CSE 311: Foundations of Computing I

Section 7: Strong Induction and Recursive Sets Solutions

1. Binary Representations

Prove that every natural number can be written as a sum of *distinct* powers of two. (I.e., that it has a unique binary representation.)

Solution:

Let $P(n)$ be "n can written as a sum of distinct powers of two, each no larger than n". We will prove $P(n)$ for all integers $n \in \mathbb{N}$ by strong induction.

Base Case $(n = 0)$**:** 0 is equal to an empty sum (no powers of two), so $P(0)$ holds.

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

Induction Step: Our goal is to show $P(k+1)$. I.e., that $k+1$ can be written as a sum of distinct powers of two, each no larger than $k+1$.

Let 2^ℓ be the largest power of two not greater than $k+1$ (i.e. $\ell = \lfloor \log_2(k+1) \rfloor$). Let $r = (k+1) - 2^\ell$, and note that $r < k+1$ since $2^\ell > 0$, so that we can apply the inductive hypothesis to r to write it as a sum $r = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_t}$, where each i_j is distinct and satisfies $2^{i_j} \leq r$.

We must have each $i_j < \ell$. Otherwise, we have $i_j \geq \ell$, which means $2^{i_j} \geq 2^\ell$ and so $r = 2^{i_1} + \cdots + 2^{i_t} \geq 2^{i_j} \geq 2^\ell$. That means $k+1 = r + 2^\ell \geq 2^\ell + 2^\ell = 2^{\ell+1}$, showing that $2^{\ell+1} \leq k+1$, which contradicts our assumption that 2^ℓ was the largest power of 2 that is less than or equal to $k+1$. (Note that this is a proof by contradiction within the larger proof.)

Now, write $k+1$ as $r + 2^\ell = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_t} + 2^\ell$, a sum of powers of two. Each of the i_j's are distinct from each other, by assumption, and from ℓ, since each satisfies $i_j < \ell$. Furthermore, we have $2^{i_j} \leq r < 2^\ell \leq k+1$, so none of the powers of two in the sum are larger than $k+1$. This shows $P(k+1)$.

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

2. Cantelli's Rabbits

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

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

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 Cases ($n = 0, 1$): $f(0) = 0$ by definition, so $\mathsf{P}(0)$ holds, and $f(1) = 1$, so $\mathsf{P}(1)$ holds.

Induction Hypothesis: Assume that for some arbitrary integer $k \geq 1$, $\mathsf{P}(j)$ holds for all $0 \leq j \leq k$. That is, for each number in this range, we have $f(j) = j$.

Induction Step: We show $\mathsf{P}(k+1)$, i.e. that $f(k+1) = k+1$.

Since $k + 1 \geq 2$, we have

$$\begin{aligned} f(k+1) &= 2f(k) - f(k-1) && \text{Definition of } f \\ &= 2(k) - f(k-1) && \text{Inductive Hypothesis} \\ &= 2(k) - (k-1) && \text{Inductive Hypothesis} \\ &= k+1 && \text{Algebra} \end{aligned}$$

which is $\mathsf{P}(k+1)$.

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

3. 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 $x_1 x_2 \cdots x_n$, 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 $x_1 x_2 \cdots x_n$, 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 $0^m 1^n$ 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 $y = 0^m 1^n$ for some $m, n \in \mathbb{N}$. We can build up the string y from the empty string by applying the rule $x \mapsto 0x$ m times and then applying the rule $x \mapsto x1$ 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 $0^m 1^n$ 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 $x = 0^m 1^m$ and $y = 1^{n-m}$, since $n \geq m$. We can build up the string x from the empty string by applying the rule $x \mapsto 0x1$ m times and then produce the string xy by applying the rule $x \mapsto x1$ $n-m$ 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.