UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS Part IB

Thursday 2 June 2016 1.30 to 4.30

COMPUTER SCIENCE Paper 6

Answer ***five*** *questions.*

Submit the answers in five ***separate*** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* ***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
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Complexity Theory

(*a*) Let $f : \mathbb{N} \to \mathbb{N}$ be a function and let $\mathrm{rng}(f)$ be defined to be the set

$$\mathrm{rng}(f) = \{y \mid f(x) = y \text{ for some } x \in \mathbb{N}\}.$$

(*i*) Define what it means to say that f is computable in polynomial time. Pay particular attention to the question of how numbers are represented as strings of symbols. [3 marks]

(*ii*) Show that if f is computable in polynomial time and increasing (i.e., for all $x \in \mathbb{N}$, $x < f(x)$), then $\mathrm{rng}(f)$ is in NP. [5 marks]

(*iii*) Show that if f is computable in polynomial time, increasing and injective, then $\mathrm{rng}(f)$ is in UP. [5 marks]

(*b*) Let $A \subseteq \mathbb{N}$ be defined as the following set of numbers

$$A = \{x \mid x = pq \text{ for distinct prime numbers } p \text{ and } q\}.$$

Prove that A is in NP and in co-NP. [7 marks]

2 Complexity Theory

The *Graph Isomorphism* problem is the problem of deciding, given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, whether there is a bijection $\beta : V_1 \to V_2$ such that

$$(u, v) \in E_1 \quad \text{if, and only if,} \quad (\beta(u), \beta(v)) \in E_2,$$

for all $u, v \in V_1$.

The Graph Isomorphism problem is not known to be in P nor known to be NP-complete.

We define GI to be the set of all languages L which are *polynomial-time reducible* to Graph Isomorphism.

What can you conclude from the above definitions and information about the truth of the following statements? If the statement is true or false, justify your answer and if you cannot conclude anything about its truth, explain why that is so.

(*a*) Graph Isomorphism is in NP. [4 marks]

(*b*) Graph Isomorphism is in co-NP. [4 marks]

(*c*) GI $\subseteq$ NP. [3 marks]

(*d*) NP $\subseteq$ GI. [3 marks]

(*e*) P $\subseteq$ GI. [3 marks]

(*f*) GI $\subseteq$ P. [3 marks]

3 Computation Theory

(*a*) Give a precise definition of the collection of *partial recursive functions*. You should define any functions, or constructions on partial functions that you use in your definition. [9 marks]

(*b*) Explain why every partial function computable by a register machine is a partial recursive function. You may assume without proof the existence of suitable primitive recursive functions for manipulating numerical codes of register machine configurations so long as you state their properties precisely. [10 marks]

(*c*) Is every partial recursive function computable by a register machine? [1 mark]

4 Computation Theory

(*a*) Define the terms M of the *λ-calculus* and the relation $M =_\beta M'$ of *β-conversion* between them. [6 marks]

(*b*) For $n \in \mathbb{N}$, what is the nth *Church numeral*? [2 marks]

(*c*) Consider encoding a non-empty list of λ-terms $M_1, M_2, \ldots, M_n$ as the λ-term

$$[M_1, M_2, \ldots, M_n] \triangleq \lambda x\, f.\, f\, M_1(f\, M_2 \ldots (f\, M_n\, x) \ldots)$$

where the variables x and f do not occur free in $M_1, M_2, \ldots, M_n$. Give, with justification, λ-terms Iter, Cons, Append and Nil satisfying

(*i*) $\mathsf{Iter}\, M\, F\, [M_1, M_2, \ldots, M_n] =_\beta F\, M_1(F\, M_2 \ldots (F\, M_n\, M))$ [2 marks]

(*ii*) $\mathsf{Cons}\, M\, [M_1, M_2, \ldots, M_n] =_\beta [M, M_1, M_2, \ldots, M_n]$ [3 marks]

(*iii*) $\mathsf{Append}\, [M_1, \ldots, M_m]\, [N_1, \ldots, N_n] =_\beta [M_1, \ldots, M_m, N_1, \ldots, N_n]$ [3 marks]

(*iv*) $\mathsf{Cons}\, M\, \mathsf{Nil} =_\beta [M]$, $\mathsf{Iter}\, M\, F\, \mathsf{Nil} =_\beta M$ and $\mathsf{Append}\, \mathsf{Nil}\, N =_\beta N$ [4 marks]

5 Logic and Proof

(*a*) Write brief notes on the use of clause methods to prove theorems. Include a description of an algorithm that can find a model of a set of clauses, if one exists. Illustrate your answer using the following example:

$$\{P, Q, \neg R\} \quad \{\neg P, R\} \quad \{\neg Q\} \quad \{P, R\}$$

[6 marks]

(*b*) For each of the following sets of clauses, either exhibit a model or show that none exists. Below, a and b are constants, while x, y and z are variables.

(*i*)

$$\begin{gathered}\{\neg P(x), Q(x,x)\}\\ \{\neg Q(x,y), \neg Q(y,x), R(x,y)\}\\ \{\neg R(x,y), \neg R(y,x)\}\\ \{P(a), P(b)\}\end{gathered}$$

[7 marks]

(*ii*)

$$\begin{gathered}\{P(x), Q(x)\}\\ \{\neg P(x), Q(f(x))\}\\ \{P(x), \neg Q(f(x))\}\\ \{\neg P(x), \neg Q(x)\}\end{gathered}$$

[7 marks]

6 Logic and Proof

(*a*) Write brief notes on Satisfiability Modulo Theories (SMT). Explain how SMT works and what sort of problem it can solve. [4 marks]

(*b*) Outline the basic ideas behind Fourier-Motzkin variable elimination, demonstrating them by solving the following set of constraints:

$$x + z \geq 5 \qquad y + z \geq 5 \qquad y - 2z \geq -2 \qquad x + y + z \leq 7$$

[8 marks]

(*c*) Briefly describe an algorithm for constructing a Binary Decision Diagram (BDD) without first constructing the full binary decision tree. Illustrate your answer by constructing the BDD for $(P \vee R) \to (P \wedge (Q \oplus R))$, where $\oplus$ denotes exclusive OR.

[8 marks]

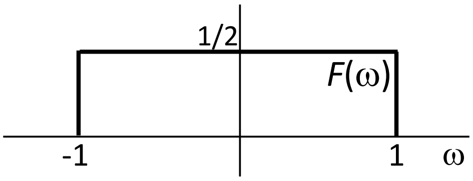
7 Mathematical Methods for Computer Science

(*a*) (*i*) Express W, the primitive N^{th} root of unity, as a complex exponential. [2 marks]

(*ii*) Express the N-point real-valued discrete sequence $f[n] = \cos(2\pi n/N)$ for $n = 1, 2, 3, \ldots, N$ in terms of W. [3 marks]

(*iii*) Using a vector sum diagram in the complex plane, show how elements of the real-valued discrete sequence $f[n]$ are represented as a sum of complex numbers related to W, each having unit length. Construct your diagram for the particular case of integer $n = N/8$. [2 marks]

(*b*) A zero-centred pulse function $F(\omega)$ in the frequency domain ω, having unit area $F(\omega) = 1/2$ for $\omega \in [-1, +1]$, and $F(\omega) = 0$ for $\mid \omega \mid > 1$, represents one ideal low-pass filter.



(*i*) Derive its inverse Fourier transform $f(x)$. [4 marks]

(*ii*) Sketch a plot of this function and specify the roots of $f(x) = 0$. [2 marks]

(*c*) Let $f(x)$ be any real-valued function whose Fourier transform $F(\omega)$ exists. Show that $F(\omega)$ has the property of Hermitian symmetry $F(-\omega) = \overline{F(\omega)}$, and comment on the computational benefits that result from this property.

Hint: Represent $f(x)$ as the sum of an even function $f_e(x)$ plus an odd function $f_o(x)$, where

$$f_e(x) = \frac{1}{2}(f(x) + f(-x))$$
$$f_o(x) = \frac{1}{2}(f(x) - f(-x))$$

and then consider the Fourier transform of $f(x) = f_e(x) + f_o(x)$. You may invoke known properties of even- and odd-symmetric functions without proof. [7 marks]

8 Mathematical Methods for Computer Science

(*a*) (*i*) Consider a random variable X with moment generating function $M_X(t)$. State Chernoff's bound for the probability $\mathbb{P}(X \geq a)$ where a is a constant. [2 marks]

(*ii*) If $X \sim \text{Binomial}(n, p)$ apply Chernoff's bound to X and minimize the upper bound over the values $t > 0$ to show that for $np < a < n$

$$\mathbb{P}(X \geq a) \leq \left(\frac{np}{a}\right)^a \left(\frac{n(1-p)}{n-a}\right)^{n-a} .$$

[8 marks]

(*b*) An online service company receives n tasks per unit time and wishes to serve these tasks using m servers. The allocation of the tasks to the servers is by a randomized load balancing strategy that assigns each of the n tasks independently and uniformly to one of the m servers. Each server can serve up to and including t tasks per unit time without becoming overloaded. Let X_i for $i = 1, 2, \ldots, m$ be the random number of tasks assigned to the i^{th} server in a given unit of time.

(*i*) What is the marginal distribution of X_i for each $i = 1, 2, \ldots, m$? [2 marks]

(*ii*) State whether or not the random variables X_i for $i = 1, 2, \ldots, m$ are mutually independent. Justify your result. [3 marks]

(*iii*) Let $Y_m = \max\{X_1, X_2, \ldots, X_m\}$ and show that

$$\mathbb{P}(Y_m \geq a) \leq m\mathbb{P}(X_i \geq a) \qquad i = 1, 2, \ldots, m\,.$$

You may assume without proof that if $A_1, A_2, \ldots, A_r$ are random events then $\mathbb{P}(\cup_{i=1}^r A_i) \leq \sum_{i=1}^r \mathbb{P}(A_i)$. [2 marks]

(*iv*) The company asks your advice about a suitable number of servers to rent so that the probability that at least one of the servers is overloaded in a given unit of time is no greater than 0.01. Determine an expression for the least value of m such that the stated criterion is met. [3 marks]

9 Semantics of Programming Languages

Consider the imperative language syntax below. Here n ranges over 32-bit numbers $\mathbb{N}_{32} = [0, .., 2^{32} - 1]$, with modular addition $\oplus$, and x ranges over an infinite set of identifiers.

$$e ::= \quad n \mid \textbf{ref}\, e \mid !e \mid e := e' \mid \textbf{skip} \mid e; e' \mid x \mid \textbf{let}\, x = e\, \textbf{in}\, e'$$

We give it two semantics. The first extends the syntax with abstract locations l (taken from some infinite set L) and has an abstract store s, a finite partial function from abstract locations to values $v ::= n \mid l$. The initial abstract store s_0 is the partial function with empty domain. The semantic rules are all standard; the most interesting are shown below for reference.

$$\boxed{\langle e_1, s_1 \rangle \longrightarrow \langle e_2, s_2 \rangle}$$

$$\frac{l \notin \textbf{dom}\,(s)}{\langle \textbf{ref}\, v, s \rangle \longrightarrow \langle l, s + \{l \mapsto v\} \rangle}\ \text{REF1} \qquad \frac{l \in \textbf{dom}\,(s) \wedge s(l) = v}{\langle !l, s \rangle \longrightarrow \langle v, s \rangle}\ \text{DEREF1}$$

$$\frac{l \in \textbf{dom}\,(s)}{\langle l := v, s \rangle \longrightarrow \langle \textbf{skip}, s + \{l \mapsto v\} \rangle}\ \text{ASSIGN1}$$

For the second semantics we have a concrete store M, a total function from concrete addresses $n \in \mathbb{N}_{32}$ to values which here are also just numbers $n' \in \mathbb{N}_{32}$, together with a counter $a \in \mathbb{N}_{32}$ that records the next unallocated address. This semantics uses the abstract syntax exactly as above, without abstract locations. The initial concrete store M_0 maps all addresses to 0; the initial $a_0 = 0$. The interesting rules are:

$$\boxed{\langle e_1, M_1, a_1 \rangle \Longrightarrow \langle e_2, M_2, a_2 \rangle}$$

$$\frac{}{\langle \textbf{ref}\, n, M, a \rangle \Longrightarrow \langle a, M + \{a \mapsto n\}, a \oplus 1 \rangle}\ \text{REF1'} \qquad \frac{M(n) = n'}{\langle !n, M, a \rangle \Longrightarrow \langle n', M, a \rangle}\ \text{DEREF1'}$$

$$\frac{}{\langle n := n', M, a \rangle \Longrightarrow \langle \textbf{skip}, M + \{n \mapsto n'\}, a \rangle}\ \text{ASSIGN1'}$$

Consider expressions e of the form $\textbf{let}\, x = \textbf{ref}\, 3\, \textbf{in}\, e'; !x$, where e' does not contain any free occurrences of x or any abstract locations l.

(*a*) Can e (with the initial store) reduce to a value different from 3, (i) in the abstract semantics or (ii) in the concrete semantics? In each case, either give an example and explain it or give a careful informal argument why not. [8 marks]

(*b*) Define a large subset of the expressions that reduce to the same value in both semantics. Explain your answer. [8 marks]

(*c*) Discuss the advantages and disadvantages of the two semantics for a C-like systems programming language. [4 marks]

10 Semantics of Programming Languages

Explain the issues involved in the design of type systems with subtyping for a language with types

$$\begin{array}{rcl} T & ::= & \textsc{integer} \mid \textsc{real} \mid \textsc{bool} \mid \textsc{unit} \mid T_1 \rightarrow T_2 \mid \\ & & \{lab_1 : T_1, .., lab_k : T_k\} \mid T \ \textsc{ref} \end{array}$$

You should include precise rules for subsumption and for the definition of the subtype relation <:. You should also include examples as appropriate, but there is no need to include the standard operational semantics or expression typing rules.

[20 marks]

END OF PAPER