

## Problem Set 2

---

This second problem set explores mathematical logic. We've chosen the questions here to help you get a more nuanced understanding for what first-order logic statements mean (and, importantly, what they don't mean) and to give you a chance to practice your proofwriting. By the time you've completed this problem set, we hope that you have a much better grasp of mathematical logic and how it can help improve your proofwriting structure.

Before attempting this problem set, we recommend that you do the following:

- Familiarize yourself with the online Truth Table Tool and play around with it a bit to get a feel for the propositional connectives.
- Read the online “Guide to Negations” and “Guide to First-Order Translations.”
- Read Handout “First-Order Translation Checklist,” to get a better sense for common errors in first-order logic translations and how to avoid them. *We will be running these checklists on your translations, so please be sure to double-check your work before submitting!*

**Checkpoint Questions Due Monday, January 21<sup>st</sup> at 2:30PM.**  
*No checkpoint problems! Enjoy your Martin Luther King Jr Day holiday.*

**Remaining Questions Due Friday, January 25<sup>th</sup> at 2:30PM.**

## Problem One: Propositional Completeness

In lecture, we covered the seven propositional connectives, which for convenience we've listed below:

$$\wedge \quad \vee \quad \neg \quad \rightarrow \quad \leftrightarrow \quad \top \quad \perp$$

We settled on this set of connectives because they're convenient and expressive. However, it turns out that we could get away with fewer connectives than this.

- i. Show how to rewrite the expressions  $\perp$ ,  $p \rightarrow q$ , and  $p \leftrightarrow q$  purely using the remaining four connectives ( $\wedge$ ,  $\vee$ ,  $\neg$ , and  $\top$ ), along with the variables  $p$  and  $q$ . This means that those four propositional connectives are *sufficient* – the other three connectives can be rewritten purely in terms of them.
- ii. There's even some redundancy in the four connectives you've used so far! Show how to rewrite the expression  $p \vee q$  without using any connectives besides  $\wedge$ ,  $\neg$ , and  $\top$ . This shows that those three connectives are actually all you need to cover all possible formulas.

We can push this further. You can rewrite any propositional formula without using any connectives besides  $\rightarrow$  and  $\perp$ .

- iii. Find a formula that's equivalent to  $\top$  and doesn't use any connectives besides  $\rightarrow$  and  $\perp$ .
- iv. Find a formula that's equivalent to  $\neg p$  and uses only the variable  $p$ , and doesn't use any connectives besides  $\rightarrow$  and  $\perp$ .
- v. Find a formula equivalent to  $p \wedge q$  that uses only  $p$  and  $q$ , and doesn't use any connectives besides  $\rightarrow$  and  $\perp$ .

*As a hint, what happens if you negate an implication?*

To recap: given the  $\rightarrow$  and  $\perp$  connectives, you can express  $\wedge$ ,  $\neg$ , and  $\top$ . From  $\wedge$ ,  $\neg$ , and  $\top$  you can get  $\wedge$ ,  $\vee$ ,  $\neg$ , and  $\top$ . And from those four connectives, you can get back the original seven. Overall, any propositional formula can be expressed purely in terms of  $\rightarrow$  and  $\perp$ . Nifty!

## Problem Two: Designing Propositional Formulas

Federal appeals courts in the United States typically assign three judges to hear each case. Each judge decides whether to *affirm* the decision (uphold the lower court's ruling) or *reverse* the decision (invalidate the lower court's ruling). The court as a whole then decides whether to affirm or reverse by going with the majority of the judges, so if two vote to affirm and one votes to reverse, the court decides to affirm.

Let  $a$ ,  $b$ , and  $c$  be propositional logic variables representing the votes of each of the judges in a particular case. Each variable is true if the judge voted to affirm, and is false if the judge voted to reverse.

- i. Write a truth table that shows the court's overall decision as a function of  $a$ ,  $b$ , and  $c$ . The output should be "true" if a majority of the judges vote to affirm, and should be "false" if a majority of the judges vote to reverse.
- ii. Design a propositional logic formula that has the above truth table. For full credit, your formula should use a total of *four* connectives.

Imagine a room with three light switches that all control the same light. If the light is on and any switch is flipped, the light turns off, and if the light is off and any switch is flipped, the light turns on. Additionally, the light is on when all three switches are in the "on" position. Let  $a$ ,  $b$ , and  $c$  be propositional variables that represent whether the first, second, and third light switches are in the "on" position, respectively.

- iii. Write a truth table that shows whether the light is on as a function of  $a$ ,  $b$ , and  $c$ . An output of "true" means that the light is turned on, and an output of "false" means that the light is turned off.
- iv. Design a propositional logic formula that has the above truth table. For full credit, your formula should use a total of *two* connectives (two connectives in total, not two different connectives; for example,  $(a \wedge b \wedge c \rightarrow a \wedge b)$  has four connectives, so that would be too many for this problem).

*Which connectives have truth values that flip when one of their inputs change?*

And now for a very fun, and very optional, coda to the previous exercise. As you've probably heard, internally, all numbers inside the computer are expressed using the **binary number system**. If you haven't seen binary numbers and binary arithmetic before, you're in for a treat! Stanford lecturer Chris Gregg has a great introduction to them available [here](#).

We can add binary numbers by using a variation of the regular algorithm for base-10 arithmetic in which we scan across the bits numbers from right to left, adding the numbers in each column and propagating a carry bit. For example, to the right we've shown how to add  $154 + 22$  (which, in binary, is  $100110101 + 00010110$ ). Here, the two input numbers are shown in purple and gold, the values carried across the columns are shown in teal, and the overall sum is in gray. (We've included a carry of zero in spots where there is no carry, just to show that it's there.)

$$\begin{array}{r}
 \begin{array}{cccccccccc}
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \\
 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & & \\
 + & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & \\
 \hline
 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & & \\
 \end{array}
 \end{array}$$

$t$     $c$   
 $a$   
 $b$   
 $s$

The calculation of adding these numbers proceeds one column at a time. In each column, we have three inputs: two bits from the numbers that we're adding (call them  $a$  and  $b$ ), plus one extra bit for the carry from the last column (call it  $c$ ). We also have two outputs: one bit representing the column sum (call it  $s$ ) and one bit representing the carry (call it  $t$ ).

And here's something really, really cool. Take a minute to convince yourself that the value of the bit  $s$

is given by the lighting control propositional formula that you came up with in part (iv) of this problem, and that the carry bit  $t$  is given by the legal propositional formula that you came up with in part (ii) of this problem. In other words, if you took the propositional logic formulas from parts (ii) and (iv) and implemented them in hardware, you'd be able to add numbers! And in fact, that's how some processors work!

Find these questions interesting? Take EE108!

### Problem Three: Symmetric Differences

The set symmetric difference operator  $\Delta$  is perhaps the most unusual set combination operator. It's not as commonly used as the others, and its behavior can be a bit surprising. Fortunately, now that we have propositional logic at our disposal, we can get a slightly better handle on how it works.

- i. Imagine you have two sets  $A$  and  $B$ . Fill in the last column of the following truth table.

$x \in A$	$x \in B$	$x \in A \Delta B$
F	F	
F	T	
T	F	
T	T	

*Don't be weirded out by the fact that the inputs to the truth table have the form  $x \in A$  and  $x \in B$  rather than single letters like  $p$  and  $q$ . Remember – the statement “ $x \in A$ ” is a proposition: it's either true or false!*

- ii. Based on your answer to part (i) of this problem, fill in the following blank with a **logic formula** (reminder, that means that the only predicate you can use is  $\in$  (no using set union, intersection, etc). Aim to get the simplest formula you can.

$x \in A \Delta B$  happens precisely when \_\_\_\_\_

*That truth table should look really familiar. Use that to guide your search.*

- iii. Generalize your answer to part (ii) by filling in the following blank. For full credit, see if you can find a **logic formula** that uses exactly *two* connectives.

$x \in (A \Delta B) \Delta C$  happens precisely when \_\_\_\_\_

*If you've done this correctly, you should have a slight sense of déjà vu.*

- iv. Using your answer to part (iii) as a starting point, simplify the expression  $(A \Delta B) \Delta B$ . Briefly justify your answer. No formal proof is necessary.

Generally speaking, if you ever find yourself in possession of a strange set-theoretic expression involving the set combination operations  $\cap$ ,  $\cup$ ,  $-$ , or  $\Delta$ , you can use the above technique to get a slightly better handle at what you're looking in fact. In fact, many rules about how these set-theoretic operators work follow directly from propositional logic!

That result in part (iv) is an interesting one. You may want to keep it in mind for later in the quarter. ☺

## Problem Four: First-Order Negations

For each of the first-order logic formulas below, find a first-order logic formula that is the negation of the original statement. Your final formula must not have any negations in it except for direct negations of predicates. For example, the negation of the formula  $\forall x. (P(x) \rightarrow \exists y. (Q(x) \wedge R(y)))$  could be found by pushing the negation in from the outside inward as follows:

$$\begin{aligned} &\neg(\forall x. (P(x) \rightarrow \exists y. (Q(x) \wedge R(y)))) \\ &\exists x. \neg(P(x) \rightarrow \exists y. (Q(x) \wedge R(y))) \\ &\exists x. (P(x) \wedge \neg\exists y. (Q(x) \wedge R(y))) \\ &\exists x. (P(x) \wedge \forall y. \neg(Q(x) \wedge R(y))) \\ &\exists x. (P(x) \wedge \forall y. (Q(x) \rightarrow \neg R(y))) \end{aligned}$$

Show every step of the process of pushing the negation into the formula (along the lines of what is done above), and *please preserve the indentation from the original formula* as you go. You don't need to formally prove that your negations are correct.

We **strongly** recommend reading over the Guide to Negations before starting this problem.

- i.  $\exists p. (Problem(p) \wedge \forall g. (Program(g) \rightarrow \neg Solves(g, p)))$   
)
- ii.  $\forall x \in \mathbb{R}. \forall y \in \mathbb{R}. (x < y \rightarrow \exists q \in \mathbb{Q}. (x < q \wedge q < y))$   
)
- iii.  $(\forall x. \forall y. \forall z. (R(x, y) \wedge R(y, z) \rightarrow R(x, z))) \rightarrow (\forall x. \forall y. \forall z. (R(y, x) \wedge R(z, y) \rightarrow R(z, x)))$

*Be careful – make sure you understand how that formula is parenthesized before you try negating it.*

- iv.  $\forall x. \exists S. (Set(S) \wedge \forall z. (z \in S \leftrightarrow z = x))$   
)
- v.  $\forall k. (SixClique(k) \rightarrow \exists t. (Triangle(t, k) \wedge (\forall e. (Edge(e, t) \rightarrow Red(e)) \vee \forall e. (Edge(e, t) \rightarrow Blue(e))))$   
)  
)

## Problem Five: Vacuous Truths

A statement is called *vacuously true* if either

- it's an implication of the form  $P \rightarrow Q$  where  $P$  is false,
- it's a universally-quantified implication  $\forall x. (P(x) \rightarrow Q(x))$  where  $P(x)$  is never true, or
- it's a universally-quantified implication  $\forall x \in S. Q(x)$  where  $S$  is the empty set.

These statements are true *by definition*. In the first case, the truth table for the  $\rightarrow$  connective says the implication is true when the antecedent is false, and in the second and third cases we just define these sorts of statements to be true.

You might be wondering why exactly this is the case.

- i. Negate the propositional formula  $P \rightarrow Q$  and push the negations as deep as possible. Now, look at the resulting formula. Explain why when  $P \rightarrow Q$  is vacuously true, its negation is false.
- ii. Negate the first-order formula  $\forall x. (P(x) \rightarrow Q(x))$  and push the negations as deep as possible. Now, look at the resulting formula. Explain why if the original formula is vacuously true, then its negation is false.
- iii. Negate the first-order formula  $\forall x \in S. Q(x)$  and push the negations as deep as possible. Now, look at the resulting formula. Explain why if the original formula is vacuously true, then its negation is false.

Your answers to parts (i), (ii), and (iii) of this problem give another justification for vacuous truths. The three classes of statements given above have false negations in the indicated cases, and therefore to keep everything consistent we choose to define the original statements as (vacuously) true. (Generally speaking, consistency is considered a virtue in mathematics. We're making up the rules, after all, and it's a lot easier when we pick them to be as simple as possible!)

## Problem Six: Translating into Logic

In each of the following, write a statement in first-order logic that expresses the indicated sentence. Your statement may use any first-order construct (equality, connectives, quantifiers, etc.), but you **must** only use the predicates, functions, and constants provided. You do not need to provide the simplest formula possible, though we'd appreciate it if you made an effort to do so. ☺ We **highly** recommend reading the Guide to First-Order Logic Translations before starting this problem.

- i. Given the predicate

$Natural(x)$ , which states that  $x$  is a natural number

and the functions

$x + y$ , which represents the sum of  $x$  and  $y$ , and

$x \cdot y$ , which represents the product of  $x$  and  $y$

write a statement in first-order logic that says “for any  $n \in \mathbb{N}$ ,  $n$  is even if and only if  $n^2$  is even.”

*Try translating this statement assuming you have a predicate  $Even(x)$ . Then, rewrite your solution without using  $Even(x)$ . Numbers aren't a part of first-order logic, so you can't use the number 2 in your solution.*

- ii. Given the predicates

$Person(p)$ , which states that  $p$  is a person;

$Kitten(k)$ , which states that  $k$  is a kitten; and

$HasPet(o, p)$ , which states that  $o$  has  $p$  as a pet,

write an FOL statement that says “someone has exactly two pet kittens and no other pets.”

*Make sure your formula requires the person to have exactly two pet kittens; look at the lecture example of uniqueness as a starting point. Good questions to ask – is your formula false if everyone has exactly one pet kitten? Is it false if everyone has exactly three pet kittens?*

- iii. The **axiom of pairing** is the following statement: given any two distinct objects  $x$  and  $y$ , there's a set containing  $x$  and  $y$  and nothing else. Given the predicates

$x \in y$ , which states that  $x$  is an element of  $y$ , and

$Set(S)$ , which states that  $S$  is a set,

write a statement in first-order logic that expresses the axiom of pairing.

*What does the Guide to Logic Translations say about translating statements about sets?*

- iv. Given the predicates

$x \in y$ , which states that  $x$  is an element of  $y$ , and

$Set(S)$ , which states that  $S$  is a set,

write a statement in first-order logic that says “every set has a power set.”

*As a warm-up, solve this problem assuming you have a predicate  $X \subseteq Y$  that says that  $X$  is a subset of  $Y$ . Once you have that working, see if you can solve the full version of this problem.*

- v. Given the predicates

$Lady(x)$ , which states that  $x$  is a lady;

$Glitters(x)$ , which states that  $x$  glitters;

$SureIsGold(x, y)$ , which states that  $x$  is sure that  $y$  is gold;

$Buying(x, y)$ , which states that  $x$  buys  $y$ ; and

$StairwayToHeaven(x)$ , which states that  $x$  is a Stairway to Heaven;

write a statement in first-order logic that says “there's a lady who's sure all that glitters is gold, and she's buying a Stairway to Heaven.”

## Problem Seven: Hereditary Sets

Let's begin with a fun little definition:

A set  $S$  is called a **hereditary set** if all its elements are hereditary sets.

This definition might seem strange because it's self-referential – it defines hereditary sets in terms of other hereditary sets! However, it turns out that this is a perfectly reasonable definition to work with, and in this problem you'll explore some properties of these types of sets.

- i. Given the self-referential nature of the definition of hereditary sets, it's not even clear that hereditary sets even exist at all! As a starting point, prove that there is at least one hereditary set.

*Try writing out some sample sets and see what you find. You'll quickly get a better handle on how the definition behaves, which will help guide your search.*

- ii. Prove that if  $S$  is a hereditary set, then  $\wp(S)$  is also a hereditary set.

*After you've written up a draft of your proofs, take a minute to read over them and apply the criteria from the Proofwriting Checklist. Here are a few specific things to watch out for:*

- *If you want to prove in part (ii) that a set  $T$  is a hereditary set, you need to prove the statement "every element of  $T$  is a hereditary set." That's a universally-quantified statement. If you're proving it via a direct proof, you'll probably need to pick some arbitrary element  $x \in T$ , then prove that  $x$  is a hereditary set by making specific claims about the variable  $x$ . Read over your proof and make sure that (1) you've introduced a new variable to refer to some arbitrarily-chosen element of  $T$  and that (2) you're making specific claims about that variable, rather than talking in general about how elements of  $T$  behave. You may need to introduce multiple variables in the course of your proofs.*
- *A common mistake we see people make when they're just getting started is to restate definitions in the abstract in the middle of a proof. For example, we commonly see people say something like "since  $A \subseteq B$ , we know that every element of  $A$  is an element of  $B$ ." When you're writing a proof, you can assume that whoever is reading your proof is familiar with the definitions of relevant terms, so statements like the one here that just restate a definition aren't necessary. Instead of restating definitions, try to apply those definitions. A better sentence would be something to the effect of "Since  $x \in A$  and  $A \subseteq B$ , we see that  $x \in B$ ," which uses the definition to conclude something about a specific variable rather than just restating the definition.*
- *Although we've just introduced first-order logic as a tool for formalizing definitions and reasoning about mathematical structures, the convention is to **not** use first-order logic notation (connectives, quantifiers, etc.) in written proofs. In a sense, you can think of first-order logic as the stage crew in the theater piece that is a proof – it works behind the scenes to make everything come together, but it's not supposed to be in front of the audience. Make sure that you're still writing in complete sentences, that you're not using symbols like  $\forall$  or  $\rightarrow$  in place of words like "for any" or "therefore," etc.*

Hereditary sets are used in *foundational mathematics*, the study of how to assemble all mathematical structures from simple structures. If you'd like to learn more about them, take Math 161!

## Problem Eight: The Star-Drawing Saga, Part Three

On Problem Set One, you saw the Simple Star Criterion, which pins down what has to happen in order for a  $p$ -point star with a step size of  $s$  to be a simple star:

**Simple Star Criterion:** A  $p$ -point star with a step size of  $s$  is a simple star if and only if for every natural number  $n$ , there is an integer  $t$  such that  $n \equiv_p s \cdot t$ .

This criterion is a little bit tricky to work with because of the universal quantifier at the front (“for every natural number  $n$ ...”) It turns out that there’s an easier way to characterize simple stars.

Take a minute and try drawing out the *non*-simple stars  $\{6 / 2\}$ ,  $\{8 / 2\}$ ,  $\{10 / 4\}$ , and  $\{12 / 3\}$ . Because those aren’t simple stars, the star you draw doesn’t end up passing through all of the points on the perimeter. Notice that, specifically, *none* of those stars end up passing through point 1. Formally speaking, if we have the star  $\{p / s\}$ , then the star passes through point 1 on the perimeter precisely if there’s an integer  $t$  such that  $1 \equiv_p s \cdot t$ .

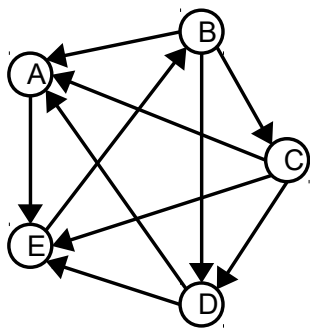
- i. Let  $p$  and  $s$  be natural numbers where  $p > 0$ . Using the simple star criterion, prove that the star  $\{p / s\}$  is simple if and only if there is an integer  $t$  where  $1 \equiv_p s \cdot t$ .

*Reminder: Q: How do you set up a proof of a biconditional statement? (A: write two proofs: one “forwards” and one “backwards”)*

*And a reminder from the previous problem – don’t use First-Order logic quantifiers or connectives in proofs. The rules for writing proofs are the same now as they were last week.*

## Problem Nine: Tournament Champions

Here’s one more problem to help you practice your proofwriting. It’s a classic CS103 problem, and we hope that you enjoy it!



A **tournament** is a contest among  $n$  players. Each player plays a game against each other player, and either wins or loses the game (let’s assume that there are no draws). We can visually represent a tournament by drawing a circle for each player and adding arrows between pairs of players to indicate who won each game. For example, in the tournament to the left, player  $A$  beat player  $E$ , but lost to players  $B$ ,  $C$ , and  $D$ .

A **tournament champion** is a player in a tournament who, for each other player, either won her game against that player, or won a game against a player who in turn won his game against that player (or both). For example, in the graph on the left, players  $B$ ,  $C$ , and  $E$  are tournament champions. However, player  $D$  is **not** a tournament champion, because he neither beat player  $C$ , nor beat anyone who in turn beat player  $C$ . Although player  $D$  won against player  $E$ , who in turn won against player  $B$ , who then won against player  $C$ , under our definition player  $D$  is **not** a tournament champion. (*Make sure you understand why!*)

- i. Let  $T$  be an arbitrary tournament and  $p$  be any player in that tournament. Prove the following statement: if  $p$  won more games than anyone else in  $T$  or is tied for winning the greatest number of games, then  $p$  is a tournament champion in  $T$ .

*This problem is a lot easier to solve if you draw the right picture. Something to think about: what happens if a player  $p$  isn’t a champion in a tournament? What would that mean about player  $p$ ? And finally, be careful not to make broad claims about tournaments or tournament structures without first proving them!*

*Once you’ve figured out the key insight and are writing up your answer, be precise with your terms.*

*For example, be careful about saying something like “player X beat player Y indirectly,” because reasonable people can disagree about what “indirectly” means. For example, in the above tournament, player D beats player E, who beats player B, who beats player C, who beats player A. In one sense, that means that player D beat player A indirectly. But D isn’t a tournament champion because that “indirectly” involves going through too many other players. If you want to talk about direct versus indirect wins, that’s fine, but be sure to precisely define what you mean! Similarly, be careful about talking about different sets of players that might have some overlap. For example, “the set of people that player C won a game against” and “the set of players that those people won games against” aren’t disjoint: player C beat player E, so E is in the first set, and player C also beat player A, who in turn beat player E, so player E is also in the second set.*

A corollary of the result you proved in part (i) is that every tournament with at least one player has at least one tournament champion – you can always pick someone who won the most games or is tied for winning the most games. However, note that you can be a champion without necessarily winning the most games – for example, player *E* in the above example is a champion, though player *E* only won a single game!

- ii. Prove that for any natural number  $n \geq 3$ , there is a tournament with exactly  $n$  players and exactly three champions.

*There’s a universal and an existential component to this theorem. How do you prove existential statements?*

### **Optional Fun Problem: Insufficient Connectives (Extra Credit)**

Every propositional logic formula could be written in terms of just  $\rightarrow$  and  $\perp$ . However, you *cannot* express every possible propositional logic formula using just the  $\leftrightarrow$  and  $\perp$  connectives. Prove why not.