CST2
COMPUTER SCIENCE TRIPOS Part II

Monday 8 June 2026 14:00 to 17:00

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*) A superscalar processor may provide many more physical registers than architectural (logical) ones, in some cases ten times as many. Why is it beneficial to have so many registers? [3 marks]

(*b*) When is it safe to release physical registers so they can be reused in a superscalar processor with a unified register file? [4 marks]

(*c*) Give three reasons why VLIW architectures are commonly used in Digital Signal Processors (DSPs). [3 marks]

(*d*) You are asked to build an analytical performance model for a processor with a scalar pipeline. You are given the following parameters for a particular program and processor:

S : number of pipeline stages
B : fraction of instructions that are branches
MP : fraction of branches mispredicted
LS : fraction of instructions that are loads or stores
IM : instruction cache miss rate
IMP : instruction cache miss penalty (in cycles)
DM : data cache miss rate
DMP : data cache miss penalty (in cycles)

Assume a base (or ideal) CPI (Clocks Per Instruction) of 1.0.

(*i*) Given the parameters above, derive an expression for CPI. Given this expression, derive the total execution time T assuming a clock period P and dynamic instruction count C. Briefly describe and justify any assumptions you make. [6 marks]

(*ii*) State two reasons why your model for the pipeline's CPI could be inaccurate in practice. [4 marks]

2 Algebraic Techniques for Programming

Suppose we have a string API with the following constant-time operations:

```
length : string -> int

head : string -> char   (* the first char of a nonempty string *)
tail : string -> string (* a substring omitting the first char *)

last : string -> char   (* the last char of a nonempty string *)
init : string -> string (* the substring omitting the last char *)
```

We can use them to define the function `pal` for computing the length of the longest palindrome which is a subsequence of the input `s`. (Recall that s_1 is a subsequence of s_2 if s_1 can be found by deleting elements of s_2.)

```
let rec pal s =
  if len s <= 1 then
    len s
  else if head s = last s then
    2 + pal (init(tail s))
  else
    max (pal (tail s)) (pal (init s))
```

(*a*) Give an ML datatype and map function implementing the functor corresponding to the recursion scheme of `pal`. [4 marks]

(*b*) Give a coalgebra and algebra implementing the divide and conquer phases of this algorithm. [5 marks]

(*c*) Prove that the coalgebra is well-founded. [5 marks]

(*d*) Give a type signature and non-memoizing implementation for the `hylo` recursion combinator, and then use it to implement `pal`. [2 marks]

(*e*) Estimate the worst-case complexity of the naive algorithm, and then give a bound for the version with a memoizing fixed point. Carefully explain your logic. [4 marks]

3 Bioinformatics

(*a*) A class of proteins recognises patterns of rich CG dinucleotides in genomes.

Similarly to the task of gene identification and transmembrane structure, you can use a Hidden Markov Model (HMM) to detect CG rich regions (islands) by exploiting different nucleotide compositions. Consider a two-state HMM with states $S = \{+, -\}$ (+ = inside CG island, − = outside) and observations $\Sigma = \{A, C, G, T\}$. The parameters are as follows.

Initial probabilities: $\pi_+ = 0.2$, $\pi_- = 0.8$.

Transition probabilities: $a_{++} = 0.8$, $a_{+-} = 0.2$, $a_{-+} = 0.1$, $a_{--} = 0.9$.

Emission probabilities: $e_+(A, C, G, T) = (0.15, 0.35, 0.35, 0.15)$,
$e_-(A, C, G, T) = (0.30, 0.20, 0.20, 0.30)$.

(*i*) Discuss how you would use the Viterbi algorithm to find the most likely state sequence for observation $O = \text{CG}$. [5 marks]

(*ii*) Give initialisation, recursion, and backtracking steps. [6 marks]

(*b*) Consider a multialignment of DNA sequences from four species. You assume to know the relationship between the species and apply the Small Parsimony Problem using Sankoff's algorithm to each column of the multialignment with a cost matrix where match = 0, transition (A ↔ G, C ↔ T) = 1, transversion (A ↔ C, A ↔ T; G ↔ C, G ↔ T) = 2. As input, use a rooted tree with leaves A: `T`, B: `G`, C: `C`, D: `G`, where X is parent of (A,B) and Y is parent of (C,D), and root R is parent of (X,Y).

(*i*) Find the optimal assignment and total parsimony score of the tree. [4 marks]

(*ii*) Discuss how to find alternative trees and then test their robustness. [5 marks]

4 Cryptography

In the following, the *index of coincidence* is the probability that two letters picked uniformly at random from different positions in a sequence of letters are identical.

(*a*) You intercept the following 20-letter sequence:

`CREATETHISNEATDEBATE`

Calculate the index of coincidence for this sequence. [3 marks]

(*b*) Calculate the expected value of the index of coincidence for letter sequences where the probability of a letter appearing at any given position is

$$p_{\mathrm{E}} = 5/20,\ p_{\mathrm{T}} = 4/20,\ p_{\mathrm{A}} = 3/20,$$
$$\text{and } p_a = 1/20 \text{ for eight other letters } a \in \{\mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{H}, \mathtt{I}, \mathtt{N}, \mathtt{R}, \mathtt{S}\}.$$

[*Hint:* the answer will be different from the one for Part (*a*).] [3 marks]

(*c*) What is the expected value of the index of coincidence for a letter sequence where each letter was chosen independently uniformly at random from the alphabet $\{\mathtt{A}, \ldots, \mathtt{Z}\}$? [2 marks]

(*d*) How is the index of coincidence of a letter sequence affected if it is encrypted with a

(*i*) shift cipher, [2 marks]

(*ii*) transposition cipher, [2 marks]

(*iii*) monoalphabetic substitution cipher, [2 marks]

(*iv*) Vigenère cipher? [2 marks]

Briefly justify your answers.

(*e*) How can the index of coincidence help to identify a likely key length used in the Vigenère cipher? [4 marks]

5 Denotational Semantics

(*a*) Let $(D, \sqsubseteq)$ be a poset and let $f\colon D \to D$ be a monotone function.

(*i*) Define the least pre-fixed point of f. [2 marks]

(*ii*) Show that it exists when D is a domain and f is continuous. [5 marks]

(*b*) The *binary join* of a pair $x, y \in D$ is the least upper bound of the set $\{x, y\}$. In other words, it is an element $(x \sqcup y) \in D$ such that

- $x \leq (x \sqcup y)$ and $y \leq (x \sqcup y)$
- for all $z \in D$, if $x \leq z$ and $y \leq z$, then $(x \sqcup y) \leq z$,

Let D be a poset with binary joins and $x \in D$, and consider the function

$$\mathrm{join}_x\colon D \to D \qquad \mathrm{join}_x(y) = x \sqcup y$$

(*i*) Show that join_x is monotone. [3 marks]

(*ii*) Show that join_x is continuous when D is a cpo. [5 marks]

(*c*) Suppose that D is a domain with binary joins, $x \in D$, and $f\colon D \to D$ is a continuous function. Show that there exists an element $y \in D$ such that

- $x \sqsubseteq y$ and $f(y) \sqsubseteq y$
- for every $z \in D$, if $x \sqsubseteq z$ and $f(z) \sqsubseteq z$, then $y \sqsubseteq z$.

[*Hint:* Is y a pre-fixed point of some function?] [5 marks]

6 E-Commerce

You are considering creating an Agentic AI browser product based business where you augment a standard web browser with an Agentic AI agent which is able to access websites and take actions on a user's behalf.

(*a*) Name five elements of Consumer Contract law that you need to keep in mind when designing the Agentic AI agent. [5 marks]

(*b*) (*i*) Choose three different E-Commerce business models and describe how each model can be applied to an Agentic AI browser product based business. [6 marks]

(*ii*) Identify the strengths and weaknesses of each business model in Part (*b*)(*i*). [6 marks]

(*iii*) Say which business model you would recommend be adopted and why. [3 marks]

7 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 $X := \texttt{alloc}(E_1, \ldots, E_n)$, heap assignment $[E_1] := E_2$, heap dereference $X := [E]$, and heap location disposal $\texttt{dispose}(E)$. 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.

(*a*) Find a command C that satisfies the following separation logic total correctness triple, or explain why no such C exists: $[\top]\ C\ [X \mapsto 1 * X \mapsto 1]$. [2 marks]

(*b*) What property asserted by the $*$ operator makes separation-logic reasoning compositional? Describe briefly how this simplifies reasoning about aliasing pointers, in a way that is not possible using Hoare logic. [3 marks]

(*c*) Define a separation logic predicate $\text{clist}(t, \alpha)$ for pointers t and mathematical lists α that represents a circular singly-linked list. You may use the provided list predicates in your definition. [3 marks]

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

(*d*) This command performs a pairwise swap over a list (i.e. swapping every two adjacent elements) by modifying its values. The list is non-empty with even length. The pw_swap function pairwise swaps a mathematical list, e.g. $\text{pw\_swap}\,[1,2,3,4] = [2,1,4,3]$.

$\{\text{list}(X, \alpha) \wedge \alpha \neq [] \wedge \exists n.\ \text{length}\ \alpha = 2n\}$
```
P := X;
while P ≠ null do
  (N1 := P; N2 := [P + 1]; NEXT := [N2 + 1];
  TMP := [N1]; [N1] := [N2]; [N2] := TMP; P := NEXT)
```
$\{\text{list}(X, \text{pw\_swap}\ \alpha)\}$ [5 marks]

(*e*) This command performs a pairwise swap over the tail of a list that starts with 0 by rearranging its structure in memory. The tail is non-empty with even length.

$\{\text{list}(X, [0]{+}{+}\alpha) \wedge \alpha \neq [] \wedge \exists n.\ \text{length}\ \alpha = 2n\}$
```
L := X; Y := [X + 1];
while Y ≠ null do
  (N1 := [Y + 1]; N2 := [N1 + 1];[L + 1] := N1; [N1 + 1] := Y; L := Y; Y := N2);
[L + 1] := null
```
$\{\text{list}(X, [0] \mathbin{++} \text{pw\_swap}\ \alpha)\}$ [7 marks]

8 Information Theory

This question concerns arithmetic coding of messages.

(*a*) You are designing an arithmetic code for a source that emits messages of random length N, where the expected length is $L = E[N]$. You implement a termination symbol ϕ such that $P(\phi) = \epsilon$, scaling the probabilities of the data symbols by $(1 - \epsilon)$.

(*i*) Explain why adding a termination symbol in this way increases the overall message length by more than the cost of emitting the termination symbol itself. [1 mark]

(*ii*) Derive an expression for the total increase in Shannon Information for a message of length N caused by the addition of the terminating character. Explain how this relates to the number of additional bits needed to encode the message. [4 marks]

(*iii*) Derive the optimal value of ϵ that minimises the total overhead for a message of average length L. You may use the approximation $\ln(1 - \alpha) \approx -\alpha$ for small α without proof. [4 marks]

(*b*) An alternative approach to termination symbols is to use a Large Language Model (LLM) as the probability model for the arithmetic coder. LLMs use a special token EoS (End of Sequence) to terminate generation. This EoS symbol can be used as the termination symbol in the coder.

(*i*) Give three disadvantages of using an LLM as the probability model for an arithmetic coder. [3 marks]

(*ii*) Explain how an LLM used in this way can result in shorter encoded messages. [3 marks]

(*iii*) Assume you have an LLM trained on long sequences ($L \sim 1000$) but you need to send messages of various lengths N, including some for which $N \leq 400$. You find that the LLM is assigning $P(\text{EoS}) = 10^{-7}$ for these shorter messages, which makes the termination symbol costly. Your colleague suggests assigning $P(\text{EoS}) = 0.5$ (neutral) for messages with $N \leq 400$. Calculate the message length at which this neutral approach adds a larger overhead than the default LLM approach. [5 marks]

9 Machine Learning and Bayesian Inference

(*a*) You have a training set $\mathbf{s} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ and wish to fit a function to it using linear regression based on the function

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i.$$

You have reason to believe that the zero-mean additive noise in your training data has a Laplace density

$$p_{\mu,b}(\epsilon) = \frac{1}{2b} \exp\left(\frac{-|\epsilon - \mu|}{b}\right)$$

where the mean μ and b are parameters. Provide a full derivation of the *Maximum Likelihood* training algorithm for this scenario. Clearly state any assumptions you make. You may make use of the fact that if ϵ has Laplace density with parameters μ, b then $\epsilon + a$ has Laplace density with parameters $\mu + a, b$. [8 marks]

(*b*) You have reason to believe that the Dirichlet density is a suitable prior on the weight vector $\mathbf{w}$, so

$$p_{\boldsymbol{\alpha}}(\mathbf{w}) = A(\boldsymbol{\alpha}) \prod_{i=0}^{n} w_i^{\alpha_i - 1}$$

where $\boldsymbol{\alpha}$ is a vector of parameters and $A(\boldsymbol{\alpha})$ is a suitable function. Explain in detail how the error derived in Part (*a*) needs to be modified to obtain a *Maximum A Posteriori* training algorithm. [5 marks]

(*c*) Which of the parameters introduced in Parts (*a*) and (*b*), excluding $\mathbf{w}$, need to be chosen as part of the training process? [1 mark]

(*d*) Describe two ways in which the task in Part (*c*) can be achieved, and explain why the error measures derived should not be optimised in a single step with respect to both the parameters and $\mathbf{w}$ simultaneously. [6 marks]

10 Optimising Compilers

(*a*) We create an abstract interpretation named "*known-bits*" analysis, which computes for each value two bitvectors (stored as `uint8_t`):

```
struct AbstractValueKB {
  uint8_t one;  // A bit is `set` if the corresponding bit of
                // the concrete value is known to be '1'.
  uint8_t zero; // A bit is `set` if the corresponding bit of
                // the concrete value is known to be '0'.
}
```

Abstract values are written as a vector of symbols where we use **1** if a given bit is known-one, **0** if a given bit is known-zero and **?** if a given bit is unknown. An example of an abstract value for our known-bits analysis is [1001????], which is stored as the pair (0b10010000, 0b01100000) or, in decimal, (144, 96).

(*i*) Briefly describe the concept of abstract interpretation. Particularly, discuss performance tradeoffs that motivate abstract interpretation. [2 marks]

(*ii*) Abstract interpretations typically define a value named $\perp$ (bottom), used for uninitialised variables. For the known-bits analysis as introduced above, explain which abstract value you would choose to represent the bottom value $\perp$. Justify your choice. [2 marks]

(*iii*) Define correct and reasonable precise abstraction functions $\alpha(a \ \& \ b)$, $\alpha(a \mid b)$, $\alpha(a \wedge b)$, where &, |, and $\wedge$ are bitwise and, or, and xor operators. [7 marks]

(*iv*) For x of form `[????11??]` and y of form `[??00????]`, compute for each variable its maximally precise abstract value.

```
uint8_t foo(uint8_t x, uint8_t y) {
  uint8_t t0 = x * 4;    // [........]
  uint8_t t1 = y >> 2;   // [........]
  uint8_t t2 = t0 && t1; // [........]
  return t2;
}
```

[3 marks]

(*b*) Describe the concept of a peephole rewrite. [2 marks]

(*c*) The following C code initialises an array. C array indexing requires a hidden arithmetic operation. Rewrite the loop to optimise it using strength reduction and loop invariant code motion.

```
int i = 0;
while (i < 100) { v[i] = 42; i++; }
```

[4 marks]

11 Principles of Communications

(*a*) In what way might we consider the goals of congestion control and peak rate price to be similar? [5 marks]

(*b*) It is well-known that when home network users are offered different prices for services depending on the time of day, they change their behaviour. A smart engineer employed by an ISP plans to implement this by adding BGP route selection by time-of-day-price. The engineer plans to carry this out using the three most common mechanisms for BGP traffic engineering, namely preferences, multi-exit discriminators, and AS-path pre-pending. Explain how these mechanisms work to control inbound and outbound traffic for the two following cases: (a) where the engineer's network is multi-homed on one provider, and (b) where the network is multi-homed on two different providers. [10 marks]

For reference, the BGP route selection rules in order are:

```
highest local preference
shortest AS Path
lowest multi-exit discriminator (MED)
i-BGP < e-BGP
lowest IGP cost to BGP egress
lowest router id
```

(*c*) The engineer is considering updating the paths on a one-hourly basis. What are the possible limitations of the traffic engineering mechanisms discussed above with that update frequency? [5 marks]

12 Quantum Computing

The Deutsch-Jozsa algorithm distinguishes the two classes of functions $f : \{0,1\}^n \rightarrow \{0,1\}$, namely those that are constant and those that are balanced. The algorithm relies on an oracle U_f which acts on an n-qubit input register and a 1-qubit target register. Let $|x\rangle$ be the state of the input register and $|y\rangle$ be the state of the target register, such that $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$.

(*a*) How should the target qubit be initialised so that the oracle U_f implements the phase map $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ on the input register? [2 marks]

(*b*) Prove that the initialisation that you proposed in Part (*a*) leads to the desired phase mapping of $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$. [2 marks]

(*c*) Let $|\psi_{out}\rangle$ be the state of the input register immediately before the final computational basis $\{0,1\}$ measurement in the standard Deutsch-Jozsa circuit. Starting from the initial state $|0\rangle^{\otimes n}$, derive the mathematical expression for the probability amplitude of the basis state $|0\rangle^{\otimes n}$ in $|\psi_{out}\rangle$. Your answer should be in the form of a normalised summation involving $f(x)$. [4 marks]

(*d*) Using your result from Part (*c*), determine the probability of measuring the state $|0\rangle^{\otimes n}$ in the following two cases:

(*i*) $f(x)$ is a constant function.

(*ii*) $f(x)$ is a balanced function.

Explain how this measurement result allows the algorithm to distinguish the two classes of functions. [4 marks]

(*e*) Suppose we run the Deutsch-Jozsa algorithm with $n = 4$ input qubits and one ancilla qubit, but the function f violates the promise, i.e. it is neither constant nor balanced. Specifically, $f(x) = 1$ for 4 of the 16 possible inputs, and $f(x) = 0$ for the remaining 12. Calculate the probability that the final measurement outcome is the state $|0000\rangle$. [4 marks]

(*f*) Consider a scenario where the oracle U_f is imperfect. Instead of applying a phase factor of -1 when $f(x) = 1$, it applies a phase factor $e^{i\phi}$ for some fixed angle ϕ. That is, the effective map on the input register is $|x\rangle \mapsto e^{i \cdot f(x) \cdot \phi}|x\rangle$.

If the function f is balanced, derive an expression for the probability of measuring the all-zero state $|0\rangle^{\otimes n}$ as a function of ϕ. Verify that your result yields the expected probabilities for the standard Deutsch-Jozsa cases where $\phi = 0$ and $\phi = \pi$. [4 marks]

13 Types

(*a*) Suppose that $\Gamma; \Delta \vdash t : \neg A$ true and $\Gamma; \Delta, u : A \vdash t' : B$ true are derivable in the proof term assignment for classical logic. Show that there is a term t'' such that $\Gamma; \Delta \vdash t'' : B$ true is derivable. [6 marks]

(*b*) Suppose that $\Gamma; \Delta \vdash t : A \vee B$ true and $\Gamma, x : A; \Delta \vdash t_1 : C$ true and $\Gamma, x : B; \Delta \vdash t_2 : C$ true are derivable in the proof term assignment for classical logic. Show that there is a term t'' such that $\Gamma; \Delta \vdash t'' : C$ true is derivable. [8 marks]

(*c*) (*i*) Give typing rules that extend the simply-typed lambda calculus with introduction and elimination forms for the natural numbers, yielding Gödel's T. [2 marks]

(*ii*) Using the rules defined in Part *(c)(i)*, define the predecessor function $\texttt{pred} : \mathbb{N} \to \mathbb{N}$. In your answer, you may use all the constructs of the simply-typed lambda calculus, including functions, products and sum types. [4 marks]

END OF PAPER