CSE 311: Foundations of Computing I

Section : Strong and Structural Induction Solutions

1. Strong Induction

(a) Prove that, for all $n \in \mathbb{N}$, every n has an unsigned binary representation.

Solution:

Let $\mathsf{P}(n)$ be "n has an unsigned binary representation". We will prove $\mathsf{P}(n)$ for all integers $n \in \mathbb{N}$ by induction.

Base Case $(n = 0)$**:** The unsigned binary representation of 0 is 0_2, so $\mathsf{P}(0)$ holds.

Induction Hypothesis: Assume that $\mathsf{P}(j)$ holds for all integers $0 \leq j \leq k$ for some arbitrary $k \in \mathbb{N}$.

Induction Step: Goal: Show $\mathsf{P}(k+1)$, i.e., $k+1$ has an unsigned binary representation

Let 2^ℓ be the largest power of two not greater than $k+1$ (i.e. $\ell = \lfloor \log_2(n) \rfloor$). Let $r = k + 1 - 2^\ell$, the remainder.
Note that $r < 2^\ell < k$, so r has some binary representation r_2 [by the Induction Hypothesis].
Then $1r_2$ is the binary expansion for $k+1$. This proves $\mathsf{P}(k+1)$.

Conclusion: $\mathsf{P}(n)$ holds for all integers $n \in \mathbb{N}$ by induction.

(b) Xavier Cantelli owns some rabbits. The number of rabbits he has in any given year is described by the function f:

$$f(0) = 0$$
$$f(1) = 1$$
$$f(n) = 2f(n-1) - f(n-2) \qquad \text{for } n \geq 2$$

Determine, with proof, the number, $f(n)$, of rabbits that Cantelli owns in year n.

Solution:

Let $\mathsf{P}(n)$ be "$f(n) = n$". We prove that $\mathsf{P}(n)$ is true for all $n \in \mathbb{N}$ by strong induction on n.

Base Case $(n = 0)$**:** $f(0) = 0$ by definition. So, $\mathsf{P}(0)$ holds.

Induction Hypothesis: Assume that for some arbitrary integer $k \geq 0$, $\mathsf{P}(j)$ holds for all $0 \leq j \leq k$.

Induction Step: We show $\mathsf{P}(k+1)$.

Case 1 ($k = 0$**):** Then, by definition $f(k+1) = f(1) = 1$. So, $\mathsf{P}(k+1)$ holds.

Case 2($k \geq 1$**):** Since $k + 1 \geq 2$, by definition of f,

$$f(k+1) = 2f(k) - f(k-1)$$

Since $0 \leq k-1, k \leq k$, by induction hypothesis,

$$\begin{aligned} f(k+1) &= 2(k) - (k-1) \\ &= k+1 && \text{[Algebra]} \end{aligned}$$

This proves $\mathsf{P}(k+1)$.

Therefore, $\mathsf{P}(n)$ is true for all $n \in \mathbb{N}$.

2. Structural Induction

(a) Consider the following recursive definition of strings.

Basis Step: `""` is a string

Recursive Step: If X is a string and c is a character then append(c, X) is a string.

Recall the following recursive definition of the function len:

$$\begin{aligned} \text{len}(\texttt{""}) &= 0 \\ \text{len}(\text{append}(c, X)) &= 1 + \text{len}(X) \end{aligned}$$

Now, consider the following recursive definition:

$$\begin{aligned} \text{double}(\texttt{""}) &= \texttt{""} \\ \text{double}(\text{append}(c, X)) &= \text{append}(c, \text{append}(c, \text{double}(X))). \end{aligned}$$

Prove that for any string X, $\text{len}(\text{double}(X)) = 2\text{len}(X)$.

Solution:

For a string X, let $\mathsf{P}(X)$ be "$\text{len}(\text{double}(X)) = 2\text{len}(X)$. We prove $\mathsf{P}(X)$ for all strings X by structural induction.

Base Case. We show $\mathsf{P}(\texttt{""})$ holds. By definition $\text{len}(\text{double}(\texttt{""})) = \text{len}(\texttt{""}) = 0$. On the other hand, $2\text{len}(\texttt{""}) = 0$ as desired.

Induction Hypothesis. Suppose $\mathsf{P}(X)$ holds for some string X.

Induction Step. We show that $\mathsf{P}(\text{append}(c, X))$ holds for any character c.

$$\begin{aligned} \text{len}(\text{double}(\text{append}(c, X))) &= \text{len}(\text{append}(c, \text{append}(c, \text{double}(X)))) && \text{[By Definition of double]} \\ &= 1 + \text{len}(\text{append}(c, \text{double}(X))) && \text{[By Definition of len]} \\ &= 1 + 1 + \text{len}(\text{double}(X)) && \text{[By Definition of len]} \\ &= 2 + 2\text{len}(X) && \text{[By IH]} \\ &= 2(1 + \text{len}(X)) && \text{[Algebra]} \\ &= 2(\text{len}(\text{append}(c, X))) && \text{[By Definition of len]} \end{aligned}$$

This proves $\mathsf{P}(\text{append}(c, X))$.

Thus, $\mathsf{P}(X)$ holds for all strings X by structural induction.

(b) Consider the following definition of a (binary) **Tree**:

Basis Step: $\bullet$ is a **Tree**.

Recursive Step: If L is a **Tree** and R is a **Tree** then $\texttt{Tree}(\bullet, L, R)$ is a **Tree**.

The function leaves returns the number of leaves of a **Tree**. It is defined as follows:

$$\begin{aligned} \text{leaves}(\bullet) &= 1 \\ \text{leaves}(\texttt{Tree}(\bullet, L, R)) &= \text{leaves}(L) + \text{leaves}(R) \end{aligned}$$

Also, recall the definition of size on trees:

$$\begin{aligned} \text{size}(\bullet) &= 1 \\ \text{size}(\texttt{Tree}(\bullet, L, R)) &= 1 + \text{size}(L) + \text{size}(R) \end{aligned}$$

Prove that $\text{leaves}(T) \geq \text{size}(T)/2$ for all Trees T.

Solution:

In this problem, we define a strengthened predicate. For a tree T, let P be leaves$(T) \geq$ size$(T)/2 + 1/2$. We prove P for all trees T by structural induction.

Base Case. We show that P$(\cdot)$ holds. By definition of leaves(.), leaves$(\bullet) = 1$ and size$(\bullet) = 1$. So, leaves$(\bullet) = 1 \geq 1/2 + 1/2 =$ size$(\bullet)/2 + 1/2$.

Induction Hypothesis: Suppose P(L) and P(R) hold for trees L, R.

Induction Step: We prove P(`Tree`$(\bullet, L, R)$) holds.

$$\begin{aligned}
\text{leaves}(\texttt{Tree}(\bullet, L, R)) &= \text{leaves}(L) + \text{leaves}(R) && \text{[By Definition of leaves]} \\
&\geq (\text{size}(L)/2 + 1/2) + (\text{size}(R)/2 + 1/2) && \text{[By IH]} \\
&= 1 + \text{size}(\texttt{Tree}(\bullet, L, R))/2 && \text{[By Definition of size]} \\
&\geq \text{size}(\texttt{Tree}(\bullet, L, R))/2 + 1/2
\end{aligned}$$

This proves P(`Tree`$(\bullet, X, R)$).

Thus, the P(T) holds for all trees T.