CCM0128

Invariants and Proofs of Correctness

Source: CS 170, UC Berkeley

March 6, 2025

Overview

- Why do we use induction?
- Reasoning about algorithms with loops
- Using loop invariants to prove correctness
- Practice problems

Why do we use induction?

Mathematical Induction

Property 1. $P(n)$: *sum of first n natural numbers is* $\frac{1}{2}n(n+1)$.

We want to prove $P(n)$ for all natural numbers n.

- ▷ **Strategy:** prove $P(0)$, $P(1)$, $P(2)$, ...?
- ▷ **Better Strategy:** use induction.
  - ○ Base case: prove $P(0)$
  - ○ Induction step: for all $n > 0$, show that $P(n-1) \Rightarrow P(n)$
  - ○ Alternative induction step: for all $n > 0$, show that

$$P(0), P(1), \ldots, P(n-1) \Rightarrow P(n)$$

Reasoning about algorithms with loops

Loop Example

```
int x = c;  // c >= 0
int y = 0;
while (x > 0) {
  x--;
  y++;
}
```

Property 2. *Value of* y *equals* c *after the loop terminates.*

- ▷ Prove *loop invariant* $x + y = c$ by induction on the no. of iterations.
  - ◦ Base case: prove induction hypothesis holds on loop entry
  - ◦ Induction step: invariant holds after $k - 1$ iterations $\Rightarrow$ it holds after k iterations
- ▷ Deduce $y = c$ when loop terminates.

Practice Problems

Problem 1

Consider the following piece of code:

```
int y = 0;
for (int i = 0; i <= n; i++)
  y += Math.pow(2, i);
return y;
```

What is the value of `y` after the loop termination?

Find a loop invariant.

Remark

The for loop

```
for (int i = 0; i <= n; i++)
  // invariant: I(i) is true
  ... loop body ...
```

is equivalent to

```
int i = 0;
loopstart:
  // invariant: I(i) is true
  if (i > n) goto end;
  ... loop body ...
  i = i + 1;
  goto loopstart;
end:
```

Step 1: Run a few iterations and find loop invariant candidate

```
int y = 0;
for (int i = 0; i <= n; i++)
  y += Math.pow(2, i);
```

At the start of each iteration:

- $i = 0$: $y_0 = 0 = 2^0 - 1$
- $i = 1$: $y_1 = 1 = 2^1 - 1$
- $i = 2$: $y_2 = 1 + 2 = 3 = 2^2 - 1$
- $i = 3$: $y_3 = 1 + 2 + 4 = 7 = 2^3 - 1$

Loop invariant candidate:

- $I(i)$: $y_i = 2^i - 1$

Step 2: Prove loop invariant is correct by induction

- **Base case:** $i = 0$: $y_0 = 2^0 - 1 = 0$
- **Induction step:** Assume that at the start of the ith iteration $y_i = 2^i - 1$. Then, at the start of the $(i+1)$st iteration, we have: $y_{i+1} = y_i + 2^i = 2^i - 1 + 2^i = 2 \times 2^i - 1 = 2^{i+1} - 1$. Q.E.D.

Step 3: Loop invariant at the last iteration

- When the loop terminates $i = n + 1$. Thus after the loop execution we have:

$$y = 2^{n+1} - 1$$

Problem 2: Binary Search

```
public static int binarySearch(int x, int[] a) {

    int lo = 0;
    int hi = a.length;

    while (lo < hi) {
        int mid = lo + (hi - lo) / 2;
        if (a[mid] == x)
            return mid;
        else if (a[mid] < x)
            lo = mid + 1;
        else
            hi = mid;
    }

    return -1; // not found
}
```

Loop invariant? (Try: *if* $\mathtt{a[i]} = \mathtt{x}$ *for some* $\mathtt{i}$*, then* $\mathtt{lo} \leq \mathtt{i} < \mathtt{hi}$.)

Referências sobre invariantes

Consulte o material de P. Feofiloff:

- *Observações sobre invariantes*
- *Documentação*, que contém a seção *Invariantes de iterações*