CSE 311: Foundations of Computing I

Section 7: Contradiction, Strong Induction, Structural Induction Solutions

1. How Contradictory!

Recall that we defined the **rational numbers** as the set $\mathbb{Q} = \{\frac{a}{b} : a, b \in \mathbb{Z}, b \neq 0\}$. For example, $\frac{5}{8}, -12$ are rational.

The **irrational numbers** are all real numbers that are not rational. For example, $\sqrt{2}, \pi$ are irrational.

Prove by contradiction that the sum of a rational number with an irrational number is always irrational.

Solution:

Suppose for the sake of contradiction that there exists some rational number a and irrational number b such that $a + b$ is rational. Since a is rational, $a = \frac{a_1}{a_2}$ for some integers a_1, a_2 such that $a_2 \neq 0$. Since $a + b$ is rational, $a + b = \frac{c_1}{c_2}$ for some integers c_1, c_2 such that $c_2 \neq 0$. Then observe:

$$\begin{aligned} b &= (a + b) - a \\ &= \frac{c_1}{c_2} - \frac{a_1}{a_2} \\ &= \frac{c_1 a_2}{c_2 a_2} - \frac{a_1 c_2}{a_2 c_2} \\ &= \frac{c_1 a_2 - a_1 c_2}{a_2 c_2} \end{aligned}$$

Since a_1, a_2, c_1, c_2 are integers, $c_1 a_2 - a_1 c_2$ is an integer and $a_2 c_2$ is an integer. Since $a_2 \neq 0$ and $c_2 \neq 0$, then $a_2 c_2 \neq 0$. So by definition, b is rational. This is a contradiction to the assumption that b was irrational. Thus the sum of a rational number with an irrational number is always irrational.

2. Walk the Dogs

Suppose that a dog walker takes care of $n \geq 12$ dogs. The dog walker is very superstitious, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Solution:

Let $P(n)$ be a group with n dogs can be split into groups of 3 dogs or 7 dogs. We will prove $P(n)$ for all natural numbers $n \geq 12$ by strong induction.

Base Cases $n = 12, 13, 14$**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12)$, . . . , $P(k)$ hold for some arbitrary $k \geq 14$.

Inductive Step: Goal: Show $k + 1$ dogs can be split into groups of 3 dogs or 7 dogs.

We first form one group of 3 dogs out of the $k + 1$ dogs. Then we can divide the remaining $k - 2$ dogs into groups of 3 or 7 by the assumption $P(k - 2)$. (Note that $k \geq 14$ and so $k - 2 \geq 12$; thus, $P(k - 2)$ is among our assumptions $P(12)$, . . . , $P(k)$.)

Conclusion: $P(n)$ holds for all integers $n \geq 12$ by by principle of strong induction.

3. Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

(a) Binary strings of odd length.

Solution:

Basis: $0 \in S, 1 \in S$
Recursive Step: If $x \in S$, then $x00, x01, x10, x11 \in S$.

Explanation: *Note that the following explanations are closer to proofs. You do not need to write such lengthy justifications in your homework.*

First observe that every element constructed by this definition will be odd length, because the basis elements are odd length (length 1), and each application of the recursive step adds 2 more characters. That means if you apply the recursive step k times, you will have length $2k + 1$, which is always odd.

Conversely, observe that every odd-lengthed binary string can be constructed this way. Let s be an arbitrary odd-lengthed binary string. If s has length 1, it is either 0 or 1, which are both in S. Otherwise, it has length $2k+1$ for some $k > 0$, and we can write s in the form $0x_1x_2...x_n$ or $1x_1x_2...x_n$, where each $x_i \in \{00, 01, 10, 11\}$ has length 2. Hence, we can see that s can be built up from the base cases of 0 or 1 by applying the recursive step with x_1, then x_2, and so on up to x_n, which shows that $x \in S$.

(b) Binary strings not containing 10.

Solution:

If the string does not contain 10, then the first 1 in the string can only be followed by more 1s. Hence, it must be of the form 0^m1^n for some $m, n \in \mathbb{N}$.

Basis: $\varepsilon \in S$.

Recursive Step: If $x \in S$, then $0x \in S$ and $x1 \in S$.

Explanation: The empty string satisfies the property, and the recursive step cannot place a 0 after a 1 since it only adds 0s on the left. Hence, every string in S satisfies the property.

In the other direction, from our discussion above, any string of this form can be written as $y = 0^m1^n$ for some $m, n \in \mathbb{N}$. We can build up the string y from the empty string by applying the rule $x \to 0x$ m times and then applying the rule $x \to x1$ n times. This shows that the string y is in S.

(c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Solution:

These must be of the form 0^m1^n for some $m, n \in \mathbb{N}$ with $m \leq n$. We can ensure that by pairing up the 0s with 1s as they are added:

Basis: $\varepsilon \in S$.

Recursive Step: If $x \in S$, then $0x1 \in S$ and $x1 \in S$.

Explanation: As in the previous part, we cannot add a 0 after a 1 because we only add 0s at the front. And since every 0 comes with a 1, we always have at least as many 1s as 0s.

In the other direction, from our discussion above, any string of this form can be written as xy, where $x = 0^m1^m$ and $y = 1^{nm}$, since $n \geq m$. We can build up the string x from the empty string by applying the rule $x \to 0x1$ m times and then produce the string xy by applying the rule $x \to x1$ nm times, which shows that the string is in S.

4. Seeing Double

Recall the following recursive definition of the set of strings Σ^* from lecture:
Basis Step: $\varepsilon \in \Sigma^*$
Recursive Step: If $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$.

Now consider the following recursive definitions of the functions len and double:
$\text{len}(\varepsilon) = 0$
$\text{len}(wa) = \text{len}(w) + 1$

$\text{double}(\varepsilon) = \varepsilon$
$\text{double}(wa) = (\text{double}(w))aa$

Prove that for every string $s \in \Sigma^*$, $\text{len}(\text{double}(s)) = 2 \cdot \text{len}(s)$

Solution:

Let $\text{P}(s)$ be "$\text{len}(\text{double}(s)) = 2 \cdot \text{len}(s)$". We prove $\text{P}(s)$ for all strings $s \in \Sigma^*$ by structural induction.

Base Case. We show $\text{P}(\varepsilon)$ holds. Observe that the LHS evaluates to $\text{len}(\text{double}(\varepsilon)) = \text{len}(\varepsilon) = 0$. Observe that the RHS evaluates to $2 \cdot \text{len}(\varepsilon) = 2 \cdot 0 = 0$. Since $0 = 0$, the base case holds.

Induction Hypothesis. Suppose $\text{P}(w)$ holds for some arbitrary string $w \in \Sigma^*$. That is, $\text{len}(\text{double}(w)) = 2 \cdot \text{len}(w)$

Induction Step. Goal: $\text{len}(\text{double}(wa)) = 2 \cdot \text{len}(wa)$ for any $a \in \Sigma$

Let $a \in \Sigma$ be arbitrary. Then observe that:

$$\begin{aligned}
\text{len}(\text{double}(wa)) &= \text{len}((\text{double}(w))aa) && \text{[By Definition of double]} \\
&= 1 + \text{len}((\text{double}(w))a) && \text{[By Definition of length]} \\
&= 2 + \text{len}(\text{double}(w)) && \text{[By Definition of length]} \\
&= 2 + 2 \cdot \text{len}(w) && \text{[By IH]} \\
&= 2(1 + \text{len}(w)) && \text{[Algebra]} \\
&= 2(\text{len}(wa)) && \text{[By Definition of length]} \\
&= 2 \cdot \text{len}(wa) && \text{[Algebra]}
\end{aligned}$$

This proves $\text{P}(wa)$.

Conclusion. Thus, $\text{P}(s)$ holds for all strings $s \in \Sigma^*$ by structural induction.

5. Reversing a Binary Tree

Recall the following recursive definition of the set of Trees from lecture:
Basis Step: null $\in$ Tree
Recursive Step: If $L, R \in$ Tree and $a \in \mathbb{Z}$, then $(L, a, R) \in$ Tree.

Now consider the following recursive definitions of the functions sum and reverse:
$\text{sum}(\text{null}) = 0$
$\text{sum}((L, a, R)) = a + \text{sum}(L) + \text{sum}(R)$

reverse(null) = null
reverse$((L, a, R)) = (\text{reverse}(R), a, \text{reverse}(L))$

Prove that for every Tree $T \in$ Tree that sum(reverse(T)) = sum(T)

Solution:

Let $P(T)$ be "sum(reverse(T)) = sum(T)". We show $P(T)$ for all $T \in$ Tree by structural induction.

Base Case. We show P(null) holds. Observe that the LHS evaluates to sum(reverse(null)) = sum(null) = 0. Observe that the RHS evaluates to sum(null) = 0. Since $0 = 0$, the base case holds.

Induction Hypothesis. Suppose P(L) and P(R) hold for some arbitrary trees $L, R \in$ Tree. That is, sum(reverse(L)) = sum(L) and sum(reverse(R)) = sum(R)

Induction Step. Goal: sum(reverse$((L, a, R))$) = sum$((L, a, R))$ for any $a \in \mathbb{Z}$

Let $a \in \mathbb{Z}$ be arbitrary. Then observe that:

$$\begin{aligned}
\text{sum}(\text{reverse}((L, a, R))) &= \text{sum}((\text{reverse}(R), a, \text{reverse}(L))) && \text{[By Definition of reverse]} \\
&= a + \text{sum}(\text{reverse}(R)) + \text{sum}(\text{reverse}(L)) && \text{[By Definition of sum]} \\
&= a + \text{sum}(R) + \text{sum}(L) && \text{[By IH]} \\
&= \text{sum}((L, a, R)) && \text{[By Definition of sum]}
\end{aligned}$$

This proves P$((L, a, R))$.

Conclusion. Thus, P(T) holds for all trees $T \in$ Tree by structural induction.