CSE 390Z: Mathematics of Computing

Week 7 Workshop

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.

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 $(a \to b) :: \mathsf{D}$ is a dictionary that maps a to b (in addition to the content of D).

Now, define the following programs on a dictionary:

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

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

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 $n \geq 1$, $f(n) = n^2$.

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 $a :: \mathsf{L}$ is a list that has the first element a followed by the elements in L.

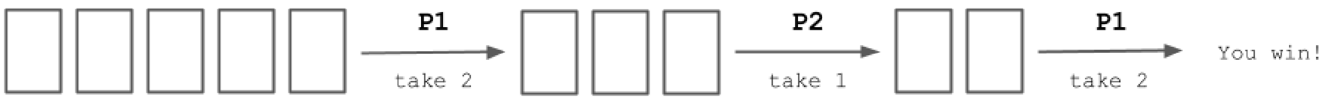
For example, $2 :: []$ is the list [2], and $1 :: 2 :: 3 :: []$ is the list [1,2,3]. Define the following recursive functions:

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

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

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?

(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.