CS270: LAB #8

Normal Forms

You may work in teams of one or two people (three is acceptable in the event of an unscheduled absence). Unless stated otherwise, the lab is due to be submitted into Gradescope at the end of the day.
In order to receive credit, follow these instructions:

[a] Every team member should be discussing simultaneously the same problem – do NOT try to divvy up the labor and assign different problems to different students since the material is cumulative.

[b] Directly edit this lab PDF using Sedja/PDFescape with your answers (extra pages can be added in the rare event you need more than the allotted space)

[c] Each lab, rotate which member has the responsibility of being the Scribe. This is the person that is typing the answers and uploading the final PDF – note that only a single copy of the filled in PDF is turned into Gradescope. Only one lab needs to be submitted for the entire team, and all members receive the same score. Make sure to use a font that your PDF editor is compatible with (otherwise you might find your answers appear as weird shapes/sizes or simply disappear entirely!)

[d] The Gradescope submission must have each answer properly tagged with the appropriate question. Moreover, every member of the team must be listed as a submitter. Although it is the Scribe which executes these actions, it is still the responsibility of the entire team to make certain this is done properly (thus it is highly recommended that the Scribe share their screen so the entire team can witness it). Answers which are improperly tagged cannot be seen by the grader and thus cannot be scored.

[e] **FOR REMOTE ONLY**: Each lab, rotate which member has the responsibility of being the Recorder. This is the person who hits the Zoom Record button (once the technical permission is granted by the TA/RCF/Professor) and ensures that everyone has their camera/microphone on. They are also the member that is responsible to make sure the DrexelStream video is marked as viewable and entered into the https://tinyurl.com/VidLinkForm webform before 11:59pm (they should also email the rest of their team as confirmation.) Note that the video file doesn't get created/processed until after the Recorder has quit Zoom.

[f] Each lab, rotate which member has the responsibility of being the Manager. This is the person that ensures that everyone is participating equally and honestly, keeps the group on task, ensures that all team members understand a solution before going on to the next question, and presses the "hand up" button in Zoom to summon a TA or the professor (but they only do so after surveying the group to make sure everyone has the same question).

Team Name (CS pioneer): Marvin Minsky

Scribe name: Terie Ha

Recorder name: Evelyn Thai

Manager name: Lixiao Yang

Other team member (if any): Jerry Li

Question 1 : 4 points

There is some terminology required for this lab.

- **Literal**: A literal is a variable, constant, or negation of a variable. In the expression $\neg A \vee B$ both $\neg A$ and B would be literals.
- **Clause**: A clause is an expression or part of an expression made up of literals connected by an operator.
- **Disjunctive Clause**: A clause made using the disjunction (or) operator. For example, $(A \vee \neg B \vee C)$ is a disjunctive clause. All literals are connected by the disjunction operator.
- **Conjunctive Clause**: A clause made using the conjunction (and) operator. For example, $(\neg A \wedge \neg B \wedge C)$ is a conjunctive clause. All literals are connected by the conjunction operator.

Both of the below questions are about the following expression.

$$(\neg A \vee B) \wedge (C \vee \neg X)$$

(a) (2 points) List all the literals in the expression.

¬A, B, C, ¬X

(b) (2 points) The expression is made up of two disjunctive clauses. What are they?

(¬A ∨ B), (C ∨ ¬X)

Question 2 : 4 points

We also talk about operators being **above** or **below** each other. This related to order of operations.

For Boolean Expressions: parenthesis always come first, then negation (not), then conjunction (and), and finally disjunction (or).

If we say "AND is above OR" that means the AND operator is evaluated after the OR. In the expression $A \wedge (B \vee C)$ the parenthesis means the OR gets evaluated first. In this expression AND is above OR. We could also also OR is below AND.

You read "f is above g" to mean operator f gets evaluated after operator g.

(a) (2 points) Fill in the blank for the expression $(A \wedge X) \vee (B \wedge X)$

In this expression OR is above AND.

(b) (2 points) Fill in the blank for the expression $X \wedge (A \vee B)$

In this expression AND is above OR.

Question 3 : 20 points

In Boolean Logic, a formula is in **Negative Normal Form** if the negation operator is only applied to variables. The expression must only contain the operators: Conjunction ($\wedge$), Disjunction ($\vee$), and Negation ($\neg$).

An expression in **NNF** must meet the following conditions.

- Negation is only applied to variables
- Only binary operators allowed are $\wedge$ and $\vee$

For each of the following expressions, write if they are in Negative Normal Form.

If the expression is in Negative Normal Form write **NNF**.

If the expression is not in Negative normal form write $\neg$**NNF**.

Hint: If an expression does not break the rules of NNF, then it is in NNF.

(a) (2 points) $\neg(A \wedge B)$

(a) ¬NNF

(b) (2 points) $A \vee \neg B$

(b) NNF

(c) (2 points) C

(c) NNF

(d) (2 points) $\neg\neg B$

(d) ¬NNF

(e) (2 points) $A \vee \neg(A \wedge B)$

(e) ¬NNF

(f) (2 points) $\neg B$

(f) NNF

(g) (2 points) $(A \vee \neg B \vee C \wedge \neg A)$

(g) NNF

(h) (2 points) $A \wedge B$

(h) NNF

(i) (2 points) $(A \vee B) \wedge \neg(B \vee C)$

(i) ¬NNF

(j) (2 points) $A \implies B$

(j) ¬NNF

Question 4 : 12 points

Any Boolean Expression can be converted to **NNF** using a small set of rules.

To convert an expression to Negative Normal Form, we replace the left side with the right side in each of the following rules. By repeating this process we will reach Negative Normal Form. This is a proof, we just don't know what our conclusion is ahead of time. We can tell when we have reached the conclusion because we will have realized the expression is in Negative Normal Form.

Like the previous lab, we will be utilizing the Boolean Algebra rules (linked on our bbLearn course page)

An example simplification is given below.

1.	$\neg(A \implies \neg B)$	Premise
2.	$\neg(\neg A \vee \neg B)$	By Rule 20 (def of implication)
3.	$\neg\neg A \wedge \neg\neg B$	By Rule 19 (deMorgans #2)
4.	$A \wedge B$	By Rule 17 TWICE (double negation elimination)

Use the rules to convert each expression to NNF. Justify each step using the Rule Number.

(a) (4 points) $\neg\neg X$

1. ¬¬X premise
2. X by Rule 17 (double negative)

(b) (4 points) $X \vee \neg(X \wedge Y)$

1. X ∨ ¬(X ∧ Y) premise
2. X ∨ (¬X ∨ ¬Y) by Rule 19 (DeMorgan 2)
3. (X ∨ ¬X) ∨ ¬Y by Rule 2 (associativity)
4. True ∨ ¬Y by Rule 16 (complementation)
5. True by Rule 10 (annihilator)

(c) (4 points) $\neg(A \iff B)$

1. ¬(A⇔ B) premise
2. ¬((A⇒ B) ∧ (B⇒ A)) by Rule 21 (def. of Equiv.)
3. ¬((¬A∨B) ∧ (¬B∨A)) by Rule 20 (def of Implies)
4. (¬(¬A∨B) ∨ ¬(¬B∨A)) by Rule 18 (DeMorgan 1)
5. (¬¬A ∧ ¬B) ∨ (¬¬B ∧ ¬A) by Rule 19 (DeMorgan 2)
5. (A ∧ ¬B) ∨ (B ∧ ¬A) by Rule 17 (double negative)

Question 5 [10 points]
Use the technique learned in lecture to write a DNF expression that would have the following truth table. You must type out the actual logic symbols (points will be deducted for ascii or spelling it out in English). You do not need to simplify your answer. (note that here 0 represents False and 1 represents True)

P	Q	R	expr
1	1	1	0
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	1
0	0	0	0

$(P \wedge \neg Q \wedge R) \vee (P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge \neg Q \wedge R)$

Question 6 : 10 points [1 pt each part]

A Boolean Expression is in **Conjunctive Normal Form** if it is a **conjunction** of one or more clauses, where a clause is a **disjunction** of **literals**.

An expression in **CNF** form must meet the following requirements.

- In NNF
- All conjunctions are above all disjunctions

One additional rule is needed to convert an NNF expression into CNF. Always replace the left side with the right to simplify.

(8) $A \vee (X \wedge Y) = (A \vee X) \wedge (A \vee Y)$

For each of the below expressions determine if it is in CNF form.

If it is in CNF form write **CNF**.

If it is not in CNF form write ¬CNF otherwise.

(a) X

(a) CNF

(b) $\neg A$

(b) CNF

(c) A $\vee (X \wedge Y \wedge A)$

(c) ¬CNF

(d) (A $\vee B) \wedge (X \vee \neg\neg A)$

(d) ¬CNF

(e) A $\wedge \neg(B \vee X)$

(e) ¬CNF

(f) X $\vee Y \vee Z$

(f) CNF

(g) (A $\wedge B) \vee A$

(g) ¬CNF

(h) (A$\vee B) \wedge (X \vee \neg B)$

(h) CNF

(i) $\neg(A \implies B)$

(i) ¬CNF

(j) (A$\vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge X$

(j) CNF

Question 7 : 18 points

Simplify each of the following expression to **CNF**. Note which rule you use at each step by its number.

(a) (6 points) $A \vee (X \wedge (Y \wedge Z))$

1.A ∨ (X ∧ (Y ∧ Z)) premise
2. (A ∨ X) ∧ (A ∨ Y) ∧ (A ∨ Z) by rule 5 twice (distributivity)

(b) (6 points) $\neg(A \wedge (X \vee Y))$

1. ¬(A ∧ (X ∨ Y)) premise
2. ¬((A ∧ X) ∨ (A ∧ Y)) by rule 5 (distributivity)
3. (¬(A ∧ X) ∧ ¬(A ∧ Y)) by rule 19 (DeMorgan 2)
4. ((¬A ∨ ¬X) ∧ (¬A ∨ ¬Y)) by rule 18 (DeMorgan 1)

(c) (6 points) $\neg(A \wedge B) \vee \neg(C \implies D)$

1. ¬(A ∧ B) ∨ ¬(C ⇒ D) premise
2. ¬(A ∧ B) ∨ ¬(¬C ∨ D) by rule 20 (def. of implies)
3.(¬A ∨ ¬B) ∧ ¬(¬C ∨ D) by rule 18 (DeMorgan 1)
4. (¬A ∨ ¬B) ∧ (¬¬C ∧ ¬D) by rule 19 (DeMorgan 2)
5. (¬A ∨ ¬B) ∧ (C ∧ ¬D) by rule 17 (Double negative)

Question 8 : 12 points

SAT in Browser

Complete this Section using the SAT Solver link in our bbLearn course page

MiniSat has a simple language based on CNF. It is called DiMACS.

It only accepts CNF expressions.

Example:

```
p cnf 4 2
1 2 0
-3 4 0
```

Translates to $(x_1 \vee x_2) \wedge (\neg x_3 \vee x_4)$

The first line says that the expression has 4 variables and 2 clauses.

Each Clause ends with a 0. You cannot use 0 as a variable.

When this problem is solved by MiniSat it prints out.

```
SATISFIABLE
v -1 2 -3 -4 0
```

This tells us to set x_1=F, x_2=T, x_3=F, x_4=F to make the statement true.

(a) (4 points) Use MiniSat to find a Satisfying assignment for the below expression.

$$(x_1 \vee x_2 \vee x_3) \wedge \neg x_2 \wedge \neg x_3$$

Write the solution provided by MiniSAT below. (Something like `v -1 2 -3 4 ...`)

SATISFIABLE

1 -2 -3 0

(b) (4 points) Covert the following expression to MiniSat CNF notation (DiMACS).

$$\neg (A \wedge B) \vee \neg (C \implies D)$$

1.¬ (A ∧ B) ∨ ¬ (C ⇒ D)	premise
2.(¬A ∨ ¬B) ∨ ¬(¬C ∨ D)	by rule 20 (def of implies)
3.(¬A ∨ ¬B) ∨ (¬¬C ∧ ¬D)	by rule 18 (DeMorgan Rule1)
4. (¬A ∨ ¬B) ∨ (C ∧ ¬D)	by rule 17 (Double negative)
5. (¬A ∨ ¬B ∨ C) ∧ (¬A ∨ ¬B ∨ ¬D)	by rule 6 (distributivity)

```
p cnf 4 2
-1 -2 3 0
-1 -2 -4 0
```

(c) (4 points) Use MiniSAT to find a satisfying assignment.
Write the solution provided by MiniSAT below. (Something like `v -1 2 -3 4 ...`)

SATISFIABLE

-1 -2 -3 -4 0

Question 9 : 10 points

Rewrite the following DiMACs expressions as Boolean Algebra Expressions.

(a) (2 points) Rewrite the following in as a Boolean Expression.

```
p cnf 4 2
1 -2 4 0
-4 -3 0
```

$(x1 \vee \neg x2 \vee x4) \wedge (\neg x4 \vee \neg x3)$

(b) (2 points) Rewrite the following in as a Boolean Expression.

```
p cnf 4 4
1 2 0
3 -2 0
2 4 0
-2 1 0
```

$(x1 \vee x2) \wedge (x3 \vee \neg x2) \wedge (x2 \vee x4) \wedge (\neg x2 \vee x1)$

(c) (2 points) Rewrite the following in as a Boolean Expression.

```
p cnf 5 1
1 -2 3 -4 5 0
```

$x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee x5$

(d) (2 points) Rewrite the following in as a Boolean Expression.

```
p cnf 2 2
-1 -2 0
1 2 0
```

$(\neg x1 \vee \neg x2) \wedge (x1 \vee x2)$

(e) (2 points) Rewrite the following in as a Boolean Expression.

```
p cnf 3 3
1 0
-2 0
3 0
```

$x1 \wedge \neg x2 \wedge x3$