# CST2
# COMPUTER SCIENCE TRIPOS Part II

---

Friday 10 June 2022 11:00 to 14:00 BST

---

COMPUTER SCIENCE Paper 9

*Answer **five** questions.*

*Submit each question answer in a **separate** PDF. As the file name, use your candidate number, paper and question number (e.g., `1234A-p9-q6.pdf`). Also write your candidate number, paper and question number at the start of each PDF.*

**You must follow the official form and conduct instructions for this online examination**

## 1 Advanced Computer Architecture

(*a*) Vector instructions extensions are added to a small 32-bit microcontroller. The vector length is 128-bits. The register bank in the processor's floating-point unit (32 x 32-bit single-precision registers) is reused for vector processing and eight 128-bit vector registers alias onto it. The processor can only issue a single instruction per cycle. It has a 32-bit wide memory datapath and a single 32-bit multiplier.

(*i*) How can adding vector instruction extensions allow us to make more efficient use of the microcontroller's memory datapath and multiplier? [2 marks]

(*ii*) What is the advantage of allowing many vector instructions to be able to access both vector registers and registers in the scalar register file? [2 marks]

(*iii*) Imagine two vector instructions are executing when the first (earlier) instruction causes an exception late in its execution. Describe two different ways in which precise exceptions could be implemented. [5 marks]

(*iv*) Describe one way in which a vector instruction-set extension may efficiently handle cases where the number of elements we wish to process is not a precise multiple of the maximum vector length supported in hardware? [3 marks]

(*b*) Imagine a 64KiB 2-way set-associative L1 data cache with a block size of 32 bytes. The cache is Virtually Indexed Physically Tagged (VIPT). The processor has a private L2 cache which is inclusive. Virtual memory uses 4KiB pages.

(*i*) What problem must be overcome to ensure correctness? [2 marks]

(*ii*) How could the problem be detected by storing a few bits of the virtual page number with each line of the processor's private inclusive L2 cache? [4 marks]

(*iii*) What is the minimum associativity the L1 cache must have to completely avoid the problem? [2 marks]

## 2 Bioinformatics

(*a*) Compute the global and local alignments for the following aminoacid sequences: SEPPT, CAPPR. Use the following scores: match = +2, mismatch = −2, gap penalty = −2. Then in the global alignment change to match = +4, mismatch = −4, gap penalty = −4. Compare and discuss the results. What is the general effect of the gap penalty? [5 marks]

(*b*) Describe in outline the *Four Russians* algorithm. Give expressions for computational complexity as the number of tiles, $t$, is changed and compare the growth order with that of a tile-less approach. [6 marks]

(*c*) Discuss with one example how an algorithm for RNA folding could use a dynamic programming approach. What are the cases when this will not work? [4 marks]

(*d*) Is it possible to remove or add just one node (a species) from/to a phylogenetic tree without actually making the tree again? Discuss using different phylogenetic methodologies. [5 marks]

## 3 Business Studies

Working with friends in college you have identified a new insulation material that you think could reduce the carbon intensity of the University's estate.

(*a*) Discuss what a product using this material needs to accomplish to be successful in the broader market of home insulation products? [6 marks]

(*b*) After considering the market for insulation materials you decide that the best way is to sell the product directly to home owners and recommend trusted builders to install the product. How would you go about creating and managing a sales pipeline for your insulation product? [14 marks]

## 4 Cryptography

(*a*) You have intercepted a ciphertext $c$ from the communications of an international criminal gang, and you need to decrypt it. By decompiling the gang's messaging app you learnt that $c$ is encrypted with AES-128 in CBC mode; the plaintext is padded to a multiple of 128 bits by setting the last plaintext byte to be the padding length in bits (encoded as an 8-bit binary number), and setting the remaining bits between the end of the message and the final length byte to zero.

Moreover, you know that the gang operates a server that is reachable over the Internet; this server internally decrypts any ciphertext you send it, and always replies with "ok" (regardless of whether the decrypted data makes sense or not). You cannot break into the server, but you notice one detail: if the decrypted message has correctly formatted padding, the reply is slightly slower than if it has incorrect padding. Presumably this is because the server spends some time storing correctly formatted messages, while malformed messages are quickly discarded without being stored.

Show that, by repeatedly sending messages to the server, you can recover the entire plaintext from $c$. Explain your technique in detail. [8 marks]

(*b*) The gang figures out that you are decrypting their messages. They decide to continue using AES-128-CBC, but in order to prevent the attack from part (*a*), they add a check to their encryption scheme so that the server rejects any message where the ciphertext has been manipulated. Explain how to securely compute such a check using the SHA-256 hash function. [3 marks]

(*c*) As part of the new check from part (*b*), the server uses the following pseudocode:

```
// tagInMessage and correctTag are byte arrays of equal length
function checkIsMessageOk(tagInMessage, correctTag) {
    for (i = 0 to correctTag.lengthInBytes - 1) {
        if (tagInMessage[i] != correctTag[i]) {
            send "rejected" reply
            return
        }
    }
    send "ok" reply
}
```

Explain the problem with this code, and show how this problem may once again allow you to recover the entire plaintext of an encrypted message. [3 marks]

(*d*) Prove that if a hash function $H(x)$ is collision resistant, then $H(H(x))$ is collision resistant as well. [6 marks]

## 5 Denotational Semantics

You may use standard results provided that you state them clearly.

($a$) For a domain $D$, let *fix* be the function mapping a continuous function $f \in (D \to D)$ to its least pre-fixed point $\mathit{fix}(f) \in D$.

Prove that $\mathit{fix} : (D \to D) \to D$ is continuous. [4 marks]

($b$) For a PCF type $\tau$, let $\Omega_\tau = \mathbf{fix}(\mathbf{fn}\ x : \tau.\ x)$ and consider the following closed PCF terms of type $(\tau \to \tau) \to (\mathit{nat} \to \tau)$.

$$\mathrm{M}_\tau = \ \mathbf{fn}\ f : \tau \to \tau.\ \mathbf{fn}\ n : \mathit{nat}.\ \mathbf{fix}(f)$$

$$\begin{aligned}\mathrm{N}_\tau = \ & \mathbf{fn}\ f : \tau \to \tau. \\ & \quad \mathbf{fix}\big(\ \mathbf{fn}\ h : \mathit{nat} \to \tau.\ \mathbf{fn}\ n : \mathit{nat}. \\ & \qquad f\big(\mathbf{if}\ \mathbf{zero}(n)\ \mathbf{then}\ \Omega_\tau\ \mathbf{else}\ h(\mathbf{pred}(n))\big)\ \big)\end{aligned}$$

Give an explicit description of the denotations $[\![\mathrm{M}_\tau]\!]$ and $[\![\mathrm{N}_\tau]\!]$ in the domain $([\![\tau]\!] \to [\![\tau]\!]) \to (\mathbb{N}_\perp \to [\![\tau]\!])$. [4 marks]

($c$) Recall that the contextual preorder $\vdash M \leq_{\mathrm{ctx}} N : \tau$ holds whenever $M$ and $N$ are closed PCF terms of type $\tau$ and for all PCF contexts $\mathcal{C}$ for which $\mathcal{C}[M]$ and $\mathcal{C}[N]$ are closed PCF terms of type $\gamma \in \{\mathit{nat}, \mathit{bool}\}$ and for all values $V$ of type $\gamma$, if $\mathcal{C}[M] \Downarrow_\gamma V$ then $\mathcal{C}[N] \Downarrow_\gamma V$.

Say whether the following statements concerning the PCF terms in Part ($b$) are true or false and, respectively, either prove or disprove them:

($i$) For all PCF types $\tau$, $\vdash \mathrm{M}_\tau \leq_{\mathrm{ctx}} \mathrm{N}_\tau : (\tau \to \tau) \to (\mathit{nat} \to \tau)$.

(Hint: Consider the case $\tau = \mathit{nat} \to \mathit{nat}$.) [6 marks]

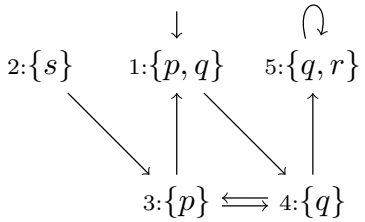
($ii$) For all PCF types $\tau$, $\vdash \mathrm{N}_\tau \leq_{\mathrm{ctx}} \mathrm{M}_\tau : (\tau \to \tau) \to (\mathit{nat} \to \tau)$.

(Hint: Recall that every PCF type is of the form $\tau_1 \to (\cdots(\tau_\ell \to \gamma)\cdots)$ where $\ell \in \mathbb{N}$, $\tau_i$ $(1 \leq i \leq \ell)$ are types, and $\gamma \in \{\mathit{nat}, \mathit{bool}\}$.) [6 marks]

## 6 Hoare Logic and Model Checking

Consider the temporal logic CTL over atomic propositions $p \in \mathit{AP}$:
$\psi \in \mathsf{StateProp} ::= \bot \mid \top \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \psi_1 \rightarrow \psi_2 \mid p \mid \mathsf{A}\ \phi \mid \mathsf{E}\ \phi$,
$\phi \in \mathsf{PathProp} ::= \mathsf{X}\ \psi \mid \mathsf{F}\ \psi \mid \mathsf{G}\ \psi \mid \psi_1\ \mathsf{U}\ \psi_2$

($a$) Consider a temporal model over atomic propositions $\mathit{AP} = \{p, q, r, s\}$, with states $\{1, 2, 3, 4, 5\}$, initial state 1 and transitions and state labelling as shown in the diagram (e.g. in state 1, atomic propositions $p$ and $q$ hold). Informally describe the meaning of each of the following CTL formulae over $\mathit{AP}$ and explain why they hold in the model or give a counter-example if they do not.




($i$) $\mathsf{AG}\ (p \vee q)$ [2 marks]

($ii$) $\mathsf{A}\ ((p \vee q)\ \mathsf{U}\ r)$ [3 marks]

($b$) Specify the following properties as CTL formulae over $\mathit{AP}$ as defined in ($a$).

($i$) Once $r$ holds, $r$ always holds. [3 marks]

($ii$) From every reachable state, it is always possible to reach another state from where on $r$ always holds. [3 marks]

($c$) John's car is getting old and parts can develop problems at any point. The car internally monitors its parts and reports, for each part, either no problem or a warning. When there is a warning for the engine (considered to be a single part) or for any three parts at once (John is lazy), John takes the car to the garage where all problems are fixed.

($i$) Describe a temporal model $M_1$ of the car's status that keeps track of exactly which parts of the car have warnings. Assume initially there are no warnings/problems, and assume that each new state has at most one additional problem compared to the previous state. Use $\mathsf{Parts}$ as the set of parts of the car. Moreover, use $\mathit{AP} = \{\mathsf{needsRepair}\}$ as the set of atomic propositions, where $\mathsf{needsRepair}$ should hold in any state where any part has a warning. [4 marks]

($ii$) Create a more abstract model $M'$ over $\mathit{AP}$ that only tracks the information John cares about, and give a simulation of $M$ by $M'$ (no proof needed). [5 marks]

## 7 Information Theory

(*a*) Explain the notions of Entropy and Mutual Information. Explain how they relate to channel capacity. For a noisy channel, how does an optimal coding affect the distribution of the input? [6 marks]

(*b*) Consider random variables $X$ and $Y$ and let $Z = X + Y$.

(*i*) Can $H(X)$ be greater than $H(Z)$? Either prove it cannot or provide a counterexample. [3 marks]

(*ii*) If $X$ and $Y$ are independent find an expression for $I(X;Z) - I(Y;Z)$ in terms of $H(X)$ and $H(Y)$ only. [5 marks]

(*c*) Consider a random variable, $X$, and a second random variable $Y = f(X)$, where $f$ is a function. Show that $H(Y) \leq H(X)$ and explain what conditions are necessary for equality. [6 marks]

## 8 Machine Learning and Bayesian Inference

You have a labelled data set $\mathbf{s} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ with $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$. The *maximum margin classifier* computes

$$f_{\mathbf{w},w_0}(\mathbf{x}) = w_0 + \mathbf{w}^T \mathbf{\Phi}(\mathbf{x})$$

$$h_{\mathbf{w},w_0}(\mathbf{x}) = \text{sgn}(f_{\mathbf{w},w_0}(\mathbf{x}))$$

where $\text{sgn}(x) = +1$ if $x > 0$ and $\text{sgn}(x) = -1$ otherwise.

(*a*) One approach to training the maximum margin classifier would be to solve the problem

$$(\mathbf{w}, w_0) = \text{argmax} \left[ \min_i \frac{y_i f_{\mathbf{w},w_0}(\mathbf{x}_i)}{||\mathbf{w}||} \right].$$

Explain how this version of the training algorithm is derived, paying particular attention to the meaning of the term $f_{\mathbf{w},w_0}(\mathbf{x}_i)/||\mathbf{w}||$. [5 marks]

(*b*) Explain why the training algorithm in Part (*a*) is not used in practice. [1 mark]

(*c*) Describe in detail two alternative ways of formulating the training of the maximum margin classifier as a constrained optimization problem. You need not describe an algorithm for solving the constrained optimization, but should explain in each case how a combination of objective function and constraints is obtained from first principles. [7 marks]

(*d*) Evil Robot has completed a course on some software called VectorDribble, and now considers himself a *Data Science Expert*. He claims that, as the *support vector machine* and *Gaussian process regressor* both involve a function of the form $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, they are essentially the same method. Explain, in as much detail as you can, why Evil Robot is mistaken. [7 marks]

## 9 Optimising Compilers

You work for a company that writes compilers for four different processors:

(*a*) A processor for an instruction set that has only six general-purpose (integer) registers.

On a function call, the first argument to the function must be placed in register `r0` and the processor automatically writes the PC to return to after the call into register `r5`.

(*b*) A microcontroller with a static branch predictor and a very small instruction memory.

When the processor encounters a branch it always predicts that it will not be taken and continues speculatively fetching instructions immediately after the branch. All binaries for this processor must fit within 8KiB of memory.

(*c*) An in-order very long instruction word (VLIW) processor with some slow integer operations and a branch-delay slot.

This processor executes four independent operations concurrently in each very long instruction. Although most integer operations take only one cycle to execute, multiplication takes three cycles and division takes eight. The branch-delay slot occurs immediately after a very long instruction containing a branch, and holds an instruction that will be executed regardless of whether the branch is taken or not.

(*d*) An out-of-order multicore with a loop cache.

Each core executes instructions out-of-order by selecting those that have their operands ready from a window of 128 instructions. The loop cache improves power consumption and performance of loops with 16 or fewer instructions. This works by keeping versions of those instructions internally, avoiding the need to fetch and decode them from the instruction cache. The processor contains four homogeneous cores to exploit parallelism within or across programs.

For each processor, describe any challenges and opportunities in register allocation, instruction scheduling and optimisation for the compiler. [5 marks each]

## 10 Principles of Communications

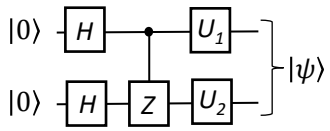
(*a*) The task is to design a Transport Protocol for the back plane – there is no packet loss and the delay between one core and another is extremely low. What are the consequences for flow and congestion control design choices? [7 marks]

(*b*) The task is now to design an Interplanetary Transport Protocol – a flow completes before a single ack gets back. There is radio noise, sometimes long lasting (e.g. solar flares) What are the design consequences for reliability, flow and congestion control? [7 marks]

(*c*) Your last task is to design a many-to-one Protocol – such a protocol might be used, for example, to deliver telemmetry data from many autonomous electric vehicles to the car manufacturer, so that they can monitor and adjust the parameters affecting motor and battery performance. What are the design consequences for reliability, flow and congestion control? [6 marks]

## 11 Quantum Computing

($a$) Consider the circuit:




($i$) What should $U_1$ and $U_2$ be to prepare the entangled state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$? [2 marks]

($ii$) Is state $|\psi\rangle$ always entangled, or does its entanglement depend on $U_1$ and $U_2$? Give reasons for your answer. [1 mark]

($iii$) If $U_1$ is a Pauli-$X$ gate and $U_2$ is a Pauli-$Z$ gate, and the first qubit is sent to 'Alice', and the second to 'Bob', then how can Alice use the shared entangled state to send two (classical) bits of information to Bob, by transmitting only a single qubit? [4 marks]

($iv$) If Alice knows only that $U_1$ is either a Pauli-$X$ or Pauli-$Z$ gate, but Bob knows which of these it is (and $U_2$ is still a Pauli-$Z$ gate which is known to both Alice and Bob), can Alice still send two bits by transmitting only a single qubit? Explain your answer. [5 marks]

($b$) Consider the three-qubit state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.

($i$) If Alice holds one of the three qubits, and Bob holds the other two, can Alice use this state to teleport one qubit to Bob? Explain your answer. [3 marks]

($ii$) If instead Alice and Bob each hold one qubit, and now some third party 'Charlie' holds the third qubit, then give a protocol to enable teleportation between Alice and Bob whereby Charlie first applies a Hadamard gate to his qubit, then measures in the computational basis and sends the measurement outcome to Alice and Bob. [4 marks]

($iii$) If Alice, Bob and Charlie each hold one of the three qubits, and there is no communication between Charlie and the other two parties, but Alice and Bob can communicate freely, is there a protocol for Alice and Bob to determine whether or not Charlie has measured his qubit? [1 mark]

## 12 Randomised Algorithms

(*a*) State the main advantages and disadvantages of Chernoff bounds in comparison to Markov's inequality and Chebyshev's inequality. [3 marks]

(*b*) Let $X_1, X_2, \ldots, X_n$ be independent random variables with $\mathbf{P}[X_i = -1] = \mathbf{P}[X_i = 1] = 1/2$ for $i = 1, 2, \ldots, n$. Let $X := \sum_{i=1}^{n} X_i$.

(*i*) What is $\mathbf{E}[X]$? [1 mark]

(*ii*) Derive the following concentration inequality: For any $t > 0$,

$$\mathbf{P}[X > t] \leq e^{-t^2/(2n)}.$$

*Hint:* You may use the inequality $\mathbf{E}\left[e^{\lambda X_i}\right] \leq e^{\lambda^2/2}$, which holds for any $1 \leq i \leq n$ and $\lambda > 0$. [5 marks]

(*c*) Consider now a random assignment of $m$ jobs to $n$ processors, such that each job is assigned to a processor chosen independently and uniformly at random from $\{1, 2, \ldots, n\}$. Independently of this assignment, each job takes 1 time unit to complete with probability $1/2$, and 3 time units otherwise.

(*i*) What is the expected number of time units assigned to one processor? [2 marks]

(*ii*) Derive an upper bound on the total sum over the time units of the $m$ jobs that holds with probability $1 - 1/m$.
*Hint:* You may use the result from (*b*)(*ii*). [4 marks]

(*iii*) Derive a concentration inequality for the number of processors which are assigned a number of time units that is at least as large as the expected number of time units per processor. For full marks, you should define the involved random variables and quantities precisely, and consider both the lower and upper tail. [5 marks]

## 13 Types

(*a*) Using the simply-typed lambda calculus with the letcont primitive, give well-typed terms corresponding to proofs of the following classical tautologies:

(*i*) $dne : \neg\neg A \to A$ [3 marks]

(*ii*) $contra : (A \to B) \to (\neg B \to \neg A)$ [4 marks]

(*iii*) $demorgan_1 : \neg(A \vee B) \to \neg A \wedge \neg B$ [5 marks]

(*iv*) $demorgan_2 : (\neg A \wedge \neg B) \to \neg(A \vee B)$ [5 marks]

(*b*) (*i*) Briefly explain what the following Agda type says.

$$\forall\{\mathtt{A} : \mathtt{Set}\}\{\mathtt{n} : \mathtt{Nat}\} \to \mathtt{Vec}\,\mathtt{A}\,\mathtt{n} \to (\mathtt{i} : \mathtt{Nat}) \to (\mathtt{i} < \mathtt{n}) \to \mathtt{A}$$

[1 mark]

(*ii*) Given the two following Agda declarations:

$$\begin{aligned}&\mathtt{zip} : \forall\{\mathtt{A} : \mathtt{Set}\}\{\mathtt{B} : \mathtt{Set}\}\{\mathtt{n} : \mathtt{Nat}\} \to \\ &\qquad (\mathtt{Vec}\,\mathtt{A}\,\mathtt{n} \times \mathtt{Vec}\,\mathtt{B}\,\mathtt{n}) \to \mathtt{Vec}\,(\mathtt{A} \times \mathtt{B})\,\mathtt{n}\end{aligned}$$

$$\begin{aligned}&\mathtt{unzip} : \forall\{\mathtt{A} : \mathtt{Set}\}\{\mathtt{B} : \mathtt{Set}\}\{\mathtt{n} : \mathtt{Nat}\} \to \\ &\qquad \mathtt{Vec}\,(\mathtt{A} \times \mathtt{B})\,\mathtt{n} \to \mathtt{Vec}\,\mathtt{A}\,\mathtt{n} \times \mathtt{Vec}\,\mathtt{B}\,\mathtt{n}\end{aligned}$$

Write a type expressing that `unzip` followed by `zip` is the identity.

[2 marks]

**END OF PAPER**