Tutorial Letter 102/2/2018

Techniques of Artificial Intelligence

COS3751

Semester 2

School of Computing

IMPORTANT INFORMATION

This tutorial letter contains Assignment 01, Assignment 02 and the self-assessment assignment.

BAR CODE

ASSIGNMENT 01
Due Date: 13 August 2018
UNIQUE ASSIGNMENT NUMBER: 801429

ONLY FOR SEMESTER 2

Study material: Chapters 1 through 4. You may skip sections 4.2 and 4.5.
Important: When we use the phrase 'formally define', we are looking for a formal definition using some form of formal notation, and not simply an English description or definition. For example: 'Define the initial state for an agent in Johannesburg.' Answer: In(Johannesburg). 'Define the actions available to this agent given that the agent simply moves between major metropolitan areas.' Answer: Actions(In(Johannesburg)) = $\{$Go(Bloemontein), Go(Durban),$\ldots\}$. When we want an English definition, we will explicitly ask for it.

Question 1

(1.1)

Consider an agent that is designed to deliver emergency supplies to miners that are trapped in a mine. The agent is small enough to navigate through most small openings, but can burrow through soft rock when necessary, because there is a lot of sand and dirt in the shafts, the agent may slip occasionally when attempting to move. Once the agent detects that it has delivered medical supplies to a survivor, it returns to the rescue services team. These agents are typically only deployed once it has been determined that no more shifts in the earth are possible, and the survivors' location has been pinned down (survivors won't move around).
Choose from among the following environmental descriptions, and justify your choices in each case.

(a) Is the environment deterministic, or stochastic?

(b) Is the environment discrete, or continuous?

(c) Is the environment static, or dynamic?

(1.2) Is this an example of a model-based, or reflex-based agent? Justify your answer.

Question 2

Three jackals and three coyotes come to a river. At the side of the river they also find a small boat. The boat can only hold two (at most) of the animals at the same time. They would like to cross the river but the coyotes are fearful that should there be more jackals than coyotes on one side of the river, they will quickly be over-powered and killed.
To avoid a long discussion, the jackals agree to a protocol for crossing the river in such a way that they will never outnumber the coyotes. That is, the number of jackals on a side will always be less than or equal to the number of jackals on the same side of the river.

The boat, being small, can only take one or two of the animals at a time. It can also not travel by itself – there has to be at least one animal in the boat for it to cross from side to side.
Suppose the state is represented as a three-tuple $S : \mathbb{N} \times \mathbb{N} \times L$, with $L \in \{left, right\}$. We don't store the right-hand side values, since we can simply assume that if a coyote or a jackal is not on the left side, they must be on the right. As coyotes and jackals move from the one side of the river to the other, the number of coyotes/jackals are simply added/subtracted from the state representation. The 'left'/'right' entry indicates the side on the river the boat is on. For example $S = (3, 3, left)$ means that there are three coyotes and three jackals, as well as the boat on the left side. Should the boat take one jackal across, we end up with $S' = (2, 3, right)$.

(2.1) Define the goal state for this problem.

(2.2) Suppose the current state is $S = (1, 2, right)$. Define the applicable actions for this state. (Hint: you will first need to define the possible actions that are available to the agent in general, then define the applicable actions for this state.)

(2.3) Provide the transition model for the search, but limit your answer to the case where $S = (1, 2, right)$.

Question 3

(3.1) Explain how a Depth First Search (DFS) ensures that it always expands the deepest node first. Provide an example to aid your discussion.

(3.2) Explain when one might want to choose DFS over Breadth First Search (BFS). (Hint: In which version of the algorithm does one have an advantage over the other?)

Question 4

Consider the search tree in Figure 1. Show the order in which the nodes will be **expanded** at each level (start with level 0 and continue until the goal test is successful), given that IDS is used. Assume the goal node is *B*, and that nodes are expanded from left to right (M is expanded before E and so on). (Hint: make sure you understand the difference between expansion and generation, and also that you understand when goal checks occur.)

Question 5

Consider the graph provided in Figure 2, and answer the questions that follow. The step cost between nodes is provided next to the edges.

(5.1) Prove that a consistent heuristic is admissable. (Hint: if you can prove that admissibility holds for nodes 1 step away from the goal, then you can prove that nodes *k*-steps away from the goal is admissible).

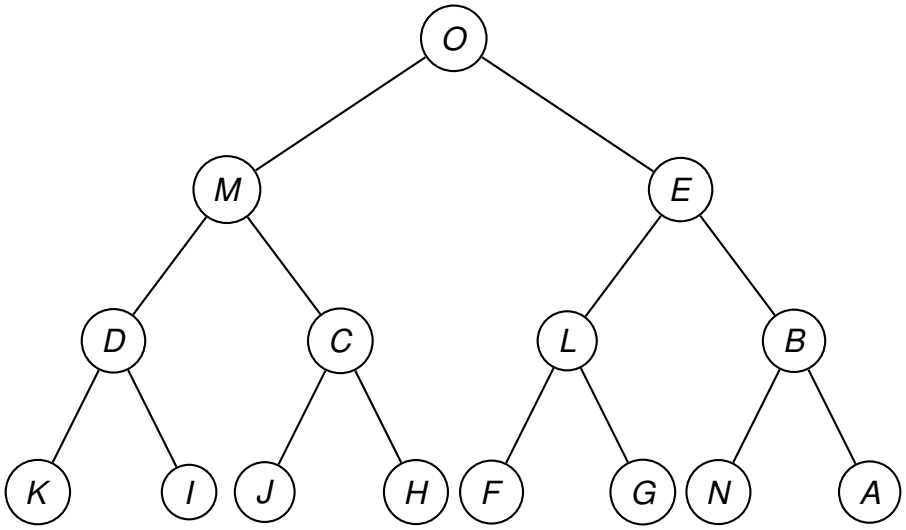

Figure 1: Search Tree (Iterative Deepening Search (IDS))

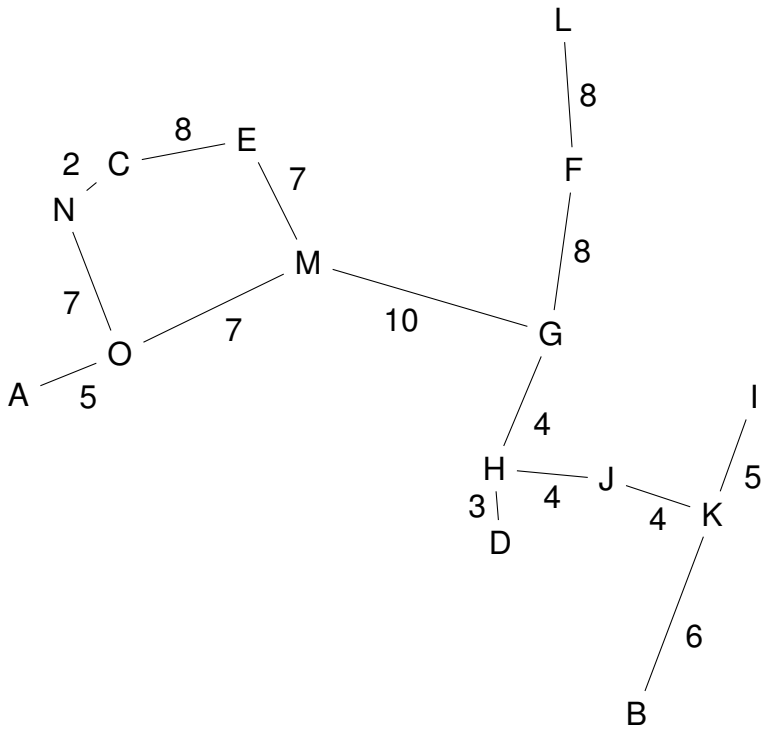

Figure 2: Search

(5.2) Perform a Uniform Cost Search (UCS) on the graph. The start node is *N* and the goal node is *F*. Provide a step-wise explanation of the search as it progresses. At each step, provide the frontier, and show which node is selected for expansion. Provide the final path from the start to the goal.
Use the following format for your answer (step 1 has been completed below):

Step	Node expanded	Frontier
1		N($\hat{g}$ = 0)
...	...	...

Remember: the first step of a well-written search algorithm is always to generate the start node (i.e. place the start node in the frontier). Step two should begin with expanding the node in the frontier with the smallest $\hat{g}$ value.

(5.3) Perform an A^* search on the graph in Figure 2. The start node is G and the goal node is C. Provide a step-wise explanation of the search as it progresses. At each step, provide the frontier, and show which node is selected for expansion. Provide the final path fromhe start to the goal. Use table 1 for the $\hat{h}$ values for each node in the graph. Use the following format for your answer (step 1 has been completed below):

Step	Node expanded	Frontier
1		G($\hat{g}$ = 0, $\hat{h}$ = 6, $\hat{f}$ = 6)
...	...	...

Remember: the first step of a well-written search algorithm is always to generate the start node (i.e. place the start node in the frontier). Step two should begin with expanding the node in the frontier with the smallest $\hat{f}$ value.

Node	Estimated distance to goal
A	3
B	10
C	0
D	7
E	2
F	6
G	6
H	6
I	9
J	8
K	9
L	6
M	3
N	1
O	3

Table 1: Estimated distance to goal for A^* search.

Question 6

Consider the four-queens problem. We would like to populate a 4×4 board with 4 queens. A queen is a piece (from chess) that can capture any other piece on the same diagonal, or in the same column or row.
Using local search algorithms, we can perform a search as follows:

- Initialise a board with queens placed randomly,
- Define an objective function to aid the local search,
- Generate a successor state from the current state by moving one queen randomly either up or down one square (in the same column),
- An optimum (minimum/maximum) is considered a goal.

Consider the following random start state (we represent the state graphically to aid your effort, and you are welcome to provide graphical representations of successor states if asked):

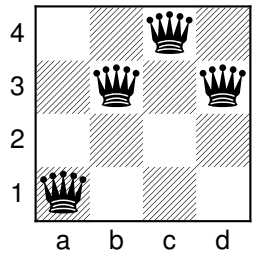

Figure 3: Four Queens Initial State

(6.1) Provide a good objective function that can be used as a minimizer (i.e. a global minimum, a loss function) for the problem.
Hint:

1. An objective function (also called either a loss, or reward function) is a function that maps the current state to some linear value which can be used to judge the *fitness* or *goodness* of the state. When we want to avoid loss, we define an objective function with respect to loss, and we try to minimize loss by minimising the evaluation of the state using the objective function (i.e. we look for a state that results in the smallest value when evaluated using the objective function). When we want reward, we define an objective function with respect to reward and we try to maximize the objective function.
2. First write down the function in plain English, and then provide the mathematical equation for it.
3. Use the rows, columns, and diagonals in the state as part of your equation.

(6.2) Now that you have defined your objective function, use it to evaluate the start state as provided in Figure 3. Show your calculations.

(6.3) A hill-climb/hill-descent local search generates a series of successors, and takes the best one from among them (using the objective function). Beginning from the initial state in Figure 3, provide two successors, one that is worse, and one that is an improvement. Show why the successor is better, or worse by using the objective function to evaluate it. Remember: generate all possible successors, and choose the better/worse from among them.
Remember, a successor state is generated by moving one queen, one row up or down.

(6.4) From the previous question, take the 'better' state you identified. Generate all possible successors (remember: a successor is generated by moving one queen, one row up or down).
Are there any improvements (show your calculations for each state)?

(6.5) What type of topographical feature is most likely being created by the situation in the previous question? (Provide a rough sketch of this type of feature in a one dimensional state-space landscape.) Can this problem be solved from this state (using hill-descent search)? Justify your answer.

ASSIGNMENT 02
Due Date: 10 September 2017
UNIQUE ASSIGNMENT NUMBER: 709589

ONLY FOR SEMESTER 2

Study material: Chapters 5, 6, 7, and 8. You may skip sections 5.5, 5.6, and 5.7, 7.6 and 8.4.

Question 1

(1.1) Clearly explain what an evaluation function is, and why it is used during adversarial searches.

(1.2) Is the ideal strategy only available if we have perfect information? Explain your answer.

(1.3) Explain how forward pruning works. Provide at least one approach to forward pruning in your explanation, as well as a problem that may be encountered with forward pruning.

(1.4) Does the order in which nodes are examined in minimax matter? Explain your answer.

Question 2

Consider Figure 4 and answer the questions that follow. (The utility value of the leaf nodes are provided below the in brackets in the leaf node.)

(2.1) Provide the minimax values for all the nodes.

(2.2) Which move should MAX make? Explain your answer.

(2.3) Write down the α/β values for all the nodes (except the leaf nodes) if alpha/beta pruning is applied to the tree.

(2.4) Write down which nodes were cut and what type of cut was made in each case (alpha, or beta).

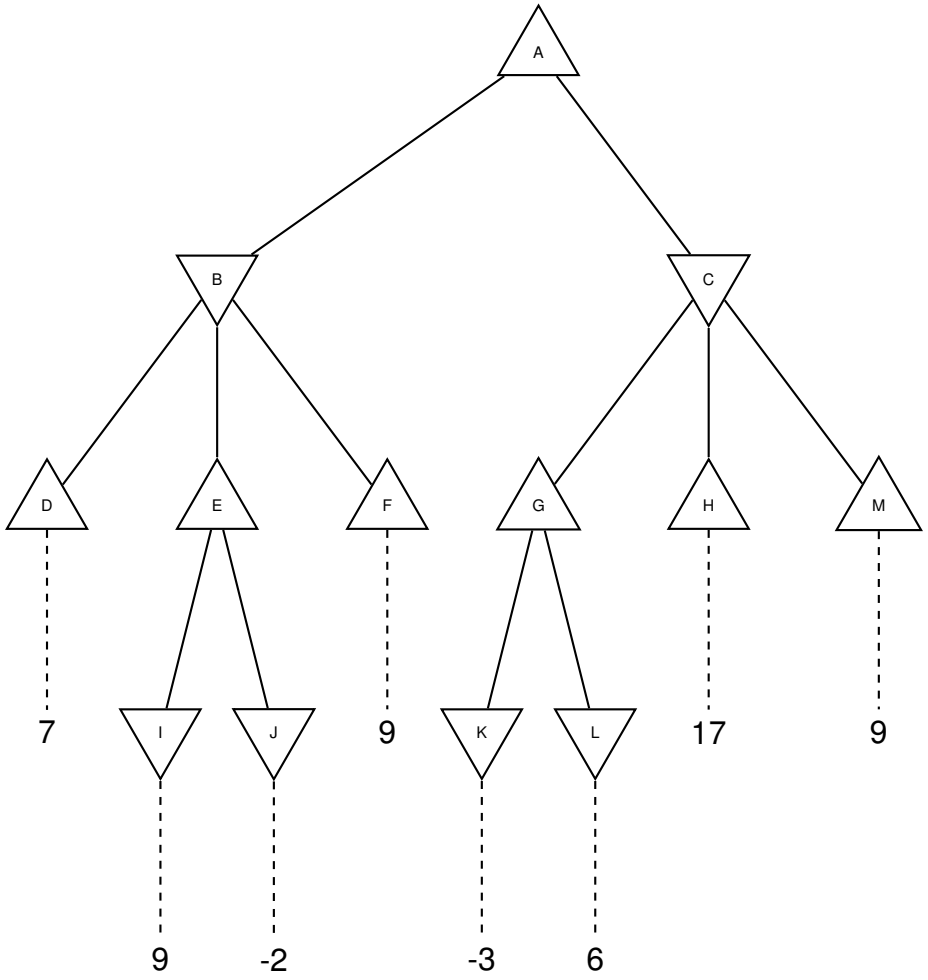

Figure 4: Minimax, alpha/beta

Question 3

Consider the *subtraction* game: two players (A and B) take turns removing items from a heap (just one heap). Each player may remove either one, two, or three items from the heap. The heap starts off with 12 items, and player A moves first. The objective of the game is to be the last player to remove items from the heap. That is, if it is your turn to move, and the heap is empty, you've lost the game[1].
The Initial state for the game is $(A, 12)$ indicating that it is A's turn to move and there are 12 items on the heap. In general then, a state is represented as (P, n) where $P \in \{A, B\}$, and $0 \leq n \leq 12$.
The evaluation function for non-terminal nodes is a simple threshold function defined as:

$$eval(S) = \begin{cases} -1 & \text{if } S.n \ mod \ (k+1) = 0, \\ 1 & \text{otherwise} \end{cases}$$

$S.n$ refers to the number of items left in the heap in state S, and *mod* is the integer modulo function. k is the maximum number of items that may be removed on a turn, so in this case $k = 3$.

[1] visit https://youtu.be/aonCsvi0LKc to see how the game is played with 21 items.

(3.1) Draw the entire game tree, starting from the initial state, down to depth two (the initial state is at depth 0), and provide the evaluation for each state at depth 2.

(3.2) Using the minimax algorithm, provide the backed-up values for states at depth 1 and 0. (**Hint**: Since you don't have the entire game tree, and thus no terminal states, you cannot use a utility value to calculate the backed up values. However, you do have an evaluation function.)

(3.3) Which node(s) would not have been evaluated at depth 2 if alpha/beta pruning was employed?

Question 4

Answer the following questions on Constraint Satisfaction Problems (CSPs).

(4.1) Define the Least Constraining Value (LCV) heuristic.

(4.2) Explain why establishing strong k-consistency is a problem.

(4.3) Define the *degree* heuristic.

(4.4) If no legal assignments for a variable remain during a solution to a CSP, does it mean that the algorithm will be able to find a solution by simply backtracking? Explain your answer.

(4.5) Explain what forward checking for a CSP is.

Question 5

CSPs are especially useful when trying to solve scheduling problems.
Consider the problem of determining how to assign aircraft in a fleet to particular flights. A flight (or a leg) is simply a scheduled transference of passengers from a departing airport to a destination airport (with no stops inbetween). In order to transfer passengers between the source and destination airport a flight needs an aircraft.
Additionally, aircraft have a minimum turn-around time of 30 minutes. That is, an aircraft cannot arrive, and then simply take of again. It has to taxi to the gate, passengers should debark, the aircraft should be cleaned, and the new passengers should embark. Only then can the aircraft taxi to the assigned runway and then take-off.
The airline in question operates a small fleet of short-haul aircraft, and all flights are only for 9 or fewer passengers.
Consider the following flight schedule:

Flight	Departs	Lands
QQ002	9:15	10:45
QQ002	14:00	14:45
QQ004	12:15	13:15
QQ008	10:45	11:45
QQ016	09:30	10:15
QQ032	11:15	13:30
QQ064	13:00	13:45
QQ128	13:15	14:15
QQ256	13:30	14:45
QQ512	10:00	10:45
QQ512	14:00	14:45

You can assume that departure happens from the same airport (you don't have to worry about the location of the aircraft).
The airline owns several Cessna 208 Grand Caravans (208B), which are:

1. XAX-344,
2. XAX-254,
3. XAX-983,
4. XAX-124
5. XAX-888

(5.1) The first step when representing a problem as a CSP is to define the variables. The variables are those elements in the problem that get assigned something else from the problem. Read the above problem description carefully to determine which elements from the problem are variables. Now provide these variables for the problem. Remember to use the correct notation!
Hint: In most cases the variables are those elements from the problem which will require the use of a limited item available.

(5.2) Once the variables have been defined, we proceed to defining the domain for each variable. That is, the 'value' that each variable can take. Define the domain for each variable in the CSP.

(5.3) The constraints determine the 'restrictions' placed on variables. Define the constraints for the variables in the CSP.

(5.4) Provide the constraint graph for this problem.

(5.5) Provide the solution to the problem. Use the Minimum Remaining Values (MRV) heuristic, and at each step establish arc-consistency for the variables. Show how the solution is calculated in a step by step fashion. Show the variables and their assigned values as your final answer!

Question 6

Consider the following statements from a knowledge base *KB*:

a. $(P \wedge Q) \Rightarrow R$

b. $T \Rightarrow Q$

c. $W \Rightarrow P$

d. $\neg R$

Prove that $KB \models (W \Rightarrow \neg T)$ using resolution refutation. (Hint: Convert the statements in *KB* to clause form first. Note that $\neg(W \Rightarrow \neg T) \equiv \neg(\neg W \vee \neg T) \equiv W \wedge T$.)

Here is a short example of what your resolution proof should look like:

1.	$\neg A \vee B$	(premise)
2.	A	(premise)
3.	$\neg B$	(negation of goal)
4.	B	(1 & 2)
5.	$\emptyset$	(3 & 4)

Question 7

Consider a vocabulary with the following symbols:

$Customer(p1, p2)$:	Predicate. Person $p1$ is a customer of person $p2$.
$Boss(p1, p2)$:	Predicate. Person $p1$ is a boss of person $p2$.
$Doctor(p)$:	Predicate. Person p is a doctor.
$Surgeon(p)$:	Predicate. Person p is a surgeon.
$Lawyer(p)$:	Predicate. Person p is a lawyer.
$Actor(p)$:	Predicate. Person p is an actor.
$Emily, Joe$:	Constants denoting people.

Use these symbols to write the following assertions in first-order logic:

a. Emily is either a surgeon or a lawyer (but not both).

b. All surgeons are doctors.

c. Joe does not have a lawyer (i.e. he is not the customer of any lawyer).

d. There exists a lawyer all of whose customers are doctors.

e. Every surgeon has a lawyer.

SELF ASSESSMENT ASSIGNMENT 01

ALL STUDENTS

Study material: Chapters 9, and 18. You may skip sections 9.3, and 9.4. You only need to study 18.1, 18.2, and 18.3.

Question 1

Convert the following First-Order Logic (FOL) sentence to clause form:

$$\forall y\ (\forall x\ P(x) \Rightarrow \exists x\ (\forall z\ Q(x, z) \vee \forall z\ R(x, y, z)))$$

Question 2

Consider the following English statements:

a. Anyone who passes their history exam and who wins the lottery is happy.

b. Anyone who studies or is lucky can pass their exams.

c. John did not study.

d. John is lucky.

e. Anyone who is lucky wins the lottery.

(2.1) Provide a vocabulary for the statements.

(2.2) Translate the above English sentences to FOL statements using the vocabulary you defined above.

(2.3) Convert the FOL statements obtained in 2.2 into clause form.

(2.4) Use resolution refutation to prove that John is happy.

Question 3

(3.1) Convert the Boolean function in Table 2 into a decision tree:

(3.2) When we construct a decision tree without the benefit of gain values, the order in which we evaluate the variables is important. Why?

x_1	x_2	x_3	f(x_1, x_2, x_3)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Table 2: Boolean function table

Question 4

The National Credit Act introduced in South Africa in 2007 places more responsibility on a bank to determine whether the loan applicant will be able to afford it. *Blue Bank* has a table of information on 14 loan applications they have received in the past.

Use the information in this table to construct a decision tree that will assist the bank in determining the RISK associated with a new loan application.

No.	Credit history	Debt	Collateral	Income	RISK
1	Bad	High	No	$<$ R15k	High
2	unknown	High	No	R15k - R35k	High
3	unknown	Low	No	R15k - R35k	Medium
4	unknown	Low	No	$<$ R15k	High
5	unknown	Low	No	$>$ R35k	Low
6	unknown	Low	Yes	$>$ R35k	Low
7	Bad	Low	No	$<$ R15k	High
8	Bad	Low	Yes	$>$ R35k	Medium
9	Good	Low	No	$>$ R35k	Low
10	Good	High	Yes	$>$ R35k	Low
11	Good	High	No	$<$ R15k	High
12	Good	High	No	R15k - R35k	Medium
13	Good	High	No	$>$ R35k	Low
14	Bad	High	No	R15k - R35k	High

Table 3: Risk information table