Design and Analysis of Algorithms

Prepared by Dr. Muhammad Imran
Modified by MN Khokhar

Loop invariants

- A loop invariant is a logical predicate such that: if it is satisfied before entering any single iteration of the loop then it is also satisfied after the iteration

Example: Loop invariant for Sum of n numbers

```
Algorithm Sum_of_N_numbers

Input: a, an array of N numbers
Output: s, the sum of the N numbers in a

s:=0;
k:=0;
While (k<N) do
  k:=k+1;
  s:=s+a[k];
end
```

Loop invariant = induction hypothesis: At step k, S holds the sum of the first k numbers

Using loop invariants in proofs

- **We must show the following 3 things about a loop invariant:**

1. **Initialization:** It is true prior to the first iteration of the loop.
2. **Maintenance:** If it is true before an iteration of the loop, it remains true before the next iteration.
3. **Termination:** When the loop terminates, the invariant gives us a useful property that helps show that the algorithm is correct.

Example: Proving the correctness of the Sum algorithm (1)

- Induction hypothesis: S= sum of the first k numbers

1. *Initialization: The hypothesis is true at the beginning of the loop:*

Before the first iteration: k=0, S=0. The first 0 numbers have sum zero (there are no numbers) => hypothesis true before entering the loop

Example: Proving the correctness of the Sum algorithm (2)

- Induction hypothesis: S= sum of the first k numbers

2. *Maintenance: If hypothesis is true before step k, then it will be true before step k+1 (immediately after step k is finished)*

We assume that it is true at beginning of step k: “S is the sum of the first k numbers”

We have to prove that after executing step k, at the beginning of step k+1: “S is the sum of the first k+1 numbers”

We calculate the value of S at the end of this step

K:=k+1, s:=s+a[k+1] => s is the sum of the first k+1 numbers

Example: Proving the correctness of the Sum algorithm (3)

- Induction hypothesis: S= sum of the first k numbers

3. *Termination: When the loop terminates, the hypothesis implies the correctness of the algorithm*

The loop terminates when k=n=> s= sum of first k=n numbers => postcondition of algorithm, DONE

Loop invariants and induction

- **Proving loop invariants is similar to mathematical induction**:
  - showing that the invariant holds before the first iteration corresponds to the base case, and
  - showing that the invariant holds from iteration to iteration corresponds to the inductive step.