Computability and Intractability (2001-2002)

CS3 June Exam Sample Answers

1. (a) Simple bookwork. *[2 marks]*

   (b) Bookwork.

   [8 marks for the first part, 4 marks for the halting problem bit.] *[12 marks]*

   (c) F is not computable, it is the halting problem in very thin disguise. G is computable by:

   Given m, n:
   run P_m on n;
   return 0

   [2 marks for giving the status of F, G correctly, 3 marks for the justification.] *[5 marks]*

   (d) Consider the program

   Given m, n:
   run P_m on n;
   (Any statement S here)

   Statement S is reachable if and only if P_m halts on n. So a solution to the reachability problem would give us a solution to the halting problem.

   [3 marks for the reduction, 3 marks for the justification.] *[6 marks]*

2. (a) Bookwork. *[3 marks]*

(b) Bookwork. *[5 marks]*

(c) Bookwork. *[4 marks]*

(d) We modify M to produce the machine M' as follows:

i. ensure that it never falls off the left hand end of the tape by employing a special symbol `$`, say. The first action of the modified machine is to insert this symbol at the first square, having moved the input to the right by one square. After that it positions its head to scan the second square and enters the initial state of M.

ii. For all state/symbol pairs for which the state is not the final state of M and for which there is no instruction we add instructions to ensure that M' will move to the right for ever.

Thus M accepts a string x if and only if M' halts on the same string.

If L is an r.e. language then it is accepted by some Turing machine M. The construction of M' gives a reduction from L to L_{halt} (we may assume w.l.o.g. that $L \subseteq \{0, 1\}^*$, alternatively convert M' to a binary Turing machine).

[7 marks for constructing M'; 5 marks for a proof that is essentially correct but overlooks a minor detail (e.g., the possibility of the head falling off the left hand end). 3 marks for the deduction.] *[10 marks]*

(e) Any non r.e. language will do, e.g., the complement of L_{halt}.

[1 mark for a correct example, 2 marks for the justification.] *[3 marks]*

3. (a) Bookwork.

   [2 marks for each definition.] [*4 marks*]

   (b) Simple bookwork.

   [No room for doubt here.] [*2 marks*]

   (c) The language is not (known to be) in NP.

   [No room for doubt here.] [*2 marks*]

   (d) The proof part is left as an exercise in the notes (however all the necessary details are given in an earlier theorem). The crucial point here is to show that the obvious transducer runs in time that is polynomial in the length of the input string.

   [3 marks for an appropriate transducer, 4 marks for showing that it runs in polynomial time.] [*7 marks*]

   (e) i. The only difference between the reduction given in the question and the one in the notes is that each vertex is expanded to 4 here and only 3 in the notes. The proof is otherwise identical and quite short.
   [4 marks for proof of correctness of the reduction, 1 mark for observing that it is indeed polynomial time.] [*5 marks*]

   ii. The graph G' of the preceding part is bipartite: take $V_1 = \{ v_1, v_3 \mid v \in V \}$ and $V_2 = \{ v_0, v_2 \mid v \in V \}$. Since DHC is NP complete this shows that BUHC is NP-hard. Of course BUHC is in NP (since this is the case for UHC on arbitrary graphs) and thus BUHC is NP-complete.
   [3 marks for showing that G' is bipartite, 2 marks for the rest.] [*5 marks*]