UNIVERSITY OF DUNDEE

University of Dundee
School of Science and Engineering
Examinations 2017

BSc Degrees in Computing

AC32008 Theory of Computation

Time allowed: **TWO** hours

Instructions

There are **SIX** questions.

Candidates must answer **FOUR** questions. All questions carry equal marks. Where appropriate, the value of each part of a question is given in square brackets.

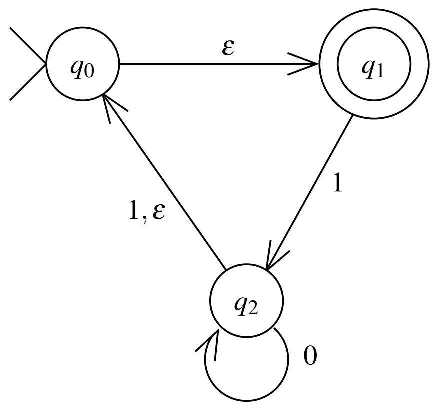
Approved calculators may be used in this examination.

Do not turn over this question paper until instructed to by the Senior Invigilator

1. **(a)** Give the definition of a non-deterministic finite automaton with ε-transitions (NFA-ε). What is meant by the $\varepsilon-$CLOSURE of a state? Explain, without formal definitions, what it means to say that an NFA-ε M accepts an input w. **[8 marks]**

   (b) Let L be the language over $\{0,1\}$ consisting of all binary strings which do not contain three consecutive symbols which are all the same. Find a regular expression $\mathbf{r}$ which represents the language L, i.e., such that $L(\mathbf{r}) = L$, and justify your answer. **[8 marks]**

   (c) Suppose that M is an NFA with ε-transitions. Explain how to obtain an NFA (without ε-transitions) M' which accepts the same language as M. Illustrate by finding an NFA which accepts the same language as the NFA-ε shown below (a transition diagram is sufficient). **[9 marks]**



2. **(a)** State the Pumping Lemma for regular languages. **[4 marks]**

   (b) Let L be the language over the alphabet $\{0,1\}$ given by

$$L = \{\, 0^i 1 0^j 1 0^{ij} \mid i, j \geq 0 \,\}$$

   that is, the set of binary strings with three blocks of zeroes, the length of the third block being the product of the length of the first two blocks (e.g. 0010001000000). Show that L is not regular. **[12 marks]**

   (c) Explain why every finite language is regular. **[9 marks]**

3. **(a)** Describe the operation of a standard one-tape deterministic Turing Machine M. **[5 marks]**

(b) Let L be the language over the alphabet $\{0,1\}$ given by

$$L = \{\, 0^i 1 0^j 1 0^{ij} \mid i, j \geq 0 \,\}$$

that is, the set of binary strings with three blocks of zeroes, the length of the third block being the product of the length of the first two blocks (e.g. 0010001000000). (This is the same language as in Question 2, part (b).)

Sketch the construction of a standard Turing Machine M which accepts the language L. (It is not necessary to give full details of the machine M; an informal description of its operation is sufficient provided it is convincing.) **[8 marks]**

(c) Explain how a Turing Machine M can be regarded as generating a language $G(M)$. **[3 marks]**

(d) Show that if a language L can be generated in standard order by some Turing Machine M (i.e. $L = G(M)$), then L is totally decidable. **[9 marks]**

4. **(a)** Give a standard code for a Turing Machine with input alphabet $\{0,1\}$, Tape alphabet $\{0,1,B\}$, states $Q = \{q_1,q_2\}$, final state q_2, and transition function δ given by:

δ	0	1	B
q_1	(q_1,B,L)	$(q_2,0,L)$	(q_2,B,R)
q_2	$-$	$-$	$-$

[6 marks]

(b) Define L_h to be the language

$$\{w_i | M_i \text{ halts on input } w_i\}.$$

Determine whether or not the string 11101001010010111 is in L_h. **[5 marks]**

(c) Define the language L_{halt}. Show that L_h is not recursive, and deduce that L_{halt} is undecidable. **[10 marks]**

(d) Say briefly what is meant by the Church-Turing thesis, and indicate three kinds of arguments which can be used to support it. **[4 marks]**

5. **(a)** Describe how the operation of a non-deterministic Turing machine (NDTM) differs from that of a deterministic Turing machine (DTM), and say what it means for a string x to be accepted by an NDTM M. Also define the time complexity function $t_M(n)$ and define the class $\mathbf{NP}$. **[8 marks]**

(b) Suppose that L_1, L_2 are binary languages. What does it mean to say that f is a polynomial transformation from L_1 to L_2? Give an example. **[7 marks]**

(c) Suppose that L_1 and L_2 are languages. Show that if $L_1 \in \mathbf{P}$ and $L_2 \propto L_1$, then $L_2 \in \mathbf{P}$. What is the significance of this result? **[10 marks]**

6. **(a)** A graph is k-colourable, for some $k \geq 1$ if it is possible to colour the vertices of the graph using k distinct colours $c_1, \ldots, c_k$ such that whenever two vertices v, w are joined by an edge, they have different colours.

The problem k-COLOURING is as follows:

k-COLOURING

Input: Graph G.

Question: Is G k-colourable?

Show that 3-COLOURING is $\mathbf{NP}-\text{complete}$. (You may assume that 3-SATISFIABILITY is $\mathbf{NP}-\text{complete}$.) **[12 marks]**

(b) Show that k-COLOURING is $\mathbf{NP}-\text{complete}$ if $k \geq 3$. **[6 marks]**

(c) Consider the following computational problem:

MINIMUM COLOURING

Input: Graph G.

Problem: What is the least integer k such that G is k-colourable?

Explain how a polynomial time algorithm for the decision problem k-COLOURING would allow the problem MINIMUM COLOURING to be solved in polynomial time.

[7 marks]

END OF PAPER