UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

COMPUTABILITY AND INTRACTABILITY

Thursday 13 June 2002

09:30 to 11:00

Examiners:

Prof. K. Turner (External)

Prof. C. Stirling (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. For this question suppose that we have a general system of computation expressed textually (i.e., as programs) over some fixed alphabet. Let $P_0, P_1, P_2, \ldots$ be an effective enumeration of all (single input) programs in the system and $I_0, I_1, I_2, \ldots$ all possible strings over the alphabet. For convenience we assume that all input strings are encoded as natural numbers in such a way that for each string there is a unique number and for each natural number there is a unique string (and it is possible to go from one to the other effectively).

   (a) Explain briefly why the assumption that our programs take a single input string rather than several strings is not a significant restriction. [*2 marks*]

   (b) Prove carefully that we cannot have a general theory of computation in which all programs are guaranteed to terminate on all inputs.

   Indicate briefly how we can modify the argument to show that the halting problem is unsolvable. [*12 marks*]

   (c) Define the functions F, G by:

$$F(m,n) = \begin{cases} 0, & \text{if } P_m \text{ halts on input } n; \\ 1, & \text{otherwise.} \end{cases}$$

$$G(m,n) = \begin{cases} 0, & \text{if } P_m \text{ halts on input } n; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

   One of these functions is computable while the other is not. State clearly which is computable, which is not computable and justify your answer. [*5 marks*]

   (d) Given a program P of a general system of computation and a particular statement of the program it is clearly of interest to know if the statement is ever executed. To be more precise, given a program P, input I to P and a statement S of P (identified in some way, e.g., by line number) is S reached during the execution of P on input I?

   Prove that this problem is unsolvable. [*6 marks*]

2. (a) Explain what is meant by saying that a language L is *recursively enumerable* (abbreviated to r.e.). *[3 marks]*

(b) Suppose $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ are languages. Prove that if L_1 is reducible to L_2, and L_2 is r.e., then L_1 is also r.e. *[5 marks]*

(c) Recall that the Halting Problem is expressed by the language

$$L_{\text{halt}} = \{\langle M \rangle \$ x \mid x \in \{\texttt{0}, \texttt{1}\}^* \text{ and } M \text{ halts on input } x\},$$

where $\langle M \rangle$ represents the encoding of the binary Turing machine M. Prove that L_{halt} is r.e. by explaining an appropriate modification of a universal Turing machine. *[4 marks]*

(d) Let M be a Turing machine with input alphabet Σ. Show that we can construct a Turing machine M', also with input alphabet Σ, such that

$$M \text{ accepts } x \iff M' \text{ halts on input } x,$$

for all $x \in \Sigma^*$.

Deduce that every r.e. language can be reduced to L_{halt}.

[**Note:** In showing how to construct M' a detailed proof at the level of quintuples is *not* expected; however, all the essential steps should be made clear.] *[10 marks]*

(e) Give an example of a language that cannot be reduced to L_{halt} and explain briefly why this is so. *[3 marks]*

3. (a) Define the classes P and NP in terms of Turing machines. [*4 marks*]

(b) Explain briefly why $P \subseteq NP$. [*2 marks*]

(c) It is a straightforward matter to produce a recursive language that is not in P and yet the P versus NP question is still unresolved. Explain briefly why this is the case. [*2 marks*]

(d) Let L_1, L_2 and L_3 be languages over some alphabets Σ_1, Σ_2 and Σ_3 respectively. Recall that the notation $L_1 \leq_P L_2$ denotes that there is a polynomial time reduction from L_1 to L_2, i.e., there is a function $f : \Sigma_1^* \to \Sigma_2^*$ such that (i) $x \in L_1$ if and only if $f(x) \in \Sigma_2^*$ for all $x \in \Sigma_1^*$ and (ii) f is computable by a polynomial-time Turing transducer.

Prove that if $L_1 \leq_P L_2$ and $L_2 \leq_P L_3$ then $L_1 \leq_P L_3$. [*7 marks*]

(e) i. Recall that the *Directed Hamiltonian Cycle* problem, DHC is defined by:

INSTANCE: A directed graph G.

QUESTION: Does G possess a Hamiltonian cycle, i.e., a directed closed path which visits every vertex of G precisely once?

The *Undirected Hamiltonian Cycle* problem, UHC is the version of this for undirected graphs.

Let $G = (V, E)$ be any directed graph. Define the undirected graph $G' = (V', E')$ by:

$$\begin{aligned}
V' &= V \times \{0, 1, 2, 3\}, \\
E' &= \{\{\langle v, 0\rangle, \langle v, 1\rangle\} \mid v \in V\} \\
&\quad \cup \{\{\langle v, 1\rangle, \langle v, 2\rangle\} \mid v \in V\} \\
&\quad \cup \{\{\langle v, 2\rangle, \langle v, 3\rangle\} \mid v \in V\} \\
&\quad \cup \{\{\langle u, 3\rangle, \langle v, 0\rangle\} \mid (u, v) \in E\}.
\end{aligned}$$

Prove that this is a polynomial time reduction from DHC to UHC.
[**Note:** This is a minor variation of the reduction given in the lecture notes.] [*5 marks*]

ii. An undirected bipartite graph is an (undirected) graph G whose vertices consist of two disjoint sets V_1, V_2 and whose edges all have one endpoint in V_1 and one endpoint in V_2.
We use BUHC to denote the Undirected Hamiltonian Cycle problem for bipartite graphs. It is known that DHC is NP-complete. Use this fact and the preceding part to deduce that BUHC is also NP-complete. [*5 marks*]