Problem Set 3

This third problem set explores graphs, relations, functions, cardinalities, and the pigeonhole principle. This should be a great way to get a feel for how we define mathematical structures and what some of the consequences of those definitions are.

Start this problem set early. It contains seven problems (plus one checkpoint question, one survey question and one extra-credit problem), several of which require a fair amount of thought.

In any question that asks for a proof, you **must** provide a rigorous mathematical proof. You cannot draw a picture or argue by intuition. If we specify that a proof must be done a certain way, you must use that particular proof technique; otherwise you may prove the result however you wish.

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

This problem set has 150 possible points. It is weighted at 7% of your total grade. The earlier questions serve as a warm-up for the later problems, so be aware that the difficulty of the problems does increase over the course of this problem set.

Good luck, and have fun!

Checkpoint due Monday, October 15 at 2:15PM

Assignment due Friday, October 19 at 2:15PM

Write your solutions to the following problems and submit them by Monday, October 15th at the start of class. These problems will be graded based on whether or not you submit solutions, rather than the correctness of your solutions. We will try to get these problems returned to you with feedback on your proof style this Wednesday, October 17th. Submission instructions are on the last page of this problem set.

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. **Note that this question has four parts.**

Checkpoint Problem: Inverse Relations (25 Points if Submitted)

If R is a binary relation over a set A, the relation R^{-1} (called the *inverse of R*) is the binary relation

$$aR^{-1}b \quad \text{iff} \quad bRa$$

For example, the inverse of $<$ is the relation $>$, since $x > y$ iff $y < x$.

i. What three properties must a relation satisfy to be an equivalence relation?
ii. Prove or disprove: If R is an equivalence relation, then R^{-1} is an equivalence relation.
iii. What three properties must a relation satisfy to be a partial order?
iv. Prove or disprove: If R is a partial order, then R^{-1} is an partial order.

The rest of these problems should be completed and submitted by Friday, October 19.

Problem One: Relations over Polygons (12 Points)

In what follows, if p is a polygon, then let $A(p)$ denote its area.

i. Define the relation $=_A$ over the set of all polygons as follows: if x and y are polygons, then $x =_A y$ iff $A(x) = A(y)$. Is $=_A$ an equivalence relation? If so, prove it. If not, explain why not.

ii. Define the relation $\leq_A$ over the set of all polygons as follows: if x and y are polygons, then $x \leq_A y$ iff $A(x) \leq A(y)$. Is $\leq_A$ a partial order? If so, prove it. If not, explain why not.

Problem Two: Meet Semilattices (20 points)

Most compilers these days don't just compile code; they improve it as well. Compilers that improve the programs they generate are called *optimizing compilers*.

Many compiler optimizations are based on a mathematical structure called a *meet semilattice*. A meet semilattice is an ordered pair $(D, \wedge)$, where D is a set of values and $\wedge$ is a binary operator called a *meet operator* that can be applied to pairs of those values. Although the symbol $\wedge$ is the same one that we will use for logical AND, in the context of meet semilattices we use $\wedge$ to represent the meet operator. For example, we would read $x \wedge y$ as "x meet y" rather than "x and y."

In order for $(D, \wedge)$ to be a meet semilattice, the following properties must hold of D and $\wedge$:

- D must be ***closed under*** $\boldsymbol{\wedge}$: If $x, y \in D$, then $x \wedge y \in D$.
- $\wedge$ must be ***idempotent***: If $x \in D$, then $x \wedge x = x$.
- $\wedge$ must be ***commutative***: If $x, y \in D$, then $x \wedge y = y \wedge x$.
- $\wedge$ must be ***associative***: If $x, y, z \in D$, then $x \wedge (y \wedge z) = (x \wedge y) \wedge z$.

As an example, the function **`min`** over $\mathbb{R}$, which takes in two real numbers and returns the smaller one, is a meet semilattice. The set intersection operator $\cap$ is also meet semilattice over the set $\wp(\mathbb{N})$.

Amazingly, these four rules about the behavior of $\wedge$, which say nothing about how elements of D rank against one another, allow us to define a partial order over the elements of D. Given a semilattice $S = (D, \wedge)$, define the relation $\leq_S$ over D as follows:

$$x \leq_S y \quad \text{iff} \quad x \wedge y = x$$

i. Prove that for any meet semilattice $S = (D, \wedge)$, that $\leq_S$ is a partial order.

The $\leq_S$ relation defined above interacts with $\wedge$ in interesting ways.

ii. Prove that for any $x, y \in D$, that $x \wedge y \leq_S x$ and $x \wedge y \leq_S y$. This proves that $x \wedge y$ is a *lower bound* of x and y.

iii. Prove that for any $x, y \in D$, that if $z \leq_S x$ and $z \leq_S y$, then $z \leq_S x \wedge y$. This proves that $x \wedge y$ is a *greatest lower bound* of x and y.

In the context of program analysis, semilattices give a mathematical model of what information is known about a program at a particular point. Larger values indicate more precise information about the program, while smaller values indicate less precise information. The meet operator then gives a mechanism for combining pieces of information together in a way that preserves as much information as possible. If you're curious how semilattices are used this way, consider taking CS243.

Problem Three: $|\mathbb{Z}| \neq |\mathbb{N}|$? (8 Points)

Recall from lecture that $|\mathbb{Z}| = |\mathbb{N}|$. But below, we have a proof of the opposite – that $|\mathbb{Z}| \neq |\mathbb{N}|$.

Theorem: $|\mathbb{Z}| \neq |\mathbb{N}|$.

Proof: Consider the function $f : \mathbb{Z} \to \mathbb{N}$ defined as follows: $f(x) = |x|$, where $|x|$ is the absolute value of x. To see that this is a valid function from $\mathbb{Z}$ to $\mathbb{N}$, note that for any $x \in \mathbb{Z}$, that $|x|$ is an integer and $|x| \geq 0$. Thus $|x| \in \mathbb{N}$, so $f(x) = |x| \in \mathbb{N}$. Thus f is a legal function from $\mathbb{Z}$ to $\mathbb{N}$.

We now prove that f is surjective. To see this, consider any $n \in \mathbb{N}$. We will show that there is an $x \in \mathbb{Z}$ such that $f(x) = n$. In particular, set $x = n$. Since n is a natural number, we have that $n \geq 0$, so $|n| = n$. This means that $f(x) = |x| = |n| = n$, as required.

However, f is not injective. To see this, note that $f(1) = |1| = 1$ and $f(-1) = |-1| = 1$, but $-1 \neq 1$. Since f is not injective, it is not a bijection. Thus $|\mathbb{Z}| \neq |\mathbb{N}|$

Of course, this proof is incorrect and contains a fatal flaw. What's wrong with this proof?

Problem Four: Set Cardinalities (20 Points)

For each of the following, show that the indicated sets have the same cardinality.

i. Prove that if A, B, C, and D are sets where $|A| = |C|$ and $|B| = |D|$, then $|A \times B| = |C \times D|$. As a hint, since $|A| = |C|$, there is a bijection $f : A \to C$. There is a similar bijection $g : B \to D$.

ii. Using your result from (i), prove that $|\mathbb{N}^k| = |\mathbb{N}|$ for all nonzero $k \in \mathbb{N}$. This result means that for any nonzero finite k, there are the same number of k-tuples of natural numbers as natural numbers. You might want to use some of the results we proved in lecture as a starting point.

Problem Five: Infinite Binary Sequences (20 Points)

An ***infinite binary sequence*** is an infinite sequence of `0`s and `1`s. For example, we can consider the infinite sequence of all zeros (`00000`...), the infinite sequence of all ones (`11111`...), or other sequences like these:

```
11011100101110111...
00110101000101000...
11110100100001000...
```

Let $\mathbb{B}^\omega$ denote the set of all all infinite binary sequences. In this problem, you will prove $|\mathbb{N}| < |\mathbb{B}^\omega|$, meaning that there are strictly more infinite binary sequences than there are natural numbers.

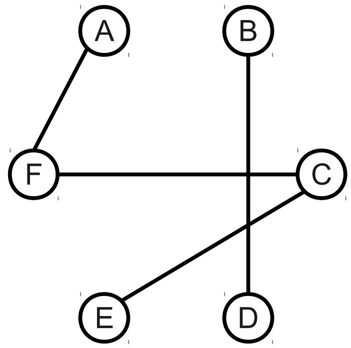
i. Prove that $|\mathbb{N}| \leq |\mathbb{B}^\omega|$ by finding an injection $f : \mathbb{N} \to \mathbb{B}^\omega$. You should formally prove that your function is an injection.

ii. Prove that $|\mathbb{N}| \neq |\mathbb{B}^\omega|$ using a proof by diagonalization, similar to the ones we covered in lecture. While it might help to draw a picture here, you should write your answer as a formal mathematical proof.

In writing up your proofs for (i) and (ii), feel free to introduce new mathematical notation if you think it would be appropriate. Just make sure to define what your notation means.

Problem Six: Pigeonhole Party! (20 Points)

Suppose that you are at a party. Any two people either have met (they are *acquaintances*) or have never met (they are *strangers*). We can therefore think of the party as an undirected graph where each node is a person and each edge connects a pair of acquaintances. For example, consider this party:



Here, person A just knows person F, person B just knows person D, and person C knows both person E and person F. However, none of A, B, or E know each other.

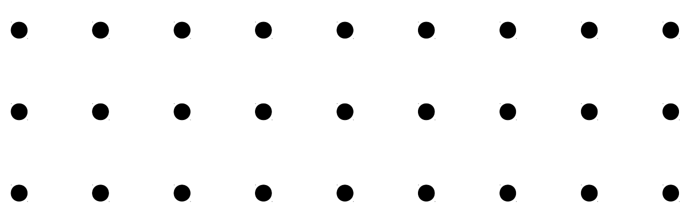
i. Show that at a party with at least two people present, there are at least two people with the same number of acquaintances at that party. *(Hint: Consider two cases: the case where someone knows no one else, and the case where everyone knows at least one person.)*

Let $\lceil x \rceil$ denote the smallest integer greater than or equal to x, so $\lceil 1 \rceil = 1$, $\lceil 1.37 \rceil = 2$, and $\lceil \pi \rceil = 4$. The ***generalized pigeonhole principle*** says that if there are n objects to be put into k boxes, then there must be some box that contains at least $\lceil n / k \rceil$ objects.*

ii. Using the generalized pigeonhole principle, show that in any group of six people that there are at least three mutual acquaintances or at least three mutual strangers. Three people are mutual acquaintances if each of them knows the other two, and three people are mutual strangers if each person does not know the other two. For example, in the above graph, A, B, and E are mutual strangers, but A, F, and C are **not** mutual acquaintances.

Problem Seven: Coloring a Grid (20 Points)

You are given a 3 × 9 grid of points, like the one shown below:



Suppose that you color each point in the grid either red or blue. Prove that no matter how you color those points, you can always find four points of the same color that form the corners of a rectangle.

---

* I'm not sure why you would be trying to put a whole bunch of pigeons into a small number of pigeonholes, but this result says what would happen if you did. Be nice to animals, folks.

Problem Eight: Course Feedback (5 Points)

We want this course to be as good as it can be, and we'd really appreciate your feedback on how we're doing. For a free five points, please answer the following questions. We'll give you full credit no matter what you write (as long as you write something!), but we'd appreciate it if you're honest about how we're doing.

i. How hard did you find this problem set? How long did it take you to finish? Does that seem unreasonably difficult or time-consuming for a five-unit class?

ii. Did you attend Monday's problem session? If so, did you find it useful?

iii. Did you read the online course notes? If so, did you find them useful?

iv. How is the pace of this course so far? Too slow? Too fast? Just right?

v. Is there anything in particular we could do better? Is there anything in particular that you think we're doing well?

Submission instructions

There are three ways to submit this assignment:

1. Hand in a physical copy of your answers at the start of class. This is probably the easiest way to submit if you are on campus.
2. Submit a physical copy of your answers in the filing cabinet in the open space near the handout hangout in the Gates building. If you haven't been there before, it's right inside the entrance labeled "Stanford Engineering Venture Fund Laboratories." There will be a clearly-labeled filing cabinet where you can submit your solutions.
3. Send an email with an electronic copy of your answers to the submission mailing list (cs103-aut1213-submissions@lists.stanford.edu) with the string "[PS3]" in the subject line.

Extra Credit Problem: $\subseteq$ is Not Well-Founded (5 Points Extra Credit)

Chapter 5.4 of the online course notes covers *well-founded orders*.

Prove that $\subseteq$ is not a well-founded order over $\wp(\mathbb{N})$.