CST2
COMPUTER SCIENCE TRIPOS Part II

Monday 3 June 2024 13:30 to 16:30

COMPUTER SCIENCE Paper 8

*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 Advanced Computer Architecture

(*a*) For an out-of-order superscalar processor, what are false dependencies on register names and what hardware technique is often used to remove them? [4 marks]

(*b*) Why do out-of-order superscalar processors use a store queue (sometimes called a store buffer) whereas simple scalar processors do not? [4 marks]

(*c*) For an out-of-order superscalar processor, what are the trade-offs between using a reorder buffer and a unified register file to hold computed register values? [4 marks]

(*d*) Consider the following C-code implementation of bubble sort:

```
void bubbleSort(int array[], int size) {
  for (int step = 0; step < size - 1; ++step) {
    for (int i = 0; i < size - step - 1; ++i) {
      if (array[i] > array[i + 1]) {
        int temp = array[i];
        array[i] = array[i + 1];
        array[i + 1] = temp;
      }
    }
  }
}
```

(*i*) The 32-bit ARM ISA allows almost all instructions to be conditional (or *predicated*) whereas the newer 64-bit ARM ISA does not. Using the above code as an example, how could predicated execution be used to avoid data-dependent branch misprediction using if-conversion? [4 marks]

(*ii*) Why do modern out-of-order superscalar processors avoid predicated execution? Why might the above code result in a lot of branch mispredictions? [4 marks]

2 Bioinformatics

(a) Discuss how to use bioinformatics algorithms to detect the specific pathogenic sequences in the genome of a pathogenic species, by comparing its genome with the genome of an evolutionarily-close non-pathogenic species.

Hint. Most pathogenic bacteria have long DNA sequences containing disease-causing genes that are not present in the genome of similar non-pathogenic species. Consider how to detect extra material, and perhaps inverted repeats (which are usually formed during the insertion of the disease-causing genes.)

[5 marks]

(b) Compute the global alignment and the best score of the sequences {CGTGT, TGGCGCC} with the following parameters: match score = +2, mismatch score = −1, gap penalty = −2. Report the final score and alignment(s). [4 marks]

(c) Dimerisation occurs when two similar proteins (P) join together to form a dimer (D), and dissociation reverses this process. The Gillespie algorithm may be applied to model dimerisation and dissociation of proteins, with species P and D, and the following rate constants:

- Dimerisation: $2\mathrm{P} \to \mathrm{D}$ with rate c_1
- Dissociation: $\mathrm{D} \to 2\mathrm{P}$ with rate c_2

Dimerisation is rare and dimers are unstable, therefore $c_2 \gg c_1$. Explain how to use the Gillespie algorithm to model dimerisation and dissociation reactions.

[7 marks]

(d) Discuss advantages and disadvantages of using DNA to store information.

[4 marks]

3 Cryptography

(*a*) *YottaVPN*, your employer's main network-encryption product, generates a master key $K \in_{\mathsf{R}} \{0,1\}^{128}$ and an initial seed $R_0 \in_{\mathsf{R}} \{0,1\}^{80}$ randomly once, when the product is installed. It then uses

$$\text{Algorithm (A):} \qquad R_i = \mathsf{Enc}_K(R_{i-1}) \qquad \text{for } i > 0$$

to generate a stream $R_1, R_2, \ldots$ of session keys for encrypting individual network connections. That algorithm then runs continuously throughout the lifetime of the product. Your colleague suggests to replace (A) with

$$\text{Algorithm (B):} \qquad R_i = \mathsf{Enc}_K(R_{i-1}) \oplus R_{i-1} \quad \text{for } i > 0$$

because they feel that would be more secure. [Enc is a government-approved blockcipher with 80-bit blocksize and $\oplus$ is bit-wise exclusive-or.]

(*i*) For each of algorithm (A) and (B), averaged over all (K, R_0), what is the expected number of different session keys $|\{R_1, R_2, \ldots\}|$ that they will be able to generate from one (K, R_0)? State your assumptions. [5 marks]

(*ii*) What is the smallest number of different values $|\{R_1, R_2, \ldots\}|$ that could be generated by (A) and (B) from any fixed pair (K, R_0)? [2 marks]

(*iii*) Suggest another deterministic key-derivation algorithm (C), using the same blockcipher, 80-bit state and fixed parameters (K, R_0), that maximises $|\{R_1, R_2, \ldots\}|$. [2 marks]

(*iv*) Years later, a worried user discovers that, due to an operator error, the state (K, R_{65535}) of their *YottaVPN* installation was accidentally committed to a publicly accessible Git repository. Compare which other values R_i were compromised by this leak, if either algorithm (A), (B), or (C) had been used. [6 marks]

(*v*) Name a security benefit that could be claimed for algorithm (B) compared to (A). [1 mark]

(*b*) Your colleagues designed a scheme that encrypts messages $M_i \in \{0,1\}^\ell$ with one-time pads $R_i \in_{\mathsf{R}} \{0,1\}^\ell$ into ciphertexts $C_i = M_i \oplus R_i$. But to help estimate the frequency of transmission errors when transferring the R_i, they decided to occasionally replace the last random bit of any R_i with a "parity" bit, with a probability of 0.01. As a result, the probability of any R_i containing an even number of one bits is 0.505. Does this encryption scheme offer *indistinguishability in the presence of an eavesdropper*? Explain your answer. [4 marks]

4 Denotational Semantics

In all parts of this question, you are allowed to use theorems from the course, provided you state them precisely beforehand.

Define the *smash product* $D \otimes E$ of two domains D and E to be the set

$$\{(x, y) \in D \times E \mid x \neq \bot_D \wedge y \neq \bot_E\} \cup \{\bot_{D\otimes E}\}$$

where $\bot_{D\otimes E}$ is a new element which is not a pair. This set is equipped with the order $\sqsubseteq_{D\otimes E}$ such that $\bot_{D\otimes E} \sqsubseteq_{D\otimes E} z$ for any z, and $(x, y) \sqsubseteq_{D\otimes E} (x', y')$ if and only if $x \sqsubseteq_D x'$ and $y \sqsubseteq_E y'$.

(*a*) Show that the smash product of two domains is a domain. [4 marks]

(*b*) Given three domains D, E and F, we call a function $f \in D \times E \to F$ *bistrict* if for any $x \in D$ and $y \in E$, $f(\bot, y) = \bot$ and $f(x, \bot) = \bot$.

Show that not all strict functions are bistrict. [3 marks]

(*c*) Let D, E and F be domains, and $f : D \times E \to F$ a function. Give a condition on the currying $\mathrm{cur}(f) : D \to (E \to F)$ of f that is necessary and sufficient for f to be bistrict. [4 marks]

(*d*) We define the function smash as follows:

$$\begin{array}{lrcll} \text{smash}: & D \times E & \to & D \otimes E & \\ & (x, y) & \mapsto & (x, y) & \text{if } x \neq \bot \text{ and } y \neq \bot \\ & (x, y) & \mapsto & \bot_{D\otimes E} & \text{otherwise} \end{array}$$

Show that if $f\colon D \times E \to F$ is continuous and bistrict, then there exists a unique $\tilde{f}\colon D \otimes E \to F$ that is strict and continuous and such that $f = \tilde{f} \circ \mathrm{smash}$. [4 marks]

(*e*) Give the definition of $X_\bot$, the flat domain on a set X. [1 mark]

Given two sets S and T, show that the domains $(S \times T)_\bot$ and $S_\bot \otimes T_\bot$ are isomorphic, *i.e.* that there exist strict continuous functions $f\colon (S \times T)_\bot \to S_\bot \otimes T_\bot$ and $g\colon S_\bot \otimes T_\bot \to (S \times T)_\bot$ such that $f \circ g = \mathrm{id}$ and $g \circ f = \mathrm{id}$. [4 marks]

5 E-Commerce

(*a*) When preparing to internationalise an E-Commerce business describe four factors that you need to consider and why. [4 marks]

(*b*) Some commentators think that E-Commerce companies should not be considered a separate category of company. Do you think that E-Commerce companies are fundamentally different to traditional companies? Using examples and referencing economic frameworks, give a reasoned argument including points for and against the proposition. [16 marks]

6 Hoare Logic and Model Checking

Consider a programming language with commands C consisting of the `skip` no-op command, sequential composition C_1`;`C_2, loops `while` B `do` C for Boolean expressions B, conditionals `if` B `then` C_1 `else` C_2, assignment X `:=` E for program variables X and arithmetic expressions E, heap allocation X `:= alloc(`E_1`,...,`E_n`)`, heap assignment `[`E_1`] :=` E_2, heap dereference X `:= [`E`]`, and heap location disposal `dispose(`E`)`. Assume `null` $= 0$, and predicates for lists and partial lists:

$$\begin{aligned}
&\text{list}(t, []) = (t = \text{null}) \wedge emp \\
&\text{list}(t, h :: \alpha) = \exists y.(t \mapsto h) * ((t+1) \mapsto y) * \text{list}(y, \alpha) \\
&\text{plist}(t_1, [], t_2) = (t_1 = t_2) \wedge emp \\
&\text{plist}(t_1, h :: \alpha, t_2) = \exists y.\ (t_1 \mapsto h) * ((t_1+1) \mapsto y) * \text{plist}(y, \alpha, t_2)
\end{aligned}$$

In the following, all triples are linear separation logic triples. No proofs are required.

(a) Precisely describe a stack and a heap that satisfy $X \mapsto Y * Y \mapsto X$. Give a (non-looping) command C that satisfies the following triple.
$\{emp\}\ C\ \{X \mapsto Y * Y \mapsto X\}$. [3 marks]

(b) Define and explain a partial correctness rule for a new command `unseq(`C_1`,`C_2`)`, which executes commands C_1 and C_2 in either order (C_1`;`C_2 or C_2`;`C_1). Maintain soundness of the proof system, and ensure the rule accurately reflects the behaviour of the new command. [3 marks]

(c) Do the same for a new command `add_to(`E_1`,`E_2`)`. If expressions E_1 and E_2 evaluate to allocated, disjoint memory locations, it increments the value stored at the first location by the value stored at the second. Otherwise it crashes. [3 marks]

For each of the following triples, give a loop invariant that would prove it.

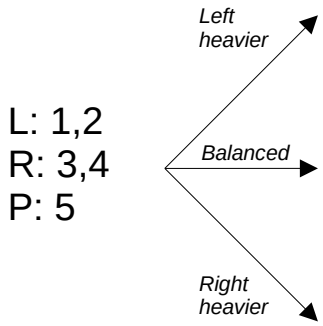
(d) This command duplicates each list element. As per precondition assume Y is initially the head X; assume dup duplicates elements, e.g. dup $[1,2] = [1,1,2,2]$.
$\{\text{list}(X, \alpha) \wedge Y = X\}$
```
while Y≠null do (V:=[Y]; N:=[Y+1]; D:=alloc(V,N); [Y+1]:=D; Y:=N)
```
$\{\text{list}(X, \text{dup}\ \alpha)\}$ [4 marks]

(e) This command removes all negative numbers in a list, assuming it starts with 0.
$\{\text{list}(X, [0]\texttt{++}\alpha)\}$
```
L:=X; Y:=[X+1];
while Y≠null do (
  V:=[Y]; N:=[Y+1];
  (if V<0 (dispose(Y); dispose(Y+1)) else ([L+1]:=Y; L:=Y)); Y:=N
); [L+1]:=null
```
$\{\text{list}(X, [0]\texttt{++}(\text{remove\_negatives}\ \alpha))\}$ [7 marks]

7 Information Theory

Consider a set of coins, identical in appearance, but where some unknown subset is heavier than the others. You have a balance scale with two pans that can be used to tell whether the contents of one pan are heavier, the same or lighter than the other.

Each weighing of coin subsets can be represented graphically as per example below, which identifies the coins on the left pan (L), right pan (R) and those put aside (P). A series of weighings can be represented by a tree of such nodes.



(*a*) Define Discrete Entropy mathematically and conceptually and explain how it can be applied in weighing problems to reduce the overall number of weighings required to find the heavy coins. How would you expect this to compare to a naive strategy of evenly partitioning the heavier set of coins on the next weighing? [5 marks]

(*b*) If the set contains six coins of which one is heavy:

(*i*) Draw the weighings tree for the naive binary partitioning strategy and compute the average number of weighings required. [3 marks]

(*ii*) Draw the weighings tree for the Entropy-based strategy and compute the average number of weighings required. [3 marks]

(*iii*) Reconcile your answer to (*a*) with your answers to (*b*)(*i*), (*b*)(*ii*). [2 marks]

(*c*) If the set contains six coins, *two* of which are heavy, draw an Entropy-based weighing strategy. You should assume the two heavy coins have the same weight as each other. Explain your answer and compute the average number of weighings needed. [7 marks]

8 Machine Learning and Bayesian Inference

Suppose a Bayesian network has the form of a *chain*: a sequence of Boolean random variables $X_1, \ldots X_n$ where $\mathsf{Parents}(X_i) = \{X_{i-1}\}$ for $i = 2, \ldots n$.

$$X_1 \to X_2 \to X_3 \to \cdots \to X_n \tag{1}$$

(*a*) Derive an expression for the probability $\mathsf{Pr}(X_1 = x_1 | X_n = \mathsf{True})$. You may neglect the normalising factor. [2 marks]

(*b*) Derive the time complexity for computing $\mathsf{Pr}(X_1 = x_1 | X_n = \mathsf{True})$ using variable elimination. Contrast against exact inference without memoization. [6 marks]

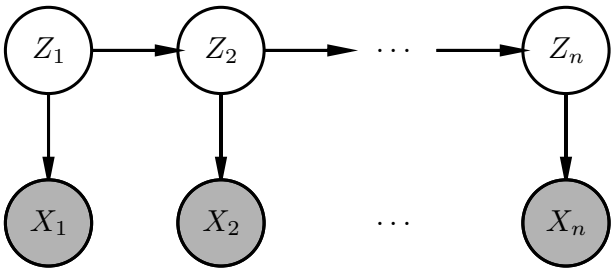
(*c*) State the **E** and **M** steps in the expectation-maximisation algorithm for parameter estimation in a problem involving latent variables Z and observed data X. [4 marks]

(*d*) Henceforth, let θ denote the parameters to be estimated and X denote data we observe. Justify why the EM algorithm locally maximises $p(X|\theta)$ with respect to θ. [3 marks]

(*e*) Consider the Bayesian network depicted in the figure below. $\{X_i\}_{i=1}^n$ denote observed variables while the $\{Z_i\}_{i=1}^n$ are unobserved. We place the following distributions on the random variables:

- Z_1 and $Z_j | Z_{j-1} = l$ are Bernoulli-distributed, where $l \in \{0, 1\}$.
- $X_j | Z_j = l$ is a univariate normal distribution, where $l \in \{0, 1\}$.

Collectively denote the parameters of the above distributions as θ. Give the factorisation of the joint probability indicated by the Bayesian network structure in terms of the given distributions. [5 marks]



9 Optimising Compilers

The following function in C-style code is optimised by a compiler. Assume that variable `arg0` is the argument to the function and so has already been defined.

```
x = arg0;
y = x * 2;
z = x * 4;
while (true) {
   if (x % 5 == 0) {
     y = z + 1;
     print(y);
     break;
   }
   x = x - 1;
}
y = arg0;
print(y);
```

(*a*) What is the live range of a variable? [2 marks]

(*b*) User variables are assumed to reside in the same virtual register in the intermediate representation throughout the entire program. How can static single assignment (SSA) form help reduce their live ranges? [2 marks]

(*c*) Put the code above into SSA form. [4 marks]

(*d*) Describe and give the dataflow equation for live variable analysis. [4 marks]

(*e*) Perform live variable analysis on the original code at the beginning of the question and use it to perform dead-code elimination, showing the *in-live* sets after the analysis. [4 marks]

(*f*) Describe how dataflow analyses, such as live variable analysis, could be simplified if the code was in SSA form. [4 marks]

10 Principles of Communications

(*a*) *Fibbing* is a technique for adding custom forwarding information base entries in a routing domain. A controller masquerades as a router, which injects more specific destinations and shorter paths by some metric, than the ones discovered by the regular link state routing algorithm. Typically, the goal is to support a traffic engineering policy for some destination or source, for example for lower latency, or higher capacity.

A consortium of Internet Service Providers propose to use the same idea for Inter-Domain routing, by announcing *specialised* paths using the Border Gateway Protocol (BGP). They have heard of path-prepending as a technique to influence inbound traffic from neighboring Autonomous Systems (ASs). Of course, they can use local preferences for outbound traffic, so that doesn't need external influence.

(*i*) How can we use the same idea as fibbing to inject BGP announcements that will influence inbound traffic, so as to create different paths for different destinations within this AS? Your answer should address the challenge that BGP is path-vector, not link-state, and that there are local filtering policies that may interfere with path advertisements. [10 marks]

(*ii*) What is the potential security problem (think about trust)? [5 marks]

(*b*) Imagine we wished to optimise a network for reliability, rather than say delay. Consider the approach of minimising delay by iteratively moving a portion of traffic from one path to others. How might this approach be adapted to provide reliability. [5 marks]

11 Quantum Computing

(*a*) In which quantum and classical computational complexity classes is factoring? [2 marks]

(*b*) Shor's algorithm is used to factor $N = 21$. Shor's algorithm requires a positive integer x, which is greater than one and less than N, to be chosen at random.

(*i*) What property must x have for Shor's algorithm not to terminate early? If $x = 14$ is chosen, when does Shor's algorithm terminate? [3 marks]

(*ii*) Instead $x = 4$ is chosen. What is the order of 4 modulo 21? [3 marks]

(*iii*) If Shor's algorithm is run with $x = 4$ explain what happens. Is the correct answer returned? [3 marks]

(*c*) Consider a three-state quantum automaton with initial state $|0\rangle$ and a single accepting state $|2\rangle$. The input letters are c and d, with transition matrices respectively:

$$M_c = \frac{1}{2}\begin{bmatrix} 1+i & 1-i & 0 \\ 1-i & 1+i & 0 \\ 0 & 0 & 2 \end{bmatrix}; \qquad M_d = \frac{1}{2}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1+i & 1-i \\ 0 & 1-i & 1+i \end{bmatrix}$$

(*i*) For quantum automata what property must hold for the transition matrices of each letter of the alphabet? [1 mark]

(*ii*) Verify that this property holds for M_c and M_d. [4 marks]

(*iii*) Give a four-letter input string containing two occurrences of c and two occurrences of d that is accepted with 100% probability. [2 marks]

(*iv*) Give an eight-letter input string containing both c and d that returns to the initial state with 100% probability. [2 marks]

12 Randomised Algorithms

Consider the allocation of n balls into n bins, both labelled $[n] := \{1, 2, \ldots, n\}$. We assume each ball is assigned to a bin chosen uniformly and independently at random.

(*a*) For any given bin, what is the expectation and variance of its load? [2 marks]

(*b*) What is known about the maximum load across all n bins? A proof or justification is not required here. [2 marks]

Assume now that each ball $j \in [n]$ has a random processing time B_j, which has a mean one exponential distribution, i.e., for any $t \geq 0$, $\mathbf{P}[\, B_j \geq t \,] = e^{-t}$. For a bin $i \in [n]$, let T_i be the sum of the processing times of balls allocated to i.

(*c*) Show that $\mathbf{E}[\, T_i \,] = 1$ for every bin $i \in [n]$. For full marks, your answer should include a justification and a formal definition of T_i. [4 marks]

(*d*) Find a constant $c > 0$ such that the probability that a fixed ball has processing time at least $c \cdot \log n$ is at least $n^{-1/2}$? [2 marks]

(*e*) Using part (*d*), argue that with high probability, at least one ball has a processing time of at least $c \cdot \log n$. [4 marks]

(*f*) Let $B := \sum_{j=1}^{n} B_j$ be the total processing time of all n balls. Prove a Chernoff Bound of the form $\mathbf{P}[\, B \geq (1+\delta) \cdot \mathbf{E}[\, B \,]\,]$, for any $\delta > 0$.
Hints: You may use the fact that for Z being exponentially distributed with mean 1, it holds for any $0 < \lambda < 1$ that $\mathbf{E}\left[\, e^{\lambda \cdot Z} \,\right] \leq \frac{1}{1-\lambda}$. Also you may want to choose $\lambda = \frac{\delta}{1+\delta}$ when optimising the tail bound. [6 marks]

13 Types

Consider the simply-typed lambda calculus with only function types and boolean types, with true, false, and if-then-else term formers for the boolean type.

(*a*) Define a logical relation suitable for establishing the termination of programs in this language. [4 marks]

(*b*) State the closure property of the logical relation. [2 marks]

(*c*) Prove closure for the case of the boolean type. [6 marks]

(*d*) State the fundamental lemma for this language. [2 marks]

(*e*) Prove the fundamental lemma for the if-then-else case. [6 marks]

END OF PAPER