CSE 20, Fall 2020 - Homework 4

Due: Monday 11/09 at 11 am PDT

Instructions

Upload a single file to Gradescope for each group. All group members' names and PIDs should be on each page of the submission. You should select appropriate pages for each question when submitting to Gradescope. Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always 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.

Reading Section 1.5 Table 1 (p. 60); Section 2.1, Definitions 1-3 (pp. 116-119), Definitions 6-8 (pp. 121-123); Section 2.2 Definitions 1-5 (pp. 127-129) and Table 1 (p. 130), Section 5.3 Definition of Structural Induction

Key Concepts Proof Strategies, Set Definitions and Induction

Problem 1 (20 points)

Consider Apple's newest product iPhone 14 (some point in the future). It has 4 types of memory defined as set $M = \{32G, 64G, 128G, 256G\}$ and it has 5 types of color defined as set $C = \{black, white, gray, green, orange\}$.

Customers are allowed to choose their favorite combination of memory type and color when purchasing the product. The domain of an iPhone 14 is defined as $D = M \times C = \{(m, c) \mid m \in M,\ c \in C\}$.

All colors the same memory type have the same price. For different memory types, the iPhone 14 are at different prices defined by the function $Price(m)$ with units in dollars:

$$Price(32G) = 200$$

$$Price(64G) = 250$$

$$Price(128G) = 300$$

$$Price(256G) = 400$$

Consider the following questions:

(a) Consider Alice wants to buy an iPhone 14. Translate the following statement in first-order logic (in predicates with quantifiers), then prove or disprove the statement. Please identify your proof strategy. (Hint: Prove for all quantifier is true, use exhaustion or universal generalization; disprove for all quantifier statements, use counterexample)

"If Alice has more than 400 dollars, then she can buy any model of iPhone 14"

Solution:
First order logic: $\forall x \in D,\ P(Alice,\ 450) \rightarrow Q(x,\ 450)$
Define P(x, y) as person x has more than y dollars.
Define Q(x, y) as iPhone 14 model x has a price less than or equal to y
(or something makes sense)

Prove the statement by **exhaustion**. There are 20 elements in the domain D. Because all the colors of the same memory type have the same price, it is sufficient to prove the statement is true for all memory types. (1) Memory type 32G in all colors has price 200 < 450, (2) Memory type 64G in all colors has price 250 < 450; (3) Memory type 128G in all colors has price 300, 300 < 450; (4) Memory type 256G in all colors has price 400 < 450.

Hence we proved that If Alice has more than 450 dollars, she can buy any model of iPhone 14 by exhaustion.

(b) Prove or disprove that all green iPhone 14's are more expensive than 380 dollars. Translate the statement in first-order logic (in predicates with quantifiers), then prove or disprove the statement. Please identify your proof strategy.

"All green iPhone 14's are more expensive than 380 dollars"

Solution:
First order logic: $\forall x \in D,\ P(x) \rightarrow Price(x) \geq 380$
Define P(x) as $P(x)\ =\ x \in \{(m,\ green),\ m \in M\}$
(or something makes sense)

Disprove the statement by **counterexample**. Consider an green iPhone 14 with 32G memory. Price((32G, green)) = 200, which is less than 380, which contradicts with the statement.
Hence we proved that the statement is false by counterexample.

(c) Consider the set $A = \{(32G,\ black),\ (32G,\ white),\ (64G,\ orange), (128G,\ orange)\}$ Prove or disprove the following statement.

$$\forall x,\ x \in A \rightarrow x \in D$$

Solution:
Proof this is true by **exhaustion**. We assume $x \in A$ is true and show that $x \in D$ is true. Therefore are 4 elements in A, (1) (32G, black) is in D because 32G is in M and black is in C, therefore (32G, black) is in M $\times$ C, hence in D. (2) (32G, white) is in D because 32G is in M and white is in C, therefore (32G, white) is in M $\times$ C, hence in D. (3) (64G, orange) is in D because 64G is in M and orange is in C, therefore (64G, orange) is in M $\times$ C, hence in D. (4) (128G, orange) is in D because 128G is in M and orange is in C, therefore (128G, orange) is in M $\times$ C, hence in D.
We showed that if x is in A, then it is in D, therefore the statement is proved.

Problem 2 (20 points)

Consider the Fibonnaci sequence, which is a number sequence starting with 0 and 1, then each number is the sum of the two preceding ones.

(a) Prove or disprove the following claim

"There are 3 consecutive odd Fibonnaci numbers"

Solution: We prove the statement is false by contradiction. Assume that there are 3 consecutive odd Fibonnaci numbers, then the third number equals the sum of previous 2 numbers. However, we know that the sum of two odd numbers results in an even number, which contradicts with the assumption. Hence there does not exist 3 consecutive odd Fibonnaci numbers.

(b) Prove or disprove the following claim

"There are 3 consecutive prime Fibonnaci numbers"

Solution: We prove this by giving a witness. 2, 3, 5 are 3 consecutive prime Fibonnaci numbers. (NOTICE: 0 and 1 are NOT prime number)

Problem 3 (20 points)

Consider sets $A = \{1, 3, 5, 6, 7\}$, $B = \{0, 1, 3, 7, 9\}$ and two arbitrary sets C and D, P denotes the power set.
Answer the following questions:

(a) Please fill-in the blank: $A \cup B =$ $\{0, 1, 3, 5, 6, 7, 9\}$

(b) Please fill-in the blank: $A - B =$ $\{5, 6\}$

(c) Write the power set $P(A \cap B) = \{\varnothing,\ \{1\},\ \{3\},\ \{7\},\ \{1, 3\},\ \{3, 7\},\ \{1, 7\},\ \{1, 3, 7\}\}$

(d) Prove or disprove the following claim:

$$A - B = B - A$$

Solution: We disprove the statement directly. A-B={5,6} and B-A={0,9}. {5,6}≠{0,9}, which implies that A-B≠B-A, therefore the statement is false.

(e) Prove or disprove the following claim:

$$P(C) \subseteq P(D) \Leftrightarrow C \subseteq D$$

Solution:
We prove the statement proving both directions.

(1) we prove that $P(C) \subseteq P(D)$ implies $C \subseteq D$

(a) Proof: We show this statement by direct proof. Assuming $P(C) \subseteq P(D)$, then every element in $P(C)$ is also an element in $P(D)$. By definition of power set, this means that all possible subsets of C is also a subset of D. Then we have for all set $A \subseteq C \rightarrow A \subseteq D$. Take $A = C$ satisfies $C \subseteq C$, therefore $C \subseteq D$.

(2) we prove that $C \subseteq D$ implies $P(C) \subseteq P(D)$

(a) Proof: We show this statement by direct proof. Assuming $C \subseteq D$. By definition of $P(C)$, there is an arbitrary set $A \in P(C)$ such that $A \subseteq C$. Because $C \subseteq D$, we have $A \subseteq C \subseteq D$, therefore $A \in P(D)$ by definition of power set. Since all elements of $P(C)$ is also an element of $P(D)$, we have $P(C) \subseteq P(D)$.

Combining (1) and (2) we have shown that $P(C) \subseteq P(D) \Leftrightarrow C \subseteq D$

Problem 4 (20 points)

Let us recall the definition for the set of linked lists L defined in class:

Basis Step: $[] \in L$

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

We now define 3 functions on this linked list as:

i) $nodes : L \to P(N)$

Basis Step: $nodes([]) = \varnothing$

Recursive Step: If $l \in L,\ n \in N$, then $nodes((n, l)) = \{n\} \cup nodes(l)$

ii) $sum : L \to N$

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

Recursive Step: If $l \in L,\ n \in N$, then $sum((n, l)) = n + sum(l)$

iii) $length : L \to N$

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

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

a) Consider the proposition given as: $\forall l \in L,\ \exists c \in N,\ \forall x \in N(x \le c \to x \in nodes(l))$. Disprove the statement by proving the negation.

b) Consider the proposition given as: $\forall l \in L,\ \exists c \in N,\ \forall x \in N(x \in nodes(l) \to x \ge c)$. Prove the statement by applying structural induction on *L*.

c) Consider the proposition given as: $\forall l \in L,\ (sum(l) = length(l))$. Disprove the statement by providing a counter-example.

Solution:

a) To disprove this proposition, we first negate the original proposition to give:

$\neg(\forall l \in L,\ \exists c \in N,\ \forall x \in N(x \le c \to x \in nodes(l))) \equiv \exists l \in L,\ \forall c \in N,\ \exists x \in N \neg(x \le c \to x \in nodes$

$\equiv \exists l \in L,\ \forall c \in N,\ \exists x \in N(x \le c \wedge \neg(x \in nodes(l))) \equiv \exists l \in L,\ \forall c \in N,\ \exists x \in N(x \le c \wedge x \notin nodes(l)$

Therefore, we need to prove that:

$\exists l \in L,\ \forall c \in N,\ \exists x \in N(x \le c \wedge x \notin nodes(l))$ is true,

We will prove this by providing a witness for L. For this purpose, consider : $L = []$

Now, we need to prove that for any $c \ge 0$, there exists an $x \ge 0$ such that $x \notin nodes(l)$

Observe that for this choice of l, the set $nodes([]) = \varnothing$, or there is no x in the set $nodes(l)$. Therefore the 2nd part of $x \le c \wedge x \notin nodes(l)$ is always true.
Further, we observe that for $x = 0$, we will always have $0 \le c$ (by the very definition of c). Hence, $x = 0$ will be a witness for any choice of c. Hence, overall,
$\exists l \in L,\ \forall c \in N,\ \exists x \in N(x \le c \wedge x \notin nodes(l))$ is true.

b) $\forall l \in L,\ \exists c \in N,\ \forall x \in N(x \in nodes(l) \to x \ge c)$
We need to use structural induction on the set L here to prove this. To this end, we define the basis and inductive step as:
$Basis\ Step : L = []$, then we need to show that $\exists c \in N,\ \forall x \in N(x \in nodes([]) \to x \ge c)$. In this case, since there does not exist any $x \in N$ that lies in $nodes(l)$, for any choice of c, the logical statement: $x \in nodes([])$ will be False. This means that $x \in nodes([]) \to x \ge c$ will always be true.
$Inductive\ Step$: Consider the arbitrary linked list l' and the natural number n. We assume by the induction hypothesis that $\exists c \in N,\ \forall x \in N(x \in nodes(l') \to x \ge c)$ is true, with c_w acting as the witness for c.
We need to show now that $\exists c \in N,\ \forall x \in N(x \in nodes((n, l')) \to x \ge c)$. In this case, let us pick the witness: $c_{w+1} = min(c_w, n)$, that is, c_{w+1} is the lesser of $c_w\ and\ n$. In this case, $x \in nodes((n, l') \Rightarrow x \in \{n\} \cup nodes(l')$. Further, since $c_{w+1} \le c_w \Rightarrow x \in nodes(l') \ge c_{w+1}$ from the induction hypothesis. Further, $n \ge c_{w+1}$ by definition. Therefore, each element $x \in \{n\} \cup nodes(l') \ge c_{w+1}$. If $x \notin \{n\} \cup nodes(l')$ then $x \in nodes((n, l'))$ is false, and the overall proposition is true regardless. Therefore, we have shown that for the witness c_{w+1} the overall proposition is true, and our induction hypothesis is true.

c) We need to disprove $\forall l \in L,\ (sum(l) = length(l))$, or effectively, prove its negation $\exists l \in L,\ (sum(l) \ne length(l))$
Consider the linked list $l = (3, [])$
In this case, we calculate $sum(l)$ as:
$sum((3, [])) = 3 + sum([]) = 3 + 0 = 3$
We calculate $length(l)$ as:
$length((3, [])) = 1 + length([]) = 1 + 0 = 1$

Hence, $sum(l) \ne length(l)$, and $l = (3, [])$ acts as a sufficient counter-example.

Problem 5 (20 points)

So far in this class, we have used recursion to define sets and functions. However, we have never offered a formal proof justifying our construction. We will now attempt to use structural induction to prove the correctness of the following functions. For each of the functions defined below, define the appropriate basis step and choose the correct recursive/inductive step to prove the correctness of the function:

a) Let us prove the correctness of the design for the set of all positive odd multiples of 4 (i.e the set {4,12,20,28…..}). To this end, we recursively define our function $multiple : N^{\neq 0} \to N$, which takes as an input an integer *x*, and computes the x^{th} odd multiple of 4 as:

Basis Step: $multiple(1) = 4$

Recursive Step: $multiple(x+1) = multiple(x) + 8$

Prove that $\forall x \in N^{\neq 0}, multiple(x) = 4(2x-1)$

b) Now, let's recall the RMSE function that we had encountered in Homework 1.

$d_{rmse,n} : ((R \times R \times ....R), (R \times R \times ......R)) \to R$

Basis Step: $d_{rmse,1}(x,y) = \left|x_1 - y_1\right|$

Recursive Step: $d_{rmse,n+1}(x,y) = \sqrt{\frac{n * d_{rmse,n}(x_{[1,n]}, y_{[1,n]})^2 + (x_{n+1} - y_{n+1})^2}{n+1}}$

Prove that $\forall n \in N^{\neq 0},\ [d_{rmse,n}((x_1, x_2 ....., x_n), (y_1, y_2 ....., y_n)) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}}]$

Solution:

a) We need to prove by induction for the function $multiple$:

$\forall x \in N^{\neq 0}, multiple(x) = 4(2x-1)$

We will use induction here.

$Basis\ Step : x = 1$

In this case: $multiple(1) = 4$

Also, $4(2x-1) = 4(2.1-1) = 4(2-1) = 4$

$Inductive\ Step$: We assume that for an arbitrary integer $k \in Z$, $multiple(k) = 4(2k-1)$. We need to prove that

$multiple(k+1) = 4(2(k+1)-1) = 4(2k+2-1) = 4(2k+1)$

Recall, from the definition of the function $multiple(k+1) = multiple(k) + 8$

Therefore, using the induction assumption, we can re-write this as:

$multiple(k+1) = 4(2k-1) + 8 = 8k - 4 + 8 = 8k + 4 = 4(2k+1) = RHS$

Hence, proved.

b) We will again use the induction hypothesis on the function $d_{rmse,n} : ((R \times R \times ....R), (R \times R \times ......R)) \to R$ to show:

$\forall n \in N^{\neq 0},\ [d_{rmse,n}((x_1, x_2 ....., x_n), (y_1, y_2 ....., y_n)) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}}]$

$Basis\ Step : d(x_1, y_1) = \sqrt{\frac{\sum_{i=1}^{1} (x_i - y_i)^2}{n}} = \sqrt{\frac{(x_1 - y_1)^2}{1}} = |(x_1 - y_1)|$, which is the definition of the function.

$Inductive\ Step$: We assume that for an arbitrary integer $k \in Z$,

$$[d_{rmse,k}((x_1, x_2 ....., x_k), (y_1, y_2 ....., y_k)) = \sqrt{\frac{\sum_{i=1}^{k} (x_i - y_i)^2}{k}}]$$

We need to prove that:

$$[d_{rmse,k+1}((x_1, x_2 ....., x_{k+1}), (y_1, y_2 ....., y_{k+1})) = \sqrt{\frac{\sum_{i=1}^{k+1} (x_i - y_i)^2}{k+1}}]$$

From the definition of the function, we can write:

$$d_{rmse,k+1}((x_1, x_2 ....., x_{k+1}), (y_1, y_2 ....., y_{k+1})) = \sqrt{\frac{k \times d_{rmse,k}(x,y)^2 + (x_{k+1} - y_{k+1})^2}{k+1}}$$

Applying the induction assumption, we get:

$$d_{rmse,k+1}((x_1, x_2 ....., x_{k+1}), (y_1, y_2 ....., y_{k+1})) = \sqrt{\frac{k \times d_{rmse,k}(x,y)^2 + (x_{k+1} - y_{k+1})^2}{k+1}}$$

Now, $[d_{rmse,k}((x_1, x_2 ....., x_k), (y_1, y_2 ....., y_k))]^2 = \frac{\sum_{i=1}^{k} (x_i - y_i)^2}{k}$

$\Rightarrow k \times [d_{rmse,k}((x_1, x_2 ....., x_k), (y_1, y_2 ....., y_k))]^2 = \sum_{i=1}^{k} (x_i - y_i)^2$

$\Rightarrow \sqrt{\frac{k \times d_{rmse,k}(x,y)^2 + (x_{k+1} - y_{k+1})^2}{k+1}} = \sqrt{\frac{\sum_{i=1}^{k} (x_i - y_i)^2 + (x_{k+1} - y_{k+1})^2}{k+1}} = \sqrt{\frac{\sum_{i=1}^{k+1} (x_i - y_i)^2}{k+1}} = RHS$

Hence Proved

Problem 6 - Bonus (20 points)

Let us assume that we have the same linked list L as provided in class:

Basis Step: $[] \in L$

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

a) Define a function *least*, which computes and returns the smallest data element in the node of the linked list (i.e., the smallest element across all nodes. Eg- (4,(1,(3,[]))) has the smallest element 1).

b) Define a function *average* that computes the mean of all data elements in the linked list. In the definition of your function, you can use any of the functions *nodes, sum* or *length* defined in Problem 4.

Solution:

a) We will use a recursively defined function here, similar to how we calculated the length and sum of the linked list:

$least : L \rightarrow N$

$Basis\ Step : least([]) = -1$

$Recursive\ Step$: If $l \in L,\ n \in N$, then

$min = least(l)$

$if\ n \leq min\ or\ min = -1,\ then\ least((n, l)) = n$

$else\ if\ min \leq n\ and\ and\ min \neq -1,\ then\ least(n, l) = min$

b) There are several possible solutions to this question

$average:\ L \rightarrow N$

$Basis\ Step: average([])\ = 0$

$Recursive\ Step$: If $l \in L,\ n \in N$, then

$average((n, l)) = \frac{average(l) \times length(l) + n}{length((n,l))}$

Note: You can avoid recursion altogether here and simply define the function as:

$average:\ L \rightarrow N$

$average(l')) = \frac{sum(l')}{length(l')}\ if\ length(l') > 0$

$= \ 0\ if\ length(l') = 0$