MATH-316301

This question paper consists of 3 printed
pages, each of which is identified by the
reference MATH 316301

No calculators allowed

Examination for the Module MATH 3163
(January 2003)

COMPUTABILITY AND UNSOLVABILITY

Time allowed : 3 hours

Do not answer more than *FOUR* questions.
All questions carry equal marks.

1. (**a**) (i) Show that $f(x,y) = x + y$ and

$$sg(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0 \end{cases}$$

are primitive recursive functions.

Deduce that if the function $f(m,n)$ is known to be primitive recursive, then so is the **bounded sum** $\sum\limits_{m \le p} f(m,n)$ defined by:

$$\sum_{m \le p} f(m,n) = f(0,n) + f(1,n) + \cdots + f(p,n).$$

(ii) Let $R(m,p)$ be a primitive recursive relation. Assume that a new relation P is defined by the following application of **bounded quantification**:

$$P(m,n) \iff (\exists p \le n) R(m,p) \quad \text{(that is, for some } p \le n,\ R(m,p) \text{ is true).}$$

Show that P is primitive recursive.

(**b**) Write a Turing program for the function $\mathbf{0} : \mathbb{N}^k \to \mathbb{N}$ defined by $\mathbf{0}(\overrightarrow{n}) = 0$ for all $\overrightarrow{n} \in \mathbb{N}^k$, $k \geq 1$, and briefly explain why your program works.

(**c**) Let $\{\varphi_x\}_{x \in \mathbb{N}}$ be a standard list of all the partial computable functions.

Let the total function f be defined by

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \text{if } \varphi_x \text{ is total,} \\ 0 & \text{otherwise.} \end{cases}$$

Explain why f is not computable.

Deduce that if

$$\mathrm{Tot} = \{x \in \mathbb{N} \mid \varphi_x \text{ is total}\},$$

then Tot is not computable.

continued ...

2. (**a**) Describe **briefly** how it is possible to effectively list all Turing programs:

$$P_0, P_1, \ldots, P_e, \ldots .$$

Prove the **Enumeration Theorem:** There exists a partial computable function $\varphi_z(x)$ (of x and z) such that for each $e \in \mathbb{N}$ φ_e is the e^{th} (unary) partial computable function.

Deduce that there exists a **Universal Turing Machine** U which, if given input (e, x), say, simulates the e^{th} Turing machine with input x (that is, $\varphi_U^{(2)}(e, x) = \varphi_e(x)$ for all $x \in \mathbb{N}$).

(**b**) We say $A \subseteq \mathbb{N}$ is **creative** if and only if

1) A is c.e., and

2) There is a computable function f such that for each e

$$W_e \subseteq \bar{A} \Rightarrow f(e) \in \bar{A} - W_e \,,$$

where $\{W_e\}_{e \in \mathbb{N}}$ is a standard list of all c.e. sets.

Show that

(i) If A is creative, then A not computable, and

(ii) K is creative, where $K = \{x \mid x \in W_x\}$.

Deduce that there exists a Turing machine whose halting problem is unsolvable.

(**c**) The **Printing Problem** for a Turing machine T and a symbol S_k is the problem of determining, for any given input x, whether T ever prints the symbol S_k.

Find a Turing machine T for which the printing problem is unsolvable.

3. (**a**) Using the terminology and notation from Question 2, part (b), show that if C is a creative set then

(i) $\overline{C} \neq \emptyset$,

(ii) For each $n \in \mathbb{N}$, if there exist n members of $\overline{C}$ then there exist $n+1$ such members,

(iii) $\overline{C}$ contains an infinite c.e. subset.

[You may assume that for any finite set X we can effectively find an i such that $X = W_i$.]

(**b**) (i) Prove the **Kleene Fixed Point Theorem** for a computable function f.

(ii) Explain why there is a computable function f such that $W_{f(n)} = \{n\}$ for every $n \in \mathbb{N}$.

Say also why it is that every c.e. set A has infinitely many distinct indices e with $A = W_e$.

Question 3 continues . . .

(iii) We say that a set A is an **index set** if for all $x, y \in \mathbb{N}$ we have

$$[x \in A \ \& \ W_x = W_y] \Longrightarrow y \in A.$$

Let K be defined as in Question 2. By using the fixed point theorem, or otherwise, show that K is not an index set.

4. (**a**) Let $A, B \subseteq \mathbb{N}$ be sets other than $\mathbb{N}$ or $\emptyset$. Define : $A \leq_m B$ (A is **many-one reducible** to B).

(i) Show that if $A \leq_m B$ and B is computable, then A is computable.

(ii) Show that if $A \leq_m B$ and B is computably enumerable, then A is computably enumerable.

(**b**) Define: $A \leq_T B$ (A is **Turing reducible to** B), $A \equiv_T B$ (A is **Turing equivalent to** B).

Given that $A \leq_m B$ via a computable function f, find an oracle Turing machine T such that A is Turing reducible to B via T.

[You may assume there is a Turing machine $\bar{T}$ which, for each input x, has output $\varphi_{\bar{T}}(x) = f(x) + 1$, printed on the tape as a block of 1s, with reading head on the leftmost 1, and in halting state q_M, where M is such that for every $q_i \neq q_M$ in T we have $i < M$.]

(**c**) The **Turing jump** B' of $B \subseteq \mathbb{N}$ is defined to be

$$B' = \{\langle m, n \rangle \mid m \in W_n^B\},$$

where $\{W_n^B\}_{n \in \mathbb{N}}$ is a standard list of all B-c.e. sets.

(i) Show that B' is c.e. in B, and that if X is c.e. in B then $X \leq_m B'$.

(ii) Show that if $B \leq_T A$ and X is computably enumerable in B, then X is computably enumerable in A.

Deduce that if $A \equiv_T B$ then $A' \equiv_T B'$.

5. (**a**) (i) Let **a** be a Turing degree.

Define: **a** is **computably enumerable** (or **c.e.**).

Show that the set $\{X \subseteq \mathbb{N} \mid X \text{ has c.e. Turing degree}\}$ is countable.

(**b**) Show that there exists a pair of incomparable Turing degrees below $\mathbf{0}'$.

6. Write an essay, covering **not more than three pages**, describing the background to, and consequences of, Alan Turing's discovery of the existence of a Universal Turing Machine.

Your answer should contain enough mathematical content to show a good grasp of the notions and results involved, and enough discussion of these to show an understanding of the broader context.

END