COMPUTER SCIENCE TRIPOS Part II

Wednesday 5 June 2019 1.30 to 4.30

COMPUTER SCIENCE Paper 9

*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 Algorithms

(a) Consider the definition of an approximation algorithm.

(i) Explain the meaning of approximation ratio in the case of a maximisation problem. [2 marks]

(ii) How is this definition adjusted to the case of a randomised approximation algorithm? [2 marks]

(b) State the definition of PTAS and FPTAS. [4 marks]

(c) Let $G = (V, E)$ be an undirected graph. For any $k \geq 1$, define $G^{(k)}$ to be the undirected graph $(V^{(k)}, E^{(k)})$, where $V^{(k)}$ is the set of all ordered k-tuples of vertices from V and $E^{(k)}$ is defined so that $(v_1, v_2, \ldots, v_k)$ is adjacent to $(w_1, w_2, \ldots, w_k)$ if and only if $\{v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k\}$ forms a clique.

(i) Argue that the graph $G^{(k)}$ can be constructed in time polynomial in n (for any fixed value of k). [3 marks]

(ii) Prove that the size of the maximum clique in $G^{(k)}$ is equal to the k-th power of the size of the maximum clique in G. [5 marks]

(iii) Argue that if there is a polynomial-time approximation algorithm that has a constant approximation ratio for finding a maximum clique, then there is a polynomial-time approximation scheme (PTAS) for the problem.
Hint: Your PTAS should be based on applying the given approximation algorithm with constant approximation ratio to $G^{(k)}$ for a proper choice of $k > 0$. Then use the equivalence in part *(ii)* to analyse its approximation ratio. [4 marks]

2 Bioinformatics

(*a*) Describe, with one example, the complexity of using dynamic programming in multiple alignment. [3 marks]

(*b*) Describe the differences between genome assembly (i.e. using a reference genome) and genome de novo sequencing from a bioinformatics perspective. [5 marks]

(*c*) The de Bruijn Graph is widely used in Bioinformatics.

(*i*) Describe with one example how to construct the paired de Bruijn Graph. [5 marks]

(*ii*) Describe the advantages of the paired de Bruijn Graph versus the non paired version of the de Bruijn Graph. [3 marks]

(*d*) Discuss the advantages of using soft k-means versus hard clustering. [4 marks]

3 Business Studies

After graduation you created a venture with £10m in sales with a 20% profit margin. After investing 10 years of your life on this endeavour you have decided that it is time to sell.

(*a*) Describe three exit routes available to you assuming you are not going to liquidate the company. [3 marks]

(*b*) Discuss how you would go about identifying and choosing the exit route to pursue. [6 marks]

(*c*) Identify your preferred exit route and discuss how you would go about managing the process. [11 marks]

4 Comparative Architectures

(*a*) As modern processors and system-on-chip (SoC) designs become more complex, so does the on-chip interconnect.

(*i*) Why does the design of an on-chip network differ greatly from that of larger scale networks? [3 marks]

(*ii*) Draw a diagram showing the datapath of an on-chip router with virtual channels. [3 marks]

(*iii*) How do virtual channels help to reduce packet latency? [3 marks]

(*iv*) For what reason, other than performance, may virtual-channel flow control be useful?

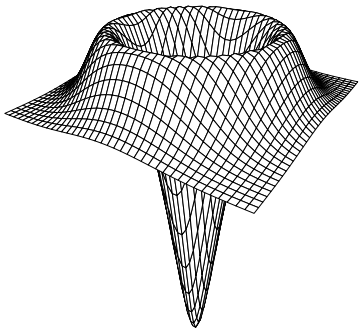
[2 marks]

(*b*) You have been asked to outline the design of a high-performance 16-core processor suitable for use in a server-class machine. Draw a clear block diagram illustrating your architecture. Include all the major building blocks, e.g. processor cores, caches, on-chip interconnects, memory controllers and the main off-chip interfaces. Provide a brief description of how cache coherence is maintained, what type(s) of on-chip interconnect are provided, a brief overview of the features of your individual cores and the characteristics of your caches. For each major component briefly justify your design decisions.

[9 marks]

5 Computer Vision

(*a*) In early stages of machine vision systems, the isotropic operator shown on the right is often applied to an image $I(x, y)$ in the following way: $[\nabla^2 G_\sigma(x, y)] * I(x, y)$.



What is the purpose of this operation? Which class of neurones in the retina does it mimic?

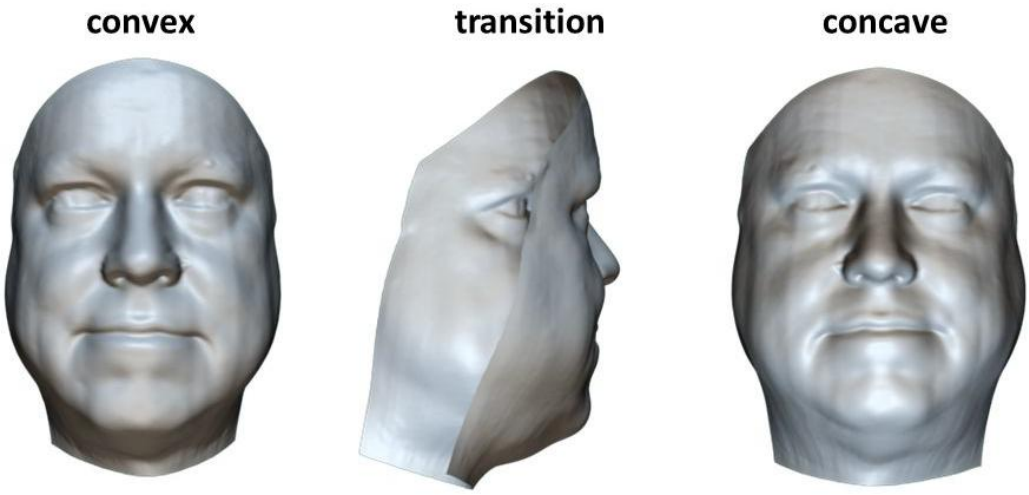
How would the results differ if instead this operation: $G_\sigma(x, y) * \nabla^2 I(x, y)$ were performed; or alternatively if this operation: $\nabla^2 [G_\sigma(x, y) * I(x, y)]$ were performed?

[6 marks]

(*b*) Computer vision colour space is usually three-dimensional, just because human vision is tri-chromatic and therefore cameras are designed with three colour planes. But suppose we added a fourth colour plane, say yellow (Y), to the standard red, green, and blue (RGB) bands. Considering that these are linearly independent but not orthogonal vectors, what would be the added capability of RGBY space? What tests would reveal it? Present a version of the Retinex algorithm for RGBY space, explaining the purpose of each step in the algorithm.

[8 marks]

(*c*) Visual inference of surface shape depends on assumptions and prior knowledge, such as "faces are mostly convex". Explain the "rotating hollow mask illusion". Why does a face mask (as pictured below) appear to reverse its direction of rotation, once it is seen from the inside instead of the outside? What is the role of Bayesian inference when interpreting a face-like surface that is actually concave in presentation instead of convex? Should visual illusions like this be considered "features" or "bugs", and should one try to design them in to a computer vision system? [6 marks]



6 Cryptography

(*a*) (*i*) Choose and briefly describe one major application of elliptic-curve group operations in cryptography. [4 marks]

(*ii*) What other group operation was previously (and still is) widely used for the same purpose? [2 marks]

(*iii*) What is a major advantage of elliptic curve group operations over the group operation you named in Part (*a*)(*ii*)? [4 marks]

(*b*) In the Galois field GF(2^8) modulo $x^8 + x^4 + x^3 + x^2 + 1$, calculate

(*i*) the sum 0011 1001 plus 0110 1100; [2 marks]

(*ii*) the product 0100 1011 times 0000 1001. [4 marks]

(*c*) In Lamport's one-time password scheme, the user is given a list of passwords $R_n, \ldots, R_0$ generated using the following algorithm:

$R_0 \leftarrow$ random
for $i := 1$ **to** n
    $R_i := h(R_{i-1})$

(*i*) State two properties required of function h. [2 marks]

(*ii*) Complete the password verification algorithm implemented in the server by filling in the ellipses (...) below:

$Q := \ldots$
while true
    $P :=$ read password
    if ...
        ...
        grant access
    else
        deny access

[2 marks]

7 Denotational Semantics

(a) Suppose that $(D, \sqsubseteq)$ is a poset which is chain-complete but does not have a least element, and that $f : D \to D$ is a continuous function.

(i) Give an example of such $(D, \sqsubseteq)$ and f for which f has no fixed point. [1 mark]

(ii) If $d \in D$ satisfies $d \sqsubseteq f(d)$, prove that there is a least element $e \in D$ satisfying $d \sqsubseteq e = f(e)$. [Hint: consider the method used to prove Tarski's fixed point theorem.] [7 marks]

(b) (i) Define the notion of *contextual equivalence* for the language PCF. (You need not describe the syntax and semantics of PCF.) [2 marks]

(ii) State the *compositionality*, *soundness* and *adequacy* properties of the denotational semantics of PCF. Explain why they imply that any two closed PCF terms of the same type with equal denotations are contextually equivalent. [8 marks]

(iii) Give, without proof, an example of two contextually equivalent PCF terms that have unequal denotation. [2 marks]

8 Hoare Logic and Model Checking

This question is about modelling a program, defined below, consisting of two threads and a single (mathematical) integer variable `X`, initially set to 0. Each thread t has its own program counter given by pc_t, initially set to 0, which describes the *current line* for that thread.

```
Thread 1          Thread 2
0:  X := X+1   ||  0:  IF IS_ODD(X) THEN STOP_ALL
1:  GOTO 0     ||  1:  GOTO 0
```

The program is executed by repeatedly carrying out execution steps, where one thread is non-deterministically selected, its entire current line is run, and its program counter is then updated appropriately. This continues until `STOP_ALL` is executed, which immediately terminates the whole program.

(*a*) The program state can be described by $(pc_1, pc_2, \texttt{X}, \textit{stopped})$, where pc_1, pc_2, and `X` are mathematical integers, and *stopped* is a boolean which is true iff `STOP_ALL` has been executed. Let S be the set of all such states.

(*i*) Define S_0, the set of initial states of the program, such that $S_0 \subseteq S$. [1 mark]

(*ii*) Define a transition relation $R \subseteq S \times S$ describing the program's execution. [2 marks]

(*iii*) Define a labelling function L that labels all states where the program has terminated with the atomic property `term`. [2 marks]

(*b*) Explain why, taking the definitions from (*a*), the model $M_{\mathbf{a}} = (S, S_0, R, L)$ is *not* a (finite) Kripke structure. [2 marks]

(*c*) Draw the finite state automaton for a model $M_{\mathbf{b}}$ which *is* a Kripke structure, such that $M_{\mathbf{a}}$ and $M_{\mathbf{b}}$ are bisimilar. Justify your answer briefly.
[*Note:* A full formal proof of bisimilarity is not required.] [5 marks]

(*d*) (*i*) Give an LTL formula ϕ such that the judgement $M_{\mathbf{b}} \vDash \phi$ corresponds to the statement "every execution of the program will eventually terminate". [2 marks]

(*ii*) Either prove that $M_{\mathbf{b}} \vDash \phi$ holds, or describe a counter-example trace. [2 marks]

(*e*) Consider the CTL formula $\psi = \mathbf{AG}(\mathbf{EF}\ \texttt{term})$. Determine whether this is equivalent to your definition of ϕ from Part (*d*). ϕ and ψ are equivalent iff, for all Kripke structures M, $(M \vDash \phi)$ iff $(M \vDash \psi)$. Justify your answer. [4 marks]

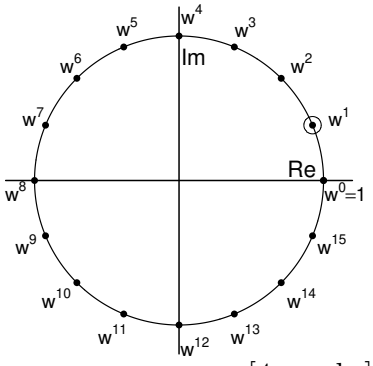
9 Information Theory

(*a*) If I pick a number n that can be any integer from 1 to ∞ whose probability distribution of being selected is $(\frac{1}{2})^n$, and you ask a series of 'yes/no' questions which I will answer truthfully, how many such 'yes/no' questions should you expect to ask before discovering which number I have picked? Justify your answer by invoking a known series limit. What sequence of questions would be the most efficient to ask, and why? [4 marks]

(*b*) An inner product space containing complex functions $f(x)$ and $g(x)$ is spanned by a set of orthonormal basis functions $\{e_i\}$. Complex coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ therefore exist such that $f(x) = \sum_i \alpha_i e_i(x)$ and $g(x) = \sum_i \beta_i e_i(x)$.

Show that the inner product $\langle f, g \rangle = \sum_i \alpha_i \overline{\beta_i}$. [4 marks]

(*c*) Consider a data sequence $f[n]$ $(n = 0, 1, \ldots, 15)$ having Fourier coefficients $F[k]$ $(k = 0, 1, \ldots, 15)$. Using the 16^{th} roots of unity labelled around the unit circle as powers of w^1, the primitive 16^{th} root of unity, construct a sequence of the w^i that could be used to compute $F[3]$ when an inner product is computed between your sequence of w^i and the data sequence $f[n]$.



[4 marks]

(*d*) Explain how vector quantisation exploits sparseness to construct very efficient codes. Use the example of encoding a natural language lexicon with a 15 bit coding budget. Contrast the strategy of using codewords for single letters versus using codewords as pointers to a sparse index of combinations of letters. [4 marks]

(*e*) A continuous signal $f(t)$ has Fourier transform $F(\omega)$. Explain why computing derivatives of $f(t)$ such as $f'(t)$ or $f''(t)$ amounts simply to high-pass filtering. For the n^{th} derivative $f^{(n)}(t)$, what exactly is this filtering operation when expressed in terms of $F(\omega)$? Show how this operation could be used to define derivatives of non-integer order (for example the 1.5^{th} derivative). [4 marks]

10 Mobile and Sensor Systems

A mobile phone software developer is programming her own behaviour tracking app and she wants to make it suitable for use in areas with intermittent and low bandwidth cellular connectivity.

(*a*) Explain the precautions the developer can take to make the app work well in this environment. Describe examples of mechanisms which can help with this. [5 marks]

(*b*) The app uses various sensors to monitor user activity and its surrounding environment. In particular, it uses the microphone to monitor location ambience (surrounding sounds). Describe a suitable machine learning approach and discuss system considerations. [6 marks]

(*c*) Describe how the developer can use location data to provide mobility prediction and how mutual information can improve the mobility prediction algorithm. [4 marks]

(*d*) A very able user wants to test the app for privacy leaks before using it on their own phone. What techniques can be used? Illustrate each technique's specific purpose and the assumptions it is predicated on. [5 marks]

11 Optimising Compilers

(*a*) Describe the phase-order problem in a compiler and illustrate your answer with some example code. [4 marks]

(*b*) You are advising a semiconductor design company on building a compiler for their latest processor. The processor has the following features:

- Sixteen 64-bit registers (`r0-r15`) and sixteen 32-bit registers (`s0-s15`), the latter corresponding to the lower 32 bits of each of the 64-bit registers.
- A one-cycle branch delay slot after each control-transfer instruction (i.e. the instruction after a branch is executed before the branch takes effect).
- Complex arithmetic instructions that implicitly use `r15` as their first source operand.

What are the challenges of code generation for this processor, given these features and how can they be addressed within the compiler? [8 marks]

(*c*) To ease compilation, the chief designer suggests that the processor's instructions could be executed directly in SSA form (i.e. all destination registers unique). This would use a small cache to provide fast access to the most recently used virtual registers. Discuss the advantages and disadvantages of such an approach from the compiler writer's viewpoint. [8 marks]

12 Principles of Communications

(*a*) Describe max-min fair share allocation of resources, explaining what it can be used for. [10 marks]

(*b*) The Transmission Control Protocol (TCP) can be seen as a distributed optimisation of the dynamic resource allocation problem.

Describe how capacity at bottlenecks (congested points in the network) is effectively allocated in proportion to demands, and how this is equivalent to the joint optimisation of user and network utilities. [10 marks]

13 Quantum Computing

A Boolean formula ϕ with n variables in it can be seen as defining a function $f : \{0,1\}^n \to \{0,1\}$, and we say that ϕ is satisfiable if there is some $x \in \{0,1\}^n$ such that $f(x) = 1$.

(*a*) Explain how f can be suitably represented as a unitary operation U_f on a complex space of dimension 2^{n+1}. [3 marks]

(*b*) Suppose that we are given a blackbox implementing U_f. Describe how this would be used to form the *Grover iterate* which can be repeated to find a value x such that $f(x) = 1$. [5 marks]

(*c*) If there is exactly one value x such that $f(x) = 1$, how many iterations of the Grover iterate would you use to find this value? What is the probability of finding it? [3 marks]

(*d*) If there are M distinct values such that $f(x) = 1$, how many iterations of the Grover iterate would you use to find one of these values? What is the probability of finding one of them? [3 marks]

(*e*) If you are able to turn an arbitrary formula ϕ into an implementation of the corresponding unitary operator U_f, how would you use this to give an algorithm for determining whether ϕ is satisfiable or not? Give an estimate of the running time of your algorithm in terms of n. [6 marks]

14 Types

(a) In System F, give a Church encoding for (i) the Boolean type, (ii) the definition of the True and False constants, and (iii) the type and definition of the if-then-else operation. [3 marks]

(b) In System F, give (i) a Church encoding Nat for the natural numbers, (ii) a Church encoding for the $\mathsf{Zero} : \mathsf{Nat}$ and $\mathsf{Succ} : \mathsf{Nat} \to \mathsf{Nat}$ constructors, and (iii) a type and definition for the iteration operator Iter for natural numbers. [3 marks]

(c) (i) In System F, give a Church encoding for (i) an Option_A type, (ii) the definitions of the $\mathsf{None} : \mathsf{Option}_A$ and $\mathsf{Some} : A \to \mathsf{Option}_A$ operations, and (iii) the type and definition of the case operation on options.

(ii) Assume that $n : B$ and $s : A \to B$, and then

(A) Prove that $\mathsf{Case}\,[B]\,n\,s\,\mathsf{None} = n$

(B) Prove that $\mathsf{Case}\,[B]\,n\,s\,(\mathsf{Some}\,x) = s\,x$

[5 marks]

(d) In System F, define a predecessor operation $\mathsf{Pred} : \mathsf{Nat} \to \mathsf{Nat}$, which returns Zero if given Zero as an argument, and return n if given $\mathsf{Succ}\,n$ as an argument. [*Hint*: The option type may be useful in formulating this definition.] [8 marks]

(e) In System F, define a subtraction operator $\mathsf{Sub} : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$, which is defined to be *saturating*. That is, $\mathsf{Sub}\,m\,n$ returns the difference if $m \geq n$, and returns 0 otherwise. [1 mark]

END OF PAPER