MATH/CSE 1019 Discrete Math for Computer Science

Assignment 6

Questions:

1. (7 points) Use mathematical induction to prove

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\cdots+\frac{1}{n(n+1)}=1-\frac{1}{n+1}$$

   whenever n is a positive integer.

   Proof;

   Basis Step: n=1: L.H.S = $\frac{1}{1*2}=\frac{1}{2}$, and R.H.S.= 1-$\frac{1}{2+1}=\frac{1}{2}$. Therefore, the statement is true.

   Inductive Step: Assume the statement is true when n=k,

   i.e. $\frac{1}{1*2}+\frac{1}{2*3}+\cdots+\frac{1}{k*(k+1)}=1-\frac{1}{k+1}$.

   Prove the statement is true when n=k+1,

   i.e. $\frac{1}{1*2}+\frac{1}{2*3}+\cdots+\frac{1}{k*(k+1)}+\frac{1}{(k+1)*(k+2)}=1-\frac{1}{k+2}$.

   L.H.S.= $\frac{1}{1*2}+\frac{1}{2*3}+\cdots+\frac{1}{k*(k+1)}+\frac{1}{(k+1)*(k+2)}$

   $= \boxed{1-\frac{1}{k+1}}+\frac{1}{(k+1)*(k+2)}$ (by the assumption)

   $=1-\frac{(k+2)-1}{(k+1)*(k+2)}$ (use common denominator)

   $=1-\frac{k+1}{(k+1)*(k+2)}$

   $=1-\frac{1}{k+2}$

   = R.H.S.

   Therefore, the statement is true when n=k+1.

By mathematical induction, the statement is true whenever n is a positive integer.

2. (7 points) Use loop invariant to prove that the program for computing the sum of 1,…,n is correct.

```
INPUT: Integer n
OUTPUT: The sum of 1,…,n

S(n)
1.  i ← 0
2.  while   n>0
3.      do    i ← i + n
4.              n ← n-1
5.  return(i)
```

Note: First we have to formulate the loop invariant. This algorithm adds n, (n-1), (n-2),… 1 to i gradually. A good loop invariant should be a statement that is always true and ensures the output is correct.

Loop Invariant: "Before iteration k, i contains the value $n_0 + (n_0 - 1) + \cdots + (n_0 - k + 2)$, and $n = n_0 - k + 1$." Here the sum is defined to be 0 if the last term is bigger than the first term.

Note: Let the initial value of n be denoted n_0, since the variable n counts down and changes value in each iteration.

Proof:

Basis Step: Before the first iteration i=0, n=n_0. The loop invariant is true.

Inductive Step:

Assume the loop invariant is true before the kth iteration. Then before iteration k, i contains the value $n_0 + (n_0 - 1) + \cdots + (n_0 - k + 2)$, and $n = n_0 - k + 1$.

If n>0, the loop does not terminate. The code in line 3 adds n to i, then i=$n_0 + (n_0 - 1) + \cdots + (n_0 - k + 2) + (n_0 - k + 1) = n_0 + (n_0 - 1) + \cdots + (n_0 - k + 2) + (n_0 - (k + 1) + 2)$. The code in line 4 subtracts 1 from n, then n=$n_0 - k = n_0 - (k + 1) + 1$. Therefore the loop invariant is true before the (k+1) the iteration.

The loop terminates when n=0. Since each time the algorithm subtracts 1 from n, the number of iteration k should be $n_0 + 1$. By the loop invariance, $i = n_0 + (n_0 - 1) + \cdots + 2 + 1$. The algorithm computes the sum correctly.

Note that if n$\leqslant$0 the loop is not executed and the program returns 0.