Section 08: Solutions

1. Recursively Defined Sets of Strings

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 even length.

Solution:

Basis: $\varepsilon \in S$.
Recursive Step: If $x \in S$, then $x00, x01, x10, x11 \in S$.
Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step.

"Brief " Justification: We will show that $x \in S$ iff x has even length (i.e., |x| = 2n for some $n \in \mathbb{N}$). (Note: "brief" is in quotes here. Try to write shorter explanations in your homework assignment when possible!)

Suppose $x \in S$. If x is the empty string, then it has length 0, which is even. Otherwise, x is built up from the empty string by repeated application of the recursive step, so it is of the form $x_1x_2...x_n$, where each $x_i \in \{00, 01, 10, 11\}$. In that case, we can see that $|x| = |x_1| + |x_2| + \cdots + |x_n| = 2n$, which is even. Now, suppose that x has even length. If it's length is zero, then it is the empty string, which is in S. Otherwise, it has length 2n for some $n > 0$, and we can write x in the form $x_1x_2...x_n$, where each $x_i \in \{00, 01, 10, 11\}$ has length 2. Hence, we can see that x can be built up from the empty string 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 $x1 \in S$.

Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step.

Brief Justification: 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 $x1 \in S$.

Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step.

Brief Justification: 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 $y = 1^{n-m}$, 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 $x \to x1$ $n-m$ times, which shows that the string is in S.

(d) Binary strings containing at most two 0s and at most two 1s.

Solution:

This is the set of all binary strings of length at most 4 *except* for these:

$$000, 1000, 0100, 0010, 0001, 0000, 111, 0111, 1011, 1101, 1110, 1111$$

Since this is a **finite set**, we can define it recursively using only basis elements and no recursive step.

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 $\mathsf{append}(c, X)$ is a string.

Recall the following recursive definition of the function len:

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

Now, consider the following recursive definition:

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

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

Solution:

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

Base Case (X = ""): By definition, $\mathsf{len}(\mathsf{double}(\texttt{""})) = \mathsf{len}(\texttt{""}) = 0 = 2 \cdot 0 = 2\mathsf{len}(\texttt{""})$, so $\mathsf{P}(\texttt{""})$ holds.

Inductive Hypothesis: Suppose $\mathsf{P}(Z)$ holds for some arbitrary string Z.

Inductive Step: Goal: Show that $\mathsf{P}(\mathsf{append}(c, Z))$ holds for any character c.

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

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

Conclusion: $\mathsf{P}(X)$ 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:

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

Also, recall the definition of size on trees:

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

Prove that $\mathsf{leaves}(T) \geq \mathsf{size}(T)/2 + 1/2$ for all Trees T.

Solution:

For a tree T, let P be $\mathsf{leaves}(T) \geq \mathsf{size}(T)/2 + 1/2$. We prove P for all trees T by structural induction on T.

Base Case (T = $\bullet$): By definition of $\mathsf{leaves}(\bullet)$, $\mathsf{leaves}(\bullet) = 1$ and $\mathsf{size}(\bullet) = 1$. So, $\mathsf{leaves}(\bullet) = 1 \geq 1/2 + 1/2 = \mathsf{size}(\bullet)/2 + 1/2$, so $\mathsf{P}(\bullet)$ holds.

Inductive Hypothesis: Suppose $\mathsf{P}(L)$ and $\mathsf{P}(R)$ hold for some arbitrary trees L, R.

Inductive Step: Goal: Show that $\mathsf{P}(\texttt{Tree}(\bullet, L, R))$ holds.

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

This proves $\mathsf{P}(\texttt{Tree}(\bullet, L, R))$.

Conclusion: Thus, $\mathsf{P}(T)$ holds for all trees T by structural induction.

(c) Prove the previous claim using strong induction. Define $P(n)$ as "all trees T of size n satisfy $\mathsf{leaves}(T) \geq \mathsf{size}(T)/2 + 1/2$". You may use the following facts:

- For any tree T we have $\text{size}(T) \geq 1$.
- For any tree T, $\text{size}(T) = 1$ if and only if $T = \bullet$.

If we wanted to prove these claims, we could do so by structural induction.

Note, in the inductive step you should start by letting T be an arbitrary tree of size $k+1$.

Solution:

Let $P(n)$ be "all trees T of size n satisfy $\text{leaves}(T) \geq \text{size}(T)/2 + 1/2$". We show $P(n)$ for all integers $n \geq 1$ by strong induction on n.

Base Case: Let T be an arbitrary tree of size 1. The only tree with size 1 is $\bullet$, so $T = \bullet$. By definition, $\text{leaves}(T) = \text{leaves}(\bullet) = 1$ and thus $\text{size}(T) = 1 = 1/2 + 1/2 = \text{size}(T)/2 + 1/2$. This shows the base case holds.

Inductive Hypothesis: Suppose that $P(j)$ holds for all integers $j = 1, 2, \dots, k$ for some arbitrary integer $k \geq 1$.

Inductive Step: Let T be an arbitrary tree of size $k+1$. Since $k+1 > 1$, we must have $T \neq \bullet$. It follows from the definition of a tree that $T = \texttt{Tree}(\bullet, L, R)$ for some trees L and R. By definition, we have $\text{size}(T) = 1 + \text{size}(L) + \text{size}(R)$. Since sizes are non-negative, this equation shows $\text{size}(T) > \text{size}(L)$ and $\text{size}(T) > \text{size}(R)$ meaning we can apply the inductive hypothesis. This says that $\text{leaves}(L) \geq \text{size}(L)/2 + 1/2$ and $\text{leaves}(R) \geq \text{size}(R)/2 + 1/2$.

We have,

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

This shows $P(k+1)$.

Conclusion: $P(n)$ holds for all integers $n \geq 1$ by the principle of strong induction.

Note, this proves the claim for all trees because every tree T has some size $s \geq 1$. Then $P(s)$ says that all trees of size s satisfy the claim, including T.

3. Reversing a Binary Tree

Consider the following definition of a (binary) **Tree**.

Basis Step `Nil` is a **Tree**.

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

The sum function returns the sum of all elements in a **Tree**.

$$\begin{aligned}
\text{sum}(\texttt{Nil}) &= 0 \\
\text{sum}(\texttt{Tree}(x, L, R)) &= x + \text{sum}(L) + \text{sum}(R)
\end{aligned}$$

The following recursively defined function produces the mirror image of a **Tree**.

$$\begin{aligned}
\text{reverse}(\texttt{Nil}) &= \texttt{Nil} \\
\text{reverse}(\texttt{Tree}(x, L, R)) &= \texttt{Tree}(x, \text{reverse}(R), \text{reverse}(L))
\end{aligned}$$

Show that, for all **Tree**s T that

$$\text{sum}(T) = \text{sum}(\text{reverse}(T))$$

Solution:

For a **Tree** T, let $P(T)$ be "$\text{sum}(T) = \text{sum}(\text{reverse}(T))$". We show $P(T)$ for all **Trees** T by structural induction.

Base Case: By definition we have $\text{reverse}(\texttt{Nil}) = \texttt{Nil}$. Applying sum to both sides we get $\text{sum}(\texttt{Nil}) = \text{sum}(\text{reverse}(\texttt{Nil}))$, which is exactly $P(\texttt{Nil})$, so the base case holds.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ hold for some arbitrary **Tree**s L and R.

Inductive Step: Let x be an arbitrary integer. Goal: Show $P(\texttt{Tree}(x, L, R))$ holds.

We have,

$$\begin{aligned}
\text{sum}(\text{reverse}(\texttt{Tree}(x, L, R))) &= \text{sum}(\texttt{Tree}(x, \text{reverse}(R), \text{reverse}(L))) && \text{[Definition of reverse]}\\
&= x + \text{sum}(\text{reverse}(R)) + \text{sum}(\text{reverse}(L)) && \text{[Definition of sum]}\\
&= x + \text{sum}(R) + \text{sum}(L) && \text{[Inductive Hypothesis]}\\
&= x + \text{sum}(L) + \text{sum}(R) && \text{[Commutativity]}\\
&= \text{sum}(\texttt{Tree}(x, L, R)) && \text{[Definition of sum]}
\end{aligned}$$

This shows $P(\texttt{Tree}(x, L, R))$.

Conclusion: Therefore, $P(T)$ holds for all **Tree**s T by structural induction.

4. Walk the Dawgs

Suppose a dog walker takes care of $n \geq 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove the dog walker can always split the n dogs into groups of 3 or 7.

Solution:

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

Base Cases $n = 12, 13, 14,$ or 15: $12 = 3 + 3 + 3 + 3$, $13 = 3 + 7 + 3$, $14 = 7 + 7$, So $P(12)$, $P(13)$, and $P(14)$ hold.

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

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

We first form one group of 3 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 $P(12), \ldots, P(k)$.)

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

5. For All

For this problem, we'll see an incorrect use of induction. For this problem, we'll think of all of the following as binary trees:

- A single node.
- A root node, with a left child that is the root of a binary tree (and no right child)

- A root node, with a right child that is the root of a binary tree (and no left child)
- A root node, with both left and right children that are roots of binary trees.

Let $P(n)$ be "for all trees of height n, the tree has an odd number of nodes"

Take a moment to realize this claim is false.

Now let's see an incorrect proof:

We'll prove $P(n)$ for all $n \in \mathbb{N}$ by induction on n.

Base Case ($n = 0$): There is only one tree of height 0, a single node. It has one node, and $1 = 2 \cdot 0 + 1$, which is an odd number of nodes.

Inductive Hypothesis: Suppose $P(i)$ holds for $i = 0, ..., k$, for some arbitrary $k \geq 0$.

Inductive Step: Let T be an arbitrary tree of height k. All trees with nodes (and since $k \geq 0$, T has at least one node) have a leaf node. Add a left child and right child to a leaf (pick arbitrarily if there's more than one), This tree now has height $k + 1$ (since T was height k and we added children below). By IH, T had an odd number of nodes, call it $2j + 1$ for some integer j. Now we have added two more, so our new tree has $2j + 1 + 2 = 2(j + 1) + 1$ nodes. Since j was an integer, so is $j + 1$, and our new tree has an odd number of nodes, as required, so $P(k + 1)$ holds.

By the principle of induction, $P(n)$ holds for all $n \in \mathbb{N}$. Since every tree has an (integer) height of 0 or more, every tree is included in some $P()$, so the claim holds for all trees.

(a) What is the bug in the proof? **Solution:**

> The proof, in trying to show something about an arbitrary of height $k+1$, builds a **particular** tree of height $k+1$, not an arbitrary one. While the tree built is indeed of height $k+1$ and has an odd number of nodes, it is not an *arbitrary* tree of height $k + 1$.
>
> Why is it that in this problem we have to start with $k + 1$, but we didn't in the "Walk the Dawgs" problem above? "Walk the Dawgs" is asking you to prove an exists statement ("can split" not "for every possible split..."). When proving an exists, you just say "here's how to do it", you don't need to introduce an arbitrary variable

(b) What should the starting point and target of the IS be (you can't write a full proof, as the claim is false). **Solution:**

> "Let T be an arbitrary tree of height $k + 1$" should be the first sentence. "T has an odd number of nodes" is the target. Notice that the only difference

6. Induction with Inequality

Prove that $6n + 6 < 2^n$ for all $n \geq 6$. **Solution:**

> Let $P(n)$ be "$6n + 6 < 2^n$". We will prove $P(n)$ for all integers $n \geq 6$ by induction on n
>
> **Base Case** $(n = 6)$**:** $6 \cdot 6 + 6 = 42 < 64 = 2^6$, so $P(6)$ holds.
>
> **Inductive Hypothesis:** Assume that $6k + 6 < 2^k$ for an arbitrary integer $k \geq 6$.

Inductive Step: Goal: Show $6(k+1)+6 < 2^{k+1}$

$$\begin{aligned} 6(k+1)+6 &= 6k+6+6 \\ &< 2^k + 6 && \text{[Inductive Hypothesis]} \\ &< 2^k + 2^k && \text{[Since } 2^k > 6\text{, since } k \geq 6\text{]} \\ &= 2 \cdot 2^k \\ &= 2^{k+1} \end{aligned}$$

So $P(k) \to P(k+1)$ for an arbitrary integer $k \geq 6$.

Conclusion: $P(n)$ holds for all integers $n \geq 6$ by the principle of induction.

7. Induction with Formulas

These problems are a little more difficult and abstract. Try making sure you can do all the other problems before trying these ones.

(a) (i) Show that given two sets A and B that $\overline{A \cup B} = \overline{A} \cap \overline{B}$. (Don't use induction.)

Solution:

Let x be arbitrary. Then,

$$\begin{aligned} x \in \overline{A \cup B} &\equiv \neg(x \in A \cup B) && \text{[Definition of complement]} \\ &\equiv \neg(x \in A \vee x \in B) && \text{[Definition of union]} \\ &\equiv \neg(x \in A) \wedge \neg(x \in B) && \text{[De Morgan's Laws]} \\ &\equiv x \in \overline{A} \wedge x \in \overline{B} && \text{[Definition of complement]} \\ &\equiv x \in (\overline{A} \cap \overline{B}) && \text{[Definition of intersection]} \end{aligned}$$

Since x was arbitrary we have that $x \in \overline{A \cup B}$ if and only if $x \in \overline{A} \cap \overline{B}$ for all x. By the definition of set equality we've shown,

$$\overline{A \cup B} = \overline{A} \cap \overline{B}.$$

(ii) Show using induction that for an integer $n \geq 2$, given n sets $A_1, A_2, \ldots, A_{n-1}, A_n$ that

$$\overline{A_1 \cup A_2 \cup \cdots \cup A_{n-1} \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_{n-1}} \cap \overline{A_n}$$

Solution:

Let $P(n)$ be "given n sets $A_1, A_2, \ldots, A_{n-1}, A_n$ it holds that $\overline{A_1 \cup A_2 \cup \cdots \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_{n-1}} \cap \overline{A_n}$." We show $P(n)$ for all integers $n \geq 2$ by induction on n.

Base Case: $P(2)$ says that for two sets A_1 and A_2 that $\overline{A_1 \cup A_2} = \overline{A_1} \cap \overline{A_2}$, which is exactly part (a) so $P(2)$ holds.

Inductive Hypothesis: Suppose that $P(k)$ holds for some arbitrary integer $k \geq 2$.

Inductive Step: Let $A_1, A_2, \ldots, A_k, A_{k+1}$ be sets. Then by part (a) we have,

$$\overline{(A_1 \cup A_2 \cup \cdots \cup A_k) \cup A_{k+1}} = \overline{A_1 \cup A_2 \cup \cdots \cup A_k} \cap \overline{A_{k+1}}.$$

By the inductive hypothesis we have $\overline{A_1 \cup A_2 \cup \cdots A_k} = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_k}$. Thus,

$$\overline{A_1 \cup A_2 \cup \cdots \cup A_k} \cap \overline{A_{k+1}} = (\overline{A_1} \cap \overline{A_2} \cap \cdots \overline{A_k}) \cap \overline{A_{k+1}}.$$

We've now shown

$$\overline{A_1 \cup A_2 \cup \cdots \cup A_k \cup A_{k+1}} = \overline{A_1} \cap \overline{A_2} \cap \cdots \overline{A_k} \cap \overline{A_{k+1}}.$$

which is exactly $P(k+1)$.

Conclusion $P(n)$ holds for all integers $n \geq 2$ by the principle of induction.

(b) (i) Show that given any integers a, b, and c, if $c \mid a$ and $c \mid b$, then $c \mid (a+b)$. (Don't use induction.)

Solution:

Let a, b, and c be arbitrary integers and suppose that $c \mid a$ and $c \mid b$. Then by definition there exist integers j and k such that $a = jc$ and $b = kc$. Then $a + b = jc + kc = (j+k)c$. Since $j + k$ is an integer, by definition we have $c \mid (a+b)$.

(ii) Show using induction that for any integer $n \geq 2$, given n numbers $a_1, a_2, \ldots, a_{n-1}, a_n$, for any integer c such that $c \mid a_i$ for $i = 1, 2, \ldots, n$, that

$$c \mid (a_1 + a_2 + \cdots + a_{n-1} + a_n).$$

In other words, if a number divides each term in a sum then that number divides the sum.

Solution:

Let $P(n)$ be "given n numbers $a_1, a_2, \ldots, a_{n-1}, a_n$, for any integer c such that $c \mid a_i$ for $i = 1, 2, \ldots, n$, it holds that $c \mid (a_1 + a_2 + \cdots + a_n)$." We show $P(n)$ holds for all integer $n \geq 2$ by induction on n.

Base Case: $P(2)$ says that given two integers a_1 and a_2, for any integer c such that $c \mid a_1$ and $c \mid a_2$ it holds that $c \mid (a_1 + a_2)$. This is exactly part (a) so $P(2)$ holds.

Inductive Hypothesis: Suppose that $P(k)$ holds for some arbitrary integer $k \geq 2$.

Inductive Step: Let $a_1, a_2, \ldots, a_k, a_{k+1}$ be $k+1$ integers. Let c be arbitrary and suppose that $c \mid a_i$ for $i = 1, 2, \ldots, k+1$. Then we can write

$$a_1 + a_2 + \cdots + a_k + a_{k+1} = (a_1 + a_2 + \cdots + a_k) + a_{k+1}.$$

The sum $a_1 + a_2 + \cdots + a_k$ has k terms and c divides all of them, meaning we can apply the inductive hypothesis. It says that $c \mid (a_1 + a_2 + \cdots + a_k)$. Since $c \mid (a_1 + a_2 + \cdots + a_k)$ and $c \mid a_{k+1}$, by part (a) we have,

$$c \mid (a_1 + a_2 + \cdots + a_k + a_{k+1}).$$

This shows $P(k+1)$.

Conclusion: $P(n)$ holds for all integers $n \geq 2$ by induction the principle of induction.

8. One-to-One and Onto

For each of these functions, state whether it is one-to-one, onto, both, or neither.

(a) $f : \mathbb{N} \to \mathbb{N}$, $f(x) = x^2$ **Solution:**

For this domain and co-domain, the function is one-to-one, but not onto.
It is one-to-one: For a natural number output (i.e., x^2), the possible inputs are $x, -x$, but only one of those can be a natural number (since natural numbers are all positive).
It is not onto: for example, $5 \in \mathbb{N}$ (i.e., the codomain) but no natural number can be put into the function to produce 5.

(b) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$ **Solution:**

For this domain and co-domain, the function is neither one-to-one nor onto.
It is not one-to-one: 16 can be produced by both $4, -4$ as inputs.
It is not onto, -5 (for example) cannot be produced as output, since real-numbers when squared are always non-negative.

(c) $f : \mathbb{R}^+ \to \mathbb{R}^+$, $f(x) = x^2$, where $\mathbb{R}^+ = \{x : x \in \mathbb{R} \wedge x \geq 0\}$, i.e., the set of non-negative real numbers. **Solution:**

For this domain and co-domain, the function is both one-to-one and onto.
It is one-to-one, if $x^2 = y^2$, then $\pm x = \pm y$, but since all inputs to f are non-negative, we must have that $x = y$.
It is onto, for an arbitrary output z, by def of f, $z = x^2$, that x is a real number, choosing x to be positive will give us a valid element of the domain.

Notice that the domain and co-domain matter! You have to know both to tell whether the function is one-to-one or onto.

9. A Bijection Proof

Let A be the set of negative integers, i.e., $A = \{-1, -2, -3, ...\}$; let B be the set of integers at least 10, i.e., $B = \{10, 11, 12, 13, ...\}$ Show that $f : A \to B$ defined by $f(x) = |x| + 9$ is a bijection.

You may use these facts:

- for negative numbers x, y: $|x| = |y| \to x = y$
- for negative numbers $|x| = -x$

that **Solution:**

One-to-One: Let x, y be arbitrary elements of A such that $f(x) = f(y)$. By definitions of f, $|x| + 9 = |y| + 9$. Cancelling the 9's, we have $|x| = |y|$. By the fact $x = y$. Since x, y were arbitrary, we have met the definitions of one-to-one.

Onto: Let y be an arbitrary element in B. Note that y is a positive integer and $y \geq 10$. Consider the value $x = -y + 9$. Observe that x is negative (since y is positive but greater than 9, negating it will give a negative greater in absolute value than 9, leaving a negative result). Since x is negative, when we calculate $|x| + 9$ we get $|x| + 9 = |-y + 9| + 9 = -(-y + 9) + 9 = y$, as required. Further note that the x we chose is in the set A: we already showed it was negative, and since integers are closed under multiplication and addition, x is an integer. Thus x is a value which gives $f(x) = y$; since y was arbitrary, f is onto.

Since f is both onto and one-to-one it is a bijection.