Midterm 2 Exam

CSCI 561 Spring 2024: Artificial Intelligence

Problems	**100 Percent total**
1- General AI Knowledge	10
2- First Order Logic	20
3- Inference	20
4- Planning	15
5- Decision Tree Learning	15
6- Neural Networks	20

<u>DO NOT OPEN EXAM UNTIL YOU ARE TOLD TO</u>

Instructions:

1. Date: **Monday Mar 25th, 3:00pm - 4:50pm**
2. Maximum credits/points/percentage for this midterm: 100
3. The percentages for each question are indicated in square brackets [ ] near the question.
4. **<u>No books</u>** (or any other material) are allowed.
5. **Write down your name, student ID and USC email address.**
6. **Your exam will be scanned and uploaded online.**
7. **Write within the boxes provided for your answers.**
8. **Do NOT write on the 2D barcode.**
9. **Do not write within less than 1" from the paper edges to avoid lost work during scanning.**
10. **The back of the pages will not be graded. You may use it for scratch paper.**
11. **The back of the pages will not be scanned. Do not write any answer there!**
12. No questions during the exam. **If something is unclear to you, write that in your exam.**
13. **Be brief: a few words are enough if precise and using correct vocabulary studied in class.**
14. When finished, raise completed exam sheets until approached by proctor.
15. **Adhere to the Academic Integrity code.**

1. [10%, 1% each, no partial]

General AI Knowledge

For each of the statements below, fill in the bubble **T** if the statement is **always and unconditionally true**, or fill in the bubble **F** if it is **always false, sometimes false, or just does not make sense**.

1.	T	F
2.	T	F
3.	T	F
4.	T	F
5.	T	F
6.	T	F
7.	T	F
8.	T	F
9.	T	F
10.	T	F

1. First order logic makes the ontological commitment that the world consists of objects, relations, functions, and properties. (T, L09, P24)

2. Propositional logic operators are also valid and retain their meanings in first order logic. (T, General understanding of PL and FOL.)

3. For FOL, any sentence entailed by another set of sentences can be proved from that set. (T, L10, P71)

4. In FOL, Backward Chaining acts like a breadth-first search at the top level, with depth-first sub-searches. (F, L10, P46)

5. The tasks of a logical reasoning system only include adding new facts to KB and decide whether a query is explicitly stored in the KB. (F, L11, P3)

6. In logical reasoning systems, FETCH and STORE actions always take at least O(n) time on n-element KB. (F, L11, P5-8)

7. In planning, linearization may derive multiple total order plans from one partial order plan. (T, L12, P16-17)

8. Fuzzy logic is a super-set of Boolean logic, building on fuzzy set theory and graded truth rather than “True/False” or “Yes/No”. (T, L13, P2)

9. The ID3 Algorithm is a greedy algorithm with bottom-up construction of a decision tree by recursively selecting the “best attribute” to use at each leaf nodes and working up to the root. (F, L14, P28)

10. Artificial Neural Networks fully simulate how the human brain works. (F, L15, General understanding)

2. [20%] First Order Logic

2A. [6%, -1% for each incorrect selection (missed a correct answer, crossed out a correct answer, or circled an incorrect answer), +1% for each correct selection, down to 0, no partial for each choice. For Question 2A where the correct answer is "C", answering "AC" will get -1 for selecting "A" and +5 for the rest and will get 4 in the end; answering "ABD" will get -4 for selecting "ABD" and not selecting "C", +2 for the rest and will get 0 in the end] Given the English sentence:

"Every student loves at least one course that some professor teaches."

Circle all correct logic translation(s):

A. ∀x(Student(x) ↔ ∃y(Course(y) ∧ Loves(x, y) ∧ ∃z(Professor(z) ∧ Teaches(z, y))))

B. ∀x(Student(x) → Loves(x, y)) ∧ ∃y(Course(y)) ∧ ∃z(Professor(z) ∧ Teaches(z, y))

C. ∀x(Student(x) → (∃y(Course(y) ∧ ∃z(Professor(z) ∧ Teaches(z, y) ∧ Loves(x, y)))))

D. ∃x(Student(x) ∧ ∀y(Course(y) → (∃z(Professor(z) ∧ Teaches(z, y)) → Loves(x, y))))

E. ∀x(Student(x) ∧ ∀y(Course(y) → ∃z(Professor(z) ∧ Teaches(z, y) ∧ Loves(x, y))))

F. (¬Student(x) ∨ Professor(z) ∨ Teaches(z, y)) ∧ (Student(x) ∨ Course(y) ∨ Teaches(z, y))

2B. [6%, same criteria as 2A] Given the English sentence:

"For any project, the project is delayed if and only if at least one of its tasks is not completed on time."

Circle all correct logic translation(s):

A. ∀x(Project(x) → Delayed(x) → ∃y(Task(y) ∧ PartOf(y, x) ∧ ¬CompletedOnTime(y)))

B. ∀x(Project(x) ∧ Delayed(x)) → (∃y(Task(y) ∧ PartOf(y, x) → ¬CompletedOnTime(y)))

C. ∀x(Project(x) ∧ (∀y(Task(y) ∧ PartOf(y, x) → CompletedOnTime(y)))) → ¬Delayed(x)

D. ¬∃x(Project(x) ∧ Delayed(x)) ∨ (∀y(Task(y) ∧ PartOf(y, x) → CompletedOnTime(y)))

E. ∀x(Project(x) → (Delayed(x) ↔ ∃y(Task(y) ∧ PartOf(y, x) ∧ ¬CompletedOnTime(y))))

F. ∀x((¬Project(x) ∨ ¬Delayed(x) ∨ (∃y(Task(y) ∧ PartOf(y, x) ∧ ¬CompletedOnTime(y))) ∧ (¬Project(x) ∨ Delayed(x) ∨ ¬ (∃y(Task(y) ∧ PartOf(y, x) ∧ ¬CompletedOnTime(y))))

2C. [8%, 1% for each step, no partial for each step. The notations, such as variables and existential quantification notations, in the transformation can be different. If the meaning is correct, the points should be granted. Each step should be graded based on its previous step. This means even if the previous step is incorrect, the current step may still be granted point if the transformation from the previous step is correct.] Transform the following logic sentence to CNF.
NOTE:

- Use lower-case letters such as **"x", "y", "z"** for variables.
- Use upper-case letters such as **"A", "B", "C"** for constants.
- Use lower-case letters such as **"g(x)", "h(x)"** for skolem functions, if needed.
- Do **NOT** use confusing notations when answering the question. For example, you should not use lower case "x" for a variable and upper case "X" for a constant at the same time.

$$(\forall x)\ (P(x) \Rightarrow (\neg(\forall y)\ (P(y) \Rightarrow P(x,y)) \wedge (\exists y)(Q(x,y) \wedge P(y))))$$

1. [1%] Eliminate =>

$(\forall x)\ (\neg P(x) \vee (\neg(\forall y)\ (\neg P(y) \vee P(x,y)) \wedge (\exists y)(Q(x,y) \wedge P(y))))$

2. [1%] Reduce scope of negation

$(\forall x)\ (\neg P(x) \vee ((\exists y)\ (P(y) \wedge \neg P(x,y)) \wedge (\exists y)(Q(x,y) \wedge P(y))))$

3. [1%] Standardize variables

$(\forall x)\ (\neg P(x) \lor ((\exists y)\ (P(y) \land \neg P(x,y)) \land (\exists z)(Q(x,z) \land P(z))))$

4. [1%] Eliminate existential quantification

$(\forall x)\ (\neg P(x) \lor ((P(g(x)) \land \neg P(x,g(x))) \land (Q(x,h(x)) \land P(h(x)))))$

5. [1%] Drop universal quantification symbols

$\neg P(x) \lor ((P(g(x)) \land \neg P(x,g(x))) \land (Q(x,h(x)) \land P(h(x))))$

6. [1%] Convert to conjunction of disjunctions

$\neg P(x) \lor P(g(x)) \land$
$(\neg P(x) \lor \neg P(x,g(x))) \land$
$(\neg P(x) \lor Q(x,h(x))) \land$
$(\neg P(x) \lor P(h(x)))$

7. [1%] Create separate clauses

$\neg P(x) \lor P(g(x))$
$\neg P(x) \lor \neg P(x,g(x))$
$\neg P(x) \lor Q(x,h(x))$
$\neg P(x) \lor P(h(x))$

8. [1%] Standardize variables

$\neg P(x) \lor P(g(x))$
$\neg P(y) \lor \neg P(y,g(y))$
$\neg P(z) \lor Q(z,h(z))$
$\neg P(w) \lor P(h(w))$

3. [20%] Inference

Given the following FOL knowledge base,
please answer following questions:

1. P(A)
2. P(B)
3. ¬P(x) ∨ Q(x)
4. ¬P(x) ∨ R(x, y) ∨ ¬Q(y)
5. ¬S(x, y) ∨ U(x, y) ∨ ¬W(y)
6. W(B)
7. ¬T(D)
8. U(A, x) ∨ T(x)
9. S(x, y) ∨ M(x)
10. ¬M(A)
11. G(x, y, z, w) ∨ U(x, y) ∨ ¬R(z, w)

NOTE:

- Your answer should contain "T" or "F" to indicate whether the query is true or false,
- If your answer is "T", you should include the numbers of the statements in the KB that are required to infer your answer.
- If your answer is "F" due to a contradiction, you should include numbers of the statements in the KB that are required to infer your answer.
- If your answer is "F" simply because it's not inferable from the KB, don't write anything else except the "F".

Example 1: "Q(A)": since you need KB statements 1 and 3 to prove it's true, your answer should be:

T, 1, 3

Example 2: "Q(H)": since Q(H) is not inferable from the KB, your answer should be:

F

3A. R(A, B)

T, 1, 2, 3, 4 (order of sentence numbers does not matter, No partial credits on 3A-3E)

3B. R(A, A)

T, 1, 3, 4

3C. U(A, B)

T, 5, 6, 9, 10

3D. ¬U(A, D)

F, 7, 8

3E. G(A, B, A, B)

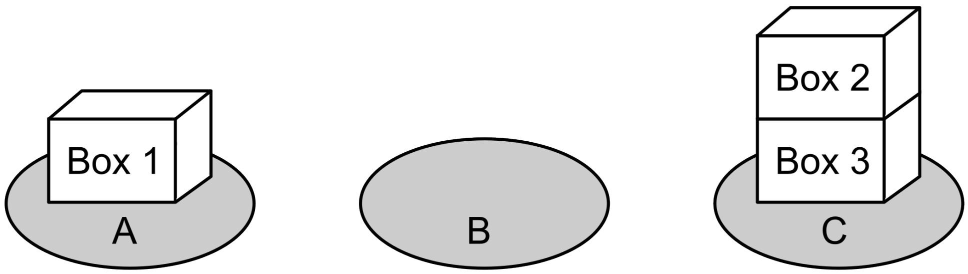
F

4. [15%] Planning

4.1 [3%] STRIPS actions

Consider a block world problem slightly
different from the one studied in class, where only one box can be directly placed onto each location A, B, or C on the table. As in the block world problem studied in class, boxes can be stacked onto other boxes and you can only move one box at a time.

Please describe the **STRIPS actions** required for a robot to rearrange the boxes in the block world shown in the figure below. (Hint: You may define an action that is to move the box to a location.)



Action: Move (box, place)
Precondition: Clear (place), Clear (box)
Effect: At (box, place)

{Optional:
Action: Move (box, place)
Precondition: At (box', place), Clear (box'), Clear (box)
Effect: At (box, place), On (box, box')

Action: Move (box, place)
If precondition has On (box, table/box'), Clear (box)
Effect: Clear (table/box')}

[If missing **Clear()** in the precondition of **Move(box, place)**, -1%; if the **STRIPS structure** is wrong, -2%; otherwise, any reasonable answers are correct]

4.2 [4%] Initial (empty) plan

Your plan is to rearrange the boxes from the starting state shown in 4.1 to a new state, where <u>Box 3 is in Place A, Box 2 is in Place B, and Box 1 is in Place C</u>.

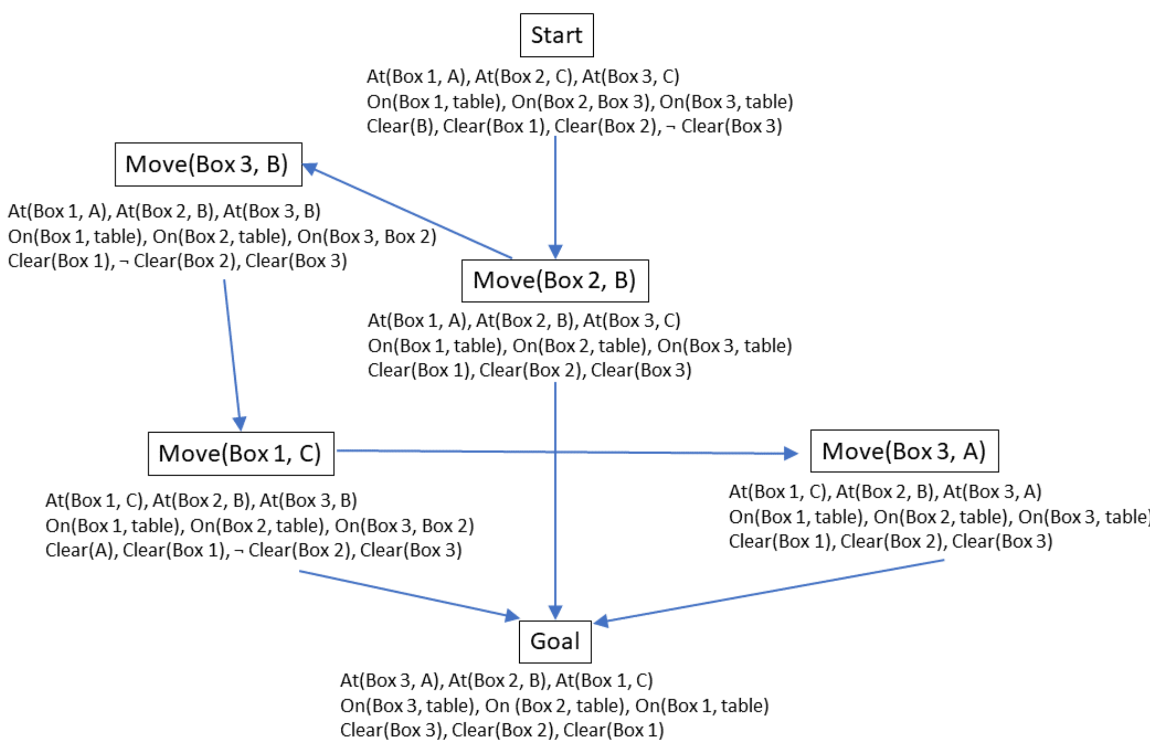
Please write down the **initial condition** and
the **goal** of this plan using your STRIPS definitions in 4.1.

Initial Condition: [2%; if missing **Clear()**, -1%; if missing **On() or At()**, -1%]
At (Box 1, A), At (Box 2, C), At (Box 3, C)
On (Box 1, table), On (Box 2, Box3), On (Box 3, table)
Clear (B), Clear (Box 1), Clear (Box 2), ¬Clear (Box 3)

Goal: [2%; if missing **Clear()**, -1%; if missing **At()**, -1%; missing On () is fine]
At (Box 3, A), At (Box 2, B), At (Box 1, C)
On (Box 3, table), On (Box 2, table), On (Box 1, table)
Clear (Box 3), Clear (Box 2), Clear (Box 1)

4.3 [8%] Complete plan

Write down **your plan** to reach the goal from the initial condition that you specified in 4.2.



[Every missing action: -2%; other reasonable answers are all accepted]

<u>5. [15%] Decision Tree Learning</u>

5.1 [3%] You are given a decision problem
with 8 "True" examples and 4 "False" examples.
What is the information $I(P)$ (or entropy)
conveyed by this distribution **in bits**? Show the formula with numbers plugged in. You do not need to work out the numeric calculation for the final result.

$$I(P) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n} = -\frac{8}{8+4} log_2 \frac{8}{8+4} - \frac{4}{8+4} log_2 \frac{4}{8+4}$$

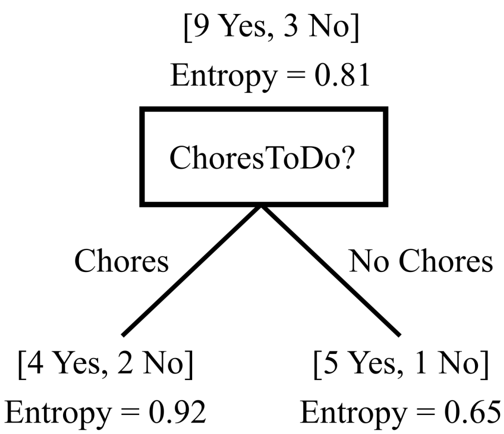
[Formula: 2%; numbers plugged in: 1%.

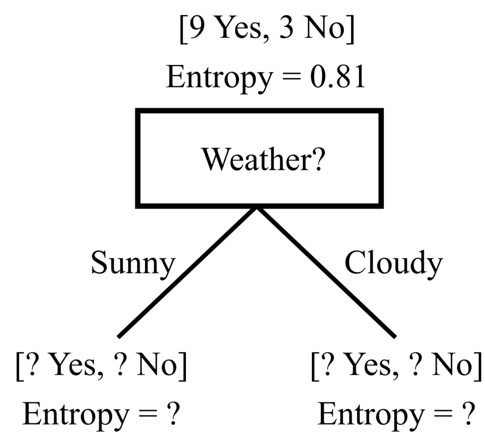
If students give the formula <u>in nats (using *ln* rather than *log*2)</u>, give them 1% for formula]

5.2 To decide whether or not to go hiking, two attributes, **weather** and **if I have chores to do**, are considered. Some example decisions are listed in the table below.

Weather	**ChoresToDo**	**Hiking**
Sunny	Chores	Yes
Sunny	Chores	Yes
Sunny	No Chores	Yes
Sunny	No Chores	Yes
Sunny	No Chores	Yes
Sunny	No Chores	Yes
Cloudy	Chores	No
Cloudy	Chores	No
Cloudy	No Chores	Yes
Cloudy	No Chores	No
Cloudy	Chores	Yes
Cloudy	Chores	Yes

With 9 positive examples of hiking and 3 negatives, the entropy of this decision **in bits** is 0.81. Consider the following two decision tree nodes, reflecting splitting on the attributes of **Weather** or **ChoresToDo**.





(a) [6%] The ChoresToDo node has been filled out. Please complete the values for the Weather node, including entropy **in bits**. (show your work)

Under Sunny: _6__ Yes, _0__ No, [1% for yes and no; 2% for entropy]

Entropy = _ $-\frac{6}{6+0} log_2 \frac{6}{6+0} - \frac{0}{6+0} log_2 \frac{0}{6+0} =$

0____

Under Cloudy: _3__ Yes, _3__ No, [1% for yes and no; 2% for entropy]

Entropy = __ $-\frac{3}{3+3} log_2 \frac{3}{3+3} - \frac{3}{3+3} log_2 \frac{3}{3+3} =$

1____

____

[For entropy, if final answer is correct but no formula, give 1% for each]

Show any details below:

(b) [4%] Calculate the information gain *IG* from splitting on ChoresToDo and Weather. Please show formulas and steps clearly.

$$IG(ChoresToDo) = I(P) - Entropy(ChoresToDo) = 0.81 - \left(\frac{6}{12} \times 0.92 + \frac{6}{12} \times 0.65\right) = 0.025$$

[1% for final answer; 1% for formulas]

$$IG(Weather) = I(P) - Entropy(Weather) = 0.81 - \left(\frac{6}{12} \times 0 + \frac{6}{12} \times 1\right) = 0.31$$

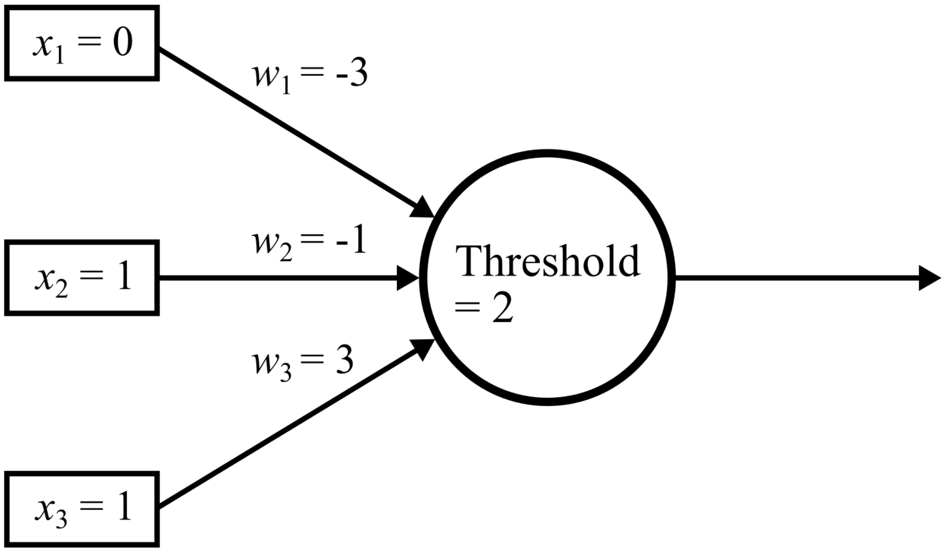
[1% for final answer; 1% for formulas]

(c) [2%] Which attribute is a better choice in constructing the decision tree? Explain why.

Weather is a better choice, because *IG*(Weather) > *IG*(ChoresToDo)
[1% for choice, 1% for the reason]

6. [20%] Neural Networks

6.1 Consider a single perceptron unit below.
Assume that the firing follows the threshold rule in class.



(a) [2%] With the inputs and weights as shown, what would the output of the perceptron be?

$0 \times (-3) + 1 \times (-1) + 1 \times 3 = 2 \geq Threshold \rightarrow 1\ (True)$

1 (or True) **[no partial credit]**

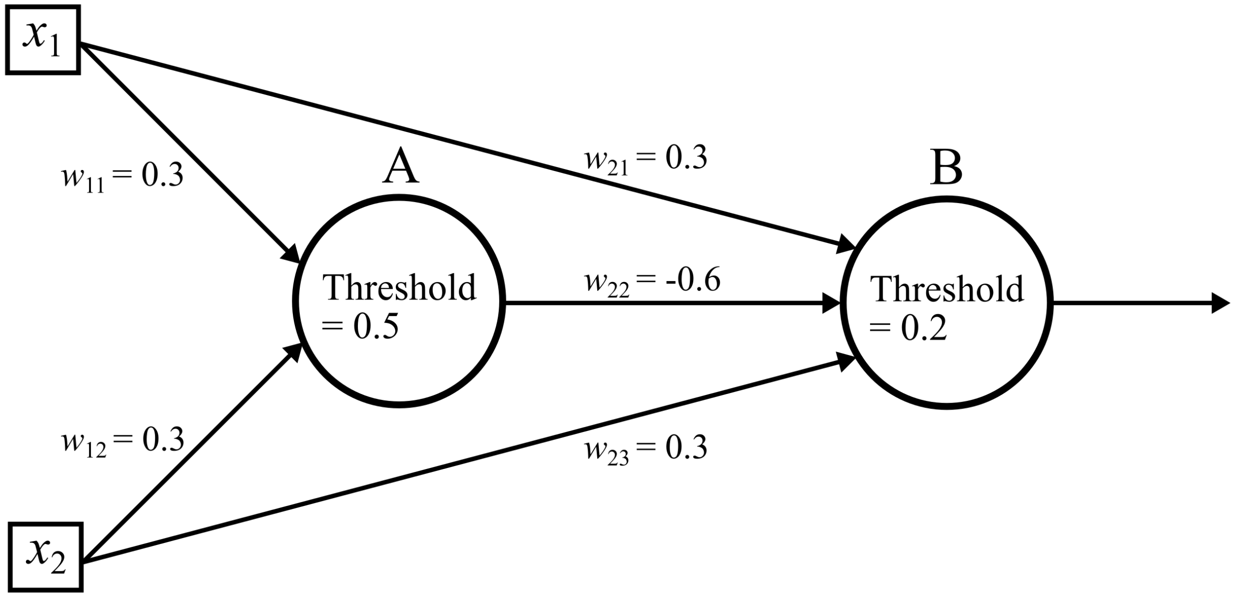
(b) [6%] If the output from the above case was higher than the desired output, how would each of the weights (w_1, w_2, w_3) on inputs be adjusted? Answers should be "increase", "decrease", or "stay the same"?

w_1: Stay the same

w_2: Decrease

w_3: Decrease [2% for each; no partial credits]

6.2 Given a diagram of a Neural Network with two perceptron units A and B, answer the questions below. **Show your steps.**



(a) [4%] When input $x_1 = 0$ and $x_2 = 0$, what does the Unit A output? What does the Unit B output?

For Unit A, $0 \times 0.3 + 0 \times 0.3 = 0 < 0.5$, output of Unit A is 0 (False) For Unit B, $0 \times 0.3 + 0 \times (-0.6) + 0 \times 0.3 = 0 < 0.2$, output of Unit B is 0 (False) **[2% for each unit. If there is only final answer but no steps, 1% for each]**

(b) [4%] When input $x_1 = 1$ and $x_2 = 0$, what does the Unit A output? What does the Unit B output?

For Unit A, $1 \times 0.3 + 0 \times 0.3 = 0.3 < 0.5$, output of Unit A is 0 (False) For Unit B, $1 \times 0.3 + 0 \times (-0.6) + 0 \times 0.3 = 0.3 > 0.2$, output of Unit B is 1 (True) **[2% for each unit. If there is only final answer but no steps, 1% for each]**

(c) [4%] What Boolean function does Unit A represent? What Boolean function does Unit B represent? Explain why?

When x1 = 0, x2 = 1, output of A is False, and output of B is True **[0.5%]**
When x1 = 1, x2 = 1, output of A is True, and output of B is False **[0.5%]**
So, the truth table is**: [1%]**

X1	X2	Output of A	Output of B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

So, unit A represents AND [1%]

Unit B represents XOR [1%].

[If the truth table is correct, but there is no separate calculation for x1=0, x2=1 and x1=1, x2=1, it is fine]