Final

**CSCI 561 Fall 2015: Artificial Intelligence**

| Student ID: | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|

Last Name: ____________________________________________

First Name: ____________________________________________

| USC email: | | | | | | | | | @usc.edu |
|---|---|---|---|---|---|---|---|---|---|

# Instructions:

1. Date: **12/9/2015 from 4:30pm – 6:30 pm in THH-101/102**
2. Maximum credits/points/percentage for this midterm: 100
3. The percentages for each question are indicated in square brackets [ ] near the question.
4. **No books** (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. **The back of the pages will not be graded. You may use it for scratch paper.**
10. No questions during the exam. **If something is unclear to you, write that in your exam.**
11. **Be brief: a few words are often enough if they are precise and use the correct vocabulary studied in class.**
12. When finished, raise completed exam sheets until approached by proctor.
13. **Adhere to the Academic Integrity code.**

| **Problems** | **100 Percent total** |
|---|---|
| 1- General AI Knowledge | 10 |
| 2- Planning | 20 |
| 3- Probability | 10 |
| 4- Game Playing | 30 |
| 5- Artificial Neural Networks | 20 |
| 6- Applications | 10 |

## 1. [10%] 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. F_It is clear and transparent to humans how learning happens in neural networks.
2. F_A single layer perceptron allows making XOR separable by one line.
3. F_A Hopfield network becomes unstable above a certain number of neurons per pattern.
4. T_Bayes' Theorem takes you from a prior belief to a posterior belief after you've collected some additional data.
5. F_Given the current state of the art as stated in class, it looks likely that a single neural network will be able to do everything the brain can do.
6. F_The number of hidden layers needed in a neural network can be calculated from the input provided.
7. F_Ockham's razor means that we reject the simplest hypothesis that is consistent with the data.
8. F_The simplest neuron models use a continuous notion of time.
9. T_If we have an algorithm or formula that solves a problem, then learning is not necessary.
10. T_Genetic algorithms are well suited to determine the number of hidden units in neural networks.

## 2. [20 %] Planning

Consider the following STRIPS action schema for planning the end of semester:

BookFlight()
  Pre: HaveExamDate(Math), HaveExamDate(Physics)
  Post: Have(Ticket)

TakeExam(subject)
  Pre: At(School), HaveExamDate(subject), Prepared(subject), Enrolled(subject)
  Post: Passed(subject)

Drive (A, B)
  Pre: At(A)
  Post: At(B), ¬At(A)

Fly()
  Pre: At(Airport), Have(Ticket), Have(Luggage)
  Post: ¬At(Airport), ¬Have(Ticket), At(Home)

LookupExamDate(subject)
  Pre: [no preconditions]
  Post: HaveExamDate(subject)

Pack()
  Pre: At (School), Have(Gift)
  Post: Have(Luggage)

Shop()
  Pre: At(Store), Prepared(Math), Prepared(Physics)
  Post: Have(Gift)

Study(subject)
  Pre: HaveExamDate(subject)
  Post: Prepared(subject)

Given the following:

  Start: At(School), Enrolled(Math), Enrolled(Physics)
  Goal: At(Home), Have(Gift), Passed(Math), Passed(Physics)

Show the partial order plan for planning the end of semester. Your answer must be a **complete written plan listing a possible sequence of actions** and also show a **graph with the actions from the Start state to the Final state using the actions above**. Your graph must clearly show the **precedence relationships** between actions and causal links as applicable, allowing the plan executor to have as much choice in the sequencing of actions without violating any causal links. For each action, **show the required preconditions on the graph**. (Hint: You can assume that both exams are scheduled at different times that don't overlap.)

You should draw the partial order plan on the next page.

**2A. [8%] Written Plan**

START

LookupExamDate(Math)

LookupExamDate(Physics)

BookFlight()

Study(Math)

Study(Physics)

Drive(School, Store)

Shop()

Drive(Store, School)

Pack()

TakeExam(Math)

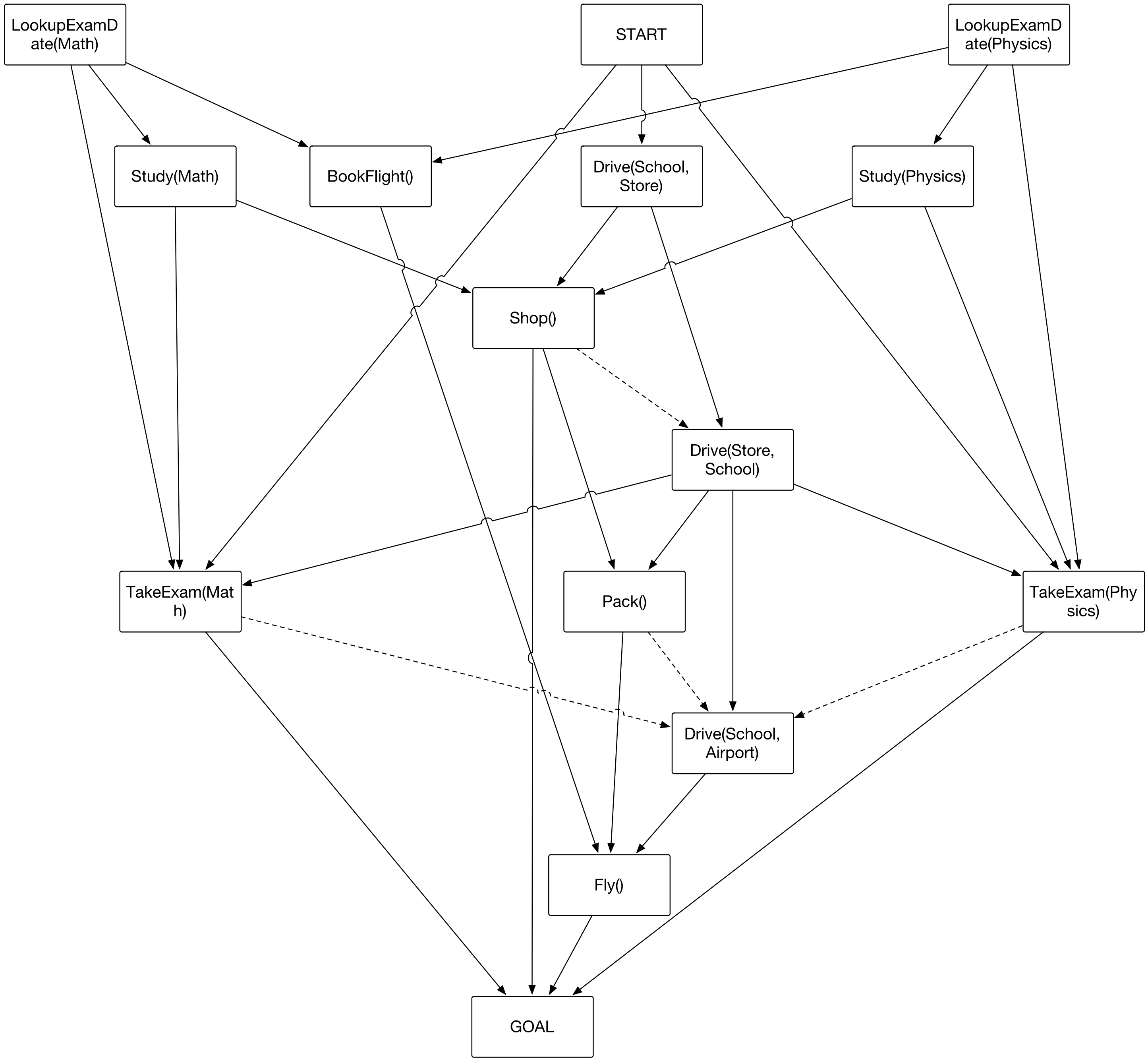
TakeExam(Physics)

Drive(School, Airport)

Fly()

GOAL

## 2B. [12%] Graph (with Actions, and Preconditions and Precedence Relationships)




The main thing to get wrong are the precedences of Drive() actions. (While the precondition to drive from A to B is just to be at A, doing this may destroy the precondition of another action.)

## 3. [10%] Probability

**A [5%]** You would like to know how useful it is for you to learn to take electric guitar lessons, so you would like to find out what the probability of getting hired by a heavy metal band is given that you took electric guitar lessons. What other probabilities would you need to know? (give a formula to show how these additional probabilities are used).

We're looking for P(hired by h-m band | can play e-guitar)

We need to know P(hired by h-m band), P(can play e-guitar), P(can play e-guitar | hired by h-m band)

P(hired by h-m band | can play e-guitar) = (P(can play e-guitar | hired by h-m band) P(hired by h-m band)) / P(can play e-guitar)

**B [5%]** (1) We have made a machine to classify whether a room smells fresh or unpleasant based on parts per million (PPM) of methane gas molecules that are in the air. A robot has taken a sample from room which has 100 PPM. From prior sampling, we have derived the following probabilities:

(a) P(100 PPM|Unpleasant) = 0.4
(b) P(100 PPM|Fresh) = 0.3
(c) P(Unpleasant) = 0.2
(d) P(Fresh) = 0.9
(e) P(100 PPM) = 0.2

Show using a Bayesian classifier whether the room's air should be classified as Fresh or Unpleasant. Be sure to show your work.

$$P(Unpleasant|100PPM) = \frac{P(100\ PPM|Unpleasant)P(Unpleasant)}{P(100\ PPM)}$$

$$P(Fresh|100PPM) = \frac{P(100\ PPM|Fresh)P(Fresh)}{P(100\ PPM)}$$

$$P(Unpleasant|100\ PPM) = \frac{0.4x0.2}{2} = 0.04$$

$P(Fresh|100\ PPM) = \frac{0.3x0.9}{0.2} = 1.35$ (*)

$$P(Fresh|100\ PPM) > P(Unpleasant|100\ PPM)$$

The air in the room is more likely to be fresh.

(*) NOTES: 1. Anyone who applies the Bayes rule correctly and gets 1.35 should get full points.
2. If in addition to that, they have indicated why is this an issue, give an additional 5 points bonus.
3. If they had 1.35, but erased it due to being >1, give full points.

## 4: [30%] Game Playing

Consider a normal version of the nim-game, a zero sum game:
The game consists of **N** piles of coins each containing ***m1, m2, …, mN*** coins respectively. Two players take turns to pick up coins from the piles. During her turn, the player must pick 1 or more coins from exactly one pile (i.e. player cannot pick coins from different stacks in one turn). The player who picks up last wins the game.

For this question, assume that
**N** = 3; ***m1*** = 1; ***m2*** = 2; ***m3*** = 1

NOTE to Graders: [A] and [B] are closely related so should be graded together.

**[A] [6%]** Describe the (a) state representation to be used for the game, (b) Initial State, and (c) Final/Terminal State

***(a)*** [2%] Array containing ***3*** numbers, each representing the number of coins in 1st, 2nd, and 3rd stack respectively. ***+ optionally, Information about whose turn it was (Player1 or Player2)***
If the information about the player is not in the state, then see [B]'s answer on how they decide the utility value. Give full points if they can determine the utility from this state representation. Otherwise deduct 1 point.
(b) [2%] Initial State: [1, 2, 1, Null] or [1, 2, 1, Player1] or [1,2,1] (depends on (a)'s answer)
(c) [2%] Final State: [0, 0, 0, x] where x can be player1 or player2 (depends on (a)'s answer)

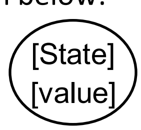
**[B] [4%]** Describe the Utility function (from Player 1's perspective) for the terminal states.

**Basically, if the last player to pick is MAX player then utility is 1, otherwise it is -1 (0 is also acceptable)**

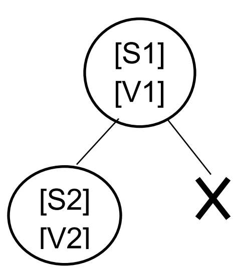
F([0, 0, 0 , x]) = $\begin{cases} 1, & if\ x = player\ 1\ (MAX\ PLAYER) \\ -1, & if\ x = player\ 2\ (MIN\ PLAYER) \end{cases}$

**[C] [20%]** Draw the minimax game tree (on the next page) for the first player, and ***select the best move***. Apply the alpha-beta pruning algorithm while generating the minimax game tree. Use the following conventions:

1. Clearly mark the MAX and MIN levels in the tree.
2. For each node, describe the ***state*** using your representation described in **[A]** and the ***backed-up value*** according to the minimax algorithm, as shown below:



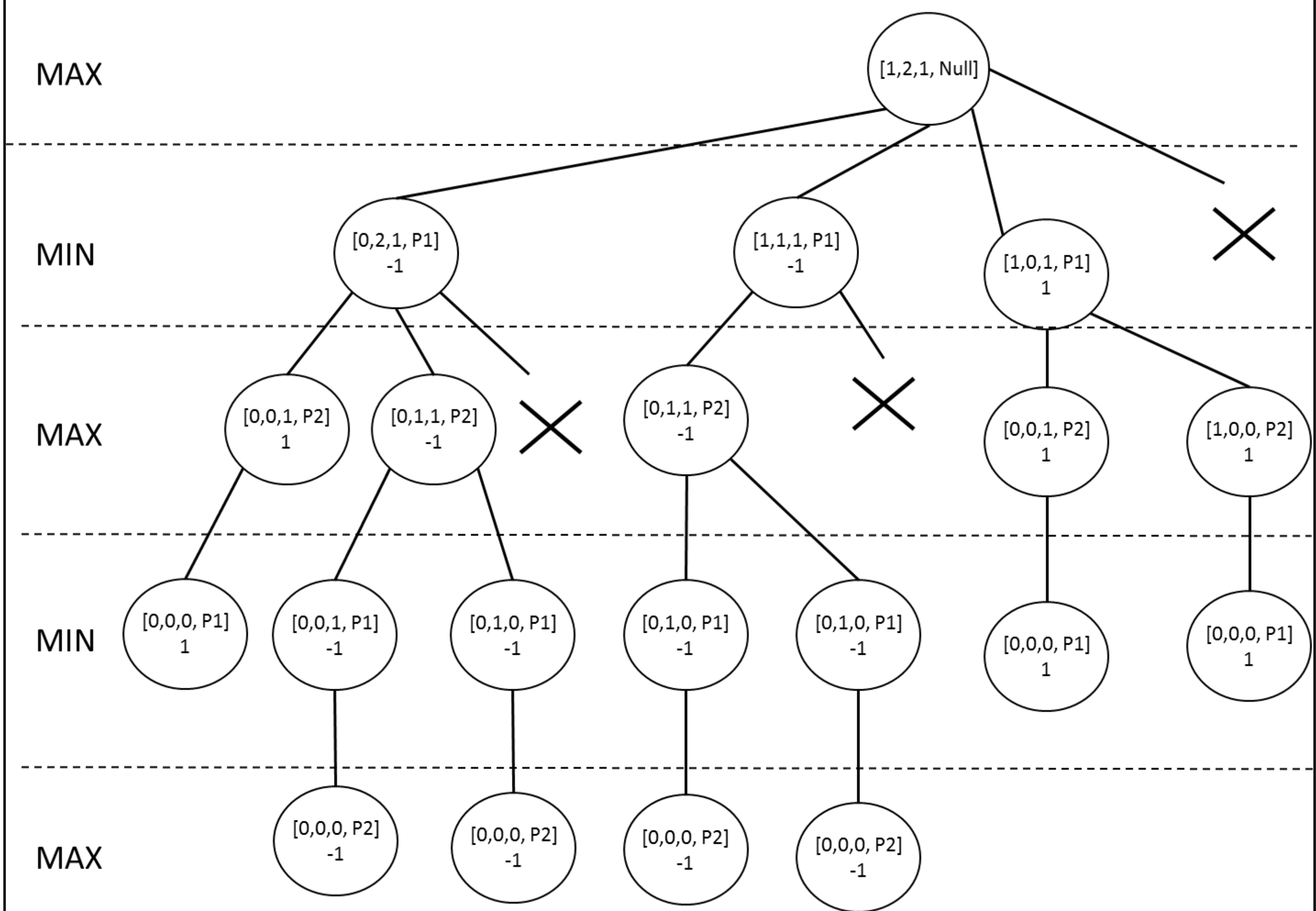

3. While applying the alpha-beta pruning algorithm, if some child nodes have been pruned, indicate that by a cross as follows:



**Hint:** You need not generate the entire tree, as you can stop as soon as the best move is found.
*NOTE: Use this page to answer **Q4 [C]**

**NOTE:** Students may build the tree in different order, so the pruning might look different. Graders should carefully evaluate the tree and contact TA (kumbhare@usc.edu) if there are any concerns since there will be no re-grading for finals.

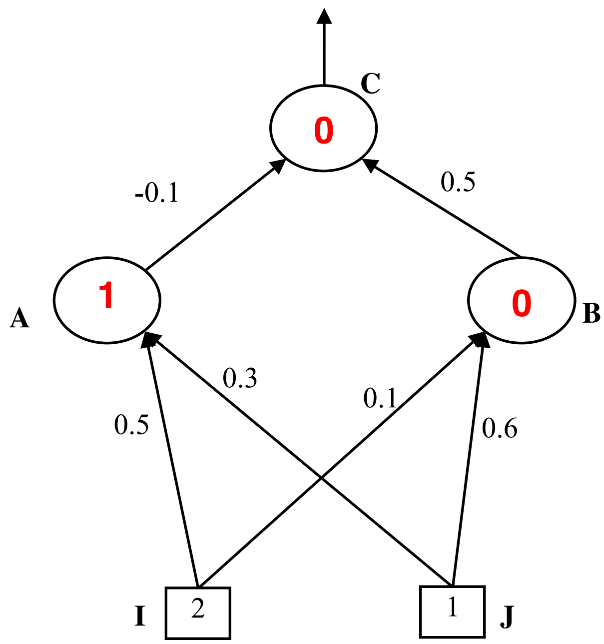
**Potential errors:** Draw the tree in BFS instead of DFS (i.e. draw all the children of the root node first, instead of expanding one child at a time). If you see this error deduct **5** points.



## 5 [20%] Artificial Neural Networks

**5.1 [8%]** Consider the neural network shown below, where I, J are inputs and A, B, C are perceptrons. For each perceptron assume it's firing is based on the threshold rule discussed in class with Threshold = 1 i.e.,

$$f(x) = \begin{cases} 1, & if\ x \geq 1 \\ 0, & otherwise \end{cases}$$




a) [3%] Write the ***output*** for each perceptron in the circles in the above figure.

NOTE: Students may have written the net value (e.g. 1.3) instead of the activation value, in that case give *0* points since the activation function was clearly mentioned.

b) [5%] Given the expected output to be 1, find the (a) error value, and the updated weights for (b) AC and (c) BC

(a) **[1%]** Error = |Expected – Observed| = |1 – 0| = 1 (If student have correct formula but incorrect answer, give 0.5 points)

(b) **[2%]** AC_new = AC + Alpha * Error * A_Output = -0.1 + 1 * 1 = 0.9 ***(NOTE: Since the exact learning rate was not given, any answer* greater than -0.1 is acceptable if the student writes the formula or gives correct reasoning in words)**

(c) **[2%]** BC_new = BC + Alpha * Error * B_Output = 0.5 + 1 * 0 = 0.5 (Note: Given B's output is 0, the only acceptable answer is 0.5)

**5.2 [12%]** You are building a set of perceptrons to mimic logic circuits. For each perceptron assume it's firing is based on the threshold rule discussed in class i.e.

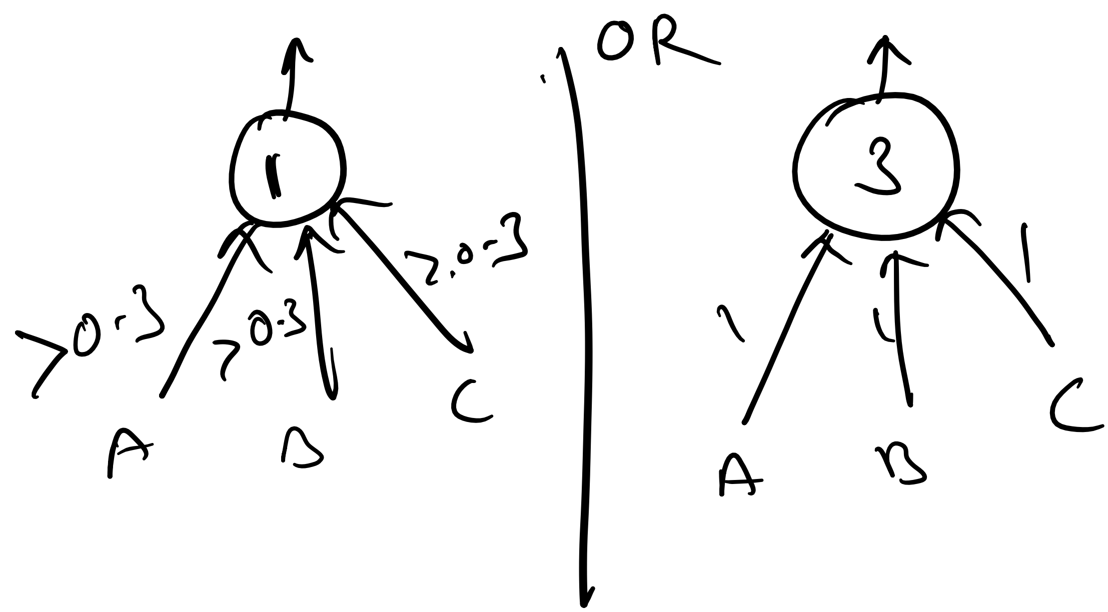
$$f(x) = \begin{cases} 1, & if\ x \geq 1 \\ 0, & otherwise \end{cases}$$

Assume all inputs have value either 0 or 1. Show how perceptrons can be used to mimic the following logic sentences. For each expression, (a) draw the perceptrons, (b) show the weights (c) write the threshold.

NOTE: (1) Looks like there was confusion regarding whether to fix threshold = 1 for all preceptrons or to allow any threshold value. We will accept either answer.

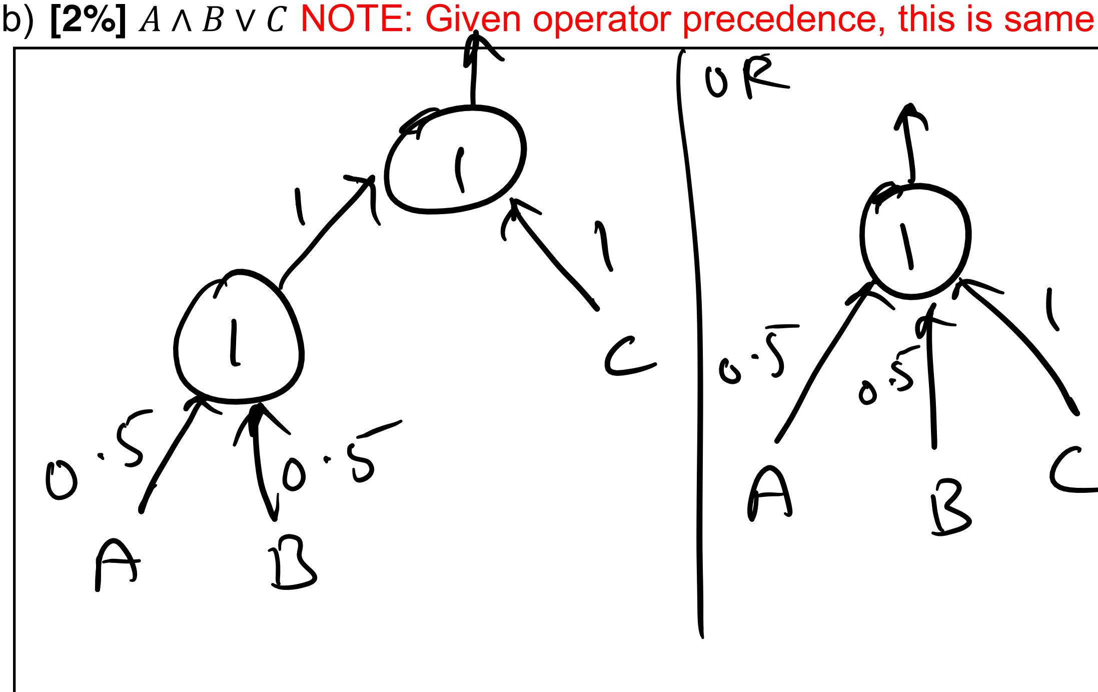
(2) The actual diagrams may differ. verify the solution using the truth table

a) **[2%]** $A \wedge B \wedge C$



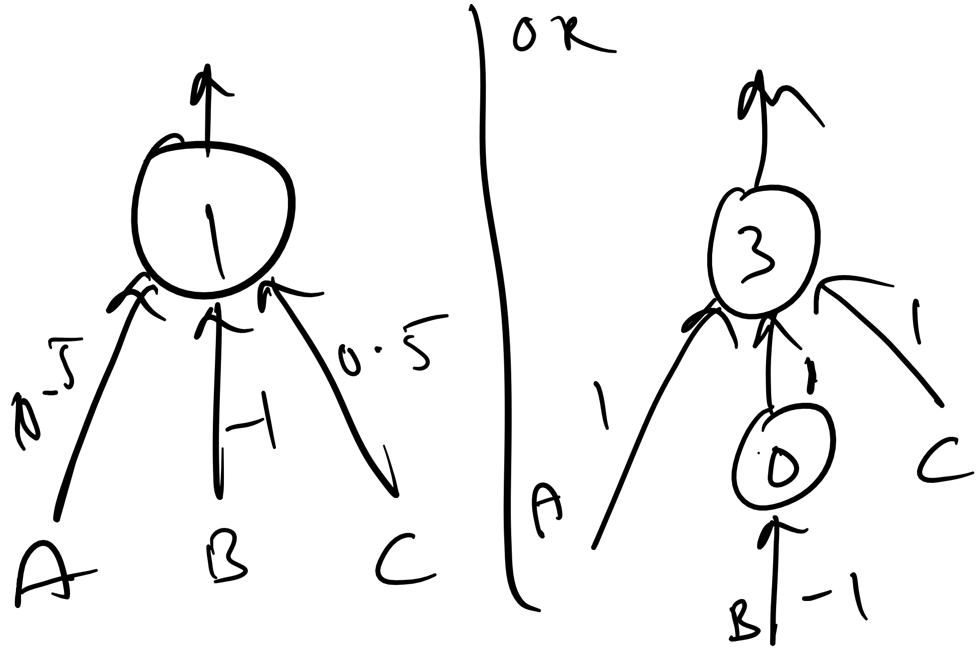

| A | B | C | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

b) **[2%]** $A \wedge B \vee C$ NOTE: Given operator precedence, this is same as $(A \wedge B) \vee C$



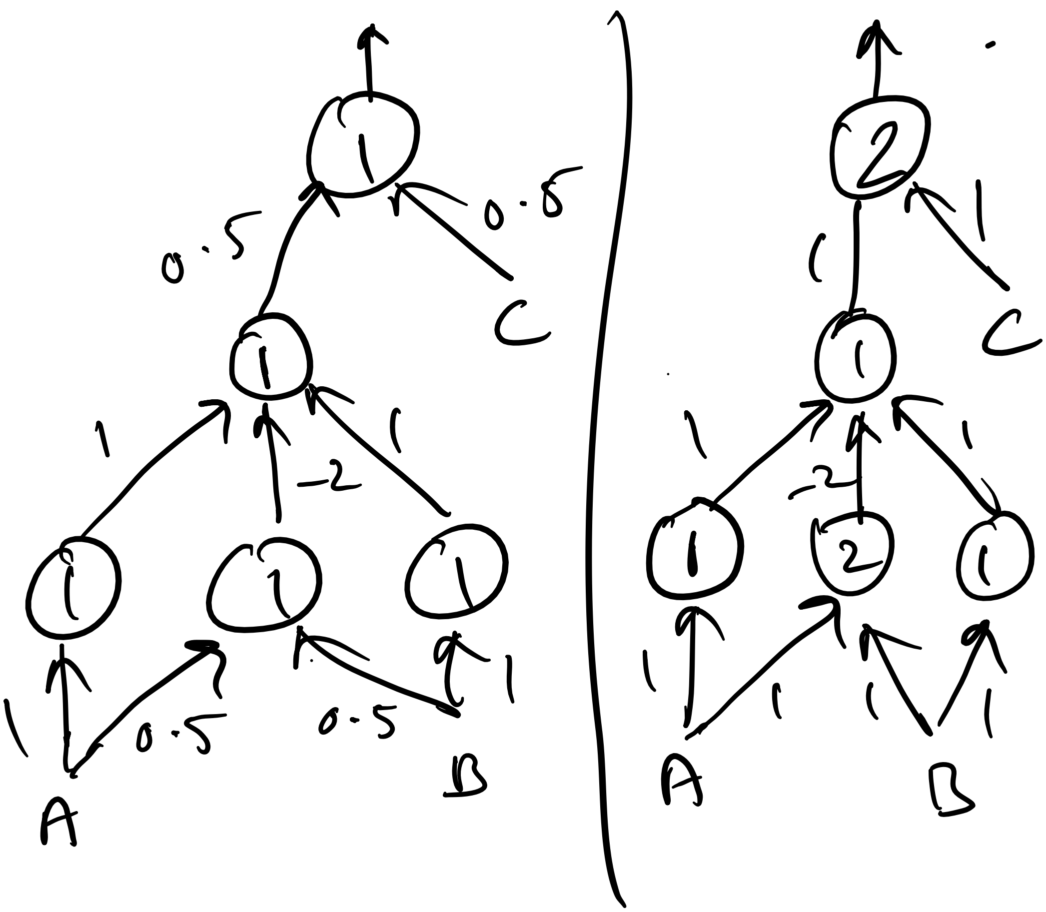

| A | B | C | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

c) **[3%]** $A \wedge \neg B \wedge C$




| A | B | C | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

d) **[5%]** $A \oplus B \wedge C$




| A | B | C | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |