CSE 390Z: Mathematics of Computing

Week 7 Workshop Solutions

Conceptual Review

Space to take notes on strong and structural induction:

1. Strong Induction: Collecting Candy

A store sells candy in packs of 4 and packs of 7. Let P(n) be defined as "You are able to buy n packs of candy". For example, $P(3)$ is not true, because you cannot buy exactly 3 packs of candy from the store. However, it turns out that P(n) is true for any $n \geq 18$. Use strong induction on n to prove this.

Hint: you'll need multiple base cases for this - think about how many steps back you need to go for your inductive step.

Solution:

1 Let P(n) be defined as "You are able to buy n packs of candy". We will prove $P(n)$ is true for all integers $n \geq 18$ by strong induction.

2 **Base Cases** ($n = 18, 19, 20, 21$)**:**

- $n = 18$: 18 packs of candy can be made up of 2 packs of 7 and 1 pack of 4 ($18 = 2 * 7 + 1 * 4$).
- $n = 19$: 19 packs of candy can be made up of 1 pack of 7 and 3 packs of 4 ($19 = 1 * 7 + 3 * 4$).
- $n = 20$: 20 packs of candy can be made up of 5 packs of 4 ($20 = 5 * 4$).
- $n = 21$: 21 packs of candy can be made up of 3 packs of 7 ($21 = 3 * 7$).

3 **Inductive Hypothesis:** Suppose for some arbitrary integer $k \geq 21$, P(j) is true for $18 \leq j \leq k$.

4 **Inductive Step:**

Goal: Show $P(k + 1)$, i.e. show that we can buy $k + 1$ packs of candy.

We want to buy $k+1$ packs of candy. By the I.H., we can buy exactly $k-3$ packs, so we can add another pack of 4 packs in order to buy $k + 1$ packs of candy, so P$(k + 1)$ is true.

Note: How did we decide how many base cases to have? Well, we wanted to be able to assume P$(k-3)$, and add 4 to achieve P$(k + 1)$. Therefore we needed to be able to assume that $k - 3 \geq 18$. Adding 3 to both sides, we needed to be able to assume that $k \geq 21$. So, we have to prove the base cases up to 21, that is: $18, 19, 20, 21$.

Another way to think about this is that we had to use a fact from 4 steps back from $k + 1$ to $k - 3$ in the IS, so we needed 4 base cases.

5 So by strong induction, P(n) is true for all integers $n \geq 18$.

2. Structural Induction: Dictionaries

Consider the following definition for a Dictionary (known in some languages as a Map):

- [] is the empty dictionary
- If D is a dictionary, and a and b are elements of the universe, then $(a \to b)$ :: D is a dictionary that maps a to b (in addition to the content of D).

Now, define the following programs on a dictionary:

$$\begin{aligned} \text{AllKeys}([\,]) &= [\,] & \text{len}([\,]) &= 0 \\ \text{AllKeys}((a \to b) :: \text{D}) &= a :: \text{AllKeys}(\text{D}) & \text{len}((a \to b) :: \text{D}) &= 1 + \text{len}(\text{D}) \end{aligned}$$

Prove that $\text{len}(\text{D}) = \text{len}(\text{AllKeys}(\text{D}))$.

Solution:

Proof. Define P(D) to be $\text{len}(\text{D}) = \text{len}(\text{AllKeys}(\text{D}))$ for a Dictionary D. We will go by structural induction to show P(D) for all dictionaries D.
Base Case: D = []: Note that:

$$\begin{aligned} \text{len}(\text{D}) &= \text{len}([\,]) \\ &= 0 && \text{[Definition of len]} \\ &= \text{len}([\,]) && \text{[Definition of len]} \\ &= \text{len}(\text{AllKeys}([\,])) && \text{[Definition of AllKeys]} \\ &= \text{len}(\text{AllKeys}(\text{D})) \end{aligned}$$

Inductive Hypothesis: Suppose P(C) to be true for an arbitrary dictionary C.
Inductive Step:
Let D' = $(a \to b)$ :: C. Note that:

$$\begin{aligned} \text{len}((a \to b) :: \text{C}) &= 1 + \text{len}(\text{C}) && \text{[Definition of Len]} \\ &= 1 + \text{len}(\text{AllKeys}(\text{C})) && \text{[IH]} \\ &= \text{len}(a :: \text{AllKeys}(\text{C})) && \text{[Definition of Len]} \\ &= \text{len}(\text{AllKeys}((a \to b) :: \text{C})) && \text{[Definition of AllKeys]} \end{aligned}$$

So P(D') holds.
Thus, the claim holds for all dictionaries D by structural induction. □

3. Strong Induction: Functions

Consider the function $f(n)$ defined for integers $n \geq 1$ as follows:
$f(1) = 1$ for $n = 1$
$f(2) = 4$ for $n = 2$
$f(3) = 9$ for $n = 3$
$f(n) = f(n-1) - f(n-2) + f(n-3) + 2(2n-3)$ for $n \geq 4$

Prove by strong induction that for all $n \geq 1$, $f(n) = n^2$.

Solution:

1 Let P(n) be defined as " $f(n) = n^2$". We will prove $P(n)$ is true for all integers $n \geq 1$ by strong induction.

2 **Base Cases** $(n = 1, 2, 3)$**:**

- $n = 1$: $f(1) = 1 = 1^2$.
- $n = 2$: $f(2) = 4 = 2^2$.
- $n = 3$: $f(3) = 9 = 3^2$

So the base cases hold.

3 **Inductive Hypothesis:** Suppose for some arbitrary integer $k \geq 3$, P(j) is true for $1 \leq j \leq k$.

4 **Inductive Step:**

Goal: Show $P(k+1)$, i.e. show that $f(k+1) = (k+1)^2$.

$$
\begin{aligned}
f(k+1) &= f(k+1-1) - f(k+1-2) + f(k+1-3) + 2(2(k+1)-3) && \text{Definition of f} \\
&= f(k) - f(k-1) + f(k-2) + 2(2k-1) \\
&= k^2 - (k-1)^2 + (k-2)^2 + 2(2k-1) && \text{By IH} \\
&= k^2 - (k^2 - 2k + 1) + (k^2 - 4k + 4) + 4k - 2 \\
&= (k^2 - k^2 + k^2) + (2k - 4k + 4k) + (-1 + 4 - 2) \\
&= k^2 + 2k + 1 \\
&= (k+1)^2
\end{aligned}
$$

So P$(k+1)$ holds.

5 **Conclusion:** So by strong induction, P(n) is true for all integers $n \geq 1$.

4. Structural Induction: Lists

Consider the following recursive definition for a List:

- `[]` is the empty list
- If L is a list, and a is an element of the universe, then a :: L is a list that has the first element a followed by the elements in L.

For example, 2 :: `[]` is the list `[2]`, and $1 :: 2 :: 3 ::$ `[]` is the list `[1,2,3]`. Define the following recursive functions:

$$\begin{aligned}
&\text{all}(x, \texttt{[]}) &&= \texttt{[]}, &\text{all}(x, y :: \text{L}) &= \text{if } x = y \text{ then } y :: \text{all}(x, \text{L}) \text{ else all}(x, \text{L}) \\
&\text{removeAll}(x, \texttt{[]}) &&= \texttt{[]}, &\text{removeAll}(x, y :: \text{L}) &= \text{if } x = y \text{ then removeAll}(x, \text{L}) \text{ else } y :: \text{removeAll}(x, \text{L}) \\
&\text{len}(\texttt{[]}) &&= 0, &\text{len}(a :: \text{L}) &= 1 + \text{len}(\text{L})
\end{aligned}$$

Prove $\text{len}(\text{removeAll}(x, \text{L})) = \text{len}(\text{L}) - \text{len}(\text{all}(x, \text{L}))$.

Solution:

Proof. Define $\text{P}(L) := \text{len}(\text{removeAll}(x, \text{L})) = \text{len}(\text{L}) - \text{len}(\text{all}(x, \text{L}))$ for all $x \in X$. We prove $\text{P}(L)$ for all Lists L by structural induction.

Base Case: L = `[]`. Note that

$$\begin{aligned}
\text{len}(\text{removeAll}(x, \texttt{[]})) &= \text{len}(\texttt{[]}) && \text{[Definition of removeAll]} \\
&= \text{len}(\texttt{[]}) - 0 \\
&= \text{len}(\texttt{[]}) - \text{len}(\texttt{[]}) && \text{Definition of len]} \\
&= \text{len}(\texttt{[]}) - \text{len}(\text{all}(x, \texttt{[]})) && \text{[Definition of all]}
\end{aligned}$$

So the $\text{P}(L)$ holds for L = `[]`.

Inductive Hypothesis: Suppose $\text{P}(K)$ holds for some arbitrary List K.

Inductive Step:

Suppose L' $= y ::$ K. Then $\text{len}(\text{removeAll}(x, \text{L'})) = \text{len}(\text{removeAll}(x, y :: \text{K}))$.

- First, consider the case where $x = y$:

$$\begin{aligned}
\text{len}(\text{removeAll}(x, y :: \text{K})) &= \text{len}(\text{removeAll}(x, \text{K})) && \text{[Definition of removeAll]} \\
&= \text{len}(\text{K}) - \text{len}(\text{all}(x, \text{K})) && \text{[IH]} \\
&= 1 + \text{len}(\text{K}) - (1 + \text{len}(\text{all}(x, \text{K}))) && \text{[Arithmetic]} \\
&= \text{len}(y :: \text{K}) - (1 + \text{len}(\text{all}(x, \text{K}))) && \text{[Definition of len]} \\
&= \text{len}(y :: \text{K}) - (\text{len}(y :: \text{all}(x, \text{K}))) && \text{[Definition of len]} \\
&= \text{len}(y :: \text{K}) - (\text{len}(\text{all}(x, y :: \text{K}))) && \text{[Definition of all, } x = y\text{]} \\
&= \text{len}(\text{L'}) - (\text{len}(\text{all}(x, \text{L'}))) && \text{[Definition of L']}
\end{aligned}$$

  So the claim holds when $x = y$.

- Now consider when $x \neq y$:

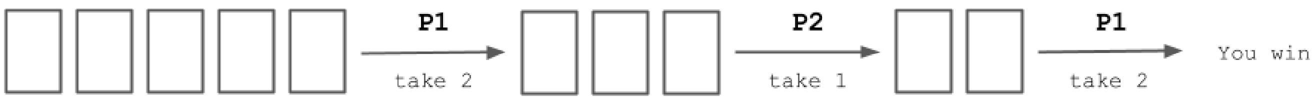
$$\begin{aligned}
\text{len}(\text{removeAll}(x, y :: \text{K})) &= \text{len}(y :: \text{removeAll}(x, \text{K})) && \text{[Definition of removeAll]} \\
&= 1 + \text{len}(\text{removeAll}(x, \text{K})) && \text{[Definition of len]} \\
&= 1 + \text{len}(\text{K}) - \text{len}(\text{all}(x, \text{K})) && \text{[IH]} \\
&= \text{len}(y :: \text{K}) - \text{len}(\text{all}(x, \text{K})) && \text{[Definition of len]} \\
&= \text{len}(y :: \text{K}) - (\text{len}(\text{all}(x, y :: \text{K}))) && \text{[Definition of all, } x \neq y\text{]} \\
&= \text{len}(\text{L'}) - (\text{len}(\text{all}(x, \text{L'})))
\end{aligned}$$

  So the claim holds when $x \neq y$.

So the claim holds no matter the relation between x and y, so P(L') holds.
Conclusion: Thus, the claim holds for all lists L by structural induction. $\square$

5. Strong Induction: Cards on the Table

I've come up with a new card game that is played between 2 players as follows. We start with some integer $n \geq 1$ cards on the table. The two players then take turns removing cards from the table; in a single turn, a player can choose to remove either 1 or 2 cards from the table. A player wins by taking the last card. For example:



The person I've been playing with has been *very* careful about dealing the cards, and keeps winning; I think they know something I don't about this game. I want to use induction to prove that if $3|n$, the second player (P2) can guarantee a win, and if n is not divisible by 3, the first player (P1) can guarantee a win.

(a) How many base cases does this proof need? What should they be?

Solution:

3 base cases; 1, 2, and 3 cards.

(b) Use strong induction to prove that if $3|n$, P2 can guarantee a win, and if n is not divisible by 3, P1 can guarantee a win.

Solution:

Proof. Let Q(n) be defined as "if $3|n$, P2 can guarantee a win, and if n is not divisible by 3, P1 can guarantee a win". We will show Q(n) holds for all integers $n \geq 1$ by *strong* induction.

Base Cases:

- **n = 1:** P1 can take 1 card and win, and 1 is not divisible by 3, so Q(1) holds.
- **n = 2:** P1 can take 2 cards and win, and 2 is not divisible by 3, so Q(1) holds.
- **n = 3:** If P1 takes 1 card, P2 can take 2 cards and win. If P1 takes 2 cards, P2 can take 1 card and win. Since 3|3, Q(3) holds.

Inductive Hypothesis: Suppose Q(j) holds for all $1 \leq j \leq k$ for an arbitrary integer $k \geq 3$.
Inductive Step:

- **case 1:** $3|k+1$
  By definition, $k + 1 = 3x$ for some integer x. If P1 takes 1 card, P2 can take 2 cards, leaving $k - 2 = 3(x - 1)$ cards for the next round. If P1 takes 2 cards, P1 can take 1 card, also leaving $k - 2 = 3(x - 1)$ cards for the next round. Since $3|k - 2$, by the Inductive Hypothesis P2 can win with $k - 2$ cards, thus P2 can win with $k + 1$ cards and Q($k + 1$) holds.
- **case 2:** $k + 1$ is not divisible by 3
  Either $k + 1 \equiv_3 1$ or $k + 1 \equiv_3 2$. If $k + 1 \equiv_3 1$, then P1 can take 1 card, leaving k cards where $3|k$, thus $k = 3x$ for some integer x. If $k + 1 \equiv_3 2$, then P1 can take 2 card, leaving k cards where $3|k$, thus $k = 3x$ for some integer x. P2 can then take 1 card or 2 cards, leaving $k - 1 = 3x - 1$ or $k - 2 = 3x - 2$ cards left, neither of which are divisible by 3. By the Inductive Hypothesis, P1 can always win with in either case, thus P1 can always win with $k + 1$ cards and Q($k + 1$) holds.

Thus, we have shown that Q(n) holds for all integers $n \geq 1$ by strong induction. □