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 **(15)** printed pages, including this page.
2. Answer **ALL** questions as indicated.
3. This is a **PARTIAL OPEN BOOK** 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 $N = \{1, \ldots, n\}$, and a set of classes that can be offered $C = \{c_1, \ldots, c_m\}$. We are given a list of potential class time slots $\{1, \ldots, t\}$. 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: $F_i \subseteq \{1, \ldots, t\}$. 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: We can have several acceptable solutions. One option is to have a variable for each class and each time τ, $y(j, \tau)$ that equals 1 if c_j is taught at time τ and is 0 otherwise. In addition, we have a variable $x(i, j)$ that equals 1 if professor i teaches class j and is zero otherwise. We omit the variable $x(i, j)$ whenever $c_j \notin T_i$; we can also omit the variable $y(j, \tau)$ if the class c_j cannot be taught at time τ: this can happen if there is no professor available to teach the class c_j at time τ, or if there is not enough time to finish the class if it starts at time τ. Constraints should be as follows:

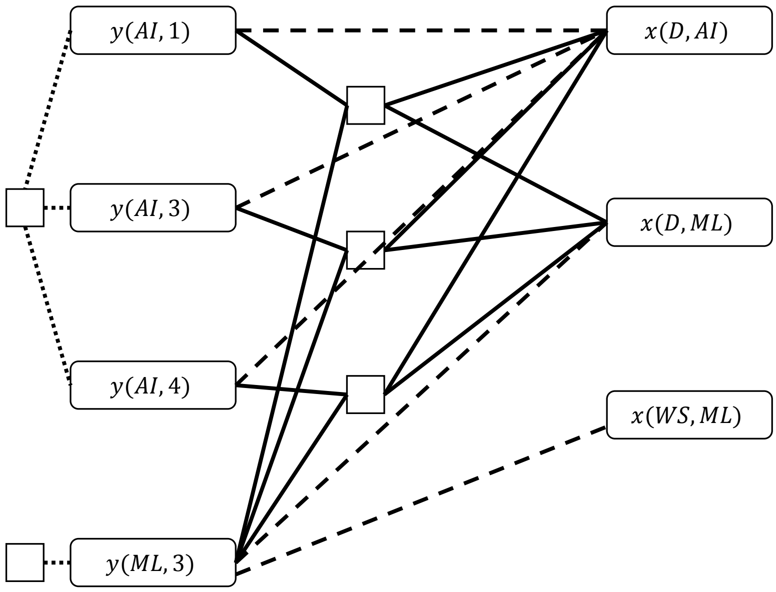
- If professor i teaches the class c_j then she should be free during that time: if $x(i, j) = 1$ and $y(j, \tau) = 1$ then
$$\tau, \ldots, \tau + d(c_j) - 1 \in F_i.$$
This constraint involves the two variables $x(i, j)$ and $y(j, \tau)$; there is such a constraint for all i, j and τ.
- If professor i teaches classes c_j and c_k then there should be no clashes between them: if $x(i, j) = x(i, k) = 1$, and $y(j, \tau) = y(k, \tau') = 1$ then
$$\{\tau, \tau + 1, \ldots, \tau + d(c_j) - 1\} \cap \{\tau', \tau' + 1, \ldots, \tau' + d(c_j) - 1\} = \emptyset$$
This constraint involves four variables $x(i, j), x(i, k), y(j, \tau), y(j, \tau')$ and there is such a constraint for all valid choices of i, j, k, τ, τ'.
- Each class is taught at least once: for every c_j, $\sum_{\tau=1}^{t} y(j, \tau) \geq 1$. Of course there is no need to teach a class more than once in our setting so it is safe to require $\sum_{\tau=1}^{t} y(j, \tau) = 1$. This is a constraint involving all $y(j, t)$; there is one such constraint for each class j.

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:



Refer to the figure above. The solid lines connecting the the squares correspond to the constraint that classes taught by the same professor should not have time clashes (not that WS has no such constraints as he can teach just one class). The dashed lines correspond to the constraint that a class can be taught in a certain time if a professor is free to teach it. The dotted constraints on the left correspond to the constraints that each class needs to be taught at least once. According to this graph there are two most constraining variables: $x(D, AI)$ and $y(ML, 3)$, both with six edges (answers ignoring the unary constraint and choosing $x(D, AI)$ will be accepted as correct as well).

$P =$	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

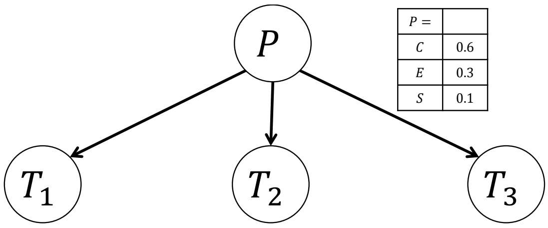
(12 points) Short essay questions. Answer in the space provided on the script.
The company MyAI is using a Spam filter. The company receives three types of emails: Spam ($Spam$), MyAI internal emails ($Company$) and emails received from external entities ($External$). To check the type of a given email, 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:



$P =$	
C	0.6
E	0.3
S	0.1

$P =$	$\Pr[T_1 = 1 \mid P]$
C	0.7
E	0.6
S	0.4

$P =$	$\Pr[T_2 = 1 \mid P]$
C	0.8
E	0.6
S	0.2

$P =$	$\Pr[T_3 = 1 \mid P]$
C	0.5
E	0.5
S	0.5

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: We compute the probabilities for each one of the events. Let $\alpha = \frac{1}{\Pr[T_1=0,T_2=0,T_3=1]}$; then:

$$\begin{aligned}
&\Pr[P = \mathit{Spam} \mid T_1 = 0, T_2 = 0, T_3 = 1] = \\
&\alpha \Pr[P = \mathit{Spam}, T_1 = 0, T_2 = 0, T_3 = 1] = \\
&\alpha \Pr[T_1 = 0, T_2 = 0, T_3 = 1 \mid P = \mathit{Spam}] \Pr[P = \mathit{Spam}] = \\
&\alpha \Pr[T_1 = 0 \mid P = \mathit{Spam}] \Pr[T_2 = 0 \mid P = \mathit{Spam}] \Pr[T_3 = 1 \mid P = \mathit{Spam}] \Pr[P = \mathit{Spam}] = \\
&\alpha \times 0.6 \times 0.8 \times 0.5 \times 0.1 = 0.024\alpha
\end{aligned}$$

Similarly we have that

$$\begin{aligned}
&\Pr[P = \mathit{External} \mid T_1 = 0, T_2 = 0, T_3 = 1] = \\
&\alpha \Pr[T_1 = 0 \mid P = \mathit{External}] \Pr[T_2 = 0 \mid P = \mathit{External}] \Pr[T_3 = 1 \mid P = \mathit{External}] \Pr[P = \mathit{External}] = \\
&\alpha \times 0.4 \times 0.4 \times 0.5 \times 0.3 = 0.024\alpha
\end{aligned}$$

and

$$\begin{aligned}
&\Pr[P = \mathit{Company} \mid T_1 = 0, T_2 = 0, T_3 = 1] = \\
&\alpha \Pr[T_1 = 0 \mid P = \mathit{Company}] \Pr[T_2 = 0 \mid P = \mathit{Company}] \Pr[T_3 = 1 \mid P = \mathit{Company}] \Pr[P = \mathit{Company}] = \\
&\alpha \times 0.3 \times 0.2 \times 0.5 \times 0.6 = 0.018\alpha
\end{aligned}$$

So there are two equally likely values for P: $P = \mathit{Spam}$ and $P = \mathit{External}$.

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

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

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

Solution: This can be done very quickly using the fact that $\Pr[A \vee B] = \Pr[A] + \Pr[B] - \Pr[A \vee B]$. Moreover, since the events are conditionally independent on $P = \mathit{External}$ we have that

$$\begin{aligned} \Pr[T_1 = 0 \vee T_2 = 0 \mid \mathit{External}] = \\ \Pr[T_1 = 0 \mid \mathit{External}] + \Pr[T_2 = 0 \mid \mathit{External}] - \Pr[T_1 = 0 \mid \mathit{External}] \Pr[T_2 = 0 \mid \mathit{External}] = \\ 0.4 + 0.4 - 0.16 = 0.64 \end{aligned}$$

Alternatively, the probability that both are 1 given $\mathit{External}$ is $0.6 \times 0.6 = 0.36$, so

$$\Pr[\neg(T_1 = 1 \wedge T_2 = 1) \mid P = \mathit{External}] = 1 - 0.36 = 0.64.$$

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

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

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

Solution: First we observe that

$$\Pr[P = \mathit{Spam} \mid S] = \frac{\Pr[P = \mathit{Spam}, S]}{\Pr[S]}.$$

First, let us compute $\Pr[S]$

$$\begin{aligned} \Pr[S] =& \Pr[S \mid \mathit{Spam}] \Pr[\mathit{Spam}] + \Pr[S \mid \mathit{External}] \Pr[\mathit{External}] + \Pr[S \mid \mathit{Company}] \Pr[\mathit{Company}] \\ =& 0.1 \Pr[S \mid \mathit{Spam}] + 0.3 \Pr[S \mid \mathit{External}] + 0.6 \Pr[S \mid \mathit{Company}] \end{aligned}$$

Note that since the events are conditionally independent this is easy to compute:

$$\begin{aligned} \Pr[S \mid \mathit{Spam}] =& \Pr[T_1 = 1, T_2 = T_3 = 0 \mid \mathit{Spam}] + \Pr[T_1 = 0, T_2 = 1, T_3 = 0 \mid \mathit{Spam}] + \\ & \Pr[T_1 = 0, T_2 = 0, T_3 = 1 \mid \mathit{Spam}] + \Pr[T_1 = T_2 = T_3 = 0 \mid \mathit{Spam}] \\ =& 0.4 \times 0.8 \times 0.5 + 0.6 \times 0.2 \times 0.5 + 0.6 \times 0.8 \times 0.5 + 0.6 \times 0.8 \times 0.5 \\ =& 0.16 + 0.06 + 0.24 + 0.24 = 0.7 \end{aligned}$$

$$\begin{aligned} \Pr[S \mid \mathit{External}] =& \Pr[T_1 = 1, T_2 = T_3 = 0 \mid \mathit{External}] + \Pr[T_1 = 0, T_2 = 1, T_3 = 0 \mid \mathit{External}] + \\ & \Pr[T_1 = 0, T_2 = 0, T_3 = 1 \mid \mathit{External}] + \Pr[T_1 = T_2 = T_3 = 0 \mid \mathit{External}] \\ =& 0.6 \times 0.4 \times 0.5 + 0.4 \times 0.6 \times 0.5 + 0.4 \times 0.4 \times 0.5 + 0.4 \times 0.4 \times 0.5 \\ =& 0.5(0.24 + 0.24 + 0.16 + 0.16) = 0.4 \end{aligned}$$

$$\begin{aligned} \Pr[S \mid \mathit{Company}] =& \Pr[T_1 = 1, T_2 = T_3 = 0 \mid \mathit{Company}] + \Pr[T_1 = 0, T_2 = 1, T_3 = 0 \mid \mathit{Company}] + \\ & \Pr[T_1 = 0, T_2 = 0, T_3 = 1 \mid \mathit{Company}] + \Pr[T_1 = T_2 = T_3 = 0 \mid \mathit{Company}] \\ =& 0.7 \times 0.2 \times 0.5 + 0.3 \times 0.8 \times 0.5 + 0.3 \times 0.2 \times 0.5 + 0.3 \times 0.2 \times 0.5 \\ =& 0.5(0.14 + 0.24 + 0.06 + 0.06) = 0.25 \end{aligned}$$

Some shortcuts: we can think of the probability that at least two tests output 1 and then take the complement. In addition, we note that $\Pr[T_3 \mid P] = 0.5$ always which simplifies things.

$$\begin{aligned}\Pr[\neg S \mid \mathit{Spam}] =& 0.5\left(\Pr[T_1 = 0, T_2 = 1 \mid \mathit{Spam}] + \Pr[T_1 = 1, T_2 = 0 \mid \mathit{Spam}] + 2\Pr[T_1 = T_2 = 1 \mid \mathit{Spam}]\right)\\ &= 0.5\left(0.6 \times 0.2 + 0.4 \times 0.8 + 2 \times 0.4 \times 0.2\right) = 0.5 \times 0.6 = 0.3\end{aligned}$$

$$\begin{aligned}\Pr[\neg S \mid \mathit{External}] =& 0.5(\Pr[T_1 = 0, T_2 = 1 \mid \mathit{External}] + \Pr[T_1 = 1, T_2 = 0 \mid \mathit{External}]\\ &+ 2\Pr[T_1 = T_2 = 1 \mid \mathit{External}])\\ =& 0.5\left(0.4 \times 0.6 + 0.6 \times 0.4 + 2 \times 0.6 \times 0.6\right) = 0.5 \times 1.2 = 0.6\end{aligned}$$

$$\begin{aligned}\Pr[\neg S \mid \mathit{Company}] =& 0.5(\Pr[T_1 = 0, T_2 = 1 \mid \mathit{Company}] + \Pr[T_1 = 1, T_2 = 0 \mid \mathit{Company}]\\ &+ 2\Pr[T_1 = T_2 = 1 \mid \mathit{Company}])\\ =& 0.5\left(0.3 \times 0.8 + 0.7 \times 0.2 + 2 \times 0.7 \times 0.8\right) = 0.5 \times 1.5 = 0.75\end{aligned}$$

Putting it together we have

$$\Pr[S] = 0.7 \times 0.1 + 0.4 \times 0.3 + 0.6 \times 0.25 = 0.34.$$

and

$$\Pr[P = \mathit{Spam}, S] = 0.07.$$

Thus, $\Pr[P = \mathit{Spam} \mid S] = 0.07/0.34 \simeq 0.205$.

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 $\textit{follows}(a, b)$, whose input is two users and whose output is `true` iff a follows b; it is assumed that $\textit{follows}(a, a) = \texttt{false}$ 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 $\textit{follows}(k, a, b)$ takes as input a parameter k and two users a and b; it outputs `true` iff a k-follows b. Describe the function $\textit{follows}(k, a, b)$ using first-order logic. Use the function $\textit{follows}(a, b)$ in your description.

   Solution: $\textit{follows}(k, a, b) \iff \exists x_1, \ldots, x_{k-1} : \textit{follows}(a, x_1) \wedge \textit{follows}(x_1, x_2) \wedge \cdots \wedge \textit{follows}(x_{k-2}, x_{k-1}) \wedge \textit{follows}(x_{k-1}, b) \wedge \neg \textit{follows}(k - 1, a, b)$

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:

$$\textit{follows}(C, A)$$
$$\forall x : \textit{follows}(x, A) \Rightarrow x = C$$
$$\textit{follows}(D, C)$$
$$\forall x : \textit{follows}(x, C) \Leftrightarrow \textit{follows}(B, x)$$

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:

$$\textit{follows}(C, A)$$
$$\neg \textit{follows}(x, A) \vee (x = C)$$
$$\textit{follows}(D, C)$$
$$\neg \textit{follows}(y, C) \vee \textit{follows}(B, y)$$
$$\neg \textit{follows}(B, z) \vee \textit{follows}(z, C)$$

The statement we wish to prove is

$$\exists L, M : \mathit{follows}(B, L) \wedge \mathit{follows}(L, M) \wedge \mathit{follows}(M, A),$$

and its negation is

$$\forall \ell, m : \neg \mathit{follows}(B, \ell) \vee \neg \mathit{follows}(\ell, m) \vee \neg \mathit{follows}(m, A).$$

We apply resolution until we arrive at a contradiction. First resolve $\mathit{follows}(C, A)$ and $\neg \mathit{follows}(B, \ell) \vee \neg \mathit{follows}(\ell, m) \vee \neg \mathit{follows}(m, A)$ by setting $m \leftarrow C$ to obtain

$$\neg \mathit{follows}(B, \ell) \vee \neg \mathit{follows}(\ell, C); \theta = [m \leftarrow C]$$

Next, we resolve this with $\mathit{follows}(D, C)$ setting $\ell \leftarrow D$ to obtain

$$\neg \mathit{follows}(B, D); \theta = [m \leftarrow C, \ell \leftarrow D]$$

We resolve this with $\neg \mathit{follows}(y, C) \vee \mathit{follows}(B, y)$ setting $y \leftarrow D$ to obtain

$$\neg \mathit{follows}(D, C).$$

This again gets resolved with $\mathit{follows}(D, C)$ to obtain a contradiction.

Part IV
Adversarial Search

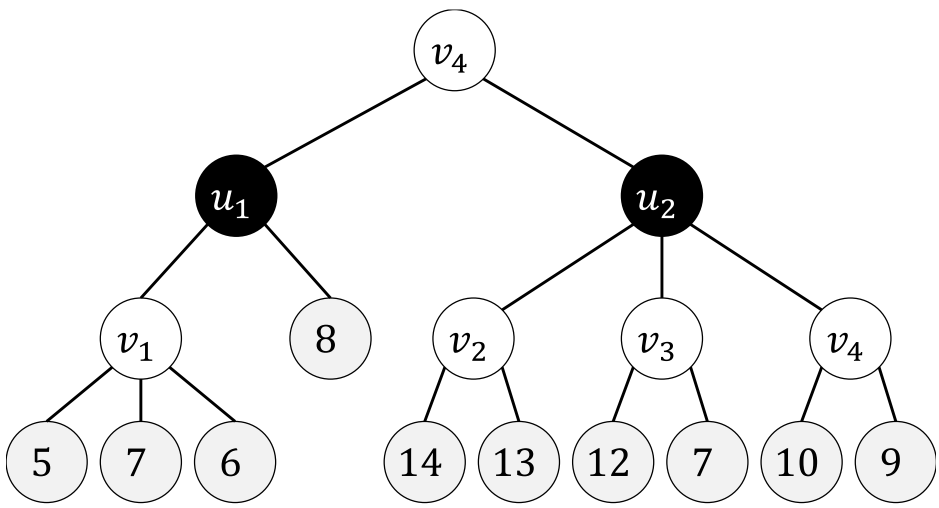
(12 points) Short essay questions. Answer in the space provided on the script.

```
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:



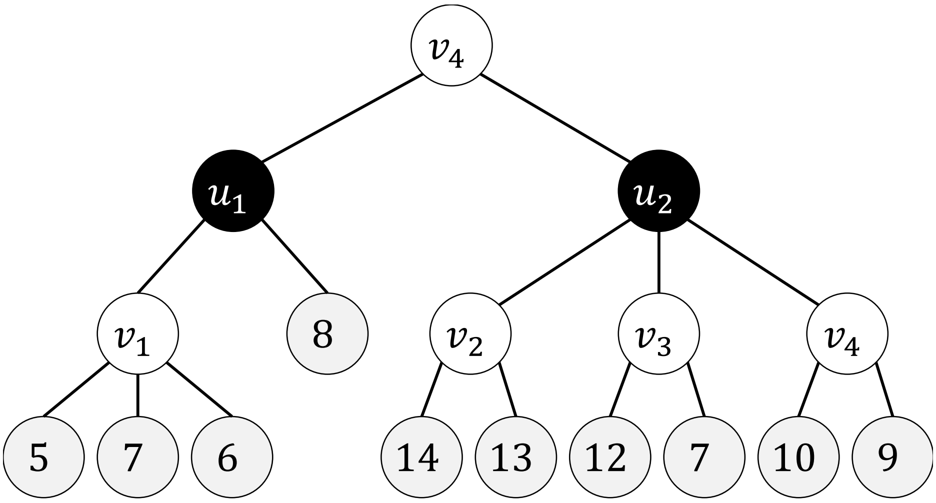
We will prune the edges $(v_2, 14)$, and (u_1, v_1).

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

Solution: 10.

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:



We will not prune any edges.

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*.

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

	True	False	
$A < B$	☐	✓	if $A = B$ then (v_1, C) gets pruned as well.
$D > E$	☐	✓	this is not necessarily true
$D > A$	☐	✓	if $D \leq A$ then we don't check $(u_2, E), (u_2, F)$
$B > E$	☐	✓	this is not necessarily true

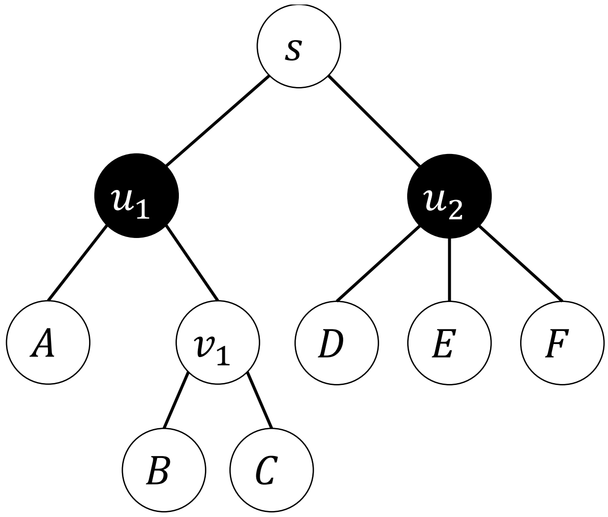


Figure 2: Minimax search tree.

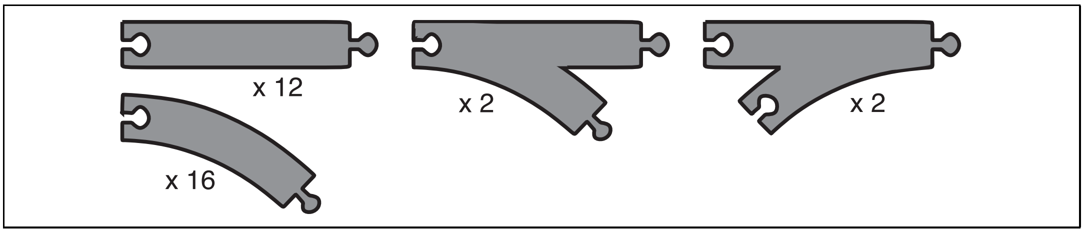


Figure 3: The track pieces in a wooden railway set; each is labeled with the number of copies in the set. Note that curved pieces and "fork" pieces ("switches" or "points") can be flipped over so they can curve in either direction. Each curve subtends at a 45 degree angle.

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)

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.

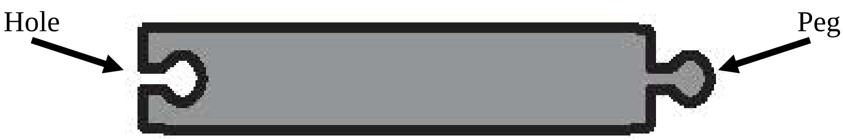


Figure 4: Pegs bulge out; holes bulge in.

Initial State: no pieces placed.

Successor Function: legally connect a piece to a loose end.

Goal Test: All pieces are used, no holes or pegs open.

Step Cost: Uniform cost for all pieces.

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

> **Solution:**
>
> All solutions are in the same depth (number of pieces) and there is a uniform cost to steps, so DFS is the best option.

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:** In every solution we need to connect every peg to a hole; thus in a given position n, we need to close each open peg in n with a piece that has holes. The maximal number of pegs that can be covered by a piece is 2 (using the fork pieces), so we are, at best, going to use a number of pieces equal to $\frac{1}{2}$ the number of open pegs in n in order to reach the goal.

END OF PAPER