hw6-proofs-cardinality-relations

CSE20S24

Due: 6/6/24 at 5pm (no penalty late submission until 8am next morning)

In this assignment, you will work with binary relations and prove properties about them, practicing proof strategies and applications.

Relevant class material: Weeks 9,10.

You will submit this assignment via Gradescope (https://www.gradescope.com) in the assignment called "hw6-proofs-cardinality-relations".

For all HW assignments: These homework assignments may be done individually or in groups of up to 3 students. Please ensure your name(s) and PID(s) are clearly visible on the first page of your homework submission, start each question on a new page, and upload the PDF to Gradescope. If you're working in a group, *submit only one submission per group*: one partner uploads the submission through their Gradescope account and then adds the other group member(s) to the Gradescope submission by selecting their name(s) in the "Add Group Members" dialog box. You will need to re-add your group member(s) every time you resubmit a new version of your assignment.

Each homework question will be graded either for **correctness** (including clear and precise explanations and justifications of all answers) or **fair effort completeness**. You may collaborate on "graded for correctness" questions only with CSE 20 students in your group; if your group has questions about a problem, you may ask in drop-in help hours or post a private post (visible only to the Instructors) on Piazza. For "graded for completeness" questions: collaboration is allowed with any CSE 20 students this quarter; if your group has questions about a problem, you may ask in drop-in help hours or post a public post on Piazza.

All submitted homework for this class must be typed. You can use a word processing editor if you like (Microsoft Word, Open Office, Notepad, Vim, Google Docs, etc.) but you might find it useful to take this opportunity to learn LaTeX. LaTeX is a markup language used widely in computer science and mathematics. The homework assignments are typed using LaTeX and you can use the source files as templates for typesetting your solutions.

Integrity reminders

- Problems should be solved together, not divided up between the partners. The homework is designed to give you practice with the main concepts and techniques of the course, while getting to know and learn from your classmates.
- You may not collaborate on homework questions graded for correctness with anyone other than your group members. You may ask questions about the homework in office hours (of the instructor, TAs, and/or tutors) and on Piazza (as private notes viewable only to the Instructors). You *cannot* use any online resources about the course content other than the class material from this quarter – this is primarily to ensure that we all use consistent notation and definitions (aligned with the textbook) and also to protect the learning experience you will have when the 'aha' moments of solving the problem authentically happen.
- Do not share written solutions or partial solutions for homework with other students in the class who are not in your group. Doing so would dilute their learning experience and detract from their success in the class.

In your proofs and disproofs of statements below, justify each step by reference to a component of the following proof strategies we have discussed so far, and/or to relevant definitions and calculations.

- A counterexample can be used to prove that $\forall x P(x)$ is **false**.
- A witness can be used to prove that $\exists x P(x)$ is **true**.
- **Proof of universal by exhaustion**: To prove that $\forall x\, P(x)$ is true when P has a finite domain, evaluate the predicate at **each** domain element to confirm that it is always T.
- **Proof by universal generalization**: To prove that $\forall x\, P(x)$ is true, we can take an arbitrary element e from the domain and show that $P(e)$ is true, without making any assumptions about e other than that it comes from the domain.
- To prove that $\exists x P(x)$ is **false**, write the universal statement that is logically equivalent to its negation and then prove it true using universal generalization.
- **Strategies for conjunction**: To prove that $p \wedge q$ is true, have two subgoals: subgoal (1) prove p is true; and, subgoal (2) prove q is true. To prove that $p \wedge q$ is false, it's enough to prove that p is false. To prove that $p \wedge q$ is false, it's enough to prove that q is false.
- **Proof of Conditional by Direct Proof**: To prove that the implication $p \rightarrow q$ is true, we can assume p is true and use that assumption to show q is true.
- **Proof of Conditional by Contrapositive Proof**: To prove that the implication $p \rightarrow q$ is true, we can assume $\neg q$ is true and use that assumption to show $\neg p$ is true.
- **Proof by Cases**: To prove q when we know $p_1 \vee p_2$, show that $p_1 \rightarrow q$ and $p_2 \rightarrow q$.

- **Proof by Structural Induction**: To prove that $\forall x \in X\, P(x)$ where X is a recursively defined set, prove two cases:

  Basis Step: Show the statement holds for elements specified in the basis step of the definition.

  Recursive Step: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

- **Proof by Mathematical Induction**: To prove a universal quantification over the set of all integers greater than or equal to some base integer b:

  Basis Step: Show the statement holds for b.

  Recursive Step: Consider an arbitrary integer n greater than or equal to b, assume (as the **induction hypothesis**) that the property holds for n, and use this and other facts to prove that the property holds for $n + 1$.

- **Proof by Strong Induction** To prove that a universal quantification over the set of all integers greater than or equal to some base integer b holds, pick a fixed nonnegative integer j and then:

  Basis Step: Show the statement holds for b, $b + 1$, $\ldots$, $b + j$.

  Recursive Step: Consider an arbitrary integer n greater than or equal to $b + j$, assume (as the **strong induction hypothesis**) that the property holds for **each of** b, $b + 1$, $\ldots$, n, and use this and other facts to prove that the property holds for $n + 1$.

- **Proof by Contradiction**

  To prove that a statement p is true, pick another statement r and once we show that $\neg p \rightarrow (r \wedge \neg r)$ then we can conclude that p is true.

  Informally The statement we care about can't possibly be false, so it must be true.

Assigned questions

1. Binary relations. In the review quiz, we considered the binary relation on $\mathbb{Z}^+$ defined by

$$\{(a, b) \mid \exists c \in \mathbb{Z}(b = ac)\}$$

Let's call that relation R_1. Consider the following other binary relations on $\mathbb{Z}^+$:

$$R_2 = \{(a, b) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid gcd(\ (a, b)\ ) = 1\}$$

$$R_3 = \{(a, b) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid a + b \leq 2024\}$$

Sample response that can be used as reference for the detail expected in your answer: To prove that R_1 is not symmetric we need to find a counterexample to

$$\forall x \in \mathbb{Z}^+ \forall y \in \mathbb{Z}^+ \, (\, (x,y) \in R_1 \rightarrow (y,x) \in R_1 \,)$$

Consider $x = 2$ and $y = 4$. Then $(x, y) \in R_1$ because $\exists c \in \mathbb{Z}(y = xc)$ is witnessed by the integer $c = 2$, since $4 = 2 \cdot 2$. However, $\exists c \in \mathbb{Z}(x = yc)$ is not true: to prove this, we consider an arbitrary integer c and consider the two (exhaustive) cases $c \geq 1$ and $c < 1$. In case 1, assume $c \geq 1$, and then $yc = 4c \geq 4 > 3$ so $yc \neq 2 = x$. In case 2, assume $c < 1$ so $c \leq 0$, and then $yc = 4c \leq 0$ so is not equal to 2, which is x. Thus, we've found positive integers x and y where $(x, y) \in R_1$ and $(y, x) \notin R_1$ so R_1 is not symmetric.

(a) (*Graded for completeness*) [1] Give example positive integers that are related according to each of these three relations. In other words, give example values (a_1, b_1), (a_2, b_2), and (a_3, b_3) such that $(a_i, b_i) \in R_i$ for each $i = 1, 2, 3$.

(b) (*Graded for correctness*) [2] Prove that R_2 is not transitive.

(c) (*Graded for correctness*) Prove that R_3 is not reflexive.

(d) (*Graded for correctness*) Prove that R_2 is symmetric.

(e) (*Graded for correctness*) Prove that R_3 is not anti-symmetric.

(f) (*Graded for completeness*) Give an example relation on $\mathbb{Z}^+$ that is both symmetric and anti-symmetric. Briefly justify your example.

2. Equivalence relations. Recall that we say a is **congruent to** b **mod** n means $(a, b) \in R_{(\textbf{mod } n)}$, that is ($a$ **mod** $n = b$ **mod** n). A common notation is to write this as $a \equiv b (\textbf{mod } n)$.

   A **modular inverse** of an integer x relative to modulus n (where n is a positive number and x is an integer between 0 and $n - 1$, inclusive) is defined to be an integer y between 0 and $n - 1$ (inclusive) such that $xy \equiv 1 (\textbf{mod } n)$.

   (a) (*Graded for completeness*) Fill in the multiplication table below so that each cell in row labelled a and column labelled b is an integer p in the set $\{0, 1, 2, 3, 4\}$ satisfying $ab \equiv p(\textbf{mod } 5)$. Then, use this multiplication table to find the **modular inverse** of each number in $\{0, 1, 2, 3, 4\}$ or to explain why it does not exist.

[1] This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers. To demonstrate your honest effort in answering the question, we expect you to include your attempt to answer *each* part of the question. If you get stuck with your attempt, you can still demonstrate your effort by explaining where you got stuck and what you did to try to get unstuck.

[2] This means your solution will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

	0	1	2	3	4
0	0	0	0	0	0
1	0				
2	0				
3	0				
4	0				

(b) (*Graded for correctness*) Find an example n and integer x between 1 and $n-1$ (inclusive) so that there is no modular inverse of x relative to n. Justify your example by exhausting through all the possible values of y between 0 and $n-1$ and showing that none of them have $xy \equiv 1 (\textbf{mod } n)$. *Note: 0 does not have a modular inverse relative to n. In this question, you're looking for a value of n where some nonzero number also doesn't have a modular inverse relative to n.*

3. (*Graded for correctness*) Partial orders. Recall the recursive definition of the set of linked lists of natural numbers (from class)

$$\begin{array}{ll} \text{Basis Step:} & [] \in L \\ \text{Recursive Step:} & \text{If } l \in L \text{ and } n \in \mathbb{N}, \text{ then } (n, l) \in L \end{array}$$

and the definition of the function which gives the length of a linked list of natural numbers $length : L \to \mathbb{N}$

$$\begin{array}{lll} \text{Basis Step:} & & length(\ []\ ) = 0 \\ \text{Recursive Step:} & \text{If } l \in L \text{ and } n \in \mathbb{N}, \text{ then} & length(\ (n, l)\ ) = 1 + length(l) \end{array}$$

For this question, we'll restrict our attention to just linked lists with data in the set $\{0, 1\}$ which have length 0, 1, or 2. Let's call this restricted set of lists L'.

Note: L' has 7 distinct elements.

Define two different partial orders with domain L'. For each of the partial orders, you can specify its definition by listing the set of ordered pairs with roster method, giving a set builder definition, giving a recursive definition, or using a clear and precise English description of which lists are related to one another. Then, draw the Hasse diagram for each of the two partial orders you defined.

4. Equivalence classes and partitions. Recall that in a movie recommendation system, each user's ratings of movies is represented as a n-tuple (with the positive integer n being the number of movies in the database), and each component of the n-tuple is an element of the collection $\{-1, 0, 1\}$. Assume there are five movies in the database, so that each user's ratings can be represented as a 5-tuple. We call Rt_5 the set of all ratings 5-tuples. Consider the binary relation on the set of all 5-tuples where each component of the 5-tuple is an element of the collection $\{-1, 0, 1\}$:

$$G = \{(u, v) \in Rt_5 \times Rt_5 \mid \text{the number of 1s in } u \text{ is the same as the number of 1s in } v\}$$

This is an equivalence relation (you do not need to prove this).

Recall that the **equivalence class** of an element $x \in X$ for an equivalence relation $\sim$ on the set X is the set $\{s \in X | (x, s) \in \sim\}$. We write this as $[x]_\sim$.

(a) (*Graded for correctness*) Find a ratings 5-tuple v such that $[v]_G = \{v\}$. Justify your choice of v.

(b) (*Graded for completeness*) Find distinct ratings 5-tuples u_1, u_2 ($u_1 \neq u_2$) whose equivalence classes $[u_1]_G$ and $[u_2]_G$ have the same size.

(c) (*Graded for completeness*) Find distinct ratings 5-tuples w_1, w_2 ($w_1 \neq w_2$) whose equivalence classes $[w_1]_G$ and $[w_2]_G$ have different sizes.

5. Modular exponentiation. Imagine you are playing the role of Alice in the Diffie Hellman key agreement (exchange) protocol. You and Bob have agreed to use the prime $p = 7$ and its primitive root $a = 3$. Your secret integer is $k_1 = 3$.

(a) (*Graded for fair effort completeness*[3]) Calculate the number you send to Bob, a^{k_1} **mod** p. Use the modular exponentiation algorithm for the calculation. Include a trace of the algorithm in your solution.

Modular Exponentation

```
1  procedure modular exponentiation(b: integer;
2                   n = (a_{k-1}a_{k-2}...a_1a_0)_2, m: positive integers)
3  x := 1
4  power := b mod m
5  for i:= 0 to k-1
6    if a_i = 1 then x:= (x · power) mod m
7    power := (power · power) mod m
8  return x {x equals b^n mod m}
```

(b) (*Graded for fair effort completeness*) Bob sends you the number 5. Compute your shared key, $\left(a^{k_2}\right)^{k_1}$ **mod** p. Hint: a^{k_2} **mod** p is what Bob sent you. Include all relevant calculations, annotated with explanations, for full credit.

(c) (*Graded for fair effort completeness*) What are some possible values for Bob's secret integer? What algorithm are you using to compute them?

[3]This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers.