Problem Set 3

This third problem set explores graphs, relations, 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).

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.

Good luck, and have fun!

Checkpoint due Monday, October 14 at 2:15 PM

Assignment due Friday, October 18 at 2:15 PM

Write your solutions to the following problems and submit them by Monday, October 14th at the start of class. These problems will be graded on a 0/12/25 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, October 16th.

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: Relations over Polygons (25 Points)

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

i. What three properties must a relation have to be an equivalence relation?

ii. 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, prove why not.

iii. What three properties must a relation have to be a partial order?

iv. 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, prove why not.

The rest of these problems should be completed and submitted by Friday, October 18th.

Problem One: 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. Let $S = (D, \wedge)$ be an arbitrary meet semilattice. Prove that the relation $\leq_S$ is a partial order.

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

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

iii. Prove for all $x, y, z \in D$ that if $z \leq_S x$ and $z \leq_S y$, then $z \leq_S x \wedge y$. This proves the value $x \wedge y$ is the *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 according to $\leq_S$ 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 Two: The Six-Color Theorem (20 Points)

In lecture, we talked about the *four-color theorem*, which says that every planar graph is 4-colorable. It took mathematicians over one hundred years to prove the four-color theorem, and the theorem was only proved with the help of computers.

Although the four-color theorem required computers to prove, it's possible to prove a slightly weaker result without such aid: every planar graph is 6-colorable, meaning that it's possible to color the nodes of any planar graph one of six different colors so that no two nodes connected by an edge are the same color as each other. This theorem is called the *six-color theorem*.

Prove the six-color theorem. You might want to use the following fact: every planar graph with at least one node has at least one node with degree five or less (the ***degree*** of a node is the number of edges connected to it). Don't worry about graphs with edges from nodes to themselves; you don't need to consider that case.

(Hint: Use induction on the number of nodes in the graph.)

Problem Three: Tournament Cycles (20 Points)

Recall from Problem Set Two that a tournament graph is a directed graph with $n \geq 1$ nodes where there is exactly one edge between any pair of distinct nodes and there are no self-loops.

The ***girth*** of a graph is defined to be the length of the shortest cycle in that graph, where the length of a cycle is measured by the number of edges it contains. If a graph has no cycles, then its girth is infinite.

Prove that if a tournament graph contains a cycle, then its girth is three. *(Hint: This is a great time to use a **proof by extremal case**. Consider the **smallest** cycle in a tournament graph containing a cycle and proceed by contradiction to show that it must have length three. Use the fact that you are considering the smallest cycle to derive the contradiction.)*

Problem Four: Complements and Connectivity (20 Points)

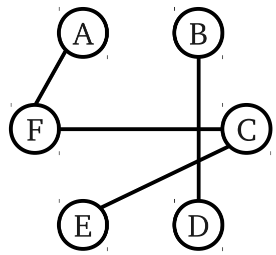
Let $G = (V, E)$ be an undirected graph. The ***complement*** of G, denoted G^c, is the graph $G^c = (V, E')$ that has the same nodes as V, but a different set of edges E' that we define as follows: for any $u, v \in V$, we have $\{u, v\} \in E'$ iff $\{u, v\} \notin E$. In other words, the set of edges in G^c is the set of edges not present in G and vice-versa.

An undirected graph $G = (V, E)$ is called ***connected*** iff for all nodes $u, v \in V$, the relation $u \leftrightarrow v$ holds.

Prove that for every undirected graph G, at least one of G and G^c must be connected. *(Hint: To prove a statement of the form "P or Q," you can instead prove the statement "if P is false, then Q is true." Show that if G isn't connected, then G^c must be connected.)*

Problem Five: 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.

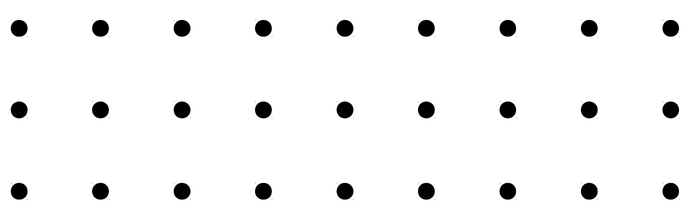
i. Prove that at a party with at least two people present, there are at least two people with the same number of acquaintances at that party. *(Hint: Consider two cases: the case where someone knows no one else, and the case where everyone knows at least one person.)*

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, prove 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, people A, B, and E are mutual strangers, but people A, C, and F are **not** mutual acquaintances.

Problem Six: 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 Seven: 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. We've put together a form online with five short feedback questions. For a free five points, please answer the questions online. 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. The link to the form is (https://docs.google.com/forms/d/1l-C5kOno73eR6FqsY_18gw02FhQPfZTRC-GyYtevVH4/viewform). We'll put a link to this form online at the course website, and we really appreciate your feedback!

Extra Credit Problem: Tournament Degrees (5 Points 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. In any graph, the sum of the indegrees of all the nodes equals the sum of the outdegrees of all the nodes, since every edge that leaves a node must enter some other node in the graph. Mathematically:

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

What's much less obvious is the following result: in any tournament graph, the sum of the *squares* of the indegrees of all the nodes is equal to the sum of the *squares* of the outdegrees of all nodes:

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

Prove that this statement holds for all tournament graphs.