NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

FINAL ASSESSMENT FOR
Semester 1 AY2022/2023

CS3243: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

SOLUTIONS

November 28, 2022 Time Allowed: 120 Minutes

INSTRUCTIONS TO CANDIDATES

1. This assessment contains FOUR (4) questions. All the questions are worth a total of 60 MARKS. It is set for a total duration of 120 MINUTES. You are to complete all 4 questions.
2. This is a CLOSED BOOK assessment. However, you may reference a SINGLE DOUBLE-SIDED A4 CHEAT SHEET.
3. You are allowed to use NUS APPROVED CALCULATORS.
4. If something is unclear, solve the question under a reasonable assumption. State your assumption clearly in the answer. If you must seek a clarification, the invigilators will only answers questions with Yes/No/No Comment answers.
5. You may not communicate with anyone other than invigilators in any way.

STUDENT NUMBER: ____

EXAMINER'S USE ONLY		
Question	Mark	Score
1	21	
2	11	
3	13	
4	15	
TOTAL	60	

1a. [1 mark] Let G denote a search problem, which is to be solved using the Uniform Cost Search (UCS) algorithm. Let G' denote a variant of G, where each step cost (i.e., in G') has a constant value, $c \in \mathbb{Z}^+$, added to the original step cost (i.e., added to the corresponding step cost in G). Consider the following statements.

- S_a: The UCS algorithm is guaranteed to return the same optimal path for G and G'.
- S_b: The UCS algorithm always returns different paths for G and G'.
- S_c: Any path the UCS algorithm returns for G' is an optimal path on G.

Specify which among the statements, S_a, S_b, and S_c, is/are true. If none are true, simply specify "none".

Solution: None

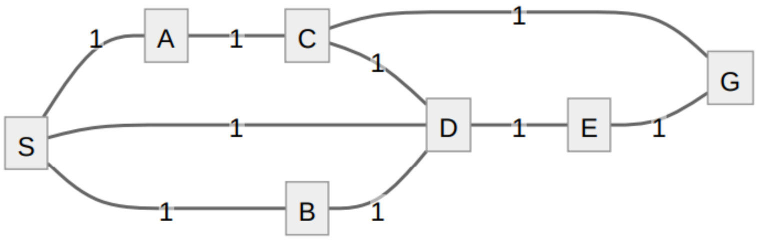
1b. [1 mark] Given a collection of admissible heuristics (for the A* search algorithm), $H = \{h_1, h_2, \ldots, h_m\}$, and a corresponding collection of weights, $W = \{w_1, w_2, \ldots, w_m\}$, such that $\forall w_i \in W$, $w_i \in \mathbb{R}$, consider the following new heuristics.

- $h_a(n) = \sum_{h_i \in H} h_i(n)$
- $h_b(n) = \frac{1}{10} \sum_{h_i \in H} h_i(n)$
- $h_c(n) = \sum_{i=1}^{m} w_i . h_i(n)$

Specify which among the new heuristics, h_a, h_b, and h_c, are admissible heuristics (again, for the A* search algorithm). If none are admissible, simply specify "none".

Solution: None

1c. Consider the following undirected graph and the accompanying heuristic h.



n	$h(n)$
S	3
A	2
B	3
C	1
D	2
E	1
G	0

Assuming that all algorithms use the same tiebreaking strategy, apply the graph and heuristic above to the **following three questions** (i.e., to **Part (i)**, **Part (ii)**, and **Part (iii)** below).

(i) [1 mark] Assume the use of the A* search algorithm using a graph-search implementation with late-goal testing. Which one statement among the following is true?

- S_a: Graph search version 3 returns the same path as version 1.
- S_b: Graph search version 3 returns the optimal path, but version 1 does not.
- S_c: Graph search version 1 returns the optimal path, but version 3 does not.
- S_d: There is not enough information; none of the above are true.

Solution: ***S_a***

(ii) [1 mark] Which one among the following statements about the heuristic, *h*, is true?

- S_a: *h* is consistent and is admissible.
- S_b: *h* is consistent but is not admissible.
- S_c: *h* is not consistent but is admissible.
- S_d: *h* is not consistent and is not admissible.

Solution: ***S_a***

(iii) [1 mark] Consider the Uniform Cost Search (UCS) algorithm implemented using graph search with late-goal testing. Running UCS, which of the following statements is/are true?

- S_a: UCS returns the optimal path *S-D-C-G*.
- S_b: UCS returns the optimal path *S-D-E-G*.
- S_c: UCS returns the same path as A* search using graph search version 3.
- S_d: There is not enough information; none of the above are true.

Solution: ***S_c***

1d. [4 marks] Prove or disprove the following statement. Given a consistent heuristic, h_1, and an admissible but inconsistent heuristic, h_2, A* search using h_1 will expand no more nodes than A* search using h_2.

Solution: False

Consider a counter example where there are 2 paths from start to goal. For example, path S > A > G and path S > B > G. Path cost S > A = 3, S > B = 1, A > G = 5, B > G = 10.

Denote h_1 as a heuristic where $h_1(n) = 0$ for n in {S, A, B, G}; h_1 is thus consistent. A* adopting h_1 on the given graph will expand every node.

Next, denote h_2 as a heuristic where $h_2(S) = 8$, $h_2(A) = 0$, $h_2(B) = 10$, and $h_2(G) = 0$. Notice that h_2 is inconsistent since $h_2(S) > c(S,A) + h_2(A)$. However, h_2 is admissible since all heuristic values under h_2 are less than the actual path costs.

Since A* adopting h_2 will only expand nodes S, A and G, we have a counterexample to the given statement.

1e. [6 marks] Mike is a founder of a web development start-up. He is looking to recruit three new staff – i.e., such that there will be a total of four in his start-up, including himself. The staff must fulfil the following roles: two JavaScript Programmers, two UX/UI Designers, one Marketing Guru, one Database Administrator, and one Systems Engineer. Assume that if a person possesses two skills, he/she can take on two roles in the company.

Mike has narrowed his recruitment options down to the following people.

Name	***Skills***
Mike (founder)	JavaScript
Samuel	JavaScript and UX/UI
Jonathan	Marketing and UX/UI
Tom	UX/UI and Systems
Susan	JavaScript and Database
Linda	Marketing and UX/UI
Wayne	Systems and JavaScript
Jimmy	Marketing and UX/UI

Model this scenario as a Constraint Satisfaction Problem (CSP). Specifically, state the variables, domains, and constraints.

Solution:

There are several ways to model the domain. Variables can either be the people, the jobs, or the skills.

For the job-based variables, there are three variables, J_1, J_2 and J_3 (with J_4 already filled) representing the three openings. The domains of all three variables are initially all the people. We can then pose the following constraints in terms of the skills of people who fill those roles.

- Number(Job, Javascript) ≥ 2
- Number(Job, UX/UI) ≥ 2
- Number(Job, Marketing) ≥ 1
- Number(Job, Database) ≥ 1
- Number(Job, Systems) ≥ 1

There is also a further constraint associated to the fact that assigning someone to one job means that this person cannot be assigned to another job opening. This constraint is as follows.

- $J_1 \neq J_2$, $J_2 \neq J_3$, $J_1 \neq J_3$

Alternatively, we can also think of the skill spots as variables – i.e., Javascript_1, Javascript_2, UX_UI_1, UX_UI_2, Marketing, Database, and Systems.

For this, we would have initial domains as follows.

- Javascript_1: Mike
- Javascript_2: Samuel, Susan, Wayne
- UX_UI_1: Samuel, Jonathan, Tom, Linda, Jimmy
- UX_UI_2: Samuel, Jonathan, Tom, Linda, Jimmy
- Marketing: Jonathan, Linda, Jimmy
- Database: Susan
- Systems: Tom, Wayne

We would additionally need constraints that say that one person cannot fill both assignments of Javascript, and that we could only hire 3 additional people.

1f. [4 marks] Continuing from **Part (f)** above, suppose that Mike decides to make Tom a co-founder. Mike and Tom discover that all the prospective staff absolutely refuse to abandon their favourite platforms. Further, given that the start-up can only afford two single-boot workstations, it may thus only operate with two distinct Operating Systems (OS). The table below shows the relevant skills (unchanged from **Part (f)**) and the OS preferences.

Name	***Skills***	***OS***
Mike (founder)	JavaScript	Windows
Samuel	JavaScript and UX/UI	Windows
Jonathan	Marketing and UX/UI	Windows
Tom (co-founder)	UX/UI and Systems	FreeBSD
Susan	JavaScript and Database	FreeBSD
Linda	Marketing and UX/UI	Linux
Wayne	Systems and JavaScript	Linux
Jimmy	Marketing and UX/UI	Windows

Given this newly added requirement, enforce node consistency and determine the domain values once node consistency has been achieved – i.e., specify the domains after all unary constraints are satisfied. Next, perform forward-checking and list the resultant domains. You are to assume that the above are done as pre-processing steps.

Note that there will be **NO Error Carried Forward** considered.

Solution:

We will model this using the skills formulation, noting that we have already hired Tom and Mike, so that means that we cannot hire either Linda or Wayne (node consistency), which leaves the following.

- Javascript_1: **Mike**
- Javascript_2: Samuel, Susan
- UX_UI_1: **Tom**
- UX_UI_2: Samuel, Jonathan, Jimmy
- Marketing: Jonathan, Jimmy
- Database: Susan
- Systems: **Tom**

Notice that we must hire Susan to satisfy the Database requirement, and as such, there is only one position left to fill, so we cannot hire Samuel (arc consistency). So, we must hire either Jonathan or Jimmy to satisfy both UX/UI and Marketing. This leaves us with the following.

- Javascript_1: **Mike**
- Javascript_2: **Susan**
- UX_UI_1: **Tom**
- UX_UI_2: Jonathan, Jimmy
- Marketing: Jonathan, Jimmy
- Database: **Susan**
- Systems: **Tom**

1g. [2 marks] Determine if the following statement is true or false and provide a clear and concise rationale. Once arc consistency is enforced as a pre-processing step, forward checking can be used during backtracking search to maintain arc consistency for all variables.

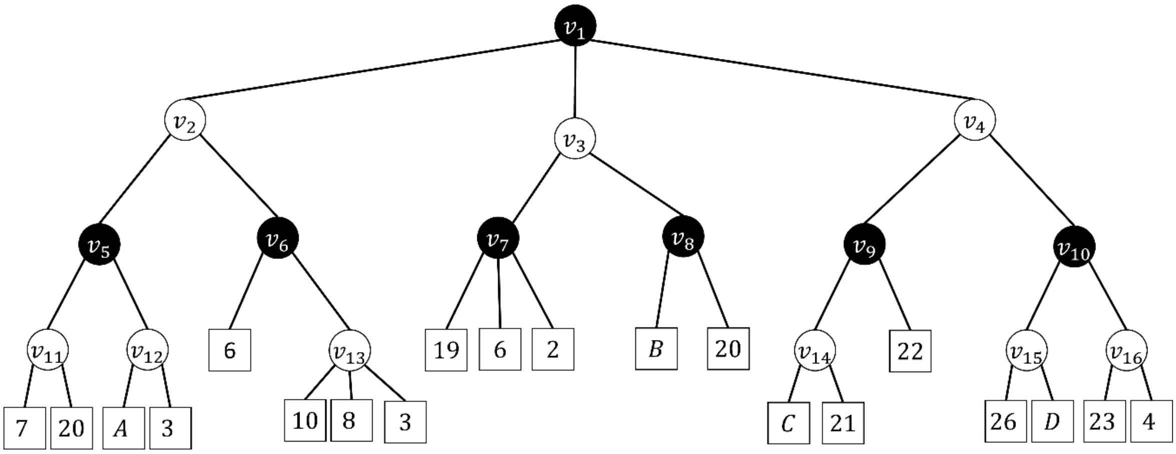
Solution: False

Consider the following counterexample with 3 variables, A, B, and C, where $D_A = \{2,3\}$, $D_B = \{3,4\}$, $D_C = \{3,4\}$, with the constraints, A = B - 1, and B ≠ C.

Notice that the domains are arc consistent.

When C is assigned to be 3, forward checking reduces B to {4}. However, ARC (A,B) is not arc-consistent as the value in {2} in D_A does not have a corresponding value in B that satisfies the constraint A = B – 1.

2a. Consider the following game tree in an adversarial search problem.



Circular black nodes correspond to the MAX player while circular white nodes correspond to the MIN player. The utility values at the terminal states are specified in the square leaf nodes. Assume that all utility values are stated in terms of the MAX player's utility and that the utility values are in the interval [-100, 100].

(i) [6 marks] Given that the Alpha-Beta Pruning algorithm is executed on the above tree from **left to right**, determine the largest range of values for the variables *A*, *B*, *C*, and *D* such that no arcs are pruned.

Solution:

$8 \le A \le 100$ $20 \le C \le 100$

$-100 \le B \le 18$ $-100 \le D \le 21$

(ii) [1 mark] Assuming that no nodes are pruned from the given game tree, determine the utility value at the root based on the execution of the Alpha-Beta Pruning algorithm.

Solution:

21 or D.

2b. [4 marks] Consider the following function.

$$F(x) = ax + b, \text{ where } a > 0, x \in \mathbb{R}$$

Prove that when F is applied to each utility value, x, in any Minimax search tree, the nodes pruned by the Alpha-Beta Pruning algorithm will remain unchanged (relative to the original utility values).

Solution: True

Given any two utility values, u_1 and u_2, if $u_1 > u_2$, $F(u_1) > F(u_2)$ since $F(x) = ax + b$, and $a > 0$.

Proof by contradiction:

- Suppose $u_1 > u_2$, but $F(u_1) \leq F(u_2)$; i.e., $u_1 - u_2 > 0$ but $F(u_1) - F(u_2) \leq 0$.
- However, we have $F(u_1) - F(u_2) = (a.u_1 + b) - (a.u_2 + b) = a(u_1 - u_2)$.
- With $u_1 - u_2 > 0$ and $a > 0$, $a(u_1 - u_2) > 0$, which is a contradiction.

When executing the α-β pruning algorithm, since α and β in the are only updated with larger and smaller utility values respectively, the values propagated must be from the original terminal nodes since $(u_1 > u_2) \Rightarrow (F(u_1) > F(u_2))$. This means that any node that was previously pruned (i.e., when satisfying $u \geq \beta$ at MAX nodes, and $u \leq \alpha$ at MIN nodes) must also be pruned now since the ordering the utility values have not changed under F.

3. Alice, Beth, Cathy, and Diana have just had lunch at a restaurant. Each of the four ladies ordered one of two possible lunch sets: *Lunch Set X* or *Lunch Set Y*.

The waiter who served the four ladies reports the following.

- Alice and Beth had different lunch sets.
- Beth and Cathy had different lunch sets.
- Cathy and Diana had different lunch sets.
- Alice, Cathy, and Diana had exactly two of *Lunch Set X*.

Apart from the above statements, the waiter does not recall who exactly had which lunch set.

We denote the following.

- *A*: Alice had *Lunch Set X*
- *B*: Beth had *Lunch Set X*
- *C*: Cathy had *Lunch Set X*
- *D*: Diana had *Lunch Set X*

Consequently, note that negation of any of the above literals refers to that person having *Lunch Set Y* instead – e.g., ¬A refers to Alice having *Lunch Set Y*.

Answer the **following three questions** (i.e., **Part (i)**, **Part (ii)**, and **Part (iii)**) based on the above context. Note that there will be **NO Error Carried Forward** considered.

(i) [4 marks] Define the knowledge base (KB) using the four statements given by the waiter. Specify the KB using propositional logic and the literals *A*, *B*, *C*, and *D*. **Do not convert the KB into Conjunctive Normal Form (CNF) for this part**.

Solution:

S_1: $(A \land \neg B) \lor (\neg A \land B)$; i.e., $A \Leftrightarrow \neg B$
S_2: $(B \land \neg C) \lor (\neg B \land C)$; i.e., $B \Leftrightarrow \neg C$
S_3: $(C \land \neg D) \lor (\neg C \land D)$; i.e., $C \Leftrightarrow \neg D$
S_4: $(A \land C \land \neg D) \lor (A \land \neg C \land D) \lor (\neg A \land C \land D)$ // disjunction of consistent states

However, on S4, it should be noted that it would probably be better to consider the negation over the disjunction of all inconsistent states, which would result in one of the following instead.

S_{4a}: $\neg((A \land C \land D) \lor (A \land \neg C \land \neg D) \lor (\neg A \land C \land \neg D) \lor (\neg A \land \neg C \land D) \lor (\neg A \land \neg C \land \neg D))$
S_{4b}: $((A \land C) \Leftrightarrow \neg D) \land ((A \land D) \Leftrightarrow \neg C) \land ((C \land D) \Leftrightarrow \neg A)$
S_{4c}: $((A \land C) \lor (A \land D) \lor (C \land D)) \land (\neg A \lor \neg C \lor \neg D)$

(ii) [6 marks] Suppose that we are given the query, α: "Did Beth have Lunch Set Y?" Assuming that we wish to prove that KB ⊨ α via resolution, update the KB accordingly, and then convert the KB such that it is fully represented in CNF.

Solution:

KB (over S_1, S_2, and S_3):
(S_1) R_1: A ∨ B
(S_1) R_2: ¬A ∨ ¬B
(S_2) R_3: B ∨ C
(S_2) R_4: ¬B ∨ ¬C
(S_3) R_5: C ∨ D // Also from (S_{4b})
(S_3) R_6: ¬C ∨ ¬D

KB (over S_4):
(S_{4b}) R_{7a}: A ∨ C
(S_{4b}) R_{8a}: A ∨ D
(S_{4b}) R_{9a}: ¬A ∨ ¬C ∨ ¬D

Or alternatively, KB (over S_4):
(S_{4a}) R_{7b}: A ∨ C ∨ D
(S_{4a}) R_{8b}: ¬A ∨ C ∨ D
(S_{4a}) R_{9b}: A ∨ ¬C ∨ D
(S_{4a}) R_{10b}: A ∨ C ∨ ¬D
(S_{4a}) R_{11b}: ¬A ∨ ¬C ∨ ¬D

And finally adding the negated query:
(¬α) R_{12}: B

(iii) [3 marks] Apply the resolution algorithm to prove or disprove the query specified in **Part (ii)**.

Solution:

- R_{13}: ¬C // resolve R_4, R_{12}
- R_{14}: A // resolve R_{7a}, R_{13}
- R_{15}: ¬A // resolve R_2, R_{12}
- R_{16}: ∅ // resolve R_{14}, R_{15}

4. You work in an oil refinery, where the temperature of the oil is codified into either 'Normal' or 'Abnormal'.

An alarm is activated when the thermal sensor, which measures the temperature of the oil in the refinery, detects an 'Abnormal' temperature. This thermal sensor will only display 'Normal' or 'Abnormal'. Further, the thermal sensor's manufacturer informs you that the temperature of the oil may cause it to fail.

The alarm is located far away from any elements that can cause its failure.

For this question, consider only the following Boolean variables:

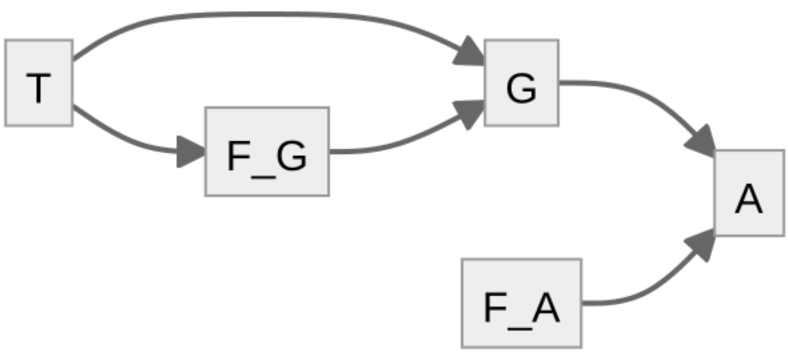
- F_A : Alarm is faulty
- F_G : Thermal sensor is faulty
- A : Alarm activates
- G : 'Abnormal' thermal sensor reading
- T : 'Abnormal' temperature of the oil in the refinery

Your boss asks you to model the system as a Bayesian Network.

Answer the **following seven questions** (i.e., **Part (a)**; **Parts (b)(i)**, **(b)(ii)**, and **(b)(iii)**; **Part (c)**; and **Parts (d)(i)** and **(d)(ii)**) based on the above context. Some additional information may also be provided before some questions. Note that there will be **NO Error Carried Forward** considered.

(a) [3 marks] Draw a Bayesian Network for this problem.

Solution:



4b. Due to a lack of information, you must model the probability that the thermal sensor gives the correct temperature using the following variables (instead of specific probabilities).

- x : when the thermal sensor is working correctly
- y : when the thermal sensor is faulty

The alarm is guaranteed to activate correctly unless it is faulty, in which case it will never activate.

(i) [2 marks] Present the conditional probability table for *G* = True.

Solution:

F_G	T	p
True	True	y
True	False	$1-y$
False	True	x
False	False	$1-x$

(ii) [2 marks] Present the conditional probability table for *A* = True.

Solution:

G	F_A	p
True	True	0
True	False	1
False	True	0
False	False	0

(iii) [1 mark] Present the conditional probability table for F_G = True. Define new variables within your table where necessary, similar to the setup in **Part (b)(i)** and **Part (b)(ii)** above, where the variables *x* and *y* were utilised.

Solution:

T	p
True	a
False	b

4c. [4 marks] Studying the literature, you discover that 'Abnormal' oil temperatures occur with probability *z*.

Suppose the alarm and thermal sensor are working and the alarm activates, what is the probability that the temperature of the oil is 'Abnormal'?

You must express this probability only in terms of the variables *x*, *y*, and *z*, and any variables you defined in **Part (b)(iii)**.

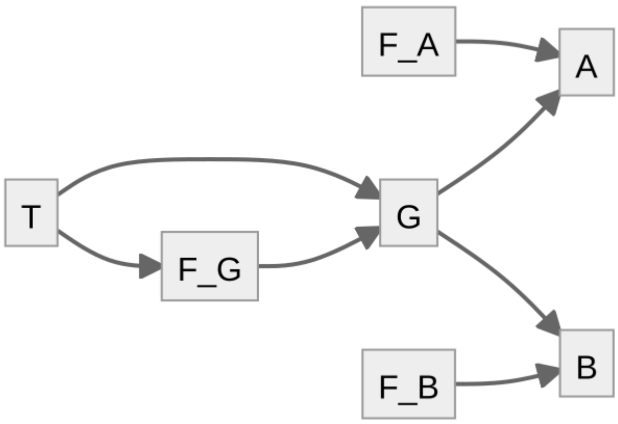
Solution:

$$\begin{aligned}
P(T|A,\neg F_A,\neg F_G) &= P(T|A,\neg F_A,\neg F_G,G)\\
&= \frac{P(T,A,\neg F_A,\neg F_G,G)}{P(A,\neg F_A,\neg F_G,G)}\\
&= \frac{P(T)P(\neg F_G|T)P(G|T,\neg F_G)P(A|\neg F_A,G)P(\neg F_A)}{P(\neg F_A)P(\neg F_G)P(G|\neg F_G)P(A|\neg F_A,G)}\\
&= \frac{P(T)P(\neg F_G|T)P(G|T,\neg F_G)}{P(\neg F_G)P(G|\neg F_G)}\\
&= \frac{P(T)P(\neg F_G|T)P(G|T,\neg F_G)}{P(G,\neg F_G)}\\
&= \frac{P(T)P(\neg F_G|T)P(G|T,\neg F_G)}{P(G,\neg F_G,T)+P(G,\neg F_G,\neg T)}\\
&= \frac{P(T)P(\neg F_G|T)P(G|T,\neg F_G)}{P(T)P(\neg F_G|T)P(G|\neg F_G,T)+P(\neg T)P(\neg F_G|\neg T)P(G|\neg F_G,\neg T)}\\
&= \frac{z\times(1-a)\times x}{z\times(1-a)\times x+(1-z)(1-b)(1-x)}
\end{aligned}$$

4d. To improve operations, the probability determined in **Part (c)** should be as high as possible. Your boss suggests purchasing a new alarm (represented by B when it activates) of the same specifications, so that it can be connected to the existing thermal sensor.

(i) [1 mark] Draw an updated Bayesian Network given the addition of this new alarm.

Solution:



(ii) [2 marks] Suppose that both alarms and the thermal sensor are working, and that both alarms activate. Considering the probability calculated in **Part (c)**, and the Bayesian Network that includes the new alarm, should your boss purchase the new alarm? Why?

Solution:

A, B, FA, FB do not have any direct relation with T, so it will never improve the probability in (e). He should not purchase the new alarm.

END OF PAPER

BLANK PAGE

BLANK PAGE

BLANK PAGE