COMPUTER SCIENCE TRIPOS Part IB

Thursday 9 June 2011 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

The following is a quotation from an Internet forum on cryptography.

> Cracking RSA is NP-complete so nothing better than brute force is possible.

Your task is to evaluate to what extent (if any) this statement is true. For full marks, you will consider the following questions.

- What would it mean, precisely, for "cracking RSA" to be NP-complete? In particular, what is the decision problem involved and what is meant by saying it is NP-complete?
- Is the problem, in fact, NP-complete? Why or why not?
- What is meant, precisely, by the conclusion, "nothing better than brute force is possible"?
- Assuming the premise is correct, i.e. "cracking RSA is NP-complete", does the conclusion follow? Why or why not?
- What is the relationship, more generally, between encryption systems and NP-completeness?

[20 marks]

2 Complexity Theory

(*a*) Recall that a Boolean formula is in 3-CNF if it is the conjunction of clauses, each of which is the disjunction of *at most three* literals. A literal is either a variable or a negated variable.

Consider the following two decision problems:

> 3-SAT: given a Boolean formula in 3-CNF, decide whether or not it is *satisfiable*.
>
> 3-VAL: given a Boolean formula in 3-CNF, decide whether or not it is *valid*.

(*i*) One of the two problems above is known to be in the complexity class **P**. Which one, and why? [2 marks]

(*ii*) Describe a polynomial time algorithm for the problem you identified in part (*i*). [6 marks]

(*iii*) What can you say about the complexity of the other problem? State precisely any standard results you use in your answer. [4 marks]

(*b*) Say that a Boolean formula is in 3-DNF if it is the disjunction of terms, each of which is the conjunction of *at most three* literals.

We now consider the following two decision problems:

> 3-DNF-SAT: given a Boolean formula in 3-DNF, decide whether or not it is *satisfiable*.
>
> 3-DNF-VAL: given a Boolean formula in 3-DNF, decide whether or not it is *valid*.

What can you conclude about the complexity of these two problems? [8 marks]

3 Computation Theory

(*a*) Explain how we can associate a unique number $\lceil P \rceil$ to each register machine program P. [5 marks]

(*b*) Consider the following two partial functions $S : \mathbb{N} \rightharpoonup \mathbb{N}$ and $T : \mathbb{N} \rightharpoonup \mathbb{N}$ on the natural numbers.

$$S(\lceil P \rceil) = \begin{cases} n & \text{if the program } P \text{ when started with 0 in all registers} \\ & \text{halts after } n \text{ steps;} \\ 0 & \text{otherwise.} \end{cases}$$

$$T(\lceil P \rceil) = \begin{cases} n & \text{if the program } P \text{ when started with 0} \\ & \text{in all registers halts after } n \text{ steps;} \\ \textit{undefined} & \text{otherwise.} \end{cases}$$

(*i*) Which, if any, of S and T is computable and which is uncomputable? [4 marks]

(*ii*) Give full justification for your answers above. State carefully any standard results that you use. [11 marks]

4 Computation Theory

(*a*) State precisely what it means for a function $f : \mathbb{N}^k \to \mathbb{N}$ to be *primitive recursive*, giving exact definitions for all operations you use. [5 marks]

(*b*) State precisely what it means for a function $f : \mathbb{N}^k \to \mathbb{N}$ to be λ-*definable*. [5 marks]

(*c*) For each of the following functions, show (using the definitions you gave) that it is primitive recursive and λ-definable.

(*i*) The function $\mathit{square} : \mathbb{N} \to \mathbb{N}$ given by $\mathit{square}(x) = x^2$. [4 marks]

(*ii*) The function $\mathit{fact} : \mathbb{N} \to \mathbb{N}$ given by $\mathit{fact}(x) = x!$. [4 marks]

(*d*) Give a definition of a function that is λ-definable but not primitive recursive. [2 marks]

5 Logic and Proof

(*a*) Recently, automated theorem provers based on the saturation algorithm have become very powerful tools.

(*i*) Exhibit a proof by resolution of the following formula in first-order logic. Include the conversion into a set of clauses and provide brief justification for each step of the proof.

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

[6 marks]

(*ii*) Prove $P(s(s(s(0))))$ by *linear* resolution from the following assumptions:

$$\forall x((P(x) \wedge Q(x)) \to P(s(x)))$$
$$\forall x(P(x) \to Q(x))$$
$$P(0)$$

[7 marks]

(*b*) Binary decision diagrams (BDDs) can be used to represent formulae in propositional logic.

Show the steps in the recursive construction of a BDD, ordered alphabetically, for the following formula:

$$((P \wedge Q) \vee R) \to (Q \vee R)$$

[7 marks]

6 Logic and Proof

(*a*) *Unification* is used to find a common instance of two terms.

(*i*) Briefly explain the unification algorithm, outlining non-trivial cases.
[4 marks]

(*ii*) What is the importance of the *occurs check* in the unification algorithm?
[2 marks]

(*iii*) Find a most general unifier for each of the following three pairs of terms or explain why none exists. Do not rename variables prior to unification.

$$j(a, z, f(x)) \quad j(y, f(y), z)$$

$$f(g(x, y), a, h(z)) \quad f(z, x, y)$$

$$f(g(x), y, x) \quad f(z, f(z), a)$$

[6 marks]

(*b*) Prove or disprove the following sequent of S4 modal logic:

$$\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \wedge B)$$

[8 marks]

7 Mathematical Methods for Computer Science

(*a*) Let X be a random variable with finite mean, $E(X)$, and variance, $\mathrm{Var}(X)$, and let $a > 0$.

(*i*) Show Markov's inequality that $P(|X| \geq a) \leq \frac{E(|X|)}{a}$. [5 marks]

(*ii*) Using Markov's inequality show that $P(|X| \geq a) \leq \frac{E(X^2)}{a^2}$. [5 marks]

(*b*) A study by a mobile phone operator shows that the expected number of simultaneous calls at a base station is 100. The actual number of simultaneous calls is a random variable, X, and so the base station is designed to handle a higher number of simultaneous calls up to a maximum of $M = 150$.

(*i*) Use the Markov inequality to bound the probability that the station will receive more than 150 calls. [5 marks]

(*ii*) Now suppose that we are given the additional information that the variance of the number of simultaneous calls is 50. Use the inequality from part (*a*)(*ii*) to give a second bound on the probability of exceeding 150 calls. [5 marks]

8 Mathematical Methods for Computer Science

Let $f[n]$ be a periodic sequence of period N with N-point Discrete Fourier Transform (DFT) $F[k]$ given by

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-2\pi i n k/N}$$

and inverse transform given by

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k] e^{2\pi i n k/N}$$

Define the *power*, P, of a periodic sequence, $f[n]$, of period N by

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |f[n]|^2$$

(*a*) For each fixed k show that the periodic sequence $\frac{1}{N} F[k] e^{2\pi i n k/N}$ has power $|F[k]|^2/N^2$. [5 marks]

(*b*) If $g[n]$ is a periodic sequence of period N with N-point DFT $G[k]$ show that

$$\sum_{n=0}^{N-1} f[n]\overline{g[n]} = \frac{1}{N} \sum_{k=0}^{N-1} F[k]\overline{G[k]}$$

[5 marks]

(*c*) Show that the power of the periodic sequence $f[n]$ is equal to $\frac{1}{N^2} \sum_{k=0}^{N-1} |F[k]|^2$.

[5 marks]

(*d*) Suppose that $f[n]$ is the periodic sequence given by $f[n] = \sin(2\pi n/N)$ of period N. Recall that $\sin(\theta) = (e^{i\theta} - e^{-i\theta})/2i$.

(*i*) Find $F[k]$ the N-point DFT of $f[n]$. [3 marks]

(*ii*) Show that the power of $f[n]$ is $1/2$. [2 marks]

9 Semantics of Programming Languages

The following grammar specifies the syntax of a simple imperative programming language. It is a fragment of L3.

Values:	v	$::=$ **skip** $\mid n \mid x \mid \ell$
		(n ranges over integers, x over variables, and ℓ over locations)
Expressions:	e	$::=$ $v \mid$ **let** $x = e$ **in** $e' \mid v + v' \mid v := v' \mid\ !v \mid$ **ref**(v)
Types:	T	$::=$ **unit** $\mid$ **int** $\mid T$ **ref**
Stores:	s	finite partial functions from locations to values
Environments:	Γ	finite partial functions from locations and variables to types

Note that the grammar is very restrictive. For instance, the expression $(3 + 4) + 7$ is not allowed.

The language is typed according to the following standard rules.

$$\frac{}{\Gamma \vdash \textbf{skip} : \textbf{unit}} \qquad \frac{}{\Gamma \vdash n : \textbf{int}} \text{ for } n \text{ an integer}$$

$$\frac{}{\Gamma \vdash x : T} \text{ if } \Gamma(x) = T \qquad \frac{}{\Gamma \vdash \ell : T\,\textbf{ref}} \text{ if } \Gamma(\ell) = T\,\textbf{ref}$$

$$\frac{\Gamma \vdash e : T \quad \Gamma, x{:}\,T \vdash e' : T'}{\Gamma \vdash \textbf{let}\ x = e\ \textbf{in}\ e' : T'} \qquad \frac{\Gamma \vdash v : \textbf{int} \quad \Gamma \vdash v' : \textbf{int}}{\Gamma \vdash v + v' : \textbf{int}}$$

$$\frac{\Gamma \vdash v : T\,\textbf{ref} \quad \Gamma \vdash v' : T}{\Gamma \vdash v := v' : \textbf{unit}} \qquad \frac{\Gamma \vdash v : T\,\textbf{ref}}{\Gamma \vdash\ !v : T} \qquad \frac{\Gamma \vdash v : T}{\Gamma \vdash \textbf{ref}(v) : T\,\textbf{ref}}$$

(a) Give a reasonable operational semantics for this language by defining a relation over configurations. [7 marks]

(b) Write down all the reduction steps of the following expression. You do not need to give their derivations.

$$\textbf{let}\ x = \textbf{ref}(0)\ \textbf{in let}\ y = !x\ \textbf{in let}\ z = y + 3\ \textbf{in}\ x := z$$

[3 marks]

(c) State and prove a Type Preservation Theorem for this language.

You may assume the following definition:

> a store s is *well-typed for* Γ, written $\Gamma \vdash s$,
> if for all locations $\ell \in \mathrm{dom}(s)$, there is a type T
> such that $\Gamma(\ell) = T\,\textbf{ref}$ and $\Gamma \vdash s(\ell) : T$

You may also assume the following substitution lemma:

$$\text{If } \Gamma \vdash v : T \text{ and } \Gamma, x : T \vdash e : T' \text{ with } x \notin \mathrm{dom}(\Gamma) \text{ then } \Gamma \vdash \{v/x\}e : T'$$

[10 marks]

10 Semantics of Programming Languages

The following grammar specifies the types and expressions of a simple functional programming language.

$$\begin{array}{lll} \textit{Types:} & T & ::= \mathbf{int} \mid T \to T' \\ \textit{Expressions:} & e & ::= n \mid x \mid e + e' \mid \mathbf{fn}(x : T) \Rightarrow e \mid e\, e' \end{array}$$

where n ranges over all integers, and x ranges over variables.

(a) Give a reasonable semantics for this language, by specifying a type system and a reduction relation. Use the call-by-name evaluation order. [9 marks]

(b) Write down all the reduction steps of the following expression. You do not need to give their derivations.

$$(\mathbf{fn}(x : \mathbf{int}) \Rightarrow (\mathbf{fn}(x : \mathbf{int}) \Rightarrow x + x))\,(1 + 2)\,(3 + 4)$$

[3 marks]

(c) Prove the following property of substitution. [Hint: Use rule induction for $\Gamma, x : T \vdash e' : T'$.]

$$\text{if } \Gamma \vdash e : T \text{ and } \Gamma, x : T \vdash e' : T' \text{ with } x \notin \mathrm{dom}(\Gamma) \text{ then } \Gamma \vdash \{e/x\}e' : T'$$

[8 marks]

END OF PAPER