

Homework 1: Propositional Logic

Due date: Friday July 28th at 11:59 PM

If you work with others (and you should!), remember to follow the collaboration policy outlined in the [syllabus](#). In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.

We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

1. Translations! [16 points]

Translate the English statements into symbolic logic. You will need to clearly define the propositions you use. Make sure the propositions you introduce are atomic (not the combination of smaller propositions).

- (a) If you are not visiting during business hours, then it is necessary to bring your Husky ID in order to enter CSE2.

Hint: remember propositions need to unambiguously assert something. In all propositions, you need a subject and verb—you may need to infer those from context, but be sure to include it explicitly in your propositions.

- (b) You should code exclusively in C++ only if you love total creative freedom and painful debugging sessions.
- (c) Define a set of *at most five* atomic propositions. Then, use them to translate all of these sentences about hand-balancing into logical notation. *Do not simplify the statements.* Note that we want you to translate the sentences as they appear; you should not add any knowledge about hand-balancing in doing the translation. [8 points]
- (i) If you can do a one-arm handstand, but you haven't spent hundreds of hours practicing, then you must have paid an influencer \$500 to learn "one weird trick."
- (ii) If you aren't a circus acrobat, but you've spent hundreds of hours practicing and can do a one-arm handstand, then you must have a weird passion for standing on your hands.
- (iii) You aren't a circus acrobat because you can't do a one-arm handstand.

2. Inequivalence [12 points]

For each part, find a truth assignment (i.e. an assignment of True or False to the variables) to show the pair of statements are not equivalent. State your assignment and then explain why your assignments work (our explanations are 1-2 sentences).

- (a) $p \rightarrow q$ vs. $q \rightarrow p$
- (b) $(p \vee q) \wedge r$ vs. $p \vee (q \wedge r)$
- (c) $p \rightarrow (q \rightarrow r)$ vs. $p \rightarrow (q \vee r)$
- (d) $(p \vee q) \wedge \neg r$ vs. $(p \wedge \neg q) \vee (\neg p \wedge \neg r)$

3. Compound Proposition [7 points]

Find a compound proposition involving the variables $a, b,$ and c that is true precisely when **exactly** one of the two conditions below is true (and false otherwise)

- at least two of $a, b,$ and c are true
- a is false and at least one of b, c is true

Note the “exactly one” above; for example, we would want the statement to evaluate to false if a is false and both b, c are true (as then both conditions would hold, and we want exactly one to hold). In addition to writing an expression, explain why your answer works (1-2 sentences).

For this problem, you may only use the logical connectives: $\neg, \vee, \wedge, \rightarrow$

Hint: There are a few different ways to approach this question, but the easiest is probably to think of all the settings which will meet the conditions and make a statement that properly combines the elements of your list.

4. Be my Bool [14 points]

The Java project you are working on contains the following function. It takes four boolean arguments, $a, b, c,$ and $d,$ and returns true if at least two of them are true or if $b, c,$ and d are all false:

```
public static boolean E(boolean a, boolean b, boolean c, boolean d) {
    if (!b && !c && !d) {
        return true;
    } else {
        int count = 0;
        if (a) count += 1;
        if (b) count += 1;
        if (c) count += 1;
        if (d) count += 1;
        return 2 <= count;
    }
}
```

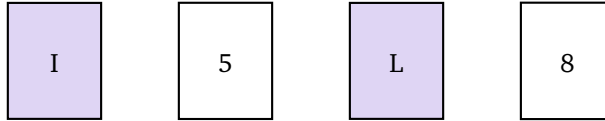
In this problem, you will write a simpler, faster version of this function.

- Write a truth table for E . Include columns showing whether at least two of the variables are true and whether $b, c,$ and d are all false before the final column showing the value of $E(a, b, c, d)$.
- Write a Boolean algebra expression for E in sum-of-products form (DNF).
- Write a Boolean algebra expression for E in products-of-sums form (CNF).
- Fill in the body of E with a Java expression that returns the same value as the original code.

```
public static boolean E(boolean a, boolean b, boolean c, boolean d) {
    return
}
}
```

5. As always, Aruna... [10 points]

You are presented with four *two-sided* (one purple, one white) cards (do you see what they spell?) :



On the purple side of each card is a letter, and on the white side is a number.

Consider the following rule:

If a card has a vowel on one side, then it has an odd number on the other side.

Which card(s) *must* be turned over to check if the rule is true? Explain your answer in a few sentences.

6. Same to you [20 points]

Prove the following assertions using a sequence of logical equivalences.

Hint: For equivalences where one side is much longer than the other, a good heuristic is to start with the longer side and try to apply the rules that will shorten it. These are Identity, Domination, Absorption, Negation, and Double Negation. (My solutions, put together, use every one of those rules at least once.)

Hint for (d): if you find yourself wanting to transform $\neg T$ into F, you may want to back up and try another approach to solving the problem. It is possible to prove $\neg T \equiv F$ using equivalences, but it takes a bit of work.

(a) $P \rightarrow (Q \rightarrow R) \equiv (P \wedge Q) \rightarrow R$

(b) $(\neg P \rightarrow Q) \wedge (Q \rightarrow P) \equiv P$

(c) $(P \rightarrow Q) \vee (P \rightarrow \neg Q) \equiv T$

(d) $((P \vee \neg P) \rightarrow P) \wedge (Q \rightarrow P) \equiv P$

7. [Extra Credit] The Poisoned Cookie

Imagine you're given 12 nearly identical cookies, with 11 of them being perfectly normal delicious cookies. However, one of them is poisoned. To determine which one is poisoned, you know that the poisoned one is slightly lighter than the rest, but by such a small amount that you won't be able to tell without a scale. You have a balancing scale, which you can use to compare the weights of the cookies against each other. Describe a way to find the poisoned cookie using at most 3 measurements of the scale.

You can measure multiple cookies at a time, for instance you could place 3 cookies on each side of the scale, if none of them are the poisoned cookie then the scale will perfectly balance. However, if one of the cookies is poisoned, then the scale will tip to the other side.

8. Feedback [1 point]

Answer these questions on the separate gradescope box for this question.

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment (excluding any extra credit questions, if applicable)? Report your estimate to the nearest hour.
- Which problem did you spend the most time on?
- Any other feedback for us?