Homework 1

Due: Wednesday, February 7, 2024

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

There are four dinosaurs: a tyrannosaurus rex, a velociraptor, a brachiosaurus, and a spinosaurus. Each of the dinosaurs has very specific sleeping habits depending on its habits and lifestyle. There are very strict rules about when each dinosaur sleeps:

  i. At least one of the tyrannosaurus rex and the velociraptor is sleeping.

 ii. If the tyrannosaurus rex is sleeping, then the brachiosaurus is sleeping.

iii. If the velociraptor is not sleeping, then the brachiosaurus and the spinosaurus are sleeping or the tyrannosaurus rex is sleeping.

Complete the following questions using the above information about the dinosaurs:

  a. For each dinosaur, assign a propositional variable to represent the proposition that it is sleeping. Using these variables and the connectives $\wedge$, $\vee$, $\rightarrow$, and $\neg$, translate the rules (i)-(iii) above into propositional logic.

  b. We can add to the list above a fourth rule *that uses implication* which, together with the first three, guarantees that the brachiosaurus is always sleeping. Write such a rule in words (like we did for (i)–(iii)), then write its translation in propositional logic.

  Explain why adding this rule guarantees that the brachiosaurus is sleeping. What proof rules from lecture 4 did you use in this argument? (For example: and introduction, or elimination, modus ponens...?)

  c. Represent the following sentence as a formula of propositional logic: "if the tyrannosaurus rex and the brachiosaurus are sleeping, then the spinosaurus is not sleeping." Under what conditions is this sentence *false*? Why?

Solution:

a. Let T be "the tyrannosaurus rex is sleeping," V be "the velociraptor is sleeping," B be "the brachiosaurus is sleeping," and S be "the spinosaurus is sleeping."

    i. $T \vee V$

    ii. $T \rightarrow B$

    iii. $\neg V \rightarrow ((B \wedge S) \vee T)$

b. We can add the following rule:

    If the velociraptor is sleeping, then the brachiosaurus is sleeping.

We represent this in propositional logic as:

$$V \rightarrow B$$

Intuitively, we argue as follows: We know that the T-rex is sleeping or the velociraptor is sleeping. We can consider these two cases:

- If the T-rex is sleeping, then we know by rule (ii) that the brachiosaurus is sleeping.

- Otherwise, the velociraptor must be sleeping, so by the above rule the brachiosaurus must be sleeping.

So, in either case (one of which must actually hold), the brachiosaurus is sleeping.

The proof rules we used above are as follows: we used disjunction elimination to break our proof into cases, then in each case we used *modus ponens* (implication elimination) to obtain the conclusion from an implication.

c. We render this proposition as follows:

$$T \wedge B \rightarrow \neg S$$

We can see its truth values under all possible truth assignments using a truth

table:

T	B	S	$T \wedge B \to \neg S$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

From this, we conclude that the proposition is false whenever all three dinosaurs are sleeping. This is because, in both such cases, the antecedent (hypothesis) T ∧ B of the conditional is true but the consequent (conclusion) ¬S is false.

Problem 2

In this problem, we'll define and investigate the properties of a new logical connective, XOR, "exclusive or." XOR is a popular connective among computer scientists, especially those who design physical circuits, but it's neglected by mathematicians. We'll use the notation $\oplus$ for this connective.

(Note: you shouldn't use this connective on any CS22 assignments or assessments outside this problem, unless it's explicitly mentioned. We wear mathematician hats in this class.)

The connective $\oplus$ is defined to have the behavior that, given propositions p and q, we have that $p \oplus q$ is true if and only if *exactly* one of p or q is true (not both!). Otherwise, $p \oplus q$ is false.

   a. Write down two natural language English sentences that are sensibly translated to propositional logic using $\oplus$ and at least one more connective. Then write down the propositional logic translation of each English sentence. You should choose two sentences that don't translate to the same propositional formula!

   You may use any atomic propositions you'd like, but please tell us which natural language propositions correspond to which propositional letters.

   b. Let p and q be propositions. *Using only conjunction and negation*, write a propositional formula that is equivalent to $p \oplus q$. Verify this equivalence using truth tables. (You may want to use Lean to generate a truth table for your formula.)

   c. The *elimination rule* for $\oplus$ tells us how we can *use* the information that an XOR statement is true. Here is the elimination rule, written in natural language:

   > If you know $p \oplus q$ and your goal is g, then in order to prove your goal, it suffices to show $(p \land \neg q) \to g$ and $(\neg p \land q) \to g$.

   An *introduction rule* for $\oplus$ gives sufficient conditions to prove a goal of the form $p \oplus q$. What are the introduction rule(s) for $\oplus$? There may be more than one! You can write the rules in natural language, as we've written the elimination rule above.

   You may find it helpful to take inspiration from the elimination rule above and the introduction rules for disjunction ($\lor$).

> **Solution:**
>
>   a. For instance:

"For dinner I eat a salad and either pizza or rice." Let S, Z, R be the propositions "I eat a salad for dinner," "I eat pizza for dinner," "I eat rice for dinner." This translates to $S \wedge (Z \oplus R)$.

b. A valid choice of formula is $\neg(\neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q))$, which we can obtain by writing down the "natural" formula $(p \wedge \neg q) \vee (\neg p \wedge q)$ and then applying De Morgan's law in reverse. The truth table is as follows:

p	q	¬q	(p ∧ ¬q)	¬(p ∧ ¬q)	¬p	(¬p ∧ q)	¬(¬p ∧ q)	(¬(p ∧ ¬q) ∧ ¬(¬p ∧ q))	¬(¬(p ∧ ¬q) ∧ ¬(¬p ∧ q))
true	true	false	false	true	false	false	true	true	false
false	true	false	false	true	true	true	false	false	true
true	false	true	true	false	false	false	true	false	true
false	false	true	false	true	true	false	true	true	false

We see that this matches our truth table for XOR:

p	q	$p \oplus q$
T	T	F
F	T	T
T	F	T
F	F	F

c. There are two introduction rules. They are as follows:

- If you can show p and $\neg q$, then you can show $p \oplus q$.

- If you can show $\neg p$ and q, then you can show $p \oplus q$.

Problem 3

This problem is a Lean question!

This homework question can be found by navigating to
`BrownCs22/Homework/Homework01.lean` in the directory browser on the left of your
screen in your Codespace. 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, download the file to your computer
(right-click on the file in the Codespace directory browser and click "Download"),
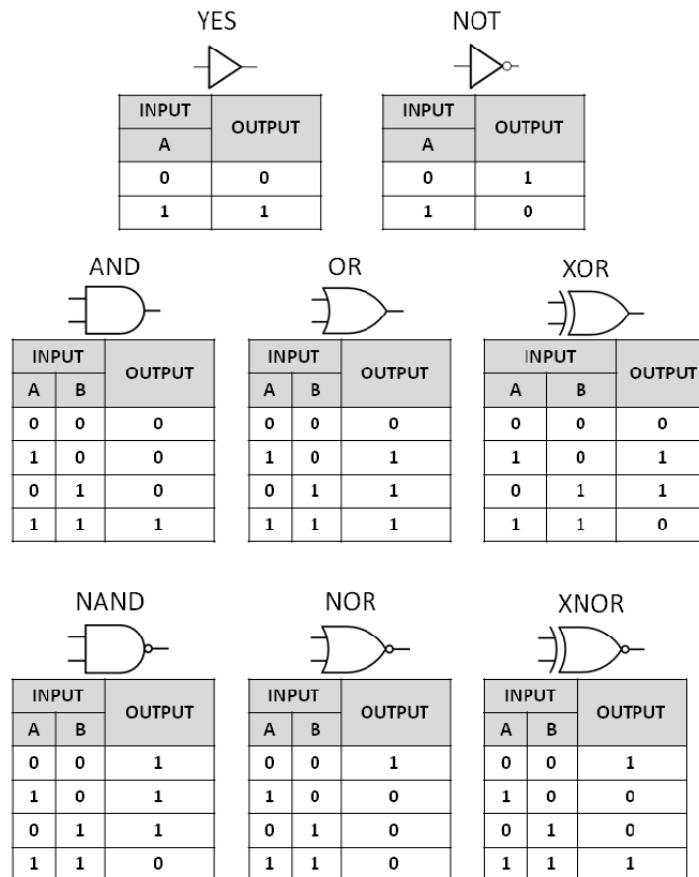and upload it to Gradescope.

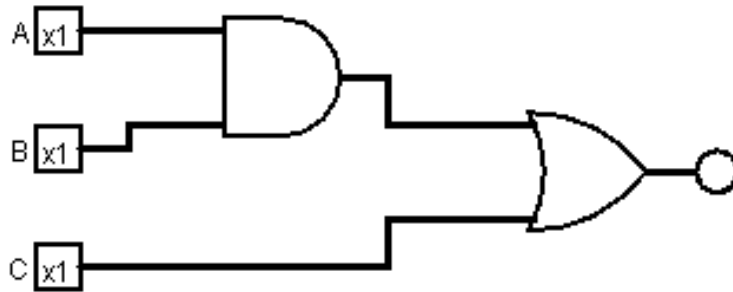# 🦕 Problem 4 (Mind Bender — *Extra Credit*)

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 represent the proposition $(A \wedge B) \vee C$ as the circuit

a. Draw circuits corresponding to the following propositions.

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

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

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

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


i.

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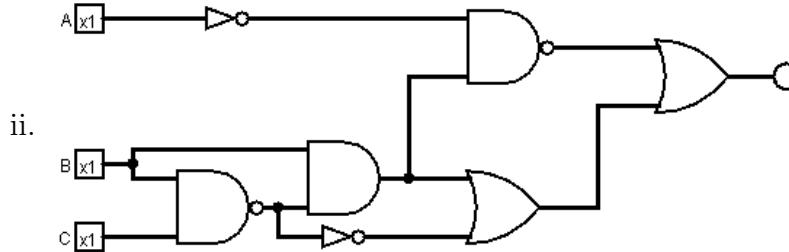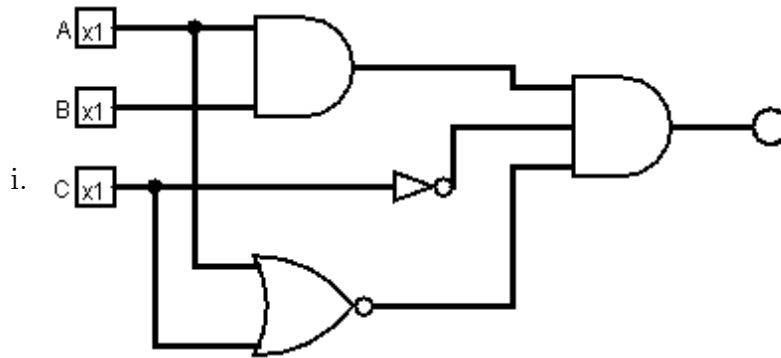
c. The NAND and NOR gates are also known as the universal logic gates since we can implement any of the other gates by just using the chosen universal gate.

    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. A simple example is an *addition circuit*, which takes in 2 one-digit Boolean numbers (representing 1 as true and 0 as false) and returns their sum. Draw a circuit that has this behavior.

HINT: Notice that while each input is only one Boolean digit, your output could have two! (We represent 2 as 10 in binary.) Therefore, utilizing two 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.
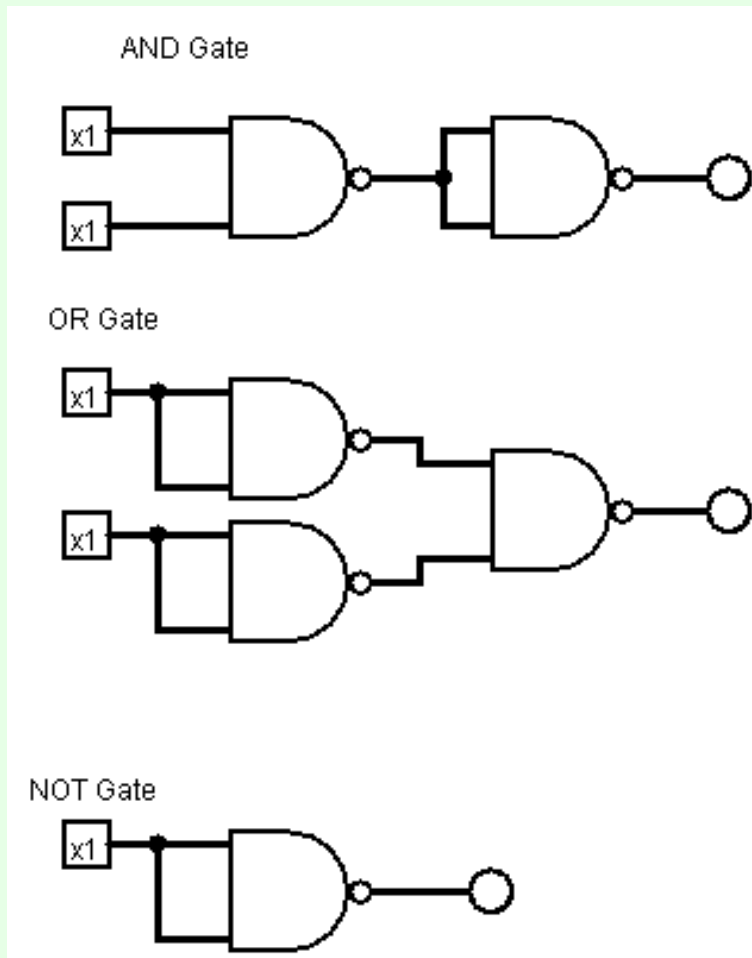
Solution:

a.   i.

ii.

9
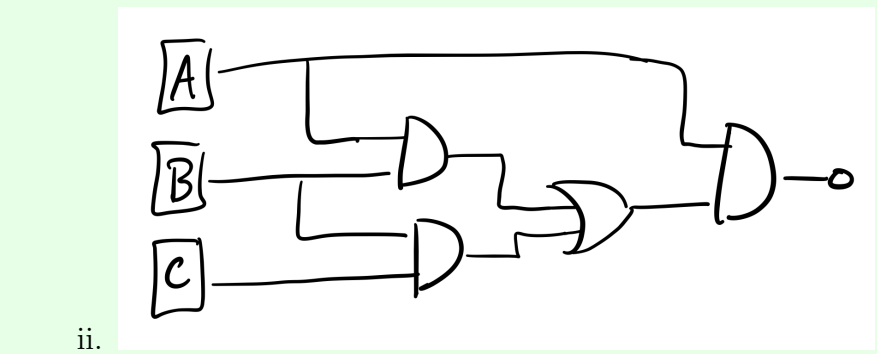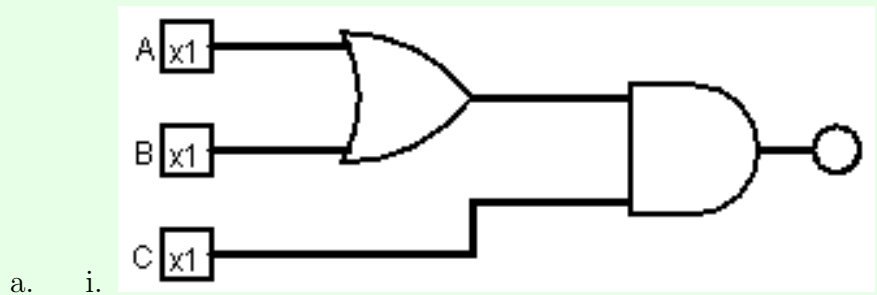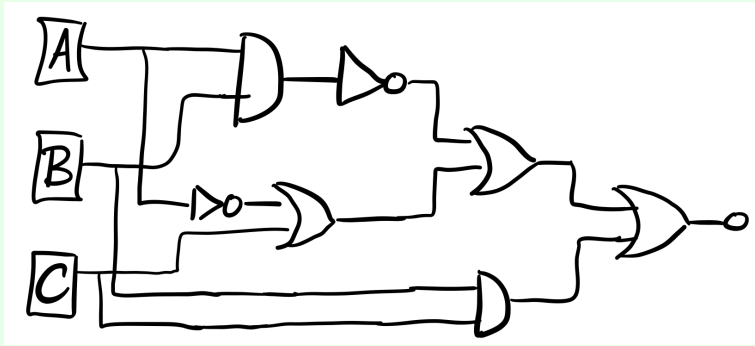
iii.

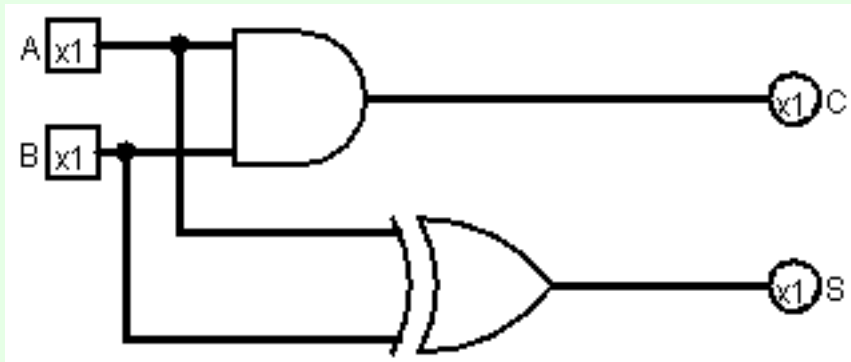b.   i. $(A \wedge B) \wedge \neg C \wedge \neg(A \vee C)$

A	B	C	Output
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	F

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

A	B	C	Output
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

AND Gate

OR Gate

NOT Gate

c. i.

  ii. Being able to make all circuits with one gate type allows for the manufacturing of circuits to be easier in terms of hardware, since we only need to make one type of gate rather than 7 gates.

d.