Well-Ordering Principle & Mathematical Induction

We define the set of positive elements of Z as

$$Z^+ = \{x \in Z \mid x > 0\} = \{x \in Z \mid x \geq 1\}$$

We can do the same for rational & real numbers.

$$Q^+ = \{x \in Q \mid x > 0\} \quad \& \quad R^+ = \{x \in R \mid x > 0\}$$

We cannot represent Q^+ or R^+ using $\geq$ as we did for Z^+.

Every nonempty subset X of Z^+ contains an integer, a, such that $a \leq x$, for all $x \in X$. In other words X contains a "least" (or smallest) element. This is not true for Q^+ or R^+ as neither of these sets contain least elements. (Hint: There is no smallest positive rational number or smallest positive real number.

The Well ordering Principle:

For every non-empty subset of Z^+, there exists a smallest element. This is often expressed by saying that Z^+ is well-ordered.

This principle is the basis of a proof technique known as mathematical induction.

The Principle of Mathematical Induction

Let $S(n)$ denote an "open mathematical statement", that involves one or more occurrences of the variable n, which represents a positive integer

a) If $S(1)$ is true; AND (basis step)

b) If whenever $S(K)$ is true (for some particular, but arbitrarily chosen, $K \in Z^+$), then $S(K+1)$ is true (the inductive step)

then $S(n)$ is true for all $n \in Z^+$.

Note: part (a) assumes that the first element, n_0, index is 1; however, as long as part (a) is true for some first element $n_0 \in Z^+$, it is sufficient for the inductive process.

The Principle of Mathematical Induction can be expressed using quantifiers as:

$$[S(n_0) \wedge [\forall K \geq n_0 [S(K) \Rightarrow S(K+1)]]] \Rightarrow \forall n \geq n_0 \, S(n)$$

Examples:

Prove: For all $n \in \mathbb{Z}^+$, $\sum_{i=1}^{n} i = 1+2+3+\dots+n = \frac{n(n+1)}{2}$

Soln:

$S(n)$: $\sum_{i=1}^{n} i = 1+2+3+\dots+n = \frac{n(n+1)}{2}$ (Statement to Prove)

$S(1)$: $\sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1$ (Basis Step)

Assume the result is true for $n=K$ (for some $K \in \mathbb{Z}^+$)
Next, establish our inductive step by showing $S(K+1)$ is also true (i.e. correct)

$$S(K+1) = \sum_{i=1}^{K+1} i = \frac{(K+1)(K+2)}{2}$$

$$= 1+2+3+\dots+K+(K+1)$$

$$= \sum_{i=1}^{K} i + (K+1)$$

$$= \frac{K(K+1)}{2} + \frac{(K+1)2}{2}$$

$$= \frac{K^2+K+2K+2}{2}$$

$$= \frac{K^2+3K+2}{2}$$

$$= \frac{(K+1)(K+2)}{2}$$

Prove that for each $n \in \mathbb{Z}^+$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Sol^n:

$$S(n): \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Basis Step: $S(1)$

$$S(1) = \sum_{i=1}^{1} i^2 = \frac{1(1+1)(2(1)+1)}{6} = \frac{1(2)(3)}{6} = \frac{6}{6} = 1$$

Inductive Step:

$$S(K) = \sum_{i=1}^{K} i^2 = \frac{K(K+1)(2K+1)}{6}$$

$$S(K+1) = \sum_{i=1}^{K+1} i^2 = \frac{(K+1)((K+1)+1)(2(K+1)+1)}{6}$$

$$= \frac{(K+1)(K+2)(2K+2+1)}{6}$$

$$= \frac{(K+1)(K+2)(2K+3)}{6}$$

Also,

$$S(K+1) = \sum_{i=1}^{K+1} i^2 = 1^2+2^2+3^2+\dots+K^2+(K+1)^2$$

$$= \sum_{i=1}^{K} i^2 + (K+1)^2$$

$$= \frac{K(K+1)(2K+1)}{6} + (K+1)^2$$

$$= (K+1)\left[\frac{K(2K+1)}{6} + \frac{(K+1)6}{6}\right]$$

$$= (K+1)\left[\frac{2K^2 + K + 6K + 6}{6}\right]$$

$$= (K+1)\left[\frac{2K^2 + 7K + 6}{6}\right]$$

$$= \frac{(K+1)(K+2)(2K+3)}{6}$$

What about:

1) $1 = 1 = 1^2$

2) $1 + 3 = 4 = 2^2$

3) $1 + 3 + 5 = 9 = 3^2$

4) $1 + 3 + 5 + 7 = 16 = 4^2$

$n \in Z^+ \quad S(n): \sum_{i=1}^{n} (2i-1) = n^2$ } Prove

From initial calculations, we can see that $S(1), S(2), S(3),$ & $S(4)$ are true (basis step).

$$S(K): \sum_{i=1}^{K} (2i-1) = K^2$$

$$S(K+1): \sum_{i=1}^{K+1} (2i-1) = \sum_{i=1}^{K} (2i-1) + [(2)(K+1) - 1]$$

$$= K^2 + 2K + 2 - 1$$

$$= K^2 + 2K + 1$$

$$= (K+1)^2$$

Applying Mathematical Induction to proving software Functionality

```
int foo (int x, int y, int n)
{
    if (n<0)    // n ∈ N , otherwise return an
      exit(1);  // error
    else
    {
      while(n!=0)      // We want to prove
      {                // that for all n ≥ 0
         x = x*y;      // this program returns
         n = n-1;      // xy^n
      }
      return x;
    }
}
```

Prove $S(n) = xy^n$

Note that the program will automatically exit for $n < 0$

The basis step is $S(0) = xy^0 = x$. When $n=0$, the above while loop will exit without executing. As such, x will equal the initial value passed in as a parameter. In other words, $S(0) = x$ in the above code.

We assume that $S(K) = xy^K$, for all $K \geq 0$.
Therefore $S(K+1)$, for all $K+1 \geq 1$, means that the program will execute the while loop at least once.
We can also see from the code that y remains unchanged.
Each iteration multiplies the previous result by y. So

$$S(K+1) = S(K)y = (xy^K)y = xy^{K+1}$$

Check out the practice problems & additional examples linked to from the lecture set web page for more practice & solutions.

Remember the format:

① Statement to prove $S(n)$
② Basis Step
③ $S(K)$ [Assume]
④ $S(K+1)$ [Prove]