COMPUTER SCIENCE TRIPOS Part IB

Thursday 6 June 2013 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	SPECIAL REQUIREMENTS
Script paper	*Approved calculator permitted*
Blue cover sheets	
Tags	

1 Complexity Theory

Consider the following decision problems:

> **Prime**: given a positive integer n written in binary, determine whether n is prime.
>
> **Factor**: given positive integers n and k, written in binary, determine whether n has a factor h such that $1 < h < k$.
>
> **Subset Sum**: given a collection of positive integers $a_1, \ldots, a_l$ and t determine whether there is a set $S \subseteq \{1, \ldots, l\}$ such that $\sum_{i \in S} a_i = t$.

(*a*) For each of the three problems above and each of the complexity classes P, NP and co-NP, state what complexity classes the problem is in, with brief justification. [8 marks]

(*b*) For each of the following statements, state whether it is true, false, or unknown. In each case, give justification for your answer and in the case where the truth of the statement is unknown, state any implications that might follow from it being true or false.

(*i*) **Prime** is NP-complete.

(*ii*) **Factor** is NP-complete.

(*iii*) Since multiplication and factorization are inverses of each other, multiplication is a one-way function.

(*iv*) **Factor** is in PSpace.

[3 marks each]

2 Complexity Theory

(*a*) State what it means for a graph $G = (V, E)$ to be 3-colourable. [2 marks]

(*b*) What is known about the complexity of deciding whether a given graph G is 3-colourable? [2 marks]

(*c*) Given a graph $G = (V, E)$ and a *partial* function $\chi : V \hookrightarrow \{1, 2, 3\}$, we define the graph G' by the following actions on G:

- for each pair $u, v \in V$ such that $\chi(u)$ and $\chi(v)$ are both defined and $\chi(u) \neq \chi(v)$, add an edge (u, v) to the graph; and
- for each pair $u, v \in V$ such that $\chi(u)$ and $\chi(v)$ are both defined and $\chi(u) = \chi(v)$, add new vertices w_1 and w_2 to the graph, along with the edges $(w_1, w_2), (u, w_1), (u, w_2), (v, w_1)$ and (v, w_2).

Prove that G' as constructed above is 3-colourable if, and only if, there is a valid 3-colouring of G that extends the partial function χ. [6 marks]

(*d*) Assume $\mathsf{P} = \mathsf{NP}$. Using this assumption and the construction in (*c*), describe a polynomial-time algorithm A which does the following:

> A takes as input a graph G. If G is not 3-colourable, A returns "no".
> If G is 3-colourable, A returns a valid 3-colouring of G.

[10 marks]

3 Computation Theory

(*a*) What does it mean for a register machine to be *universal*? [4 marks]

(*b*) Define what it means for a partial function $f \in \mathbb{N}^n \rightharpoonup \mathbb{N}$ to be *register machine computable.* [3 marks]

(*c*) Show that the following functions f, g, h, k are register machine computable.

(*i*) The partial function $f \in \mathbb{N} \rightharpoonup \mathbb{N}$ that is everywhere undefined. [1 mark]

(*ii*) $g(x_1, x_2) = \begin{cases} x_1 - x_2 & \text{if } x_1 \geq x_2 \\ 0 & \text{if } x_1 < x_2 \end{cases}$ [4 marks]

(*iii*) $h(x_1) = \begin{cases} 2^{x_1 - 1} & \text{if } x_1 > 0 \\ \text{undefined} & \text{if } x_1 = 0 \end{cases}$ [4 marks]

(*iv*) $k(x_1, x_2) = 1$ if the register machine program with index x_1, when started with 0 in all registers, halts in at most x_2 steps; and $k(x_1, x_2) = 0$ otherwise. [4 marks]

4 Computation Theory

(*a*) (*i*) What does it mean for a λ-term to be a *β-normal form*? Defining the sets of canonical (C) and neutral (U) λ-terms by the grammar

$$\begin{aligned} C &::= \lambda x.\, C \mid U \\ U &::= x \mid U\, C \end{aligned}$$

show that a λ-term is a β-normal form if and only if it is canonical. [5 marks]

(*ii*) Carefully stating any standard properties of β-reduction, explain why a λ-term reduces to at most one β-normal form (up to α-equivalence). [4 marks]

(*iii*) Give an example of a λ-term that does not reduce to any β-normal form. [2 marks]

(*b*) (*i*) Define what it means for a closed λ-term F to *represent* a partial function $f \in \mathbb{N} \rightharpoonup \mathbb{N}$. [4 marks]

(*ii*) The composition of partial functions $f, g \in \mathbb{N} \rightharpoonup \mathbb{N}$ is the partial function $g \circ f = \{(x, z) \mid (\exists y)\ (x, y) \in f \wedge (y, z) \in g\} \in \mathbb{N} \rightharpoonup \mathbb{N}$. Suppose F represents f, G represents g, and f and g are totally defined. Show that $\lambda x.\, G\,(F\,x)$ represents $g \circ f$. [2 marks]

(*iii*) Give an example to show that $\lambda x.\, G\,(F\,x)$ need not represent $g \circ f$ when f and g are not totally defined. [3 marks]

5 Logic and Proof

(*a*) In the context of clause methods in theorem proving, define and discuss the concept of a *pure literal.* [3 marks]

(*b*) Use the DPLL method to find a model satisfying the following set of formulas or to prove that no such model exists.

$$(P \land R) \to Q$$
$$\neg(P \land Q \land R)$$
$$(R \lor \neg Q) \to P$$
$$P \to R$$

[5 marks]

(*c*) For each of the following formulas, either exhibit a formal proof (in a sequent or tableau calculus) or exhibit a falsifying interpretation.

(*i*)

$$\forall x(P(x) \to Q(x)) \to (\exists x P(x) \to \exists x Q(x))$$

[6 marks]

(*ii*)

$$\exists x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$$

[6 marks]

6 Logic and Proof

(*a*) In the context of resolution theorem proving, describe the steps involved in transforming a formula of first-order logic into clause form, briefly justifying each step. [4 marks]

(*b*) For each of the following sets of clauses, either derive the empty clause or demonstrate that the set is satisfiable by exhibiting a model. Below, a and b are constants, while x, y and z are variables.

(*i*)

$$\begin{array}{rl} \{P(a), P(b)\} & \{\neg P(x), Q(f(x)), \neg P(y)\} \\ \{\neg Q(z), R(z)\} & \{\neg Q(x), \neg R(y)\} \end{array}$$

[8 marks]

(*ii*)

$$\begin{array}{rl} \{P(a)\} & \{\neg P(x), Q(f(x)), \neg P(y)\} \\ \{\neg Q(z), R(z)\} & \{\neg R(y), \neg P(y)\} \end{array}$$

[8 marks]

7 Mathematical Methods for Computer Science

(*a*) For vectors $u, v \in V$ in linear space $V = \mathbb{R}^n$ with $u = (u_1, u_2, \ldots, u_n)$, define the Euclidean norm $||u||$, and state the triangle inequality for $||u + v||$. [2 marks]

(*b*) Define cyclical convolution of two periodic sequences $f[n]$ and $g[n]$. [2 marks]

(*c*) If $\Psi(x)$ is a generating (or "mother") wavelet, give the dyadic shifting and scaling operations that generate her "daughter" wavelets $\Psi_{jk}(x)$ in terms of dilates j and translates k of $\Psi(x)$. [2 marks]

(*d*) Why is the dyadic property of wavelets useful for analysing naturally-arising data that often exhibits self-similarity across scales? [2 marks]

(*e*) Derive the Fourier series of a periodic triangle wave, $f(x) = |x|$ for $x \in [-\pi, \pi]$ [4 marks]

(*f*) The Modulation Theorem asserts that if $f(x)$ has Fourier transform $F(\omega)$, then modulating $f(x)$ at frequency c (multiplying it by e^{icx}) simply shifts its transform up by c to become $F(\omega - c)$. Prove this, and explain one important practical application of this property. [4 marks]

(*g*) Show how Fourier methods enable solution of differential equations such as the following, in which the function $g(x)$ is known (hence its Fourier transform $G(\omega)$ can be computed), and a, b, c are constant coefficients. Derive an expression for $f(x)$ that solves this differential equation.

$$a\frac{d^2 f(x)}{dx^2} + b\frac{df(x)}{dx} + cf(x) = g(x)$$

[4 marks]

8 Mathematical Methods for Computer Science

(*a*) Given a random variable, X, with mean μ, variance σ^2 and a constant $c \geq 0$ prove *Chebyshev's inequality* in the form

$$\mathbb{P}(|X - \mu| \geq c) \leq \frac{\sigma^2}{c^2}$$

[5 marks]

(*b*) Suppose now that X is a random variable taking values in the interval $[a, b]$ with a mean μ and a variance σ^2. Define the function $f(\alpha) = \mathbb{E}((X - \alpha)^2)$ for $\alpha \in \mathbb{R}$ and show that $f(\alpha)$ is minimized by the choice $\alpha = \mu$. Show that

$$f\left(\frac{a+b}{2}\right) = \mathbb{E}((X-a)(X-b)) + \frac{(b-a)^2}{4}$$

and hence that $\text{Var}(X) \leq (b-a)^2/4$. In the case that X is a Bernoulli random variable show that $\text{Var}(X) \leq 1/4$. [5 marks]

(*c*) Let p be the fraction of computers that are running normally on some network and $1 - p$ the fraction that need rebooting. Suppose that you test n of the computers choosing independently and without replacement. Let X_i be the Bernoulli random variable recording the result of the ith test for $i = 1, \ldots, n$. Write $P_n = \sum_{i=1}^{n} X_i/n$ for the proportion of computers in your sample that were found to be running normally and show that

$$\mathbb{P}(|P_n - p| \geq \epsilon) \leq \frac{p(1-p)}{n\epsilon^2}$$

if p is known. However, if p is unknown show that

$$\mathbb{P}(|P_n - p| \geq \epsilon) \leq \frac{1}{4n\epsilon^2}$$

[5 marks]

(*d*) Now suppose that you wish to determine the least sample size n such that

$$\mathbb{P}(|P_n - p| \geq \epsilon) \leq \delta$$

for given choices of ϵ and δ. What happens to the value of n as recommended by the Chebyshev inequality in part (*c*) in each of the following two cases?

(*i*) the value of ϵ is halved

(*ii*) the probability δ is halved

[5 marks]

9 Semantics of Programming Languages

This question is about a language that is like L1 but with a stack instead of a store.

(*a*) Consider the following grammars for expressions e and values v:

$$\begin{aligned} e &::= \mathsf{push}(e) \mid \mathsf{pop}() \mid \mathsf{skip} \mid e_1 \,;\, e_2 \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\, e_1 \,\mathsf{then}\, e_2 \,\mathsf{else}\, e_3 \\ v &::= \mathsf{skip} \mid \mathsf{true} \mid \mathsf{false}. \end{aligned}$$

The configurations for this language are pairs $\langle e\ ,\ bs\rangle$ where e is an expression and bs is a finite list of booleans.

The operational semantics of $\mathsf{push}(e)$ and $\mathsf{pop}()$ are defined by the following rules:

$$\frac{-}{\langle \mathsf{push}(\mathsf{true})\ ,\ bs\rangle \longrightarrow \langle \mathsf{skip}\ ,\ (\mathsf{true} :: bs)\rangle} \qquad \frac{\langle e\ ,\ bs\rangle \longrightarrow \langle e'\ ,\ bs'\rangle}{\langle \mathsf{push}(e)\ ,\ bs\rangle \longrightarrow \langle \mathsf{push}(e')\ ,\ bs'\rangle}$$

$$\frac{-}{\langle \mathsf{push}(\mathsf{false})\ ,\ bs\rangle \longrightarrow \langle \mathsf{skip}\ ,\ (\mathsf{false} :: bs)\rangle} \qquad \frac{-}{\langle \mathsf{pop}()\ ,\ b :: bs\rangle \longrightarrow \langle b\ ,\ bs\rangle}$$

Write down rules for the other language constructs, to define a reasonable operational semantics. [5 marks]

(*b*) The types for this language are

$$T \ ::= \ \mathsf{unit} \mid \mathsf{bool}$$

We define a relation $e : T$ between expressions and types. The types of $\mathsf{push}(e)$ and $\mathsf{pop}()$ are given by the following rules:

$$\frac{e : \mathsf{bool}}{\mathsf{push}(e) : \mathsf{unit}} \qquad \frac{-}{\mathsf{pop}() : \mathsf{bool}}$$

Write down rules for the other language constructs to define a reasonable type system. [5 marks]

(*c*) Consider the following statements:

(*i*) For all pairs of configurations $\langle e\ ,\ bs\rangle$, $\langle e'\ ,\ bs'\rangle$, and all types T:
if $e : T$ and $\langle e\ ,\ bs\rangle \longrightarrow \langle e'\ ,\ bs'\rangle$ then $e' : T$.

(*ii*) For all configurations $\langle e\ ,\ bs\rangle$ and all types T:
if $e : T$ then either e is a value or there is a configuration $\langle e'\ ,\ bs'\rangle$ such that $\langle e\ ,\ bs\rangle \longrightarrow \langle e'\ ,\ bs'\rangle$.

For each of these two statements, state whether it holds. If it holds, prove it. If it doesn't hold, explain why and suggest a change to the semantics that would make the theorem hold. [10 marks]

10 Semantics of Programming Languages

This question is about a variation on a fragment of the L2 language in which functions take two arguments. The language has the following expressions:

$$e \quad ::= \quad x \;\mid\; \mathsf{fn}\,(x_1, x_2) \Rightarrow e \;\mid\; e_0\,(e_1, e_2) \;\mid\; n$$

where x ranges over variables and n ranges over integers. As usual, $\mathsf{fn}\,(x, y) \Rightarrow e$ is binding: we work up-to α-equivalence and require that x and y are different.

(*a*) Write down a call-by-name operational semantics for this language. [2 marks]

(*b*) Consider the following type system. The types are

$$T \;::=\; \mathsf{int} \mid \mathsf{ret} \mid (T_1, T_2) \to \mathsf{ret}$$

A context Γ is a finite partial function from variables to types. The type system is given by the following rules:

$$\frac{-}{\Gamma, x : T, \Gamma' \vdash x : T} \qquad \frac{\Gamma \vdash e_0 : (T_1, T_2) \to \mathsf{ret} \quad \Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash e_0\,(e_1, e_2) : \mathsf{ret}}$$

$$\frac{-}{\Gamma \vdash n : \mathsf{int}}\ (n \text{ is an integer}) \qquad \frac{\Gamma, x_1 : T_1, x_2 : T_2 \vdash e : \mathsf{ret}}{\Gamma \vdash \mathsf{fn}\,(x_1, x_2) \Rightarrow e : (T_1, T_2) \to \mathsf{ret}}$$

(The idea is that $(T_1, T_2) \to \mathsf{ret}$ is a type of functions taking arguments of type T_1 and T_2. However, there are no expressions of type ret in the empty context, and so rather than returning a result you pass it to a 'continuation'.)

(*i*) Find a type T for which $\vdash \mathsf{fn}\,(x, k) \Rightarrow k\,(3, x) : T$, giving a derivation. [3 marks]

(*ii*) Give a derivation of the following judgement: [2 marks]

$$k : (\mathsf{int}, \mathsf{ret}) \to \mathsf{ret} \vdash \mathsf{fn}\,(x, l) \Rightarrow l\,(7, k) : (\mathsf{int}, (\mathsf{int}, (\mathsf{int}, \mathsf{ret}) \to \mathsf{ret}) \to \mathsf{ret}) \to \mathsf{ret}$$

(*c*) Prove the following 'progress' theorem for this language: [6 marks]

> If $\vdash e : T$ then either $e = (\mathsf{fn}\,(x, y) \Rightarrow e')$, or e is an integer, or there is e' such that $e \longrightarrow e'$.

(*d*) We now consider the situation where there is a type posint of positive integers which is a subtype of int.

Define a subtyping relation <: and extend the type system to accommodate it. Demonstrate it by giving a derivation of the following judgement:

$$k : (\mathsf{int}, \mathsf{ret}) \to \mathsf{ret} \vdash \mathsf{fn}\,(x, l) \Rightarrow l\,(7, k) : (\mathsf{int}, (\mathsf{int}, (\mathsf{posint}, \mathsf{ret}) \to \mathsf{ret}) \to \mathsf{ret}) \to \mathsf{ret}$$

[7 marks]

END OF PAPER