YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR.

Answer ALL FOUR Questions

Examiners: Prof. Geraint A. Wiggins; Prof. Edmund Robinson

Answer ALL FOUR of the following questions.

1. **Reasoning using Bayes' Rule**

   Bayes' Rule,

   $$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)},$$

   can be used to compute the probabilities of outcomes in situations such as the following:

   > Prof. Smith is crossing the Pacific Ocean on a plane, on her way to a conference. The Captain has just announced that an unusual engine fault has been signalled by the plane's computer; this indicates a fault that only occurs once in 10,000 flights. If the fault report is true, then there's a 70% chance the plane will have to crash-land in the Ocean, which means certain death for the passengers. However, the sensors are not completely reliable: there's a 2% chance of a false positive; and there's a 1% chance of the same fault occurring without the computer flagging the error report.

   (a) Formulate this problem in terms of conditional probabilities of outcomes, existence of a fault and whether or not it is reported. **[10 marks]**

   (b) Use Bayes' rule to compute Prof. Smith's chances of survival. Give your full reasoning and explain it. **[15 marks]**

2. **Knowledge Representation and Inference**

Consider the following short story.

There is a Bear. The Bear either sleeps in its cave or hunts in the forest. If the Bear is hungry then it does not sleep. If the Bear is tired then it does not hunt.

(a) List the minimal predicates and/or objects required to specify the premises given.
[3 marks]

(b) Formulate the story, above, in FOPC, using your predicates and/or objects. **[3 marks]**

(c) Convert your FOPC into conjunctive normal form, showing and explaining your working. **[8 marks]**

(d) Write down a fully explained resolution proof that the Bear hunts if it is hungry.
[11 marks]

3. **Genetic Algorithms applied to Logical Reasoning**

Consider the following 8-bit chromosomes, to be used as the initial state of a very small genetic algorithm.

Bit Number	0	1	2	3	4	5	6	7
Chromosome C0	0	1	1	0	0	0	1	1
Chromosome C1	1	1	0	0	1	0	1	1
Chromosome C2	0	0	0	1	1	0	0	1
Chromosome C3	0	1	1	1	0	0	1	1

Answer the following questions, using this data.

(a) What would be result of applying crossover to **C0** and **C1** between bits **2** and **3**?
[2 marks]

(b) Explain why crossover is not enough to explore the entire search space described by this representation from this initial state, and propose a solution. Explain your solution in full detail. **[6 marks]**

(c) Explain how the above representation could be used to explore a search space of logical literals of the form $P(A, B)$, representing the application of 2-place (binary) predicates to a set of logical constants. What constraint will the representation you describe place on the predicates and constants that you can represent? **[6 marks]**

(d) Explain the component of a genetic algorithm that directs search towards successful solutions. **[4 marks]**

(e) Consider a domain containing the following symbols.

Objects:	A B C D E G H J K L
Predicates:	M P Q R S U V W

All predicates are binary. Objects A…G (only) can occur as the first argument of a predicate, and objects H…L (only) can occur as the second.

Suppose you are given a resolution theorem prover, with a suitable programmable interface (API), and a set, $\mathcal{E}$, of logical expressions in the above language. Describe a chromosome representation that would allow you to use the genetic algorithm above to search the space of possible literals to find literals that *contradict* $\mathcal{E}$. State any assumptions you make in your design, and any requirements that your design places on the other components of the system. **[7 marks]**

4. **Adversarial Search**

   Consider the game of Noughts and Crosses. The game is played on a board consisting of nine squares arranged in a 3 x 3 matrix. We number those squares like this:

1	2	3
4	5	6
7	8	9

   To play, two players, **X** and **O**, take turns to place a piece on the board. A new piece may only be placed on an empty square. The winner is the first player to achieve a line of 3 of their pieces, horizontally, vertically or diagonally. Thus, a winning board for **X** would be:

		O
X	X	X
O		

   Now consider a computer program that is to play **X** in the game, from the following position:

		O
O	X	X
O		

   The program always considers squares in the same order: 1, 2, 3, 4, 5, 6, 7, 8, 9 in the diagram above. Draw the game tree, as generated by a minimax procedure using alpha-beta pruning to optimise the number of states that are considered. Number your states to indicate the order in which they are expanded, using numbers in circles, like this:

   (5)

   Do not number states which are added to the agenda but not expanded under minimax. Use the following utility function:

   Utility = number of lines of 3 with 2 **X**s $+$
   10 $\times$ number of lines of 3 with 3 **X**s $-$
   number of lines of 3 with 2 **O**s $-$
   10 $\times$ number of lines of 3 with 3 **O**s

   So the utility value for the starting state, above, is -1, as **X** has one horizontal row of 2 pieces, while **O** has one vertical and one diagonal row of 2 pieces each.

   Indicate next to each state what utility value is passed back up the tree in the minimax/alpha-beta pruning process. **[25 marks]**

End of Paper