGULAR_{TM} = \{<M>| \ M \text{ is a TM and } L(M) \ \}$.

PROBLEM 2:

Define $E_{TM} = \{(M)| \text{ M is a TM and } L(M) = \phi\}$. Show that E_{TM} is undecidable.

(**Total marks for this question:** ***18 marks***)

QUESTION TWO: Complexity

For one of the two problems below, prove that it is NP-complete by reduction from SAT, 3-SAT, VERTEX-COVER or 3-COLOR. Answer only one of the two problems. If answers to both problems are attempted, only the first will be marked.

PROBLEM 1:

A cyber security taskforce is looking at the structure of a wide area network to determine which parts are the most vulnerable to threats. It is determined that computers that are directly inter-connected with a large number of other computers are most at risk, and thus such sections in the network are to be located. The main security coding team is tasked with writing an application that can locate all such cliques in a network.

PROBLEM 2:

A $n \times n$ game board has each field randomly filled with a black counter, a white counter, or nothing at all, with each being equally likely. On each turn, one counter is removed from the board. A game is won when each column contains only white counters or black counters and is lost when a row becomes empty.

Let *BLACK-AND-WHITE* = { <*G*> | *G* is a winnable game configuration }.

(**Total marks for this question:** ***18 marks***)

QUESTION FOUR: Short Answer

For each of the statements below, say whether it is true or false and explain your reasoning. No marks will be given for an answer that does not include an appropriate level of reasoning.

STATEMENT 1:
A problem is known to be verifiable in polynomial time but there is no known non-deterministic polynomial time algorithm to solve it. This problem is in NP.

(3 marks)

STATEMENT 2:
A problem has been shown to have a solution with O(n^3) time complexity on a 3 tape deterministic Turing machine. We can conclude that this problem is in P.

(3 marks)

STATEMENT 3:
An algorithm with time complexity O(n^3) will always take longer to complete than one with time complexity O($n \log n$).

(3 marks)

STATEMENT 4:
A new research paper claims to have proved that a problem known to have a polynomial-time average time complexity is NP-Complete. This is sufficient for proving that P = NP. **(3 marks)**

STATEMENT 5:
If a client asks you to develop a program for a large instance of a problem that you know to be NP-complete you have to refuse because such a program is only feasible for reasonably small inputs. **(4 marks)**

(**Total marks for this question: *16 marks***)

END OF EXAMINATION