Homework 1

Due: Wednesday, February 8 2023

All homeworks are due at 11:59 PM on Gradescope.

Please do not include any identifying information about yourself in the handin, including your Banner ID.

Be sure to fully explain your reasoning and show all work for full credit.

Problem 1

The CS22 gardeners are planting trees to fill a new forest. There are five kinds of trees they can plant: ash, birch, cherry, dogwood, and elm. They have very strict rules about which kinds of trees can go together:

i. If ash trees are not planted, birch trees are planted.

ii. If cherry and birch trees are both planted, then elm trees cannot be planted.

iii. If dogwood trees are not planted then elm trees must be planted.

Complete the following questions regarding the trees they plant:

(a) Using a variable to represent each atomic proposition "ash trees are planted," "birch trees are planted," etc., write logical expressions using $\wedge$, $\vee$, $\rightarrow$, and $\neg$ to demonstrate the above rules.

(b) If neither dogwood nor birch trees are planted, which kinds of trees *are* planted? Which are not? Which can we not determine from these rules? Explain your reasoning.

(c) An *inference* is a single step of reasoning: "because fact 1 holds, fact 2 must hold." Choose two inferences in your above explanation. Which proof rules from lecture 4 do they correspond to? (For example: an introduction, or elimination, modus ponens.)

Problem 2

(a) Suppose x and y are propositions such that the truth value of $x \to y$ is false. Determine the truth values of the following, and briefly explain your answer:

  i. $(\neg x) \to y$

  ii. $x \vee y$

  iii. $y \to x$

(b) We have seen the proof rule *modus ponens*, which says that if we know $x \to y$ and we know x, we can conclude y. This rule makes sense according to our truth table semantics: if $x \to y$ and x are both true, then y must be true as well.

Rob is a greedy logician and he wants more proof rules! Below are a few of his proposals. For each proposal, decide whether it makes sense according to our truth table semantics. If yes, explain why. (Showing some truth tables is a good way to explain this!) If no, give an example of a situation where this rule would allow us to conclude something false.

For example: a bad proof rule would be one that says "if we know $x \vee y$ then we know y." But we know $(1 = 1) \vee (1 = 2)$, and so this rule would let us conclude that $1 = 2$, which is false.

  i. If we know $x \to y$ and we know y, we can conclude x.

  ii. If we know $x \to y$ and we know $\neg y$, we can conclude $\neg x$.

  iii. If we know $x \to y$ and we know $\neg x$, we can conclude $\neg y$.

Problem 3

This problem is a Lean question! Please go to `https://github.com/brown-cs22/CS22-Lean-2023` where you will find instructions for setting up the course Lean project in Gitpod.

This homework question can be found by navigating to `BrownCs22/Homework/Hw1.lean` in the directory browser on the left of your screen in Gitpod. The comment at the top of that file provides more detailed instructions.

You will submit your solution to this problem separately from the rest of the assignment. Once you have solved the problem, copy the entire file to your computer, and upload it to Gradescope.

Mind Bender

Mind Benders are extra credit problems intended to be more challenging than usual homework problems. They are denoted with a symbol. Occasionally, some parts from problem 1-3 might also be extra credit problems.

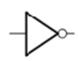
These problems are sometimes quite challenging. Try them on your own—don't look to the TAs for too much help!

One application of propositional logic is in the design of logic circuits, which allow computers to perform operations on the binary data that they work with. Often in computers, this is accomplished by using transistors, but for our modeling purposes we can use the following notation.

YES

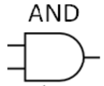
INPUT	OUTPUT
A	
0	0
1	1

NOT



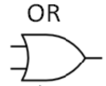
INPUT	OUTPUT
A	
0	1
1	0

AND



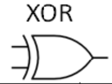
INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR



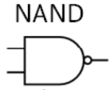
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR



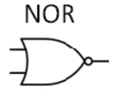
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND



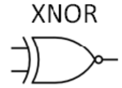
INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR



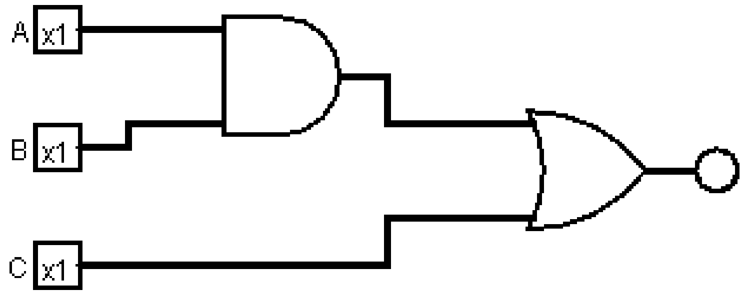
INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR



INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

For example, we can write the statement $(A \wedge B) \vee C$ as



a. Draw the following circuits.

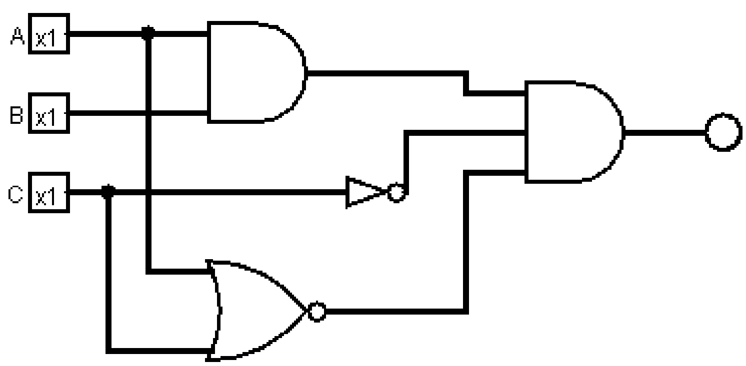
   i. $(A \vee B) \wedge C$

   ii. $(A \wedge B) \vee ((B \vee C) \wedge (C \wedge B))$

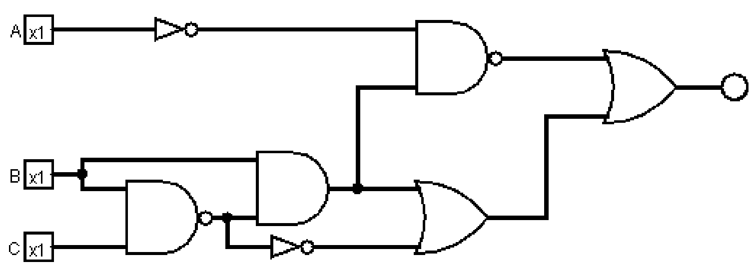
   iii. $((\neg A \wedge B) \wedge (\neg A \wedge \neg C)) \vee (B \wedge C)$

b. Write the following logic circuits as logic statements and create a truth table for each.

   i.



   ii.



c. The NAND and NOR gates are also known as the universal logic gates since we can implement any of the other gates by just using the chosen universal gate.

   i. Implement AND, OR, and NOT by only using NAND gates.

   ii. Why can it be helpful to use universal logic gates in computers rather than all of the gates separately?

d. Logic circuits can also be used to perform Boolean arithmetic. Let's make the simplest addition circuit, which takes in 2 one digit Boolean numbers and returns the sum.

Hint: Since each output can only return one Boolean digit, utilising 2 outputs, one for each digit of the result, can be a useful strategy. It can also be useful to create a truth table for Boolean addition before making the circuit.