UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS Part II

Thursday 2 June 2016 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*) Explain the difference between PTAS and FPTAS, and give one example of a problem for which a FPTAS is known, and one example of a problem for which a PTAS is known but no FPTAS. [4 marks]

(*b*) We consider an extension of the MAX-3-CNF problem, called MAX-4-CNF problem, where we are given a 4-CNF formula with m clauses, e.g., $(x_1 \vee \overline{x_3} \vee x_4 \vee x_5) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3 \vee \overline{x_5}) \wedge \cdots$, and the goal is to find an assignment of the variables $x_1, x_2, \ldots, x_n$ that satisfies as many clauses as possible.

(*i*) Design a randomised approximation algorithm and analyse its approximation ratio. (For full marks, the approximation ratio must be smaller than 10/9.) [4 marks]

(*ii*) Express the MAX-4-CNF problem as an integer program. [4 marks]

(*iii*) Based on the construction from Part (*b*)(*ii*) or otherwise, describe an algorithm that performs randomised rounding on the solution of a linear relaxation. [3 marks]

(*iv*) Analyse the expected approximation ratio of the algorithm from Part (*b*)(*iii*).
Hint: You may want to use the following two inequalities. Firstly, for any non-negative numbers $a_1, a_2, \ldots, a_k$, we have

$$\left(\prod_{i=1}^{k} a_i\right)^{1/k} \leq \frac{\sum_{i=1}^{k} a_i}{k}.$$

Secondly, for any integer $k \geq 2$ and $0 \leq a \leq 1$,

$$1 - \left(1 - \frac{a}{k}\right)^k \geq \left(1 - (1 - \frac{1}{k})^k\right) \cdot a.$$

[5 marks]

2 Bioinformatics

(*a*) Explain the steps and the complexity of the Hirschberg algorithm and illustrate them with an example. [7 marks]

(*b*) Give one example why the multiple alignment, as implemented in the software Clustal, needs a guide tree. [5 marks]

(*c*) Explain what an amino acid exchange propensity matrix is and how you would construct it. [3 marks]

(*d*) Explain with an example why a compression algorithm is often needed in genome assembly. [5 marks]

3 Computer Systems Modelling

Consider a M/M/1 queueing system with an arrival rate $\lambda > 0$ and a service rate $\mu > 0$ where $\rho = \lambda/\mu < 1$.

(*a*) Derive the distribution for N, the total number of customers present in the queueing system in equilibrium. [6 marks]

(*b*) Suppose that the queueing system is in equilibrium. For each of the following terms define the quantity and determine its value:

(*i*) utilization

(*ii*) throughput

(*iii*) mean number of customers present in the system

(*iv*) mean time spent by a customer in the system

[8 marks]

(*c*) Now suppose that the arrival rate and service rate are both scaled by a factor of $s > 0$. For each of the four quantities in part (*b*) determine their new values and explain your findings. [6 marks]

4 Computer Vision

(*a*) Present five experimental observations about human vision that support the thesis that what we see is explicable only partly by the optical image itself, but is more strongly determined by top-down knowledge, model-building and inference processes. [5 marks]

(*b*) Discuss the use of texture gradients as a depth cue in computer vision. How can texture gradients be measured? What role can Fourier analysis play in this? What ancillary "metaphysical" assumptions must be invoked by a vision algorithm in order to make the inference task well-posed and thereby make such computations possible? You may find it helpful to refer to the following figures: [5 marks]

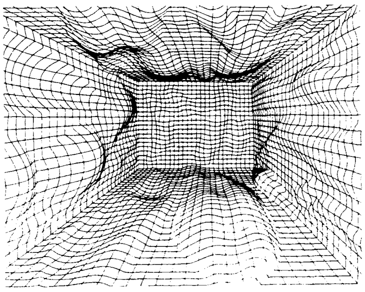


figure 1

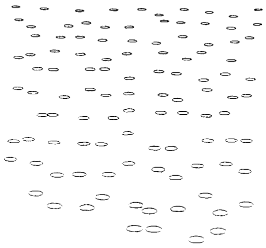


figure 2

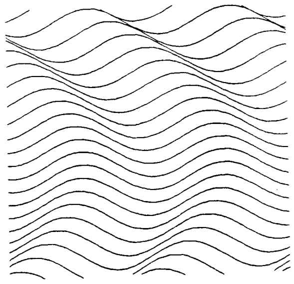


figure 3

(*c*) Briefly define each of the following concepts as it relates to vision:

(*i*) "signal-to-symbol converter" [2 marks]

(*ii*) Hadamard's criteria for well-posed problems [2 marks]

(*iii*) correspondence problem [2 marks]

(*iv*) reflectance map [2 marks]

(*v*) Bayesian prior and its role in visual inference [2 marks]

5 Denotational Semantics

For all PCF types τ and all $M \in \mathrm{PCF}_{\tau \to bool}$, let $M^{\#} \subseteq [\![\tau]\!]$ be defined as

$$M^{\#} = \{ \, d \in [\![\tau]\!] \mid [\![M]\!](d) = true \, \}$$

Indicate whether the following statements are true or false, respectively providing a proof or a counterexample. You may use any standard results provided that you state them clearly.

(*a*) For all PCF types τ and all $M, N \in \mathrm{PCF}_{\tau \to bool}$, if $M^{\#} \subseteq N^{\#}$ then $\vdash M \leq_{\mathrm{ctx}} N : \tau$. [5 marks]

(*b*) For all PCF types τ and all $M, N \in \mathrm{PCF}_{\tau \to bool}$, if $\vdash M \leq_{\mathrm{ctx}} N : \tau$ then $M^{\#} \subseteq N^{\#}$. [5 marks]

(*c*) For all PCF types τ and all $M, N \in \mathrm{PCF}_{\tau \to bool}$, there exists $P \in \mathrm{PCF}_{(\tau \to bool) \to ((\tau \to bool) \to (\tau \to bool))}$ such that $(P\, M\, N)^{\#} = M^{\#} \cap N^{\#}$. [5 marks]

(*d*) For all PCF types τ and all $M, N \in \mathrm{PCF}_{\tau \to bool}$, there exists $P \in \mathrm{PCF}_{(\tau \to bool) \to ((\tau \to bool) \to (\tau \to bool))}$ such that $(P\, M\, N)^{\#} = M^{\#} \cup N^{\#}$.

Hint: Consider the fact, for which you need not provide a proof, that there is no PCF-definable function $f \in (\mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp))$ such that $f\, true\, \perp = f\, \perp\, true = true$ and $f\, false\, false \neq true$. [5 marks]

6 Digital Signal Processing

(*a*) Let δ be the Dirac delta function and $T, b > 0$ be time intervals. Give the Fourier transform

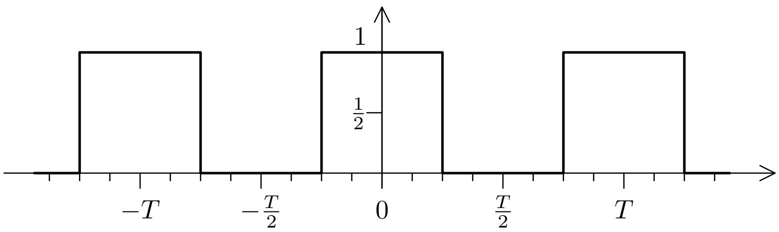
$$X(f) = \int_{-\infty}^{\infty} x(t)\, \mathrm{e}^{-2\pi \mathrm{j} f t}\, \mathrm{d}t$$

of the following two functions:

(*i*) $x(t) = c_T(t)$, where $c_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ [3 marks]

(*ii*) $x(t) = r_b(t)$, where $r_b(t) = \begin{cases} 1 & \text{if } |t| < b \\ \frac{1}{2} & \text{if } |t| = b \\ 0 & \text{otherwise} \end{cases}$ [5 marks]

(*b*) Consider this periodic, binary, square-wave clock signal $p(t)$, with period T, duty cycle 0.5 and maximum amplitude 1:

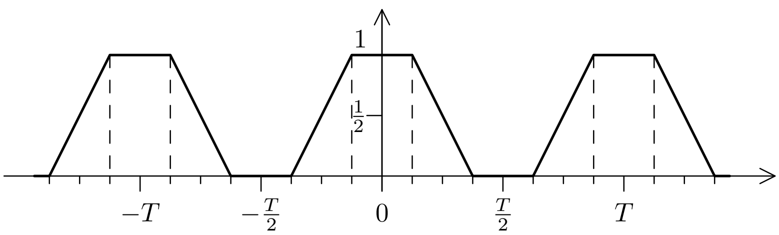


Show that its Fourier transform is

$$P(f) = \frac{1}{2}\delta(f) + \frac{1}{2\pi} \cdot \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{2k+1}{T}\right) \cdot \frac{(-1)^k}{k + \frac{1}{2}}.$$

Hint: Use the answers from part (*a*). [8 marks]

(*c*) Real-world digital signals need some time to transition between low and high. What is the Fourier transform of the periodic, trapezoid-wave clock signal $q(t)$, shown below, with period T and transition time $T/4$?



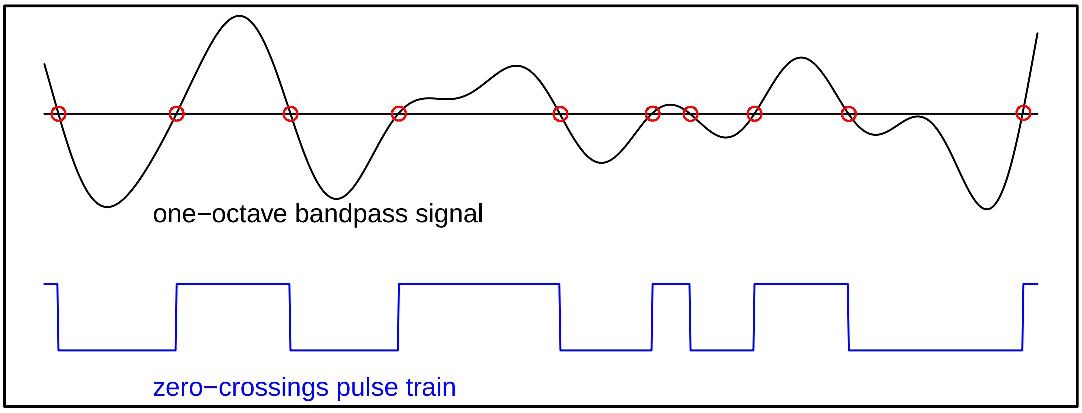
[4 marks]

7 Information Theory

(*a*) A continuous communication channel adds Gaussian white noise to signals transmitted through it. The ratio of signal power to noise power is 30 decibels, and the frequency bandwidth of this channel is 10 MHz. Roughly what is the information capacity C of this channel, in bits/second? [5 marks]

(*b*) Explain the comb function, $\text{comb}(t) = \delta_X(t)$, its role in the sampling theorem, its self-Fourier property, and the constraint on the spacing of the comb's tines that is required in both the signal domain and consequently in the Fourier domain in order to reconstruct exactly, from discrete samples, a signal having no frequency components higher than W. [10 marks]

(*c*) Explain what Logan's Theorem asserts about the richness of the zero-crossings in signals strictly bandlimited to one octave (as illustrated in the figure below). Consider an amplitude-modulated signal such as $f(t) = [1 + a(t)]\, c(t)$, where $c(t)$ is a pure sinusoidal carrier wave and its modulating function is $[1 + a(t)] > 0$. What would Logan's Theorem say about the information contained in its zero-crossings? Name one intended application, and at least one algorithmic difficulty, of Logan's theorem.



[5 marks]

8 Natural Language Processing

(*a*) Give three examples of NLP tasks, apart from sentiment classification, that are typically treated as supervised classification problems. Describe the tasks briefly. [3 marks]

(*b*) Describe the Naïve Bayes classifier and how it would apply to sentiment classification. [5 marks]

(*c*) What is the "naïve" assumption behind Naïve Bayes and does this assumption hold for language-based classification tasks? [3 marks]

(*d*) What are the limitations of the bag-of-words features in sentiment classification? What other features would you use to address these limitations? Illustrate your suggestions with linguistic examples. [5 marks]

(*e*) Discuss the evaluation of sentiment classification systems. [4 marks]

9 Optimising Compilers

(*a*) Explain the scenario in which a strictness analyser is used to optimise a program. Your answer should consider the following: for what languages strictness optimisation is useful, where it is beneficial to be able to place strict or non-strict annotations on a program (seeing the strictness analyser as a black-box oracle), and how such annotations can safely allow an optimiser to represent strict or non-strict values differently at run time. Give an example program which has different run-time space complexity before and after strictness optimisation. [5 marks]

(*b*) One implementation for a strictness analyser determines *strictness functions* associated with each user-defined or built-in function. Given a user-defined function taking n integer arguments to an integer result, state the domain and range of its associated strictness function. How can such a strictness function be used to produce the strict or non-strict annotations in Part (*a*)? [3 marks]

(*c*) Give a data structure suitable for representing strictness functions within a strictness analyser. Can ordinary functions be used? Would your data structure represent strictness functions $\lambda(x, y, z).\ x \wedge (y \vee z)$ and $\lambda(x, y, z).\ (x \wedge y) \vee (x \wedge z)$ differently? Would these two strictness functions enable different strictness optimisations in Part (*a*)? [4 marks]

(*d*) Give the strictness functions for the following source-language functions.

(*i*) The built-in addition and 3-argument conditional functions. [2 marks]

(*ii*) A built-in *parallel-if* function, which evaluates all its three arguments in parallel, and returns a result as soon as enough of its arguments terminate. This includes returning value v when the second and third arguments evaluate to v even if the first argument is still computing. [2 marks]

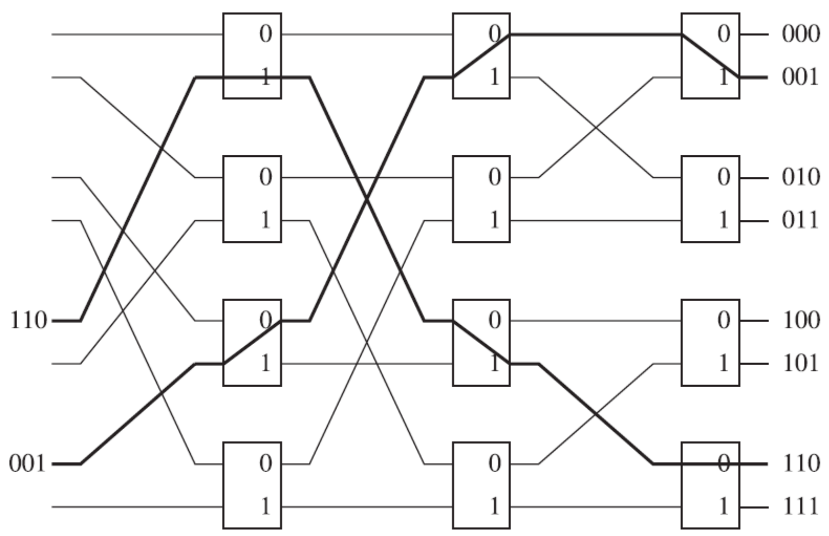
(*iii*) The user-defined function `f` defined by

```
f(x,y,z,t,u) =
  if x=0 then y else f(x-1, t+2, u+3, y*4, z*5);
```

[4 marks]

10 Principles of Communications

(*a*) A simple packet switch fabric, such as that illustrated below, can experience head of line blocking. Explain what this is, and how it can be mitigated by clocking the output lines faster, or by combining queuing and scheduling access to the inputs carefully.



[10 marks]

(*b*) Weighted fair queueing is a complex system devised to support performance guarantees, e.g. for latency for some users and capacity for other users, in routers in the Internet. Explain how and why we can make use of a much simpler queuing system for data centre networks.

[10 marks]

11 System-on-Chip Design

(*a*) Describe two key features of each of these three forms of RTL: Structural, Behavioural, Synthesisable. [6 marks]

(*b*) What is the purpose of an RTL '`generate`' statement and what is its equivalent in Bluespec or Chisel ? [4 marks]

(*c*) Define high-level logic synthesis and high-level modelling of hardware, saying what purpose they serve. [6 marks]

(*d*) Which forms of RTL or high-level model are best for estimating the performance and energy use of a hardware design and why? [4 marks]

12 Hoare Logic and Model Checking

(*a*) Suppose we have a representation of a computer system, either as a set of axioms Γ specifying its behaviour or as a model $\mathcal{M}$, along with a property ϕ which we expect to hold (but which may not hold due to programming errors). Give two reasons why we might prefer to model-check $\mathcal{M} \models \phi$ rather than use logical inference to prove $\Gamma \vdash \phi$. [2 marks]

(*b*) Assuming a given set AP of atomic properties, ranged over by p, give the syntax of LTL formulae ϕ. (It is not necessary to be encyclopaedic—full marks can be obtained by including four constructs not present in classical logic.) Explain how an LTL formula is interpreted as true or false in a model. It suffices to consider two temporal operators along with conjunction and an atomic property p. [7 marks]

(*c*) Suppose p is an atomic property. Give informal explanations of the two properties $\mathbf{G}(\mathbf{F}\ p)$ and $\mathbf{F}(\mathbf{G}\ p)$. State, giving reasons, whether the properties are equivalent or whether one implies the other. [3 marks]

(*d*) Consider a program consisting of the following two threads where `WORK` is an unspecified unit of work not involving variables `A` or `B`. The threads are executed on a scheduler which first sets `A` and `B` to zero and then repeatedly and non-deterministically chooses to execute a *whole line* of code from either the left or right thread. An `AWAIT` e statement can only be scheduled if its condition e evaluates to true.

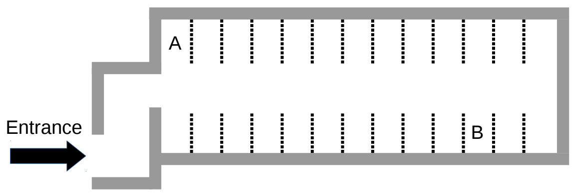
```
L:  AWAIT A=0;  A:=1;        | M:  AWAIT B=0;  B:=1;
    AWAIT B=0;  B:=1;        |     AWAIT A=0;  A:=1;
    WORK;                    |     WORK;
    A:=0; B:=0; GOTO L;      |     B:=0; A:=0; GOTO M;
```

Determine a Kripke structure model for this program, and draw it as a finite-state automaton. You should label one or more states of the automaton as satisfying the atomic property of `deadlock`. [5 marks]

(*e*) Give a temporal logic formula expressing that `deadlock` does not occur. For the program in Part (*d*), would a model checker prove this formula or produce a counterexample trace? [3 marks]

13 Topical Issues

This question concerns tracking a vehicle to a parking space in a covered car park, where satellite navigation systems are unavailable. The vehicle is assumed to have an electronic map of the car park and to have been fitted with odometry sensors that provide the accumulated number of rotations made by a wheel and the direction the wheel is pointing in at any given moment. A particle filtering approach is used to track the car from the car park entrance. The car park layout is shown below, with dotted lines used to show parking bays:



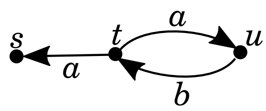
(*a*) Briefly describe the purpose of the prediction, correction and resampling stages of a particle filter. [6 marks]

(*b*) Describe how the odometry information could be incorporated into the prediction stage and the map information into the correction stage. Why is odometry not typically incorporated into the correction stage, despite being a measurement? [6 marks]

(*c*) If the wheel radius were unknown, one solution is to use an average wheel radius and associate a large noise with any derived displacements in the particle filter. Explain why this is not optimal and suggest a better approach. You should illustrate your answer by considering a vehicle that parks in space A and another that parks in space B as shown in the figure above. [8 marks]

14 Topics in Concurrency

(*a*) For each of the following modal-μ assertions, write down the set of states in the following transition system which satisfy the given assertion.



(*i*) $[a]\langle b\rangle T$

(*ii*) $\langle a\rangle[b]F$

(*iii*) $\nu X.\langle-\rangle X$

(*iv*) $\mu X.\langle-\rangle X$

(*v*) $\nu X.\langle-\rangle\langle-\rangle X$ [5 marks]

(*b*) Let the set of states of an arbitrary transition system be S. The operation $\varphi : \mathcal{P}(S) \to \mathcal{P}(S)$ is defined, for a set of states $X \subseteq S$, as follows:

$$\varphi(X) = [-]X = \{y \ : \ \forall x \forall a.\ \text{if } y \xrightarrow{a} x \text{ then } x \in X\}.$$

Prove that $x \in \varphi^n(\emptyset)$ if, and only if, all sequences (including the empty sequence) of transitions starting from state x are of length less than n. [6 marks]

(*c*) For the operation defined in part (*b*) there are transition systems for which

$$\bigcup_{n\in\omega} \varphi^n(\emptyset) \quad \neq \quad \mu X.[-]X$$

(*i*) What can you immediately infer about the operation φ? [2 marks]

(*ii*) Explain whether such transition systems can be finite. [2 marks]

(*iii*) Give an example of a transition system for which

$$\bigcup_{n\in\omega} \varphi^n(\emptyset) \quad \neq \quad \mu X.[-]X$$

[3 marks]

(*iv*) State when, in general, a state satisfies $\mu X.[-]X$. [2 marks]

15 Types

(a) Explain briefly what is meant by the *Curry-Howard correspondence*. [3 marks]

(b) The Calculus of Constructions (λC) is the Pure Type System whose specification has two sort symbols `Prop` and `Set`, one axiom ($\mathtt{Prop}, \mathtt{Set}$) and four rules ($\mathtt{Prop}, \mathtt{Prop}, \mathtt{Prop}$), ($\mathtt{Set}, \mathtt{Prop}, \mathtt{Prop}$), ($\mathtt{Prop}, \mathtt{Set}, \mathtt{Set}$) and ($\mathtt{Set}, \mathtt{Set}, \mathtt{Set}$).

(i) Give the λC typing rule for sort symbols. [1 mark]

(ii) Give the λC typing rules for pseudo-terms of the form $\Pi x : t\,(t')$ and $\lambda x : t\,(t')$. [4 marks]

(c) Suppose $\Gamma \vdash A : \mathtt{Set}$ holds in λC. Give a pseudo-term Eq_A for *Leibniz equality* on A, explaining any abbreviations that you use. What is the type of Eq_A? [5 marks]

(d) With Γ, A and Eq_A as in part (c), give pseudo-terms refl_A and symm_A that have the following typings (which you do not have to prove):

(i) $\Gamma \vdash \mathit{refl}_A : \Pi x : A\,(Eq_A\,x\,x)$ [3 marks]

(ii) $\Gamma \vdash \mathit{symm}_A : \Pi x : A\,(\Pi y : A\,(Eq_A\,x\,y \to Eq_A\,y\,x))$ [4 marks]

END OF PAPER