Birkbeck
(University of London)

MSc Examination

Department of Computer Science and Information Systems

Fundamentals of Computing (COIY058H7)

CREDIT VALUE: 15 credits

Date of Examination: Thursday, 19 May 2016
Time of Examination: 10:00 – 13:00
Duration of Paper: Three hours

*This paper is split into **two** sections.*
There are 50 marks for each section.
There are 5 questions in Section A and 5 questions in Section B.

Use a separate answer book for each section.

Answer all questions.
Start each answer on a new page.

The use of electronic calculators is not permitted.

The breakdown of marks per question is as follows:

Question	1	2	3	4	5	6	7	8	9	10
Marks	10	9	10	11	10	4	15	11	12	8

Section A

1. (a) Construct the truth-table for the Boolean function given by the formula

$$\neg(A_1 \to A_2) \vee (A_1 \wedge A_2) \vee (A_2 \wedge A_3).$$ **(3 marks)**

(b) Find a Boolean circuit with AND, OR and NOT gates only that computes the Boolean function in (a) above and contains as few gates as possible. **(4 marks)**

(c) Determine whether the formula in (a) is equivalent to the formula

$$A_1 \vee (A_2 \wedge (A_2 \to A_3)).$$

Show your working. **(3 marks)**

2. (a) Represent -44 as a 32-bit two's complement binary number. Show your working. **(3 marks)**

(b) Represent -15.375 as an IEEE 754 32-bit floating-point number. Show your working. **(3 marks)**

(c) Find the decimal number represented by the 32-bit word

1100 0010 1100 1010 0000 0000 0000 0000

assuming it is a single precision IEEE 754 floating-point number. **(3 marks)**

3. (a) Design a deterministic finite automaton A such that $L(A)$ consists of all words over the alphabet $\{0, 1\}$ that contain at least two 0's and end with 11. **(6 marks)**

(b) Find a regular expression representing the language over the alphabet $\{a, b\}$ consisting of strings that have exactly two or three b's and end with a. **(4 marks)**

4. (a) Give a context-free grammar for the language over the alphabet $\{a, b\}$ containing all words with at most three a's. Show the derivation of *abba* in your grammar. **(5 marks)**

(b) What is the language over $\{0, 1\}$ defined by the following context-free grammar with start variable S? **(5 marks)**

$$S \to TS, \quad S \to 1T, \quad S \to 1S$$
$$T \to TT, \quad T \to 0T1, \quad T \to 1T0 \quad T \to \varepsilon$$

Is this language regular? Give an informal explanation of your answer. **(1 mark)**

5. Consider the following $\mathbb{N} \to \mathbb{N}$ function f:

$$f(n) = \begin{cases} 4n+1 & \text{if } n \text{ is odd} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

(a) Give an implementation level description in English of a Turing machine that computes this function f. **(4 marks)**

(b) Give the complete transition table of this Turing machine. **(6 marks)**

Section B

6. (a) Are there any circumstances in which it would be preferable to represent a stack using a doubly-linked list rather than a singly-linked list? Give brief reasons for your answer. **(2 marks)**

   (b) What are the advantages and disadvantages of using sequential representation (an array) or linked representation (a linked list) to represent a queue? **(2 marks)**

7. (a) Draw the forest F represented by the list expression $(H(G(C(A), D)), X(P(K), S, T))$. In what order will the nodes of F be visited if it is traversed in

   (i) forest/tree preorder, (ii) forest/tree postorder?

   Draw the binary tree B corresponding to the forest F using the *natural correspondence* between forests and binary trees. **(5 marks)**

   (b) Consider the following binary tree traversal algorithm:

```
PointerStack S      //S is initially empty
while (true)
{  if (P ≠ nil)
   {  Visit(P)
      Stack(S,P)
      P ← P↑Rlink
   }
    else
   {  if Empty(S) return
      P ← Unstack(S)
      P ← P↑Llink
   }
}
```

   Trace the execution of this algorithm on the binary tree B that you constructed in part (a) and write down the order in which the nodes are visited. Does this have any relationship to any of the standard binary tree traversal orders? **(5 marks)**

   (c) Suppose, by mistake, the stack was replaced by a *queue*, and the `Stack` and `Unstack` operations by the `Queue` and `Unqueue` operations, respectively. In what order would the nodes of B now be visited?

   Is this order related to any of the standard traversal orders of B?

   What meaning does this order have if considered as a traversal of the forest F? **(5 marks)**

8. (a) Consider any three successive nodes X, Y and Z in the inorder traversal of a threaded binary tree, i.e. $Y = insucc(X)$, $Z = insucc(Y)$. There are four possible relationships between these three nodes in the tree depending on whether there is a thread from X to Y or one from Y to X, and similarly for Y and Z. Draw diagrams to illustrate each of the four cases. **(4 marks)**

   (b) In a *triply-linked* binary tree, as well as links to its two children, each node has a link to its parent. Write a function `insuccTrip(P)` that returns a pointer to the inorder successor of a given *leaf* node `P` in a triply-linked binary tree. **(5 marks)**

(c) When trying to print out in order the values in a *binary search tree*, a programmer by mistake uses preorder traversal instead of inorder. Nevertheless, the values still come out in the correct order. How is this possible? Illustrate your answer using a tree with 5 nodes. **(2 marks)**

9. (a) (i) Why is *directly-addressed* file organisation rarely used?

(ii) If *random* access is required but *key-ordered sequential* access is not required, would *random* access be faster using *directly-addressed* file organisation or using a *B-tree*? Give brief reasons for your answer.

(iii) Are there any circumstances in which *directly-addressed* file organisation would be suitable if *key-ordered sequential* access was also required? **(3 marks)**

(b) A B-tree of order 5 with a maximum of 4 records per data page currently contains 17 records. The keys are two-digit positive integers, e.g. 74.

(i) What is the *minimum* number of *pages* that there could currently be in the B-tree?
What is the *maximum* number of *pages* that there could currently be in the B-tree?
Draw a possible B-tree R0 with the *minimum* number of pages and a possible B-tree R1 with the *maximum* number of pages.
Note: You do not have to include any of the keys in your diagrams, but should indicate below each *data* page how many records it contains. **(5 marks)**

(ii) Write in suitable key values in the *pointer* pages of R0 and identify a new key value such that inserting a record with this key would cause the number of pages to increase by as much as possible. Draw the resulting B-tree R0*.
Note: In both R0 and R0* you should include the keys on all the *pointer* pages, but only include the keys on those *data* pages that are changed after the insertion. **(4 marks)**

10. (a) *Quicksort* uses a procedure that partitions the elements of a sub-array, $a[L], a[L+1], \ldots, a[U]$, around a pivot value t and assigns a value to the variable p, so that, after executing the call $partition(a, L, U)$,

$$a[k] \leq t \text{ for } L \leq k < p, \quad a[p] = t, \quad \text{and} \quad a[k] \geq t \text{ for } p < k \leq U.$$

An alternative partitioning method uses *three-way* partitioning, so that, after executing the call $partition3(a, L, U)$, the values assigned to the variables p and q are such that

$$a[k] < t \text{ for } L \leq k < p, \quad a[k] = t \text{ for } p \leq k \leq q, \quad \text{and} \quad a[k] > t \text{ for } q < k \leq U.$$

Write a *Quicksort* procedure for sorting the array a that uses the three-way partitioning procedure *partition*3.

Note: You do not have to provide code for the procedure *partition*3. **(3 marks)**

(b) The time taken by *partition*3 is approximately 50% longer than that taken by *partition*. Are there any circumstances in which it would be advantageous to use the *partition*3 version of *Quicksort* rather than the *partition* version? Would it make any difference if the *partition* version used *one-sided* or *two-sided* partitioning? Give your reasons for your answers, including comparative time complexities where appropriate. **(5 marks)**

