

hw4-proofs-sets-recursion

CSE20W26

Due: 02/26/2026 at 5pm (late submission until 8am next morning)

In this assignment, you will analyze statements and determine if they are true or false using valid proof strategies. You will also determine if candidate arguments are valid. You will work with recursively defined sets and functions and prove properties about them, practicing induction and other proof strategies.

Relevant class material: Weeks 5, 6, and 7.

You will submit this assignment via Gradescope (<https://www.gradescope.com>) in the assignment called “hw4-proofs-sets-recursion”.

For all HW assignments: These homework assignments may be done individually or in groups of up to four 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.

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.

- Use only resources explicitly allowed for each assignment. Resources not affiliated with this quarter's version of the class may use inconsistent notation or definitions. When consulting resources, balance getting the help you need while also protecting 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.

When evaluating your homework solutions, we will be considering:

- The correctness of any final answers.
- Whether ideas are presented 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. A rule of thumb for the level of detail to include is to imagine someone who is taking the class alongside you but missed about a week's worth of content: does your writeup have enough detail for them to learn from it?

To demonstrate your honest effort in answering homework questions, 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 (e.g. reviewing annotated notes from class, checking the relevant textbook sections, looking up comparable questions on the relevant weekly review quizzes, asking questions on Piazza, attending (instructor/TA/tutor) office hours, consulting generative AI tools, etc.).

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 $b, b + 1, \dots, b + j$.
 - 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** $b, b + 1, \dots, n$, 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. Number properties. Consider the predicate $F(a, b) = \text{"}a \text{ is a factor of } b\text{"}$ over the domain $\mathbb{Z}^{\neq 0} \times \mathbb{Z}$. Consider the following quantified statements

- | | |
|--|---|
| (i) $\forall x \in \mathbb{Z} (F(1, x))$ | (v) $\forall x \in \mathbb{Z}^{\neq 0} \exists y \in \mathbb{Z} (F(x, y))$ |
| (ii) $\forall x \in \mathbb{Z}^{\neq 0} (F(x, 1))$ | (vi) $\exists x \in \mathbb{Z}^{\neq 0} \forall y \in \mathbb{Z} (F(x, y))$ |
| (iii) $\exists x \in \mathbb{Z} (F(1, x))$ | (vii) $\forall y \in \mathbb{Z} \exists x \in \mathbb{Z}^{\neq 0} (F(x, y))$ |
| (iv) $\exists x \in \mathbb{Z}^{\neq 0} (F(x, 1))$ | (viii) $\exists y \in \mathbb{Z} \forall x \in \mathbb{Z}^{\neq 0} (F(x, y))$ |

- (a) Which of the statements (i) - (viii) is being **proved** by the following proof:

By universal generalization, **choose** e to be an **arbitrary** integer. We need to show that $F(1, e)$. By definition of the predicate F , we can rewrite this goal as $\exists c \in \mathbb{Z} (e = c \cdot 1)$. We pick the **witness** $c = e$, which is an integer and therefore in the domain. Calculating, $c \cdot 1 = e \cdot 1 = e$, as required. Since the predicate $F(1, e)$ evaluates to true for the arbitrary integer e , the claim has been proved. $\square$

Hint: it may be useful to identify the key words in the proof that indicate proof strategies.

- (b) Which of the statements (i) - (viii) is being **disproved** by the following proof:

To disprove the statement, we need to find a counterexample. We choose 2, a nonzero integer so in the domain. We need to show that $\neg F(2, 1)$. By definition of the predicate F , we can rewrite this goal as $1 \bmod 2 \neq 0$. By definition of integer division, since $1 = 0 \cdot 2 + 1$ (and $0 \leq 1 < 2$), $1 \bmod 2 = 1$ which is nonzero so the counterexample works to disprove the original statement. $\square$

Hint: it may be useful to identify the key words in the proof that indicate proof strategies.

- (c) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\exists x \in \mathbb{Z}^{\neq 0} \exists y \in \mathbb{Z}^{\neq 0} (\neg(x = y) \wedge F(x, y) \wedge F(y, x))$$

- (d) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\forall x \in \mathbb{Z}^{\neq 0} \forall y \in \mathbb{Z}^{\neq 0} (F(x, y) \rightarrow \neg F(y, x))$$

- (e) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation) (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\exists x \in \mathbb{Z}^{\neq 0} \exists y \in \mathbb{Z} (F(x, y) \wedge F(x + 1, y) \wedge F(x + 2, y))$$

- (f) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\forall x \in \mathbb{Z}^{\neq 0} (F((x, x^2)) \wedge F((x, x^3)))$$

2. Structural induction. Recall that we define the set of bases as $B = \{\mathbf{A}, \mathbf{C}, \mathbf{U}, \mathbf{G}\}$. The set of RNA strands S is defined (recursively) by:

$$\begin{array}{ll} \text{Basis Step:} & \mathbf{A} \in S, \mathbf{C} \in S, \mathbf{U} \in S, \mathbf{G} \in S \\ \text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B, \text{ then } sb \in S \end{array}$$

where sb is string concatenation. The function $rnalen$ that computes the length of RNA strands in S is defined recursively by $rnalen : S \rightarrow \mathbb{Z}^+$

$$\begin{array}{ll} \text{Basis step:} & \text{If } b \in B \text{ then } rnalen(b) = 1 \\ \text{Recursive step:} & \text{If } s \in S \text{ and } b \in B, \text{ then } rnalen(sb) = 1 + rnalen(s) \end{array}$$

The function $basecount$ that computes the number of a given base b appearing in a RNA strand s is defined recursively:

$$\begin{array}{ll} & basecount : S \times B \rightarrow \mathbb{N} \\ \text{Basis Step:} & \text{If } b_1 \in B, b_2 \in B \quad basecount((b_1, b_2)) = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases} \\ \text{Recursive Step:} & \text{If } s \in S, b_1 \in B, b_2 \in B \quad basecount((sb_1, b_2)) = \begin{cases} 1 + basecount((s, b_2)) & \text{when } b_1 = b_2 \\ basecount((s, b_2)) & \text{when } b_1 \neq b_2 \end{cases} \end{array}$$

- (a) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\forall b \in B \exists s \in S (rnalen(s) = basecount((s, b)))$$

- (b) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\forall s \in S (rnalen(s) > basecount((s, \mathbf{A})))$$

- (c) Translate the statement to English, state whether is it true or false, and then justify your answer (by proving the statement or its negation). (*Graded for correctness of translation and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

$$\forall s \in S \forall b \in B (basecount((s, b)) > 0)$$

3. Games and induction. The game of Nim-Var is a two-player game (which is a variant of Nim). At the start of the game, there are two piles, each containing n jelly beans (n is a positive integer). On a player's turn, that player picks one of the two piles and does **one** of the following: either

- removes some positive number of jelly beans from that pile, **or**
- moves some positive number of jelly beans (that is less than the current total number of jelly beans in the pile) from that pile to the other. *Note: if there is only one jelly bean left in a pile, the player cannot move this jelly bean to the other pile.*

The player to take the last jelly bean wins. Use strong induction to prove that the second player always has a winning strategy in Nim-Move. A complete and correct solution will first identify what the strategy is, and then prove that following this strategy will lead the second player to win the game (no matter what the first player chooses to do at each turn).

4. Linked Lists. Recall the recursive definition of the set of linked lists of natural numbers (from class)

Basis Step: $[] \in L$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$

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

Basis Step: $length([]) = 0$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$, then $length((n, l)) = 1 + length(l)$

the function $append : L \times \mathbb{N} \rightarrow L$ that adds an element at the end of a linked list

Basis Step: If $m \in \mathbb{N}$ then $append([], m) = (m, [])$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$ and $m \in \mathbb{N}$, then $append((n, l), m) = (n, append(l, m))$

and the function $prepend : L \times \mathbb{N} \rightarrow L$ that adds an element at the front of a linked list

$$prepend((l, n)) = (n, l)$$

Sample response that can be used as reference for the detail expected in your answer: To evaluate the result of $length((4, (2, (7, []))))$ we calculate:

$$length((4, (2, (7, [])))) = 1 + length((2, (7, [])))$$

using recursive step of definition of $length$, with $n = 4, l = (2, (7, []))$

$$= 1 + 1 + length((7, []))$$

using recursive step of definition of $length$, with $n = 2, l = (7, [])$

$$= 1 + 1 + 1 + length([])$$

using recursive step of definition of $length$, with $n = 7, l = []$

$$= 1 + 1 + 1 + 0$$

using basis step of definition of $length$, since input to $length$ is $[]$

In this question, we'll consider the combination of these functions with a new function, one that removes the element at the front of the list (if there is any). We define $remove : L \rightarrow L$ by

Basis Step: $remove([]) = []$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$, then $remove((n, l)) = l$

(a) What is the result of $remove(append(prepend(((10, []) , 5)), 20))$? For full credit, include all intermediate steps with brief justifications for each.

(b) Prove the statement

$$\forall l \in L (remove(prepend((l, 0))) = l)$$

(c) Disprove the statement

$$\forall l \in L (remove(append((l, 0))) = l)$$

5. Primes, divisors, and proof strategies. For each statement below, identify the **main logical structure or connective** of the statement, list the proof strategies that could be used to prove and to disprove a statement with that structure, then identify whether the statement is true or false and justify with a proof of the statement or its negation. (*Graded for correctness of identification of logical structure and proof strategies and evaluation of statement (is it true or false?) and fair effort completeness of the proof*)

Sample response that can be used as reference for the detail expected in your answer:

Consider the statement: There is a greatest negative integer.

The main logical structure for this statement is that it is an **existential** statement, as we can see by translating it to symbols:

$$\exists g \in \mathbb{Z}^- \forall x \in \mathbb{Z}^- (g \geq x)$$

To prove an existential statement, the main proof strategy we could use is to find a witness. Proof by cases and proof by contradiction could also be used, because both can be used to prove any statement (no matter its logical structure).

To disprove an existential statement, we would need to prove its negation, which (using DeMorgan's Laws) can be written as a universal statement. Therefore, to disprove this statement the strategies we could use are universal generalization or structural induction (because $\mathbb{Z}^-$ is a recursively defined set) or proof by cases or proof by contradiction. Notice that proof by exhaustion is not possible because the domain is not finite.

The statement is true, as we can see from the witness $g = -1$, since it is in the domain $\mathbb{Z}^-$ and when we evaluate

$$\forall x \in \mathbb{Z}^- (-1 \geq x)$$

we can proceed by universal generalization and take an arbitrary negative integer x , which by definition means $x < 0$, and since x is an integer, guarantees $x \leq -1 = g$, as required.

- (a) The quotient of any even number with any nonzero even number is even.
- (b) There are two odd numbers (not necessarily distinct) whose sum is even.
- (c) The greatest common divisor of 5 and 23 is 1 and the greatest common divisor of 7 and 19 is 1.
- (d) There are two positive integers greater than 20 that have the same greatest common divisor with 20.