UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS Part IB

Thursday 4 June 2015 1.30 to 4.30 pm

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
Rough work pad

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Complexity Theory

(*a*) Give precise definitions of each of the following:

(*i*) the complexity class NP;

(*ii*) polynomial-time reduction; and

(*iii*) NP-complete problem.

[3 x 2 marks]

(*b*) An instance of a *linear programming* problem consists of a set $X = \{x_1, \ldots, x_n\}$ of variables and a set of *integer constraints*, each of which is of the form

$$\sum_{1 \le i \le n} a_i x_i \ \le \ b,$$

where each a_i and b is an integer.

The 0-1 Integer Linear Programming feasibility problem (ILP) is, to determine, given such a linear programming problem, whether there is an assignment of values from the set $\{0, 1\}$ to the variables in X so that substituting these values in the constraints leads to all constraints being simultaneously satisfied.

(*i*) Consider a *clause* c, i.e. a disjunction of Boolean literals. Show how such a clause can be converted to an integer constraint which has a $\{0, 1\}$-solution if, and only if, c is satisfiable. [4 marks]

(*ii*) Use part (*b*)(*i*) to show that there is a polynomial-time reduction from the problem CNF-SAT to ILP. [4 marks]

(*iii*) Is there a polynomial-time reduction from ILP to CNF-SAT? Justify your answer. [4 marks]

(*iv*) What can you conclude about the complexity of ILP? [2 marks]

2 Complexity Theory

(*a*) Give precise definitions of each of the following complexity classes:

(*i*) P;

(*ii*) L; and

(*iii*) NL.

[3 x 2 marks]

(*b*) State the Space Hierarchy Theorem. [2 marks]

(*c*) For the purposes of this question, let $\mathsf{L2}$ denote the complexity class $\mathsf{SPACE}((\log n)^2)$.

For each of the following inclusions between complexity classes, state whether it is true, false or unknown, giving full justification for your answer.

(*i*) $\mathsf{L} \subseteq \mathsf{L2}$;

(*ii*) $\mathsf{L2} \subseteq \mathsf{L}$;

(*iii*) $\mathsf{L2} \subseteq \mathsf{P}$; and

(*iv*) $\mathsf{NL} \subseteq \mathsf{L2}$.

[4 x 3 marks]

3 Computation Theory

(*a*) What does it mean for a partial function to be *register machine computable*? [3 marks]

(*b*) Give definitions of bijective codings of pairs of numbers $(x, y) \in \mathbb{N}^2$ as numbers $\langle x, y\rangle \in \mathbb{N}$; and of finite lists of numbers $\ell \in \mathit{list}\,\mathbb{N}$ as numbers $\ulcorner \ell \urcorner \in \mathbb{N}$. [3 marks]

(*c*) Let T be the subset of $\mathbb{N}^3$ consisting of all triples $(e, \ulcorner[x_1, x_2, \ldots, x_m]\urcorner, t)$ such that the computation of the register machine with index e halts after t steps when started with $\mathtt{R}_0 = 0, \mathtt{R}_1 = x_1, \ldots, \mathtt{R}_m = x_m$ and all other registers zeroed. Define a function $s \in \mathbb{N} \to \mathbb{N}$ as follows. For each $n \in \mathbb{N}$, $s(n) \in \mathbb{N}$ is the maximum of the finite set of numbers $\{t \mid \exists e, x \in \mathbb{N}.\ \langle e, x\rangle \leq n \ \wedge\ (e, x, t) \in T\}$.

Prove that for all recursive functions $r \in \mathbb{N} \to \mathbb{N}$, there exists some $n \in \mathbb{N}$ with $r(n) < s(n)$. Any standard results about register machines and about recursive functions that you use should be clearly stated, but need not be proved. [14 marks]

4 Computation Theory

(*a*) Give inductive definitions of the relations $M \to N$ and $M \twoheadrightarrow N$ of *single-step* and *many-step* β*-reduction* between λ-terms M and N. (You may assume the definition of α-conversion, $M =_\alpha N$.) [6 marks]

(*b*) *Turing's fixed point combinator* is the λ-term $\mathtt{A}\,\mathtt{A}$ where $\mathtt{A} = \lambda x.\lambda y.\, y(x\,x\,y)$. Use it to show that given any λ-term M, there is a λ-term X satisfying $X \twoheadrightarrow M\,X$. [2 marks]

(*c*) The sequence of λ-terms $\mathtt{N}_0, \mathtt{N}_1, \mathtt{N}_2, \ldots$ is defined by $\mathtt{N}_0 = \lambda x.\lambda f.\, x$ and $\mathtt{N}_{n+1} = \lambda x.\lambda f.\, f\,\mathtt{N}_n$. Say that a function $\mathtt{f} \in \mathbb{N}^k \to \mathbb{N}$ is *Scott definable* if there is a λ-term F satisfying that $F\,\mathtt{N}_{n_1} \cdots \mathtt{N}_{n_k} \twoheadrightarrow \mathtt{N}_{\mathtt{f}(n_1,\ldots,n_k)}$ for all $(n_1, \ldots, n_k) \in \mathbb{N}^k$.

(*i*) Show that the successor function , $\mathtt{succ}(n) = n + 1$, is Scott definable. [2 marks]

(*ii*) Show that for any λ-terms M and N, $\mathtt{N}_0\,M\,N \twoheadrightarrow M$ and $\mathtt{N}_{n+1}\,M\,N \twoheadrightarrow N\,\mathtt{N}_n$. Deduce that the predecessor function

$$\mathtt{pred}(n) = \begin{cases} 0 & \text{if } n = 0 \\ n-1 & \text{if } n > 0 \end{cases}$$

is Scott definable. [2 marks]

(*iii*) By considering the λ-terms $P_m = \mathtt{A}\,\mathtt{A}\,(\lambda f.\lambda y.\, y\,\mathtt{N}_m\,(\lambda z.\, S(f\,z)))$ for a suitable choice of S, or otherwise, prove that the addition function $\mathtt{plus}(m, n) = m + n$ is Scott definable. [8 marks]

5 Logic and Proof

(*a*) Give and explain the inference rules of resolution and factoring, in the context of automated theorem proving. Why is factoring necessary for completeness? [5 marks]

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

(*i*)

$$\{P, \neg Q(a), \neg Q(b), R(a)\}$$
$$\{\neg P, Q(x), R(b)\}$$
$$\{\neg R(b), \neg R(x)\}$$

[5 marks]

(*ii*)

$$\{\neg P(x,y), Q(x,y,f(x,y))\}$$
$$\{\neg R(y,z), Q(a,y,z)\}$$
$$\{R(y,z), \neg Q(a,y,z)\}$$
$$\{P(x,g(x)), Q(x,g(x),z)\}$$
$$\{\neg R(x,y), \neg Q(x,w,z)\}$$

[10 marks]

6 Logic and Proof

(*a*) Describe how to test a propositional formula A for satisfiability, exhibiting a model if possible, based on

(*i*) converting A to disjunctive normal form

(*ii*) converting A to a binary decision diagram (BDD)

Briefly describe these alternative forms and state their respective advantages. [6 marks]

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

(*i*) $[\forall x\, \exists y\, Q(x, y)] \Rightarrow \exists y\, Q(y, y)$ [4 marks]

(*ii*) $[\forall x\, (P(x) \to \neg P(x))] \land [\exists y\, P(y)] \to \exists y\, Q(y)$ [4 marks]

(*iii*) $\Box(A \lor B) \to (\Diamond\Box\neg A \to \Diamond\Box B)$ [6 marks]

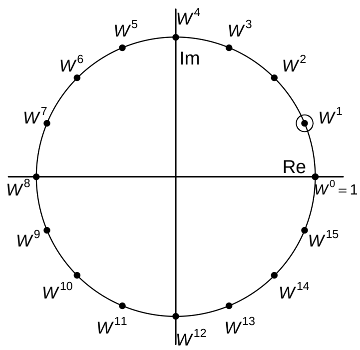
7 Mathematical Methods for Computer Science

(*a*) An inner product space E containing piecewise continuous complex functions $f(x)$ and $g(x)$ on some interval is spanned by the orthonormal basis functions $\{e_i\}$ used in the Fourier series. Thus complex coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ exist such that $f(x) = \sum_i \alpha_i e_i(x)$ and $g(x) = \sum_i \beta_i e_i(x)$.

(*i*) Show that $\langle f, g \rangle = \sum_i \alpha_i \overline{\beta_i}$. [5 marks]

(*ii*) Would the same result hold if the orthonormal basis functions $\{e_i\}$ that span E were *not* the Fourier basis? Justify your answer, and provide the name for coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ in such a case. [2 marks]

(*b*) Consider a sequence $f[n]$ $(n = 0, 1, \ldots, 15)$ with Fourier coefficients $F[k]$ $(k = 0, 1, \ldots, 15)$. Using the 16^{th} roots of unity as labelled around the unit circle in powers of w^1, the primitive 16^{th} root of unity, construct a sequence of these w^i that could be used to compute $F[3]$.



[4 marks]

(*c*) From the well-known fact that a periodic square wave ($f(x) = 1$ for $0 < x < \pi$, $f(x) = -1$ for $\pi < x < 2\pi$, $\cdots$) has the following Fourier series

$$f(x) = \frac{4}{\pi}\left[\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \frac{\sin(7x)}{7} + \cdots\right]$$

produce the first four terms of the Fourier series for the triangle wave whose derivative is this square wave. [4 marks]

(*d*) What sets of frequencies are required to perform the following analyses?

- Fourier transform of a non-periodic continuous function
- Fourier analysis of a piecewise continuous periodic function with period 2π
- Wavelet transform of a non-periodic function, either continuous or discrete

Comment on the relationship between the density of frequencies required and the role of "locality" in the analysis. [5 marks]

8 Mathematical Methods for Computer Science

(*a*) Let X be a random variable with finite mean $\mu = \mathbb{E}(X)$ and finite variance $\sigma^2 = \mathrm{Var}(X)$. State and prove Chebyshev's inequality for the random variable X. You may assume Markov's inequality without proof. [5 marks]

(*b*) Now suppose that X is a continuous random variable with probability density function $f_X(x)$ and finite mean $\mu = \mathbb{E}(X)$ such that

- $f_X(x) = 0 \quad \forall x \notin [\alpha, \beta]$
- $x f_X(x) \leq \gamma \quad \forall x \in [\alpha, \beta]$

where α, β and γ are non-negative real constants with $\alpha < \beta$. Suppose that (A_i, B_i) for $i = 1, 2, \ldots, n$ is a sequence of independent and identically distributed 2-dimensional random variables where A_i and B_i are independent with marginal distributions $A_i \sim U[\alpha, \beta]$ and $B_i \sim U[0, \gamma]$ for each $i = 1, 2, \ldots, n$.

(*i*) Define random variables I_i for $i = 1, 2, \ldots, n$ such that

$$I_i = \begin{cases} 1 & \text{if } B_i \leq A_i f_X(A_i) \\ 0 & \text{otherwise} \end{cases}$$

and set $Z_n = \frac{1}{n}\sum_{i=1}^{n} I_i$. Show that $\mathbb{E}(Z_n) = \mu/(\gamma(\beta - \alpha))$ and that $\mathrm{Var}(Z_n) \leq \frac{1}{4n}$. [5 marks]

(*ii*) Using Chebyshev's inequality show that Z_n converges in probability to the degenerate random variable with value $\mu/(\gamma(\beta - \alpha))$. [5 marks]

(*iii*) Describe an algorithm to estimate the mean μ of the random variable X. You may assume for the purpose of your algorithm that you have a function that returns random points of the given form (A_i, B_i). [5 marks]

9 Semantics of Programming Languages

Consider the following language syntax:

Booleans $b \in \mathbb{B} = \{\textsf{true}, \textsf{false}\}$
Natural numbers $n \in \mathbb{N} = \{0, 1, ...\}$
Locations $\ell \in \mathbb{L} = \{l, l_0, l_1, l_2, ...\}$
Operations $op ::= + \mid - \mid \geq$
Expressions

$$e \quad ::= \quad n \mid b \mid e_1 \;\; op \;\; e_2 \mid \textsf{if} \;\; e_1 \;\; \textsf{then} \;\; e_2 \mid \ell := e \mid !\ell \mid \textsf{skip} \mid e_1; e_2 \mid \textsf{print} \;\; e$$

Types $T ::= \textsf{nat} \mid \textsf{bool} \mid \textsf{unit} \mid T\,\textsf{ref}$

(*a*) Define a reasonable operational semantics and type system for this syntax. Your operational semantics should be expressed as a relation

$$\langle e, s \rangle \xrightarrow{L} \langle e', s' \rangle$$

where s models the store and the label L is either n (for a **print** of that natural number) or τ (for an internal transition). Your type system should be expressed as a relation

$$\Gamma \vdash e : T$$

Make clear what s and Γ range over in your semantics.

[12 marks]

(*b*) Explain the main design choices you made in part (*a*), giving alternative rules or examples (of expressions, transitions, or derivations of transitions or typing) as appropriate. Discuss whether your semantics has type preservation and progress properties. [8 marks]

10 Semantics of Programming Languages

Consider the following syntax:

Booleans $b \in \mathbb{B} = \{\textsf{\textbf{true}}, \textsf{\textbf{false}}\}$
Integers $n \in \mathbb{Z} = \{..., -1, 0, 1, ...\}$
Variables $x \in \mathbb{X} = \{x, y, ...\}$
Expressions $e ::= b \mid n \mid x \mid \textsf{\textbf{fn}}\ x \to e \mid e_1\ e_2 \mid \textsf{\textbf{print}}\ e \mid \textsf{\textbf{skip}}$
(considered up to alpha equivalence, with x binding in e in $\textsf{\textbf{fn}}\ x \to e$)

The set of free variables of an expression $\mathrm{fv}(e)$ are defined in the normal way as follows.

$$\begin{array}{lcl} \mathrm{fv}(b) & = & \{\} \\ \mathrm{fv}(n) & = & \{\} \\ \mathrm{fv}(x) & = & \{x\} \\ \mathrm{fv}(\textsf{\textbf{fn}}\ y \to e) & = & \mathrm{fv}(e) - \{y\} \\ \mathrm{fv}(e_1\ e_2) & = & \mathrm{fv}(e_1) \cup \mathrm{fv}(e_2) \\ \mathrm{fv}(\textsf{\textbf{print}}\ e) & = & \mathrm{fv}(e) \\ \mathrm{fv}(\textsf{\textbf{skip}}) & = & \{\} \end{array}$$

(*a*) Define capture-avoiding substitution $\{e/x\}e'$. [3 marks]

(*b*) Define a small-step right-to-left call-by-value operational semantics for this syntax. Your semantics should be expressed as a relation

$$e \xrightarrow{L} e'$$

where the label L is either n (for a **print** of that integer) or τ (for an internal transition). [5 marks]

(*c*) Explain how a call-by-name semantics would differ, giving any changes required to the rules and giving an example expression that has different output in the two semantics (you should give its transitions in each but need not give their derivations). [3 marks]

(*d*) We are normally interested in closed programs (with no free variables). Prove, with respect to your call-by-value semantics of part (*b*), that if e is closed and $e \xrightarrow{L} e'$ then e' is closed. You can omit the cases for **print**. [9 marks]

END OF PAPER