UNIVERSITY OF DUNDEE

BSc EXAMINATION
School of Computing
2 hours

May 2014

Theory of Computation (AC32008)

This paper contains SIX questions.
Answer FOUR questions only.

Only calculators approved by the School of Computing for exam use may be used in this exam.

1. **(a)** Define 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 x. **[9 marks]**

(b) Let L be a language over $\{0,1\}$ consisting of all binary strings which do not contain two consecutive zeros. Find a regular expression $\mathbf{r}$ which represents the language L, i.e. so that $L(\mathbf{r}) = L$, and justify your answer. **[7 marks]**

(c) Sketch a proof that for any DFA M, there is a regular expression $\mathbf{r}$ which represents the language accepted by M, i.e. $L(\mathbf{r}) = L(M)$. In particular, say what is meant by the language R^k_{ij} and the regular expression $\mathbf{r}^k_{ij}$ and explain why

$$\mathbf{r}^k_{ij} = \mathbf{r}^{k-1}_{ij} + \mathbf{r}^{k-1}_{ik}(\mathbf{r}^{k-1}_{kk})^*\mathbf{r}^{k-1}_{kj}.$$

[9 marks]

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

(b) Let L be the binary language consisting of all strings in which the longest string of consecutive zeros is the same length as the longest string of consecutive ones. Thus, for example, the following strings are in L:

0101010	(1 consecutive zero, 1 consecutive one)
111100001	(4 consecutive zeros, 4 consecutive ones)

while (for example) the following strings are not in L:

0	(1 consecutive zero, 0 consecutive ones)
111100000	(5 consecutive zeros, 4 consecutive ones)

Show that L is not regular. **[12 marks]**

(c) Let L be a regular language. Say what is meant by the language L^*, and show that L^* is also regular. **[9 marks]**

3. **(a)** Describe the operation of a standard one-tape deterministic Turing Machine M and define the language $L(M)$ accepted by M. **[7 marks]**

(b) Describe the operation of a k-tape Turing machine (where $k \geq 2$). Give a brief outline of a proof that any language which can be accepted by a k-tape Turing machine can also be accepted by a standard one-tape Turing machine. Why is this result often useful? **[9 marks]**

(c) Construct a multitape Turing Machine M which accepts the language L_{pal} of all binary palindromes, i.e. all binary strings x such that $x = x^R$, where x^R is the reverse of x:

$$L_{\text{pal}} = \{x \,|\, x = x^R\}.$$

(It is not necessary to give full details of the machine M; an informal description of its operation is sufficient provided it is convincing. You may assume M has two-way infinite tapes if you wish.) **[9 marks]**

4. **(a)** Give a standard enumeration of binary strings. Explain how any binary string can be regarded as a description of a Turing Machine with tape alphabet $\{B, 0, 1\}$, input alphabet $\{0, 1\}$ and set of states $Q = \{q_1, \ldots, q_n\}$ for some n, where q_2 is the only final state, and use this to give an enumeration of all such Turing Machines.

[7 marks]

(b) Define the universal language L_u, and briefly explain how it can be shown that L_u is partially decidable. **[8 marks]**

(c) Let L_h be the following language:

$$\{w_i \,|\, M_i \text{ halts on } \ w_i\}$$

Show that L_h is not recursive. **[10 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 **NP**. **[8 marks]**

(b) What does it mean to say that f is a polynomial transformation from a language L_1 to another language L_2? Show that polynomial transformations are transitive, i.e. that if $L_1 \propto L_2$ and $L_2 \propto L_3$, then $L_1 \propto L_3$. **[8 marks]**

(c) Consider the following two decision problems.

HAMILTONIAN CIRCUIT

Input: Graph G.

Question: Does G have a Hamiltonian Circuit?

TSP

Input: A collection $c_1, \ldots, c_n$ of cities, and an integer distance $d(c_i, c_j)$ for each pair of cities, and an integer B.

Question: Is there an ordering $i_1, \ldots, i_n$ of the numbers $1, \ldots, n$ such that

$$d(c_{i_1}, c_{i_2}) + \cdots + d(c_{i_{n-1}}, c_{i_n}) + d(c_{i_n}, c_{i_1}) \leq B?$$

Given that HAMILTONIAN CIRCUIT is **NP**$-$complete, show that TSP is **NP**$-$complete.

[9 marks]

[Question 6 on next page]

6. **(a)** Define the term **NP**−complete.

Many problems (languages) have been shown to be **NP**−complete. Give a brief account of how this has been done, and why it is useful to know that that a problem is **NP**−complete. **[7 marks]**

(b) Show that any language in **NP** can be accepted by a *deterministic* Turing machine which uses at most exponentially many steps in its computation, i.e. for an input of length n, the computation takes at most $2^{q(n)}$ steps, where q is a fixed polynomial. **[8 marks]**

(c) Say what is meant by a k-colouring of the vertices of a graph.

The GRAPH COLOURING problem is defined as follows:

GRAPH COLOURING

Input: A graph G and an integer k.

Question: Is there a k-colouring of G?

Also consider the following problem:

TIMETABLING

Input: A set $E = \{e_1, \ldots, e_n\}$ of events (each lasting one hour), and a set of **clashes**, where a clash is a pair of events e_i, e_j which must not happen at the same time.

Question: Can all the events be scheduled in k time slots (each of one hour), so that no pair of events which form a clash are scheduled in the same time slot.

Explain how we could show that TIMETABLING is **NP**−complete. It is not necessary to give a formally defined reduction, but you should explain what reduction could be used, and why it would work. **[10 marks]**

End of examination paper