UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MEng Honours Degrees in Computing Part IV
MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the City and Guilds of London Institute

PAPER 4.38

COMPLEXITY

Thursday, May 8th 1997, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 In this question, Σ is an arbitrary finite alphabet containing at least two characters. If a Turing machine is 'obviously' p-time you need not prove that it is.

a Explain what we mean when we say that:

i) a non-deterministic Turing machine *accepts,* or *rejects,* its input,

ii) a non-deterministic Turing machine runs in p-time (polynomial time),

iii) a language $L \subseteq \Sigma^*$ is in NP,

iv) a language $L \subseteq \Sigma^*$ is in co-NP.

b A *strong non-deterministic Turing machine* is a non-deterministic Turing machine that has three possible outcomes (halting states): *yes, no,* and *maybe.*

Let $L \subseteq \Sigma^*$ be a language. We say that such a machine *decides* L if

- when it is given as input a word in L, all computations end up with *yes* or *maybe,* and at least one with *yes,* and
- when given a word of Σ that is not in L, all computations end up with *no* or *maybe*, and at least one with *no.*

i) Show that if L is decided by some p-time strong non-deterministic Turing machine then $L \in$ NP$\cap$co-NP (that is, $L \in$ NP and $L \in$ co-NP).

ii) Outline an argument to show the converse: that if $L \in$ NP$\cap$co-NP then L is decided by some p-time strong non-deterministic Turing machine.

c A *coding* is a map $f : \Sigma \to \Sigma$, *not necessarily 1-1.*

Let $f : \Sigma \to \Sigma$ be a coding. If $w = a_1a_2\ldots a_n$ is a word of Σ, write f(w) for the word $f(a_1)f(a_2)\ldots f(a_n)$ of Σ. That is, f(w) is got by replacing each 'letter' a of w by f(a).

If $L \subseteq \Sigma^*$ is a language, write f(L) for the language $\{f(w) : w \in L\}$. That is, f(L) is made by replacing each word w of L by f(w).

Prove that NP is closed under codings — that is, if $L \in$ NP, and $f : \Sigma \to \Sigma$ is a coding, then $f(L) \in$ NP.

The three parts carry, respectively, 30%, 40%, 30% of the marks.

2 In this question, if a construction is 'obviously' p-time you need not prove that it is.

a i) Explain briefly what it means for a decision problem A to *reduce* to another decision problem, B, *in p-time* (in symbols, $A \leq B$).

Let A, B be arbitrary decision problems and let '$\leq$' be as in part a(i). Briefly explain why:

ii) If A is NP-complete, $B \in NP$, and $A \leq B$, then B is NP-complete. (You may assume without proof that '$\leq$' is transitive.)

iii) If $A \leq B$ and $B \in P$ then $A \in P$.

b Consider the following decision problem:

WGE (weak graph embedding)
Given: a pair (G,H) of undirected graphs.
Problem: is there is a 1-1 map $i : \text{nodes}(G) \rightarrow \text{nodes}(H)$ that preserves edges forwards: that is, if (x,y) is an edge of G then (i(x),i(y)) is an edge of H?
[Such an i is called a *weak* embedding, as it is not required that if (i(x),i(y)) is an edge of H then (x,y) is an edge of G.]

Prove, using part a(ii) or otherwise, that WGE is NP-complete.

(You may assume that standard problems such as Hamiltonian Circuit are NP-complete.)

c Consider the following decision problem:

2COL (2-colourability)
Given: an undirected graph G.
Problem: can each node of G be coloured either white or black in such a way that for any edge (x,y) of G, the nodes x and y have different colours?

Prove that $2COL \in P$.

(You may wish to reduce 2COL to 2SAT and then use part a(iii). You may assume standard facts such as that $2SAT \in P$.)

Turn over...

3a i) What does it mean for a multi-tape deterministic Turing machine M to be *logspace bounded*?

ii) What does it mean for a family of Boolean circuits C_n to be *uniform*?

iii) Define the class NC_j, for $j \geq 1$.

b Show that if L_1, L_2 are NC_j then so is $L_1 \quad L_2$ (any $j \geq 1$).

c Let Σ be an alphabet, and let f, g be functions: $\Sigma^* \rightarrow \Sigma^*$. Show that if f and g are logspace functions then their composition g(f(x)) is also logspace.

d In what follows let f, g be functions: $\{0,1\}^* \rightarrow \{0,1\}^*$.
Say that f is computed by the family of Boolean circuits C_n if for each n and each input x of size n, C_n computes f(x) (so C_n must have multiple output gates, and the output size must be the same for all inputs of a given size). Say that f is *uniform computable* if it is computed by some uniform C_n.

Show that if f and g are uniform computable then so is g(f(x)).

4a i) Let Σ be an alphabet, and let R be a binary relation on Σ^*. What does it mean for R to be *polynomially balanced*?

ii) Define the function problem classes FNP and FP.

b i) Define FHC, the function problem associated with the Hamiltonian circuit problem HC, and explain why FHC   FNP.

ii) Show that if HC   P then FHC   FP.

c i) Define a *one-way* function and a *trapdoor* function.

ii) Explain why public key cryptography is impossible unless FNP $\neq$ FP.

iii) Why is worst-case complexity in a sense irrelevant for cryptography? Suggest a more appropriate notion.

The three parts carry, respectively, 25%, 40%, 35% of the marks.

End of paper

