COMPUTER SCIENCE TRIPOS Part IB

Wednesday 6 June 2018 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.*

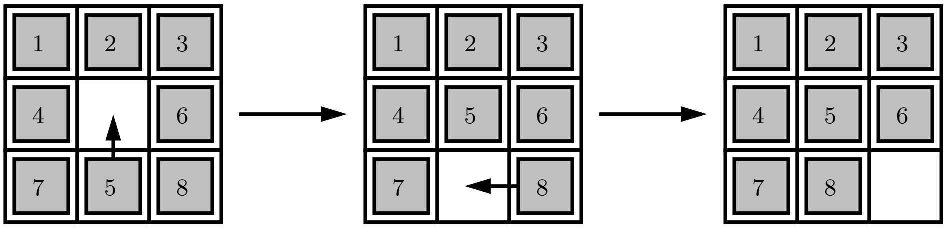
You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Artificial Intelligence

Consider the standard 3×3 *sliding blocks puzzle*.



The aim is to find a sequence of moves that re-arranges the puzzle to the state shown on the right, where each move involves sliding a single square into the empty space.

(*a*) Explain in detail how this problem can be treated as a *planning problem* by translating it into a *Boolean satisfiability (SAT)* problem. Your answer should address the following issues, and in each case should provide specific examples of the SAT representation:

(*i*) The representation of the *start state* and *goal state*. [4 marks]

(*ii*) The representation of the relevant *actions* using *successor-state* axioms. [4 marks]

(*iii*) The need for *precondition axioms*. [2 marks]

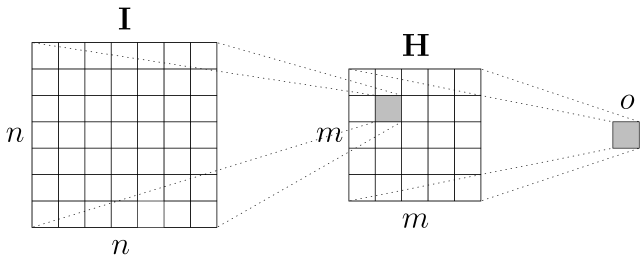
(*iv*) The need for *action-exclusion* or *state-constraint* axioms, and why one might be preferred over the other. [3 marks]

(*v*) The algorithm that can be used to employ a SAT-solver to solve a given sliding blocks problem, and the method for extracting a solution. [3 marks]

(*b*) You do not have a SAT-solver available. You do however have a solver for general *local search* problems. Explain how you might use the latter to solve the SAT problem obtained in Part (*a*). [4 marks]

2 Artificial Intelligence

Evil Robot is updating his visual system. He has a single camera that produces an $n \times n$ matrix $\mathbf{I}$ of pixel values. His visual system is arranged as follows:



The input $\mathbf{I}$ is reduced to an $m \times m$ matrix $\mathbf{H}(\mathbf{I})$. The elements $H_{i,j}$ are

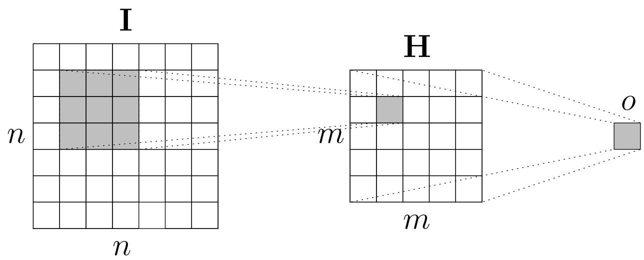
$$H_{i,j}(\mathbf{I}) = \sigma\left(\sum_{k=1}^{n}\sum_{l=1}^{n} w_{k,l}^{(i,j)} I_{k,l} + b^{(i,j)}\right)$$

where σ is an appropriate function, and $w_{k,l}^{(i,j)}$ and $b^{(i,j)}$ are the weights and bias for element (i,j). A single output $o(\mathbf{H})$ is computed as

$$o(\mathbf{H}) = \sigma\left(\sum_{k=1}^{m}\sum_{l=1}^{m} w_{k,l} H_{k,l} + b\right).$$

(*a*) If Evil Robot has a training example $(\mathbf{I}', y')$ and is using an error $E(\mathbf{w})$ where $\mathbf{w}$ is a vector of all weights and biases available, derive an algorithm for computing $\frac{\partial E}{\partial \mathbf{w}}$ for the example. [12 marks]

(*b*) A modification to the system works as follows:



The mapping from $\mathbf{I}$ to $\mathbf{H}$ is replaced by an $n' \times n'$ *convolution kernel*. This has a single set of parameters $v_{k,l}$ and c used to compute every element of $\mathbf{H}$ as the weighted sum of a patch of elements in $\mathbf{I}$

$$H_{i,j}(\mathbf{I}) = \sigma\left(\sum_{k=1}^{n'}\sum_{l=1}^{n'} v_{k,l} I_{i+k-1,j+l-1} + c\right).$$

Provide a detailed description of how the algorithm derived in Part (*a*) must be updated to take account of this modification. [8 marks]

3 Complexity Theory

(*a*) Give a precise definition of each of the complexity classes NP and co-NP. [4 marks]

(*b*) Give an example each of

(*i*) an NP-complete language; and

(*ii*) a co-NP-complete language,

in each case giving a precise statement of the decision problem involved. [4 marks]

(*c*) If A and B are the two languages identified in Part (*b*), give an example of a language that is polynomial-time reducible to both A and B. Justify your answer. [4 marks]

(*d*) Consider the following statement:

> There is a polynomial p such that every valid Boolean formula of length n has a proof of length at most $p(n)$. Moreover, there is a polynomial-time algorithm that can check the correctness of the proofs.

This statement is not known to be true or false. Explain what would be the consequences of this statement being true or false for the relationship between NP and co-NP, giving full justification for your answer. [8 marks]

4 Complexity Theory

Consider the following two decision problems:

- **Reach** – the problem of deciding, given a *directed* graph G and two vertices a and b in G, whether there is a path in G from a to b.
- **UReach** – the problem of deciding, given an *undirected* graph G and two vertices a and b in G, whether there is a path in G from a to b.

It is known that **Reach** is NL-complete (under logarithmic-space reductions) and that **UReach** is in the complexity class L.

(*a*) Based on the above information, for each of the following statements, state whether it is true, false, or unknown. In each case, give justification for your answer and in the case where the truth of the statement is unknown, state any implications that might follow from it being true or false.

(*i*) **Reach** $\leq_L$ **UReach**, i.e. **Reach** is reducible in logarithmic-space to **UReach**.

(*ii*) **UReach** $\leq_L$ **Reach**.

(*iii*) **UReach** is in P.

(*iv*) If **Reach** is in L, then P=NP.

[3 marks each]

(*b*) Let us say that a nondeterministic Turing machine M is *symmetric* if for any two configurations c_1 and c_2 of M, if $c_1 \rightarrow_M c_2$, then $c_2 \rightarrow_M c_1$. We write SL for the class of all languages that are accepted by a symmetric Turing machine using $O(\log n)$ work space on inputs of length n.

By considering the configuration graph of a machine and using the fact that **UReach** is in L, explain why it follows that SL$\subseteq$L. [8 marks]

5 Computation Theory

(*a*) Give a precise definition of the class of *partial recursive functions.* [3 marks]

(*b*) We can associate with each natural number $i \in \mathbb{N}$ the partial recursive function $f_i : \mathbb{N} \rightharpoonup \mathbb{N}$ computed by the register machine coded by the number i. Explain why

(*i*) for every partial recursive function $f : \mathbb{N} \rightharpoonup \mathbb{N}$, there is an i such that $f = f_i$; and

(*ii*) the partial function $g : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ given by $g(i, n) = f_i(n)$ is computable.

[8 marks]

(*c*) Show that the total function $T : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given by:

$$T(i, n) = \begin{cases} 1 & \text{if } f_i(n)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

is uncomputable. Here f_i refers to the partial function associated with $i \in \mathbb{N}$ as in (*b*). [9 marks]

6 Computation Theory

(*a*) What does it mean to say that a partial function $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is *register machine computable*? [2 marks]

(*b*) Show that the following functions are register machine computable:

(*i*) $\text{add}(x, y) \triangleq x + y$;

(*ii*) $\text{max}(x, y) \triangleq \begin{cases} y & \text{if } x \leq y \\ x & \text{otherwise} \end{cases}$; and

(*iii*) $\text{comp}(x, y) \triangleq \begin{cases} 0 & \text{if } x \leq y \\ 1 & \text{otherwise.} \end{cases}$

[9 marks]

(*c*) What does it mean to say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is λ-*definable*? [2 marks]

(*d*) Is every λ-definable function register-machine computable? Give a detailed justification for your answer. [7 marks]

7 Foundations of Data Science

Let $X_1, \ldots, X_{100}$ be independent samples drawn from the $\mathrm{Exp}(\lambda)$ distribution, for some unknown parameter $\lambda > 0$.

[*Note:* The $\mathrm{Exp}(\lambda)$ distribution has density function $f(x) = \lambda e^{-\lambda x}$, for $x > 0$. It has mean $1/\lambda$, and variance $1/\lambda^2$.]

(*a*) Show that the maximum likelihood estimator for λ is $\hat{\lambda} = 100/\sum_{i=1}^{100} X_i$. [3 marks]

(*b*) Using the central limit theorem, find a and b such that

$$\mathbb{P}\big(1/\hat{\lambda} \in [a, b]\big) \approx 0.95$$

explaining your calculations carefully. Hence find real numbers α and β such that

$$\mathbb{P}\big(\lambda \in [\alpha\hat{\lambda}, \beta\hat{\lambda}]\big) \approx 0.95\,.$$

[6 marks]

(*c*) Explain how to use the bootstrap resampling method to approximate the probability

$$\mathbb{P}\Big(\lambda \in \big[\hat{\lambda}(1-\varepsilon), \hat{\lambda}(1+\varepsilon)\big]\Big)$$

where ε is given. In your answer, include an explanation of what is meant by 'resampling'. [6 marks]

(*d*) Using your answer to Part (*c*), give pseudocode to compute ε such that

$$\mathbb{P}\Big(\lambda \in \big[\hat{\lambda}(1-\varepsilon), \hat{\lambda}(1+\varepsilon)\big]\Big) \approx 0.95\,.$$

Comment your code appropriately. [5 marks]

8 Foundations of Data Science

Fisher's Iris dataset contains, among other things, measurements of Petal.Length and Sepal.Length for samples from each of three species of iris. Suppose we want to fit the model

$$\text{Petal.Length} = \alpha_s + \beta_s \,\text{Sepal.Length} + \text{Normal}(0, \sigma^2)$$

where s is the species.

Note: The Normal(μ, σ^2) distribution has density function

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}.$$

(*a*) Explain what is meant by 'linear model', 'feature', and 'orthogonal projection'. Rewrite the above model as a linear model made up of linearly independent features, and explain why they are linearly independent. [7 marks]

(*b*) You are given a library function proj$(y, [e_1, \ldots, e_n])$. It returns a list $[\lambda_1, \ldots, \lambda_n]$ such that $\lambda_1 e_1 + \cdots + \lambda_n e_n$ is the orthogonal projection of the vector y onto the subspace spanned by vectors $\{e_1, \ldots, e_n\}$. Explain what is meant by the 'least squares method', and give pseudocode using proj to find the least squares estimators for α_s and β_s. [2 marks]

(*c*) Explain how to compute the maximum likelihood estimators of α_s, β_s, and σ. In your answer, you should explain the relationship between the least squares method and maximum likelihood estimation. [5 marks]

(*d*) We wish to know whether the β_s coefficients for the three species are noticeably different. Outline the Bayesian approach to answering this question. [6 marks]

9 Logic and Proof

(a) Outline the basic ideas behind Fourier-Motzkin variable elimination, demonstrating them by applying the technique to the following set of constraints:

$$x - z \leq 2 \qquad x + y - z \geq 5 \qquad y + 2z \leq 6 \qquad x + 2 \geq 3y$$

[8 marks]

(b) Give and explain the inference rules of binary resolution and factoring, in the context of automated theorem proving. [4 marks]

(c) For the following clauses in Kowalski form, express each clause as a set of literals. For the resulting set of clauses, either exhibit a model or show that none exists. Notice that a, b and c are constants, while x, y and z are variables. Briefly justify your answer.

$$\begin{aligned} P &\rightarrow Q(a) \vee S(x) \vee T(y) \\ T(b) &\rightarrow \\ Q(z) &\rightarrow \\ U(b) \wedge S(c) &\rightarrow T(y) \\ U(y) &\rightarrow T(y) \vee P \\ &\rightarrow U(b) \end{aligned}$$

[8 marks]

10 Logic and Proof

(*a*) Describe the DPLL method, explaining briefly each of its four steps. [4 marks]

(*b*) Use the DPLL method to find a model satisfying the following set of clauses, or to prove that no such model exists.

$$\{P, Q\}\,\{R\}\,\{Q, \neg S, \neg Q\}\,\{\neg Q, \neg R, S\}\,\{P, \neg Q\}\,\{P, Q, \neg S\}\,\{P, \neg R, S\}\,\{\neg P, \neg S\}$$

[6 marks]

(*c*) Describe briefly the procedure for constructing a BDD. Illustrate your answer by constructing the BDD (using the variable order $P < Q < R < S$) for:

$$(\neg P \lor (Q \land S)) \lor (S \lor (\neg R \lor S))$$

[10 marks]

END OF PAPER