COMPUTER SCIENCE TRIPOS Part IB

Wednesday 5 June 2019 1.30 to 4.30

COMPUTER SCIENCE Paper 6

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.*

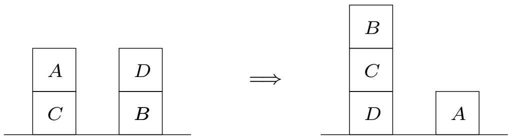
You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Artificial Intelligence

Evil Robot has been kidnapped by experimental psychologists, who are forcing him to solve problems involving the stacking of blocks. For example, given the start state on the left, he is asked to re-arrange the blocks into the state shown on the right.



You are to help him solve these problems by designing a system using *planning graphs*. A block can only be moved if it does not have another block on top of it. Only one block can be placed directly on top of another, although stacks of multiple blocks are allowed.

(*a*) Explain how this problem can be represented as a planning problem, such that it can be analyzed using a planning graph. Describe how state should be represented, and how actions should be represented, giving a specific example relevant to the stated problem in each case. [5 marks]

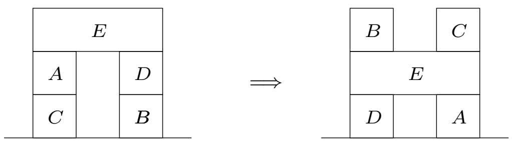
(*b*) Using the start state in the diagram above, draw the initial planning graph for the problem, including the initial state level, the first action level, and the state level resulting from the first action level. Do not add any mutex links at this stage. [4 marks]

(*c*) Define an *inconsistent effects mutex* and an *interfering actions mutex*. Add to your diagram for Part (*b*) a single example of each, or explain why this is not possible. [4 marks]

(*d*) Define a *competing for preconditions mutex*. By adding a small number of actions to the second action level of your planning graph, give a single example of such a mutex, or explain why this is not possible. [2 marks]

(*e*) How many more action levels would you expect to need before a valid plan could be extracted to solve the problem stated? Explain your answer. [2 marks]

(*f*) Give two examples of the difficulties that might arise if we also wish to include long blocks as follows:



In each case explain why it might be difficult to address such an extension using planning graphs. [3 marks]

2 Artificial Intelligence

(*a*) Describe the *state-variable* representation for planning by discussing the following, giving specific examples of each concept:

(*i*) Rigid relations and state variables. [2 marks]

(*ii*) Actions. [2 marks]

(*iii*) The representation of state. [2 marks]

(*iv*) Goals and solutions. [2 marks]

(*b*) Your boss has suggested using *heuristic search* to solve a planning problem expressed in the state-variable representation. Explain how this might be achieved. You do not need to suggest a specific heuristic at this stage. [3 marks]

(*c*) Comment on whether or not heuristic search is a good approach to solving planning problems in the state-variable representation, compared with the alternative of using a constraint satisfaction solver. [5 marks]

(*d*) Suggest an admissible heuristic that could be used when applying $A^\star$ search to solving a planning problem in the state-variable representation. Show that it is admissible. [4 marks]

3 Complexity Theory

A Boolean formula ϕ is said to be *satisfiable* if there is an assignment $\sigma : V \to \{\text{true}, \text{false}\}$ of values to the variables of ϕ that makes it true.

A *quantified Boolean formula* θ is an expression that is *(i)* either a Boolean formula; or *(ii)* $\exists X \phi$ where ϕ is a quantified Boolean formula and X is variable; or *(iii)* $\forall X \phi$ where ϕ is a quantified Boolean formula and X is variable.

We say that a quantified Boolean formula θ is satisfied by an assignment $\sigma : V \to \{\text{true}, \text{false}\}$ if either

- θ is a Boolean formula that is made true by σ; or
- θ is $\exists X \phi$ and either $\sigma[X/\text{true}]$ or $\sigma[X/\text{false}]$ make ϕ true; or
- θ is $\forall X \phi$ and both $\sigma[X/\text{true}]$ and $\sigma[X/\text{false}]$ make ϕ true.

Here, $\sigma[X/v]$ denotes the assignment that is the same as σ for all variables apart from X, and it maps X to the truth value v.

We write QBF for the decision problem of determining whether a given quantified Boolean formula is satisfiable. In answering the questions below, you may assume the NP-completeness of any standard problem, as long as you state your assumptions clearly.

(*a*) Show that QBF is NP-hard. [4 marks]

(*b*) Show that QBF is co-NP-hard. [6 marks]

(*c*) Show that QBF is in PSPACE. [6 marks]

(*d*) Is QBF NP-complete? Why or why not? [4 marks]

4 Complexity Theory

A Boolean formula ϕ is in *conjunctive normal form* (CNF) if it is the conjunction of clauses, each of which is the disjunction of literals. It is said to be in k-CNF (for $k \in \mathbb{N}$) if each clause has exactly k literals in it.

An assignment $\sigma : V \to \{\text{true}, \text{false}\}$ of truth values to the variables is a *satisfying assignment* for a CNF formula ϕ if it makes at least one literal in each clause of ϕ true. It is said to be a *not-all-equals* assignment for ϕ if it makes at least one literal in each clause of ϕ true *and* at least one literal in each clause of ϕ false.

Let CNF-SAT denote the problem of determining, given a formula in CNF, whether it has a satisfying assignment.

Let k-SAT denote the problem of determining, given a formula in k-CNF, whether it has a satisfying assignment.

Let k-NAE denote the problem of determining, given a formula in k-CNF, whether it has a not-all-equals assignment.

(*a*) Explain why CNF-SAT is NP-complete. Your explanation should include a full definition of NP-completeness and a brief sketch of the proof of the Cook-Levin theorem. [5 marks]

(*b*) Show that 3-SAT is NP-complete by means of a suitable reduction. [3 marks]

(*c*) Give a polynomial-time reduction from 3-SAT to 4-NAE. What can you conclude about the complexity of the latter problem?
(*Hint:* consider introducing one new variable and adding it to every clause.) [8 marks]

(*d*) Show that the problem 3-NAE is NP-complete.
(*Hint:* consider a reduction from 4-NAE) [4 marks]

5 Computation Theory

For each $e \in \mathbb{N}$, let φ_e denote the partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ computed by the register machine with index e.

(*a*) What is meant by a *universal register machine* for computing partial functions $\mathbb{N}^k \rightharpoonup \mathbb{N}$ of any number of arguments k. [3 marks]

(*b*) How would you modify the machine from Part (*a*) to compute the partial function $u : \mathbb{N}^2 \rightharpoonup \mathbb{N}$ satisfying $u(e, x) \equiv \varphi_e(x)$ for all $e, x \in \mathbb{N}$? [2 marks]

(*c*) Given a register machine computable partial function $g : \mathbb{N}^2 \rightharpoonup \mathbb{N}$, show that there is a *total* function $\bar{g} : \mathbb{N} \to \mathbb{N}$ which is register machine computable and which satisfies $u(\bar{g}(x), y) \equiv g(x, y)$ for all $x, y \in \mathbb{N}$. [7 marks]

(*d*) Suppose $h : \mathbb{N} \to \mathbb{N}$ is a total function which is register machine computable. Show that there exists a number $n \in \mathbb{N}$ such that φ_n and $\varphi_{h(n)}$ are equal partial functions.
[*Hint*: let g be the computable partial function defined by $g(x, y) \equiv u(h(u(x, x)), y)$ and consider $\bar{g}(e)$ where $\bar{g}$ is the function obtained from g as in Part (*c*) and e is the index of some register machine that computes it.]
[8 marks]

6 Computation Theory

(*a*) (*i*) Give an inductive definition of the relation $M =_\beta N$ of *β-conversion* between λ-terms M and N. [3 marks]

(*ii*) What is meant by a term in *β-normal form*? [1 mark]

(*iii*) If M and N are in β-normal form, explain why $M =_\beta N$ implies that M and N are α-equivalent λ-terms. [2 marks]

(You need not define notons such as α-equivalence and capture-avoiding substitution.)

(*b*) Show that there are λ-terms True, False and If satisfying $\mathsf{If}\,\mathsf{True}\,M\,N =_\beta M$ and $\mathsf{If}\,\mathsf{False}\,M\,N =_\beta N$ for all λ-terms M and N and with $\mathsf{True} \neq_\beta \mathsf{False}$. [4 marks]

(*c*) Define *Curry's fixed point combinator* Y and prove its fixed point property. [3 marks]

(*d*) Consider the following two properties of a λ-term M:

(I) there exist λ-terms A and B with $M\,A =_\beta \mathsf{True}$ and $M\,B =_\beta \mathsf{False}$

(II) for all λ-terms N, either $M\,N =_\beta \mathsf{True}$ or $M\,N =_\beta \mathsf{False}$.

Prove that M cannot have both properties (I) and (II). [Hint: if M has property (I), consider $M\,(Y(\lambda x.\,\mathsf{If}\,(M\,x)\,B\,A))$.] [4 marks]

(*e*) Deduce that there is no λ-term E such that for all λ-terms M and N

$$E\,M\,N =_\beta \begin{cases} \mathsf{True} & \text{if } M =_\beta N \\ \mathsf{False} & \text{otherwise} \end{cases}$$

[3 marks]

7 Foundations of Data Science

(*a*) Let $X_1, \ldots, X_n$ be independent binary random variables, $\mathbb{P}(X_i = 1) = \theta$, $\mathbb{P}(X_i = 0) = 1 - \theta$, for some unknown parameter θ. Using Uniform$[0,1]$ as the prior distribution for θ, find the posterior distribution. [*Note:* For your answer, and in answer to parts (*b*) and (*d*), give either a named distribution with its parameters, or a normalised density function.] [3 marks]

I have collected a dataset of images, and employed an Amazon Mechanical Turk worker to label them. The labels are binary, nice or nasty. To assess how accurate the worker is, I first picked 30 validation images at random, found the true label myself, and compared the worker's label. The worker was correct on 25 and incorrect on 5.

(*b*) Let θ be the probability that the worker labels an image incorrectly. Using Beta$(0.1, 0.5)$ as the prior distribution for θ, find the posterior. [3 marks]

I next ask the worker to label a new test image, and they tell me the image is nice. Let $z \in \{\textsf{nice}, \textsf{nasty}\}$ be the true label, and let the prior distribution for z be $\Pr(\textsf{nice}) = 0.1$, $\Pr(\textsf{nasty}) = 0.9$.

(*c*) For both $z = \textsf{nice}$ and $z = \textsf{nasty}$, find

$$\mathbb{P}\big(\text{worker says } \textsf{nice} \,|\, z, \theta\big).$$

Hence find the posterior distribution of (z, θ). Your answer may be left as an un-normalised density function. [5 marks]

(*d*) Find the posterior distribution of z. [5 marks]

My colleague has more grant money and she can employ 3 workers to rate each image. On a test set of 30 images, she found that they all agreed on 15 images, worker 1 was the odd one out on 8 of the images, worker 2 was the odd one out on 4, and worker 3 was the odd one out on 3.

(*e*) Let θ_i be the probability that worker i labels an image incorrectly. Find the posterior distribution of $(\theta_1, \theta_2, \theta_3)$. Your answer may be left as an un-normalised density function. [4 marks]

Hint. The Beta(α, β) *distribution has mean* $\alpha/(\alpha + \beta)$ *and density*

$$\Pr(x) = \binom{\alpha + \beta - 1}{\alpha - 1} x^{\alpha - 1}(1 - x)^{\beta - 1}, \quad x \in [0, 1].$$

8 Foundations of Data Science

The exam paper at Oxbridge Academy has three questions, and students are asked to choose two questions and answer them. Each question is marked out of 20. The results for four students were

	question 1	question 2	question 3
student 1	13	15	
student 2	14	12	
student 3	18		10
student 4		16	8

The examiners are concerned that question 3 was harder than the other two questions. Consider the model

$$X_{ij} \sim \text{Normal}\big(\alpha_i + \beta_j, \sigma^2\big)$$

where X_{ij} is the mark for student i on question j. Here, α_i represents the ability of student i, and β_j represents the easiness of question j.

(*a*) Write this as a linear model, and identify the feature vectors. [3 marks]

(*b*) Are the feature vectors in your model linearly independent? Justify your answer. If they are not independent, rewrite your model in a form with linearly independent feature vectors. [*Hint:* You should have 6 linearly independent feature vectors.] [5 marks]

(*c*) In order to grade the students fairly, the examiners wish to fill in the blanks in the table using predicted marks. Give pseudocode to find $\alpha_1 + \beta_3$, the predicted mark for student 1 on question 3, using your model from Part (*b*). Describe briefly any standard library routines you use in your answer. [4 marks]

(*d*) What is meant by *parametric resampling*? Explain how to use parametric resampling to generate a resampled version of this dataset. [4 marks]

(*e*) To find out whether question 3 is indeed harder, the examiners wish to find a confidence interval for $\beta_3 - (\beta_1 + \beta_2)/2$. Suggest a confidence interval, and give pseudocode to find its error probability. [4 marks]

9 Logic and Proof

(*a*) Prof Blunder was using a SAT solver to solve some propositional logic problems he found in a textbook, presented in clause form. Unfortunately, he typed in the problems incorrectly, making five types of error.

In each of the following cases, briefly indicate what can be deduced about the original problem from the SAT solver output for the modified problem. Consider both possible outputs for the SAT solver: reporting "unsatisfiable" and outputting a model.

(*i*) Mistyping some occurrences of a propositional symbol so that it becomes two different symbols.

(*ii*) Mistyping two different propositional symbols such that they become the same symbol.

(*iii*) Splitting a clause in two, e.g. replacing $\{P, \neg Q, R\}$ by $\{P, \neg Q\}$ and $\{R\}$.

(*iv*) Deleting a clause.

(*v*) Moving a literal from one clause to another.

[10 marks]

(*b*) This part is concerned with Binary Decision Diagrams. Use the variable ordering P, Q, R.

(*i*) Write down the BDDs for $P \wedge Q \wedge R$ and $(\neg R \wedge Q) \rightarrow P$. There is no need to show your work. [2 marks]

(*ii*) Use the results above to obtain the BDD of

$$[P \wedge Q \wedge R] \leftrightarrow [(\neg R \wedge Q) \rightarrow P] \leftrightarrow P,$$

showing your working. [8 marks]

Hint: In $A \leftrightarrow B \leftrightarrow C$, the order of the operands is insignificant.

10 Logic and Proof

(*a*) For each of the following formulas, present either a formal resolution proof or a falsifying interpretation. Note that a and b are constants.

$$\forall x[Q(x) \to R(x)] \land \neg R(a) \land \forall x[\neg R(x) \land \neg Q(x) \to P(b) \lor Q(b)] \to P(b) \lor R(b)$$

[4 marks]

$$\exists x[\forall yz[(P(y) \to Q(z)) \to (P(x) \to Q(x))]]$$

[4 marks]

(*b*) For each of the following formulas, present a proof in a sequent or tableau calculus, or alternatively, a falsifying interpretation. In Part (*b*)(*iii*) the modal logic is S4.

(*i*) $\exists y\, \forall x\, P(x, y) \to \exists z\, P(z, z)$ [3 marks]

(*ii*) $\forall x\, [P(x) \land \exists y\, \neg P(y)] \to Q$ [5 marks]

(*iii*) $(\Box\Diamond P \land \Box\Diamond Q) \to \Box\Diamond(P \land Q)$ [4 marks]

END OF PAPER