NATIONAL UNIVERSITY OF SINGAPORE

CS3243 - INTRODUCTION TO ARTIFICIAL INTELLIGENCE

(Semester 2: AY2017/18)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. This assessment paper contains **FIVE (5)** parts and comprises of **(12)** printed pages, including this page.
2. Answer **ALL** questions as indicated.
3. This is a **CLOSED BOOK WITH APPROVED MATERIALS** assessment.
4. You are allowed to use **NUS APPROVED CALCULATORS**.
5. Please write your **Student Number** below. DO NOT WRITE YOUR NAME.

STUDENT NUMBER: ______________________________

EXAMINER'S USE ONLY		
Part	Mark	Score
I	9	
II	12	
III	10	
IV	12	
V	7	
TOTAL	50	

In Part I, II, III, IV, and V, you will find a series of short essay questions. For each short essay question, give your answer in the reserved space in the script.

Part I
Constraint Satisfaction Problems

(9 points) Short essay questions. Answer in the space provided on the script.

The Class Scheduling Problem: There is a set of professors $N = \{1, \dots, n\}$, and a set of classes that can be offered $C = \{c_1, \dots, c_m\}$. We are given a list of potential class time slots $\{1, \dots, t\}$. Each class c_j has an integer duration $1 \leq d(c_j) \leq t$. Each professor i has a list of classes that they can teach, $T_i \subseteq C$, and a list of time slots during which they are free: $F_i \subseteq \{1, \dots, t\}$. A class schedule is valid if

- Every class is scheduled
- Every class c_j is assigned a professor i who can teach c_j at the time that it is scheduled.
- If a professor i is assigned to teach a class c_j, she has to be free for all time slots that the class takes (so all time slots the class takes are in F_i).
- A professor cannot be teaching two classes at the same time slot.

Our objective is to output a valid schedule, or output that no such schedule exists.

1. (5 points) Write the class scheduling problem as a CSP.

Solution:

Variables:

Constraints:

2. (4 points) Consider the following instance of the class scheduling problem. We have two professors: David (D) and Wee Sun (WS), and two classes: Artificial Intelligence (AI) and Machine Learning (ML). There are four time slots: $t_1, \ldots, t_4$ (of one hour each); David is free on slots t_1, t_3 and t_4 (so $F_D = \{t_1, t_3, t_4\}$) and Wee Sun is free on slots t_3 and t_4 (so $F_{WS} = \{t_3, t_4\}$). David can teach both AI and ML, but Wee Sun can only teach ML. AI is a one hour class, and ML is a two hour class.

   (a) Draw the constraint graph for this instance of the class scheduling problem.

   (b) What is the most constraining variable in this CSP?

Solution:

(a) Draw the constraint graph here:

(b) The most constraining variable:

$P =$	T_1=1 T_2=0 T_3=0	T_1=0 T_2=1 T_3=0	T_1=0 T_2=0 T_3=1	T_1=1 T_2=1 T_3=0	T_1=1 T_2=0 T_3=1	T_1=0 T_2=1 T_3=1	T_1=1 T_2=1 T_3=1
Company	0.07	0.12	0.03	0.28	0.07	0.12	0.28
External	0.12	0.12	0.08	0.18	0.12	0.12	0.18
Spam	0.16	0.06	0.24	0.04	0.16	0.06	0.04

Table 1: The data collected by MyAI.

Part II
Uncertainty and Bayesian Networks

(12 points) Short essay questions. Answer in the space provided on the script.
The company MyAI is using a Spam filter. The company receives three types of emails: Spam ($Spam$), MyAI internal emails ($Company$) and emails received from external entities ($External$). To check the type of a given email, MyAI runs three independent tests denoted T_1, T_2, T_3; an email that fails at least two of the three tests is flagged as Spam. Let P be the random variable for email type ($Spam$, $Company$ or $External$); for every $i \in \{1, 2, 3\}$, the random variable T_i takes a value of 1 if an email passes the test T_i and a value of 0 otherwise.

The company MyAI has collected the following data, summarized in Table 1. The entries in Table 1 are to be read as follows: each entry is the conditional probability of the event given in the top column occurring, given that the email is of a given type. For example, looking at the $External$ row, and the $T_1 = 1, T_2 = 0, T_3 = 1$ column, we have that $\Pr[T_1 = 1, T_2 = 0, T_3 = 1 \mid P = External] = 0.12$. In addition, it is known that

$$\Pr[P = Spam] = 0.1; \Pr[P = Company] = 0.6; \Pr[P = External] = 0.3.$$

1. (4 points) Construct and draw a Bayesian network in the following order: P, T_1, T_2, T_3. Remember to include the **conditional probability tables** (CPTs).

Solution:

2. (4 points) Suppose that an email has failed T_1 and T_2 but passed T_3. What is the likeliest email type? That is, decide which of the following three probabilities is the greatest:

$$\Pr[P = \mathit{Spam} \mid T_1 = 0, T_2 = 0, T_3 = 1]$$
$$\Pr[P = \mathit{Company} \mid T_1 = 0, T_2 = 0, T_3 = 1]$$
$$\Pr[P = \mathit{External} \mid T_1 = 0, T_2 = 0, T_3 = 1]$$

Explain your answer, including computational steps.

Solution:

The likeliest email type is:

Explanation:

3. (1 point) What is the probability that an email failed tests T_1 or T_2 given that it is an external email? That is, compute the probability

$$\Pr[(T_1 = 0) \vee (T_2 = 0) \mid P = \textit{External}]$$

Your answer should be accurate up to 3 decimal places; present your calculation in the space below as well.

Solution:

$\Pr[(T_1 = 0) \vee (T_2 = 0) \mid P = \textit{External}] =$

Explanation:

4. (3 points) What is the probability that an email is actually a Spam email given that it was flagged as spam? That is, let S be the event that at least two tests outputted 0. Compute the probability

$$\Pr[P = \textit{Spam} \mid S]$$

Your answer should be accurate up to 3 decimal places; present your calculation in the space below as well.

Solution:

$\Pr[P = \textit{Spam} \mid S = 1] =$

Explanation:

Part III
Logical Agents

(10 points) Short essay questions. Answer in the space provided on the script.

The social networking platform InstaFun maintains a database of all its users; this includes a list of all the people that a user follows. InstaFun implemented the function $follows(a, b)$, whose input is two users and whose output is `true` if and only if a follows b; it is assumed that $follows(a, a) =$ `false` for any user a. Finally, the set of all users on the InstaFun database is denoted U.

1. (2 points) InstaFun wants to know, given two users, what is their *follow distance*. If a follows b, then a 1-follows b; if a is not following b, but is following a person x who is following b, then a 2-follows b. More generally, we say that a k-follows b if a is not ℓ-following b for all $\ell < k$, and there exist $k - 1$ people $x_1, \ldots, x_{k-1}$ such that a follows x_1; x_i follows x_{i+1} for all $i < k - 1$, and x_{k-1} follows b. The binary function $follows(k, a, b)$ takes as input a parameter k and two users a and b; it outputs `true` if and only if a k-follows b. Describe the function $follows(k, a, b)$ using first-order logic. Use the function $follows(a, b)$ in your description.

Solution:

2. (3 points) Suppose that we know the following facts

   1. Claire is the only person who follows Alice.
   2. Danielle follows Claire
   3. Bob follows people if and only if they follow Claire.

   Represent these facts as sentences in first-order logic.

Solution:

3. (5 points) Convert the sentences derived in the previous question to CNF form, and apply resolution in order to derive the statement

   "Bob follows someone, who follows someone, who follows Alice."

   At each resolution step, state the variable substitutions required to obtain the resolvent.

Solution:

Write down the logical statements (as well as the query) in CNF form below:

In each line in the below table, write the two FOL sentences you wish to resolve as (Statement 1) and (Statement 2); their resolvent should appear in both in the marked column, and in the next line as Statement 1. The required substitution should be written in the last column. Note that you might not need to fill all table rows.

Statement 1	Statement 2	Resolvent	Substitution (θ =?)

Part IV
Adversarial Search

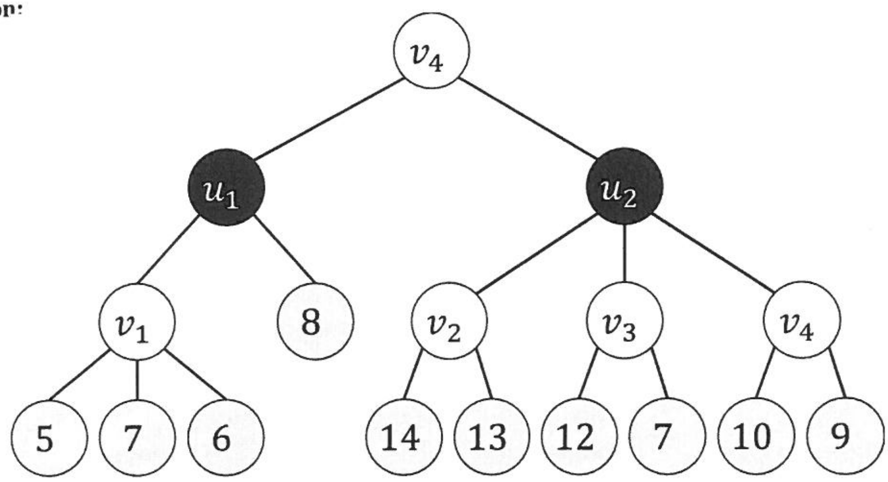
(12 points) Short essay questions. Answer in the space provided on the script.

```
function ALPHA-BETA-SEARCH(state) returns an action
  v ← MAX-VALUE(state, −∞, +∞)
  return the action in ACTIONS(state) with value v

function MAX-VALUE(state, α, β) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← −∞
  for each a in ACTIONS(state) do
    v ← MAX(v, MIN-VALUE(RESULT(s,a), α, β))
    if v ≥ β then return v
    α ← MAX(α, v)
  return v

function MIN-VALUE(state, α, β) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← +∞
  for each a in ACTIONS(state) do
    v ← MIN(v, MAX-VALUE(RESULT(s,a) , α, β))
    if v ≤ α then return v
    β ← MIN(β, v)
  return v
```

Figure 1: Alpha-beta pruning algorithm (note that $s = state$).

1. (4 points) Consider the minimax search tree shown in the solution space below; the utility function values are specified with respect to the MAX player and indicated at all the leaf (terminal) nodes. The MAX player controls the white nodes (s, v_1, v_2, v_3 and v_4), and the MIN player controls all the black nodes (u_1 and u_2). Suppose that we use alpha-beta pruning algorithm, given in Figure 5.7 of AIMA 3rd edition (reproduced in Figure 1), in the direction from **right to left** to prune the search tree. **Mark (with an "X") all ARCS** that are pruned by alpha-beta pruning, if any.

Solution:

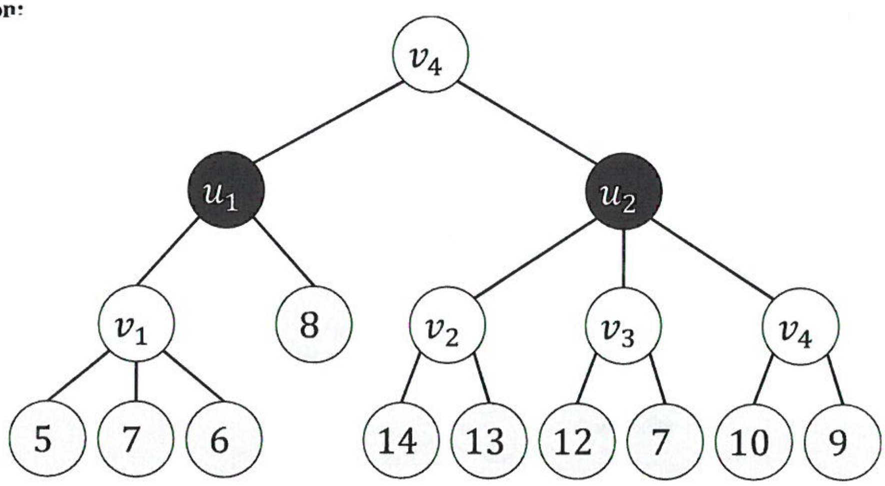


State the **EXACT** minimax value at the root node.

Solution:

2. (4 points) Consider again the same minimax search tree discussed in question 1; suppose that we again use the alpha-beta pruning algorithm (Figure 1) to prune edges, but this time we iterate over nodes in the direction **from left to right**. **Mark (with an "X") all ARCS** that are pruned by alpha-beta pruning, if any.

Solution:



3. (4 points) Consider the minimax search tree shown in Figure 2 below; the utility function values are specified with respect to the MAX player and indicated at all the leaf (terminal) nodes. The MAX player controls the white nodes s and v_1, and the MIN player controls the black nodes u_1 and u_2. Suppose that alpha-beta pruning algorithm, given in Figure 5.7 of AIMA 3rd edition (reproduced in Figure 1), is used. We inspect nodes in order from **left to right**.

For each of the statements below, mark it as either *true* or *false*.

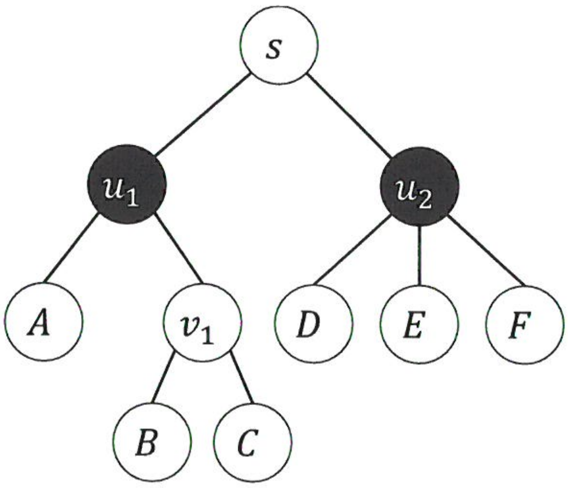


Figure 2: Minimax search tree.

Solution: In order to ensure that the **maximal** number of arcs is pruned, we **must** have that:

	True	False
$A < B$	☐	☐
$D > E$	☐	☐
$D > A$	☐	☐
$B > E$	☐	☐

Part V
Informed Search

(7 points) Short essay questions. Answer in the space provided on the script.
A basic wooden railway set contains the pieces shown in Figure 3. The task is to connect these pieces into a railway that has no overlapping tracks and no loose ends where a train could run off onto the floor. Every loose end in a state can be either a 'peg' (bulging out) or a 'hole' (bulging in) (see Figure 4)

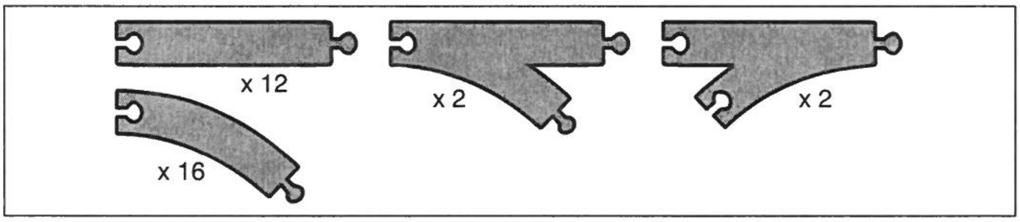


Figure 3: The track pieces in a wooden railway set; each is labeled with the number of copies in the set. Note that curved pieces and "fork" pieces ("switches" or "points") can be flipped over so they can curve in either direction. Each curve subtends at a 45 degree angle.

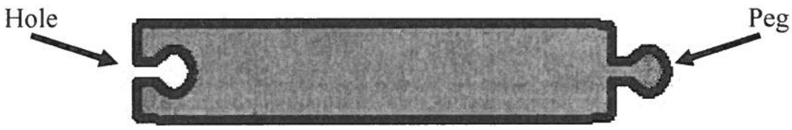


Figure 4: Pegs bulge out; holes bulge in.

1. (3 points) Suppose that the pieces fit together *exactly* with no slack; give a precise formulation of the task as a search problem.

Solution: Fill out your answers to each of the items below.

Initial State:

Successor Function:

Goal Test:

Step Cost:

2. (2 points) Identify a suitable **uninformed** search algorithm for this task; justify your choice.

Solution:

3. (2 points) Consider the following heuristic function $h(n)$:

$$h_1(n) = \frac{1}{2}\text{number of open pegs in } n$$

Prove that h_1 is admissible; what kind of relaxation on the problem constraints does n assume?

Solution:

END OF PAPER