Problem Set 3

This third problem set explores mathematical logic and graph theory. 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 some exposure to the rich world of graph theory. By the time you've completed this problem set, we hope that you have a much better grasp of first-order logic and an appreciation for some of the beautiful theory behind graphs.

As always, please feel free to drop by office hours, ask questions on Piazza, or send us emails if you have any questions. We'd be happy to help out.

This problem set has 35 possible points. It is weighted at 5% of your total grade.

Good luck, and have fun!

Checkpoint due Monday, April 20 at the start of lecture.

Assignment due Friday, April 24 at the start of lecture.

Write your solutions to the following problems and submit them electronically by Monday, April 20th at the start of class. These problems will be graded on a 0/1/2 basis based on the effort you have demonstrated in solving all of the problems. We will try to get these problems returned to you with feedback on your proof style this Wednesday, April 22nd.

Please make the best effort you can when solving these problems. We want the feedback we give you on your solutions to be as useful as possible, so the more time and effort you put into them, the better we'll be able to comment on your proof style and technique.

Checkpoint Problem: Love, Love Me Do (2 Points)

This question explores what happens when you interchange the order of quantifiers in a first-order logic statement.

Consider the following two statements in first-order logic:

$$\exists x \in P.\ \forall y \in P.\ \mathit{Loves}(x, y)$$

$$\forall y \in P.\ \exists x \in P.\ \mathit{Loves}(x, y)$$

Here, we'll imagine that P is a set of people, and $\mathit{Loves}(x, y)$ means that person x loves person y.

i. Translate these two statements into English.

ii. Prove that these statements are *not* equivalent to one another. To do so, find a group of people P where one of these statements is true and the other is false.

iii. One of these statements implies the other. Determine which statement implies the other, then prove it.

The rest of these problems should be completed and submitted by Friday, April 24th.

Problem One: Simplifying Propositional Formulas (3 Points)

All of the following propositional formulas can be significantly simplified. For each formula, find the simplest formula you can that's equivalent to it. Then, give an intuitive justification for why your new formula is equivalent to the original one. Your justification should not just be a truth table – see if you can give a concise explanation for why the formulas are equivalent.

i. $(p \to q) \land (p \to \neg q)$

ii. $(p \to q) \lor (q \to p)$

iii. $(p \lor q) \to (p \land q)$

Problem Two: Ternary Conditionals (3 Points)

Many programming languages support a *ternary conditional operator*. For example, in C, C++, and Java, the expression

$$x \ \texttt{?}\ y \ \texttt{:}\ z$$

means "evaluate the boolean expression x. If it's true, the entire expression evaluates to y. If it's false, the entire expression evaluates to z."

In the context of propositional logic, we can introduce a new *ternary* connective `?:` such that

$$p \ \texttt{?}\ q \ \texttt{:}\ r$$

means "if p is true, the connective evaluates to the truth value of q, and otherwise it evaluates to the truth value of r."

i. Based on this description, write a truth table for the `?:` connective.

ii. Find a propositional formula equivalent to $p \ \texttt{?}\ q \ \texttt{:}\ r$ that does not use the `?:` connective. Justify your answer by writing a truth table for your new formula. This shows that adding the `?:` connective to propositional logic does not enable propositional logic to express any concepts it couldn't already express.

It turns out that it's possible to rewrite any formula in propositional logic using only `?:`, $\top$, and $\bot$. The rest of this question will ask you to show this.

iii. Find a formula equivalent to $\neg p$ that uses only `?:`, $\top$, and $\bot$. No justification is necessary.

iv. Find a formula equivalent to $p \to q$ that uses only `?:`, $\top$, and $\bot$. No justification is necessary.

It turns out that the remaining connectives ($\lor$, $\land$, and $\leftrightarrow$) can be written purely in terms of $\neg$ and $\to$, so any propositional formula using the seven standard connectives can be rewritten using only the three connectives `?:`, $\top$, and $\bot$.

Problem Three: First-Order Negations (3 points)

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 $\forall x.\ (p(x) \to \exists y.\ (q(x) \land r(y)))$ could be found by pushing the negation in from the outside inward as follows:

$$\neg(\forall x.\ (p(x) \to \exists y.\ (q(x) \land r(y))))$$
$$\exists x.\ \neg(p(x) \to \exists y.\ (q(x) \land r(y)))$$
$$\exists x.\ (p(x) \land \neg\exists y.\ (q(x) \land r(y)))$$
$$\exists x.\ (p(x) \land \forall y.\ \neg(q(x) \land r(y)))$$
$$\exists x.\ (p(x) \land \forall y.\ (q(x) \to \neg r(y)))$$

Show every step of the process of pushing the negation into the formula (along the lines of what is done above). You don't need to formally prove that your negations are correct.

i. $\forall x \in \mathbb{R}.\ \forall y \in \mathbb{R}.\ (x < y \to \exists q \in \mathbb{Q}.\ (x < q \land q < y))$
ii. $(\forall x.\ \forall y.\ \forall z.\ (R(x, y) \land R(y, z) \to R(x, z))) \to (\forall x.\ \forall y.\ \forall z.\ (R(y, x) \land R(z, y) \to R(z, x)))$
iii. $\forall x.\ \exists S.\ (Set(S) \land \forall z.\ (z \in S \leftrightarrow z = x))$

Problem Four: 'Cause I'm Happy (4 Points)

In what follows, let's suppose that P is a nonempty set of people and that $Happy(x)$ means that person x is happy.

Below are six statements in first-order logic. Each of these statements is one of two types:

- The statement is true only when either everyone in P is happy or no one in P is happy.
- The statement is always true, regardless of which people in P are happy.

For each statement, determine which of the two types that statement is and write a short proof explaining why. Your proofs should definitely demonstrate why each statement is of the type you choose; for the purposes of this problem, disproving that a statement is of one type is not sufficient to establish that it must be of the other type.

i. $(\forall x \in P.\ Happy(x)) \to (\exists y \in P.\ Happy(y))$
ii. $(\exists x \in P.\ Happy(x)) \to (\forall y \in P.\ Happy(y))$
iii. $\forall x \in P.\ (Happy(x) \to (\exists y \in P.\ Happy(y)))$
iv. $\exists x \in P.\ (Happy(x) \to (\forall y \in P.\ Happy(y)))$
v. $\forall x \in P.\ \exists y \in P.\ (Happy(x) \to Happy(y))$
vi. $\exists y \in P.\ \forall x \in P.\ (Happy(x) \to Happy(y))$

Problem Five: Translating into Logic (7 points)

In each of the following, you will be given a list of first-order predicates and functions along with an English sentence. In each case, 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. ☺

i. Given the predicate

   Natural(x), which states that x is an 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."

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 a statement in first-order logic that says "someone has exactly two pet kittens and no other pets."

iii. Given the predicate

   Integer(x), which states that x is an integer

   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 "the square root of two is irrational."

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."

v. Given the predicates

   Lady(x), which states that x is a lady;
   Glitters(x), which states that x glitters;
   IsSureIsGold(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."*

* Let's face it – the lyrics to Led Zeppelin's "Stairway to Heaven" are impossible to decipher. Hopefully we can gain some insight by translating them into first-order logic!

Problem Six: The Raven Paradox (2 Points)

Go to Wikipedia and look up the *Raven Paradox*, a strange result that occurs when you start mixing deductive reasoning (formal logic) with inductive reasoning (observation and generalization). Talk about the Raven Paradox with your teammates (or with anyone you'd like, if you're working alone).

Write a short (one to two paragraph) essay with your thoughts about the Raven Paradox. We're looking for your honest thoughts and opinions here, and there's no need to use formal mathematical notation. Do you agree that there's something fishy going on with the Raven Paradox, or do you think there's nothing troubling about it? Why? There's no right or wrong answers here – we're genuinely interested to see what you think!

Problem Seven: Graph Coloring (3 Points)

The ***degree*** of a node v in a graph G is the number of edges that have v as an endpoint. In other words, it's the number of edges v is directly connected to. Interestingly, there usually isn't much of a connection between the degree of the nodes in a graph and the number of colors necessary to color that graph.

i. Give an example of a 2-colorable graph where some node has degree seven. Briefly justify why your graph meets these criteria; no proof is necessary.

ii. Give an example of a 2-colorable graph where every node has degree three. Briefly justify why your graph meets these criteria; no proof is necessary.

iii. Give an example of a graph where every node in the graph has degree two and the graph is 3-colorable, but the graph is *not* 2-colorable. Briefly justify why your graph meets these criteria; no proof is necessary.

Problem Eight: Tournament Cycles (4 Points)

A ***tournament*** is a directed graph with n nodes where there is exactly one edge between any pair of distinct nodes and there are no self-loops. Prove that if a tournament graph contains a cycle of any length, then it contains a cycle of length three.

Problem Nine: Bipartite Graphs (4 Points)

In lecture, we saw planar graphs as one special class of graphs with nice properties. Another class of graphs called ***bipartite graphs*** are used extensively in computer science.

An undirected graph $G = (V, E)$ is called ***bipartite*** if there is a way to partition the nodes V into two sets V_1 and V_2 so that every edge in E has one endpoint in V_1 and the other in V_2.

To help you get a better intuition for bipartite graphs, let's look at an example. Suppose that you have a group of people and a list of restaurants. You can illustrate which people like which restaurants by constructing a bipartite graph where V_1 is the set of people, V_2 is the set of restaurants, and there's an edge from a person p to a restaurant r if person p likes restaurant r.

Bipartite graphs have many interesting properties. One of the most fundamental is this one:

An undirected graph is bipartite iff it contains no cycles of odd length.

Intuitively, a bipartite graph contains no odd-length cycles because cycles alternate between the two groups V_1 and V_2, so any cycle has to have even length.

The trickier step is proving that if G contains no cycles of odd length, then G has to be bipartite. For now, assume that G has just one connected component; if G has multiple connected components, we can treat each one as a separate graph for the purposes of determining whether G is bipartite. (You don't need to prove this, but I'd recommend taking a minute to check why this is the case.)

Suppose G is an undirected graph with no cycles of odd length. Choose any node $v \in V$. Let V_1 be the set of all nodes that are connected to v by a path of odd length and V_2 be the set of all nodes connected to v by a path of even length.

i. Prove that V_1 and V_2 have no nodes in common.

ii. Using your result from part (i), prove that if G has no cycles of odd length, then G is bipartite.

Extra Credit Problem: Tournament Degrees (1 Point Extra Credit)

In a directed graph, the *indegree* of a node (denoted $deg^-(v)$) is the number of edges entering node v, and the *outdegree* of a node (denoted $deg^+(v)$) is the number of edges leaving node v.

Let $G = (V, E)$ be an arbitrary tournament graph. Prove that the sum of the squares of the indegrees of all the nodes in V is equal to the sum of the squares of the outdegrees of all nodes in V. Mathematically, prove that the following result is true for all tournament graphs G:

$$\sum_{v \in V} deg^-(v)^2 = \sum_{v \in V} deg^+(v)^2$$

(Here, the notation $\sum_{v \in V} deg^-(v)^2$ means "the sum of $deg^-(v)^2$ for all nodes $v \in V$). Note that this statement is not true about graphs in general; it's specific to tournaments.