HW3

CS40 Summer '24

Due: Monday, July 15, 2024 at 11:59PM on Gradescope

Integrity reminders for individual homeworks

- "Individual homeworks" must be solely your own work.
- You may not collaborate on individual homeworks with anyone or seek help from online tutors or entities outside the class.
- You may ask questions about the homework in office hours (of the instructor, TAs, and/or tutors) and on Piazza. However, the staff will only answer clarifying questions on these homeworks. You *cannot* use any online resources about the course content other than the text book and class material from this quarter.
- Do not share written solutions or partial solutions for homework with other students. Doing so would dilute their learning experience and detract from their success in the class.

You will submit this assignment via Gradescope (https://www.gradescope.com) in the assignment called "HW3".

Summary of Proof Strategies (so far)

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 $p \to q$ 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 $p \to q$ is true, we can assume $\neg q$ is true and use that assumption to show $\neg p$ is true.

You will also work with recursively defined sets and functions and prove properties about them, practicing induction and other proof strategies. We include induction-related strategies here.

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 $n+1$.

Linked list related definitions

Definition The set of linked lists of natural numbers L is defined by:

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

Definition The function $\mathit{removeTail} : L \to L$ that removes the last node of a linked list (if it exists) is defined by:

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

Definition The function $\mathit{remove} : L \times \mathbb{N} \to L$ that removes a single node containing a given value (if present) from a linked list is defined by:

$$\begin{array}{llll} & & \mathit{remove} : L \times \mathbb{N} & \to L \\ \text{Basis Step:} & \text{If } m \in \mathbb{N} \text{ then} & \mathit{remove}([\,], m) & = [\,] \\ \text{Recursive Step:} & \text{If } l \in L, n \in \mathbb{N}, m \in \mathbb{N}\text{, then} & \mathit{remove}((n, l), m) & = \begin{cases} l & \text{when } n = m \\ (n, \mathit{remove}(l, m)) & \text{when } n \neq m \end{cases} \end{array}$$

Definition: The function $\mathit{prepend} : L \times \mathbb{N} \to L$ that adds an element at the front of a linked list is defined by:

$$\mathit{prepend}(l, n) = (n, l)$$

Definition The function $\mathit{append} : L \times \mathbb{N} \to L$ that adds an element at the end of a linked list is defined by:

$$\begin{array}{llll} & & \mathit{append} : L \times \mathbb{N} & \to L \\ \text{Basis Step:} & \text{If } m \in \mathbb{N} \text{ then} & \mathit{append}([\,], m) & = (m, [\,]) \\ \text{Recursive Step:} & \text{If } l \in L \text{ and } n \in \mathbb{N} \text{ and } m \in \mathbb{N}\text{, then} & \mathit{append}((n, l), m) & = (n, \mathit{append}(l, m)) \end{array}$$

Assigned Questions

1. **Theorem**: If n and m are odd integers, then $n \cdot m$ is odd.

   For each of the following proof attempts of the theorem, explain where the proof uses invalid reasoning or skips essential steps.

   (a) Let n and m be odd integers. Then $n = 2k + 1$ and $m = 2j + 1$. Plugging into the expression $n \cdot m$ gives
   $$n \cdot m = (2k + 1)(2j + 1) = 2(j \cdot k + j + k) + 1$$
   Since k and j are integers, $j \cdot k + j + k$ is also an integer. Then $n \cdot m$ equals to two times an integer plus one, therefore $n \cdot m$ is odd.

   (b) Let n and m be odd integers. Since n is an odd integer, then $n = 2k + 1$ for some integer k. Since m is an odd integer, then $m = 2j + 1$ for some integer j. Plugging in $2k + 1$ for n and $2j + 1$ for m into the expression $n \cdot m$ gives
   $$n \cdot m = (2k + 1)(2j + 1)$$
   Since $n \cdot m$ is equal to two times an integer plus one, then $n \cdot m$ is an odd integer.

   (c) Let n and m be odd integers. Since n is an odd integer, then $n = 2k + 1$ for some integer k. Since m is an odd integer, then $m = 2k + 1$ for some integer k. Plugging into the expression $n \cdot m$ gives
   $$n \cdot m = (2k + 1)(2k + 1) = 2(2k^2 + 2k) + 1$$
   Since k is an integer, $2k^2 + 2k + 1$ is also an integer. Since $n \cdot m$ is equal to two times an integer plus one, then $n \cdot m$ is an odd integer.

2. **Theorem** For all non-zero integers, x, y, z, if x does not divide yz, then x does not divide y

   For each of the following proof attempts of the theorem, explain where the proof uses invalid reasoning or skips essential steps.

   (a) We prove this by contrapositive. Let x, y, z be non-zero integers. We assume that x does divide y. Then $y = kx$. So,
   $$zy = z(kx) = (kz)x$$
   Since z, k are integers, zk is also an integer. So, x divides yz.

   (b) We prove this by constrapositive. Let x, y, z be non-zero integers. We assume that x does divide y. So, $y = kx$ for some integer k. Since zk is also an integer, we conclude that x divides zy, thus proving the theorem.

   (c) We prove this by constrapositive. Let x, y, z be non-zero integers. Assume that x divides y. So, $x = ky$ for some integer k. Therefore, x divides yz if and only if $ky = ayz$ for some integer a. We divide this out to get $a = \frac{k}{z}$. Since k, z are both integers, kz is also an integer, thus proving that x divides yz.

3. Consider the statement: The sum of any two integers is odd if and only if at least one of them is odd.

   (a) Define predicates as necessary and write the symbolic form of the statement using quantifiers.

   (b) Prove or disprove the statement. Specify which proof strategy is used.

4. Consider the statement: If x and y are integers such that $x + y \geq 5$, then $x > 2$ or $y > 2$.

   (a) Write the symbolic form of the statement using quantifiers.

   (b) Prove or disprove the statement. Specify which proof strategy is used.

5. Consider the statement: The average of two odd integers is an integer.

   (a) Write the symbolic form of the statement using quantifiers.

   (b) Prove or disprove the statement. Specify which proof strategy is used.

6. Consider the statement: For any three consecutive integers, their product is divisible by 6.

   (a) Write the symbolic form of the statement using quantifiers.

   (b) Prove or disprove the statement. Specify which proof strategy is used.

7. We define the set of balanced parentheses S recursively as follows:

   Basis: the string () is in S

   Recursive rules:

   (a) $\forall x \in S((x) \in S)$

   (b) $\forall x \in S \forall y \in S(xy \in S)$ where xy means the concatenation of x and y

   Using structural induction, prove that for any string x in S, the number of left parentheses in x is equal to the number of right parentheses in x.

   Definitions related to linked list are listed on page 3 and will be used in Qs 9-11

8. (*Graded for correctness*)

   Calculate the following function applications. Include all intermediate steps, with justifications.

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

   Calculating $append(\ (1,(2,[\,]))\ ,3)$, we have

$append(\ (1,(2,[\,]))\ ,3) = (1, append(\ (2,[\,])\ ,3))$	By recursive step of $append$: $n = 1$, $l = (2,[\,])$, $m = 3$
$= (1,\ (2,\ append(\ [\,]\ ,3)))$	By recursive step of $append$: $n = 2$, $l = [\,]$, $m = 3$
$= (1,\ (2,\ (3,[\,])))$	By basis step of $append$: $m = 3$

   (a) Calculate $removeTail(\ append(\ (2,(3,[\,])\ ), 1\ )\ )$

   (b) Calculate $prepend(\ remove(\ (1,(2,(2,(3,[\,])))), 2\ ), 3\ )$

9. Consider the following statement and attempted proof:

$$\forall l \in L \, \exists n \in \mathbb{N} \; (\; append(removeTail(l), n) = l \;)$$

Attempted proof: By structural induction on L, we have two cases:

Basis Step: Consider $l = []$ and choose the witness $n = 0$ (in the domain $\mathbb{N}$ since it is a nonnegative integer). We need to show that $append(removeTail(\; (0, []) \;), 0) = (0, [])$. By the definition of $removeTail$, using the recursive step with $l = []$ and $n = 0$, we have $removeTail(\; (0, []) \;) = []$. By the definition of $append$, using the basis step with $l = []$ and n, we have $append([], 0) = (0, [])$ as required.

Recursive Step Consider an arbitrary list $l = (x, l')$, $l' \in L$, $x \in \mathbb{N}$, and we assume as the **induction hypothesis** that:

$$\exists n \in \mathbb{N} \; (\; append(removeTail(l'), n) = l' \;)$$

Our goal is to show that $\exists n \in \mathbb{N} \; (\; append(removeTail((x, l')), n) = (x, l') \;)$. Choose the witness $n = x$, a nonnegative integer so in the domain. We need to show that

$$append(removeTail((x, l')), x) = (x, l')$$

Applying the definitions:

$$\begin{aligned}
LHS &= append(removeTail((x, l')), x) \\
&= append((x, removeTail(l')), x) \qquad \text{by recursive step of } removeTail\text{, with } l = l' \text{ and } n = x \\
&= (x, l') \qquad \text{by the recursive definition of } append\text{, with } l = l' \text{ and } n = x \\
&= RHS
\end{aligned}$$

as required.

Thus, the recursive step is complete and we have finished the proof by structural induction. ■

(a) (*Graded for correctness*) Demonstrate that this attempted proof is invalid by providing and justifying a **counterexample** (disproving the statement).

(b) (*Graded for fair effort completeness*) Explain why the attempted proof is invalid by identifying in which step(s) a definition or proof strategy is used incorrectly, and describing how the definition or proof strategy was misused.

10. (*Each part graded for correctness in evaluating statement and for fair effort completeness in the justification)* Statements like these are used to build the specifications for programs, libraries, and data structures (API) which spell out the expected behavior of certain functions and methods. In this HW question, you're analyzing whether and how order matters for the *remove* and *prepend* functions.

(a) Prove or disprove the following statement:

$$\forall l \in L \, \forall m \in \mathbb{N} \; (\; prepend(\; remove(l, m) \; , \; m) = l \;).$$

(b) Prove or disprove the following statement:

$$\exists l \in L \, \exists m \in \mathbb{N} \; (\; prepend(\; remove(l, m) \; , m) = l \;).$$