1 Inductive Proofs

Prove each of the following claims by induction

Claim 1. *The sum of the first* n *even numbers is* n^2+n. *That is,* $\sum_{i=1}^{n} 2i = n^2+n$.

Proof. Base case $i = 1$

$$\begin{aligned} LHS &= 2 * 1 \\ &= 2 \\ RHS &= 1^2 + 1 \\ &= 2 \\ LHS &= RHS \end{aligned}$$

The base case is established.

Inductive hypothesis: Assume that

$$\sum_{i=1}^{n} 2i = n^2 + n \text{ for all } 1 \leq n \leq k$$

Inductive step: Show that

$$\begin{aligned} \sum_{i=1}^{k+1} 2i &= \sum_{i=1}^{k} 2i + 2(k+1) \\ &= k^2 + k + 2k + 2 \text{ This is true on our Inductive hypothesis} \\ &= k^2 + 2k + 1 + k + 1 \\ &= (k+1)^2 + (k+1) \end{aligned}$$

We have shown that $\sum_{i=1}^{k+1} 2i = (k+1)^2 + (k+1)$.

□

Claim 2. $\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}$

Proof. Base case $i = 1$

$$\begin{aligned} LHS &= \frac{1}{2^1} \\ &= \frac{1}{2} \\ RHS &= 1 - \frac{1}{2^1} \\ &= \frac{1}{2} \\ LHS &= RHS \end{aligned}$$

The base case is established.

Inductive hypothesis: Assume that

$$\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n} \text{ for all } 1 \leq n \leq k$$

Inductive step: Show that

$$\begin{aligned} \sum_{i=1}^{k+1} \frac{1}{2^i} = \sum_{i=1}^{k} &+ \frac{1}{2^{k+1}} \\ &= 1 - \frac{1}{2^k} + \frac{1}{2^{k+1}} \text{ This is true on our Inductive hypothesis} \\ &= 1 + \frac{-2}{2^{k+1}} + \frac{1}{2^{k+1}} \\ &= 1 - \frac{1}{2^{k+1}} \end{aligned}$$

We have shown that $\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}$.

□

Claim 3. $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

Proof. Base case $i = 0$

$$\begin{aligned} LHS &= 2^0 \\ &= 1 \\ RHS &= 2^{0+1} - 1 \\ &= 2 - 1 \\ &= 1 \\ LHS &= RHS \end{aligned}$$

The base case is established.

Inductive hypothesis: Assume that

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \text{ for all } 0 \leq n \leq k$$

Inductive step: Show that

$$\begin{aligned}\sum_{i=0}^{k+1} 2^i &= \sum_{i=0}^{k} 2^i + 2^{k+1} \\ &= 2^{k+1} - 1 + 2^{k+1} \text{ This is true on our Inductive hypothesis} \\ &= 2 * 2^{k+1} - 1 \\ &= 2^{(k+1)+1} - 1\end{aligned}$$

We have shown that $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$.

□

2 Recursive Invariants

The function `minEven`, given below in pseudocode, takes as input an array A of size n of numbers. It returns the smallest *even* number in the array. If no even numbers appear in the array, it returns positive infinity ($+\infty$). Using induction, prove that the `minEven` function works correctly. Clearly state your recursive invariant at the beginning of your proof.

```
Function minEven(A,n)
  If n is 0 Then
    Return +infinity
  Else
    Set best To minEven(A,n-1)
    If A[n-1] < best And A[n-1] is even Then
      Set best To A[n-1]
    EndIf
    Return best
  EndIf
EndFunction
```

Proof. P(n) = the function minEven(A,n) returns the smallest even number in the frist n values of the array, otherwise it returns positive infinity.

Base case $n = 0$

The function returns positive infinity when $n = 0$. This is true because there is no value inside the array. The base case is established.

Inductive hypothesis: Assume that the function minEven is correct for all arrays of size n where $0 \leq n \leq k$.

Inductive step: Show that the function minEven is correct for all arrays of size k+1.

minEven(A,k+1)

case 2

Set the variable best to the result of minEven(A,k). We know minEven(A,k) returns the smallest even number in the frist k values of the array, otherwise it returns positive infinity base on the inductive hypothesis.

case 2a

If A[k] is less than the best and A[k] is even, then it set best to A[k]. The function returns best.

case 2b

If A[k] is greater than best or A[k] is odd, then best remains the same. The function returns best.

The function has compared the element A[k+1] to minEven(A,k). We know that minEven(A,k) returns the smallest even number in the frist n values of the array, otherwise it returns positive infinity through the inductive hypothesis. Thus, minEven(A,k+1) will return the smallest even number in the frist k+1 values of the array, otherwise it returns positive infinity

$\square$