COMPUTER SCIENCE TRIPOS  Part I\textbf{b}

Thursday 7 June 2012    1.30 to 4.30

COMPUTER SCIENCE  Paper 6

Answer \textit{five} questions.

\textit{Submit the answers in five \textit{separate} bundles, each with its own cover sheet. On each cover sheet, write the numbers of \textit{all} attempted questions, and circle the number of the question attached.}

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS  
\begin{itemize}
  \item Script paper
  \item Blue cover sheets
  \item Tags
\end{itemize}

SPECIAL REQUIREMENTS
\begin{itemize}
  \item Approved calculator permitted
\end{itemize}
1 Complexity Theory

(a) Suppose $L_1$ and $L_2$ are languages in $\text{P}$. What can you say about the complexity of each of the following? Justify your answer in each case.

(i) $L_1 \cup L_2$. [3 marks]

(ii) $L_1 \cap L_2$. [3 marks]

(iii) The complement of $L_1$. [2 marks]

(b) Suppose $L_1$ and $L_2$ are languages in $\text{NP}$. What can you say about the complexity of each of the following? Justify your answer in each case.

(i) $L_1 \cup L_2$. [3 marks]

(ii) $L_1 \cap L_2$. [3 marks]

(iii) The complement of $L_1$. [2 marks]

(c) Give an example of a language in $\text{NP}$ that is not $\text{NP}$-complete and prove that it is not. [4 marks]
2 Complexity Theory

(a) Consider the following decision problem.

Given positive integers $x_1, \ldots, x_n, y$, determine whether there is a set $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} x_i = y$.

(i) Prove that, if the integers $x_1, \ldots, x_n, y$ are written in unary, then the problem is in $\mathbf{P}$. [Hint: consider a recursive algorithm which checks whether either of $y$ or $y - x_1$ can be expressed as the sum of a subset of $x_2, \ldots, x_n$.] [6 marks]

(ii) What can you say about the complexity of the decision problem when the integers $x_1, \ldots, x_n, y$ are written in binary? You do not need to prove your answer, but state clearly any standard results you use. [2 marks]

(b) What does it mean for a language $L$ to be $\mathbf{NP}$-hard? What does it mean for $L$ to be $\mathbf{NP}$-complete? [2 marks]

(c) We write $[M]$ to be the string encoding a Turing machine $M$ using some standard coding scheme. Consider the language $A$ defined by:

$$A = \{[M], x \mid M \text{ accepts } x\}$$

where $[M], x$ denotes the string $[M]$ followed by a comma and then $x$.

(i) Prove that $A$ is $\mathbf{NP}$-hard. [8 marks]

(ii) Is $A$ $\mathbf{NP}$-complete? Justify your answer. [2 marks]
3 Computation Theory

(a) Define what is a *Turing machine* and a *Turing machine computation.* [7 marks]

(b) What is meant by a *partial function* from $\mathbb{N}^n$ to $\mathbb{N}$? Define what it means for such a partial function to be *Turing computable.* [4 marks]

(c) Describe the *Church-Turing Thesis* and some evidence for its truth. [4 marks]

(d) Assuming the existence of a universal *register* machine, give an example, with justification, of a partial function that is not Turing computable. [5 marks]
4 Computation Theory

(a) Define what it means for a set of numbers $S \subseteq \mathbb{N}$ to be register machine decidable. Why are there only countably many such sets? Deduce the existence of a set of numbers that is not register machine decidable. (Any standard results that you use should be clearly stated.) [4 marks]

(b) A set of numbers $S \subseteq \mathbb{N}$ is said to be computably enumerable if either it is empty or equal to $\{f(x) \mid x \in \mathbb{N}\}$ for some total function $f : \mathbb{N} \to \mathbb{N}$ that is register machine computable.

(i) Show that if $S$ is register machine decidable, then it is computably enumerable. [Hint: consider separately the cases when $S$ is, or is not empty.] [4 marks]

(ii) Show that if both $S$ and its complement $\{x \in \mathbb{N} \mid x \not\in S\}$ are computably enumerable, then $S$ is register machine decidable. [Hint: consider a register machine that interleaves the enumeration of $S$ and its complement.] [6 marks]

(c) Let $\varphi_e : \mathbb{N} \to \mathbb{N}$ denote the partial function computed by the register machine with code $e \in \mathbb{N}$ and consider the set $T = \{e \in \mathbb{N} \mid \varphi_e$ is a total function$\}$.

(i) Suppose that $f : \mathbb{N} \to \mathbb{N}$ is a register machine computable total function such that $f(x) \in T$ for all $x \in \mathbb{N}$. Define $\hat{f}(x)$ to be $\varphi_{f(x)}(x) + 1$. Show that $\hat{f} = \varphi_e$ for some $e \in T$. [3 marks]

(ii) Deduce that $T$ is not computably enumerable. [3 marks]
5 Logic and Proof

(a) Exhibit a model for the following set of formulas, or prove that none exists.

\[ P \rightarrow Q \land R \quad P \land Q \rightarrow S \quad \neg R \leftrightarrow S \quad P \lor Q \]

[8 marks]

(b) Consider the following set of clauses:

\{\neg(x < y), \neg y < \neg x\} \quad \{\neg(x < y), x + z < y + z\} \quad \{0 < 1\}

(i) What is the Herbrand universe of these clauses? [3 marks]

(ii) What semantics must any Herbrand interpretation of the clauses attach to the function symbols? [3 marks]

(iii) Specify an Herbrand model by giving a semantics of the relation \(<\), justifying your choice with reference to a natural model of the set of clauses. [6 marks]
6 Logic and Proof

(a) Demonstrate the sequent calculus, the free-variable tableau calculus and resolution by using each of them to prove the following formula:

\[(P(a, b) \lor \exists z P(z, z)) \rightarrow \exists x \exists y P(x, y)\]

Comment briefly on the similarities and differences among these three methods. [12 marks]

(b) Prove \(\Box \Diamond P \rightarrow \Diamond \Box P\) using the sequent calculus for S4 modal logic, or exhibit a falsifying interpretation. [4 marks]

(c) Briefly outline the procedure for converting a formula to a BDD, illustrating your answer by constructing the BDD that represents the conjunction of those below. [4 marks]
7 Mathematical Methods for Computer Science

(a) Define linear independence and linear dependence for the set of vectors 
{v_1, v_2, \ldots, v_n} of a vector space V over a field \( \mathbb{F} \) of scalars \( a_1, a_2, \ldots, a_n \in \mathbb{F} \).

(b) Using the Euclidean norm on an inner product space \( V = \mathbb{R}^3 \), for the following vectors \( u, v \in V \) whose span is a linear subspace of \( V \),

\[
\begin{align*}
  u &= \left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \\
v &= \left( \sqrt{3}, -\frac{\sqrt{3}}{2}, -\frac{\sqrt{3}}{2} \right)
\end{align*}
\]

demonstrate whether \( u, v \) form an orthogonal system, and also whether they form an orthonormal system.

(c) Using a diagram in the complex plane showing the \( N \)th roots of unity, explain why all the values of complex exponentials that are needed for computing the Discrete Fourier transform of \( N \) data points are powers of a primitive \( N \)th root of unity (circled here for \( N = 16 \)), and explain why such factorisation greatly reduces the number of multiplications required in a Fast Fourier transform.

(d) For the function \( f(x) = e^{-a|x|} \) with \( a > 0 \), derive its Fourier transform \( F(\omega) \).

(e) For a function \( f(x) \) whose Fourier transform is \( F(\omega) \), what is the Fourier transform of \( f^{(n)}(x) \), the \( n \)th derivative of \( f(x) \) with respect to \( x \)? Explain how Fourier methods make it possible to define non-integer orders of derivatives, and name one scientific field in which it is useful to take half-order derivatives.
8 Mathematical Methods for Computer Science

(a) Consider the Markov Chain, \( X_n \), on the states \( i = 0, 1, 2, \ldots \) with transition matrix given by

\[
\begin{align*}
p_{i,i-1} &= p & i &= 1, 2, \ldots \\
p_{i,i+1} &= 1 - p & i &= 0, 1, \ldots \\
p_{0,0} &= p
\end{align*}
\]

where \( 0 < p < 1 \).

(i) Show that the Markov chain is irreducible. \([2 \text{ marks}]\)

(ii) Show that the Markov chain is aperiodic. \([2 \text{ marks}]\)

(iii) Find a condition on \( p \) to make the Markov chain positive recurrent and find the stationary distribution in this case. \([6 \text{ marks}]\)

(b) Consider the PageRank algorithm.

(i) Describe PageRank as a Markov chain model for the motion between nodes in a graph where the nodes correspond with web pages. \([5 \text{ marks}]\)

(ii) Explain the main mathematical results that underpin the relevance of PageRank to a notion of web page importance. \([5 \text{ marks}]\)
9 Semantics of Programming Languages

This question is about a very simple programming language with support for exceptions. The syntax and semantics is given to you.

The language has the following syntax.

\( \text{Exceptions: } \gamma, \gamma_1, \gamma_2, \ldots \)

\( \text{Expressions: } e ::= \text{true} \mid \text{false} \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \)

\( \mid e_1 \text{ handle } \gamma \Rightarrow e_2 \mid \text{raise } \gamma \)

\( \text{Types: } \) There is only one type: \text{bool}

A typing context is a set \( \Gamma \) of exceptions. The typing rules are as follows:

\[
\begin{align*}
\Gamma \vdash \text{true} : \text{bool} & \quad \Gamma \vdash \text{false} : \text{bool} \quad \Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \text{bool} \quad \Gamma \vdash e_3 : \text{bool} \\
\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \text{bool} & \quad \Gamma \vdash e_1 \text{ handle } \gamma \Rightarrow e_2 : \text{bool} \\
(\gamma \notin \Gamma) & \quad \Gamma \vdash \text{raise } \gamma : \text{bool} \quad (\gamma \in \Gamma)
\end{align*}
\]

The reduction relation \( \rightarrow \) is defined as follows:

\[
\begin{align*}
\text{if true then } e_2 \text{ else } e_3 & \rightarrow e_2 \quad \text{if false then } e_2 \text{ else } e_3 & \rightarrow e_3 \\
\text{if } e_1 \text{ then } e_2 \text{ else } e_3 & \rightarrow \text{if } e'_1 \text{ then } e_2 \text{ else } e_3 \\
(\text{true handle } \gamma \Rightarrow e_2) & \rightarrow \text{true} \quad (\text{false handle } \gamma \Rightarrow e_2) & \rightarrow \text{false} \\
(\text{false handle } \gamma \Rightarrow e_2) & \rightarrow \text{false} \quad (\text{true handle } \gamma \Rightarrow e_2) & \rightarrow \text{true} \\
((\text{raise } \gamma) \text{ handle } \gamma \Rightarrow e_2) & \rightarrow e_2 \quad (\text{raise } \gamma) \text{ handle } \gamma \Rightarrow e_2 & \rightarrow ((\text{raise } \gamma') \text{ handle } \gamma \Rightarrow e_2) \\
\text{if (raise } \gamma \text{) then } e_2 \text{ else } e_3 & \rightarrow \text{raise } \gamma \quad (\gamma \neq \gamma')
\end{align*}
\]

(a) Consider the program \( e_0 \overset{\text{def}}{=} \text{if } (\text{raise } \gamma) \text{ handle } \gamma \Rightarrow \text{true} \text{ then false else true}. \)

(i) Give a derivation for \( \vdash e_0 : \text{bool}. \) [4 marks]

(ii) Give a derivation for all of the transition steps for \( e_0. \) [3 marks]

(b) Prove the following theorem about this language:

If \( \Gamma \vdash e : \text{bool} \) then one of the following four conditions holds: (1) \( e = \text{true}; \) (2) \( e = \text{false}; \) (3) there is \( \gamma \in \Gamma \) such that \( e = (\text{raise } \gamma); \) (4) there is \( e' \) such that \( e \rightarrow e'. \) [13 marks]
10 Semantics of Programming Languages

This question is about a simple functional programming language with the following syntax.

Expressions: \( e ::= x \mid \text{skip} \mid \text{fn} \ x : T \Rightarrow e \mid e \ e' \)

Types: \( T ::= \text{unit} \mid T \rightarrow T' \)

(a) Give rules defining a typing relation (\(\vdash\)) for this language. [5 marks]

(b) Give a brief illustration of the following concepts: free variables and closed expression. [2 marks]

(c) Give rules defining a transition relation (\(\rightarrow\)) for this language. Use the call-by-value evaluation order, and take care to say what the values are. [5 marks]

(d) State and prove a Type Progress theorem for this language. [8 marks]

END OF PAPER