Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Advanced Algorithms

Date: Thursday 22\textsuperscript{nd} May 2008

Time: 14:00 – 16:00

Please answer any THREE Questions from the FIVE questions provided

This is a CLOSED book examination

The use of electronic calculators is NOT permitted.
1. a) Describe the Knuth-Morris-Pratt algorithm for searching for a string pattern in a
   text. Your description should include:
   i) the principles behind the pre-processing of the pattern,
   ii) the way the pre-processing results are used to search the text.

   You need not write a program but the main steps in the algorithm should be
clearly explained. (10 marks)

b) In a string searching context, sometimes (for example, when searching English
texts) some characters are known to appear rarely and others are common.
Suppose that the expected frequency of each character is known in advance.
Describe two string searching algorithms which make use of this frequency data
to improve on the naive string searching algorithm, as follows:
   i) an algorithm which, when a mis-match occurs, the pattern is moved one
      position rightwards, (5 marks)
   ii) an algorithm which, when a mis-match occurs, the pattern may move
      multiple positions rightwards. (5 marks)

2. a) i) Explain clearly what is meant by a Depth-First Search (DFS) of an
       undirected graph.

   ii) Describe a recursive method for calculating a DFS of an undirected graph.
       You should write a program or explain your algorithm in pseudocode.

   iii) How would you modify your DFS algorithm to calculate the number of
        connected components of an undirected graph? (7 marks)

b) What is meant by an Eulerian circuit of an undirected graph?

   Describe clearly a linear algorithm for computing an Eulerian circuit of an
   undirected graph, if such a circuit exists, or reporting failure. You need not
   describe a program but the steps of your algorithm and its correctness should be
clearly explained. (7 marks)

c) For one NP-complete task on undirected graphs, describe an exponential
   algorithm for the task. You should write a program or explain your algorithm in
   pseudocode. (6 marks)
3. a) Kripke structures can be used to represent models of finite state programs in model checking based verification. Informally explain what a Kripke structure is, provide a formal definition for it, and indicate how it may be used to represent the behaviour of finite state programs. (6 marks)

b) Consider the following concurrent program

\begin{verbatim}
1:   L0: while (y>0) do
2:     L1: cobegin
3:           L11: x = y+1;
4:           L12: ||
5:     L21: y = x-3;
6:     L22: coend
\end{verbatim}

in which the assignment statements from lines 3 and 6 are executed concurrently, but atomically, within the body of the while loop. Assume a program counter variable pc that takes values according to the following table.

<table>
<thead>
<tr>
<th>Label Pairs</th>
<th>pc Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>L11, L21</td>
<td>1</td>
</tr>
<tr>
<td>L12, L21</td>
<td>2</td>
</tr>
<tr>
<td>L11, L22</td>
<td>3</td>
</tr>
<tr>
<td>L12, L22</td>
<td>1</td>
</tr>
</tbody>
</table>

Using logical variables pc, x and y, and their primed counterparts to represent their next state values, write out a logical formula that characterises the transition relation over program states defined by the above concurrent program. (4 marks)

c) Assuming the initial values of variables x and y, are, respectively, 1 and 2, and using the transition relation given in answer to part b), construct a Kripke structure that will represent the computation state graph of the program from that initial state. (4 marks)

d) What are maximally strongly connected components (MSCCs) of a graph? Briefly describe an algorithm for finding MSCCs. Explain the significance of the existence of reachable MSCCs in a Kripke structure representing a program’s computation state. (6 marks)

[PTO]
4. This question is concerned with infinite word automata.

a) Briefly explain what an infinite word automaton is, in particular the differences between it and a finite word automaton, and the role that such automata can play in the specification of program properties. (4 marks)

b) Buchi automata are a modification of finite word automata that accept infinite words. Formally define Buchi automata, and then present, in graphical form, a Buchi automaton that accepts the language described by the omega regular expression

\[ a^* (bc^*)^\omega \]  (4 marks)

c) Given two Buchi automata, BA1 and BA2, describe and explain an algorithm for constructing a Buchi automaton from BA1 and BA2 that will accept the intersection of the languages accepted by BA1 and BA2. (6 marks)

d) Consider the infinite word languages represented by the omega regular expressions

i) \((bc^*)^\omega\)

ii) \((b^*c)^\omega\)

For each language, give a Buchi automaton that accepts the language, then using the algorithm given in answer to part c) above, construct a Buchi automaton that accepts the intersection of the two languages given by i) and ii). (6 marks)
5. a) In terms of complexity theory, explain what is meant by a computationally “easy” algorithm. Give one example of a linear space-bounded algorithm, and one example of a quadratic time-bounded algorithm. (2 marks)

b) What is meant by an algorithm being

i) NP-complete
ii) PSPACE-hard

Give an example of an algorithm in both classes. (3 marks)

c) What is meant by a polynomial-time reduction of one computational task to another? (2 marks)

d) Explain clearly how, given a task that is known to be NP-complete, another task can be established to be NP-complete by reduction. (3 marks)

e) Using the NP-completeness of the general Boolean satisfiability problem, describe in detail how to establish the NP-completeness of one task on directed, or undirected, graphs. (10 marks)