

THE CALCULUS OF COMPUTATION:  
Decision Procedures with  
Applications to Verification

by  
Aaron Bradley  
Zohar Manna

Springer 2007

## 6. Program Correctness: Strategies

# Developing Inductive Assertions

Some structured techniques for developing inductive annotations for proving partial correctness. Just heuristics.

## Basic Facts

### Example: LinearSearch

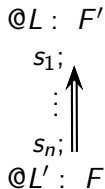
```
for
  @L :  $l \leq i \leq u + 1$ 
  (int  $i := l; i \leq u; i := i + 1$ ) {
  if ( $a[i] = e$ ) return true;
}
```

## Example: BubbleSort

```
for
  @L1 :  $-1 \leq i < |a|$ 
  (int  $i := |a| - 1$ ;  $i > 0$ ;  $i := i - 1$ ) {
  for
    @L2 :  $0 < i < |a| \wedge 0 \leq j \leq i$ 
    (int  $j := 0$ ;  $j < i$ ;  $j := j + 1$ ) {
    if ( $a[j] > a[j + 1]$ ) {
      int  $t := a[j]$ ;
       $a[j] := a[j + 1]$ ;
       $a[j + 1] := t$ ;
    }
  }
}
```

## The Precondition Method

- Given annotation  $@L : F$
- Compute the precondition of  $F$  backward
- Find new annotation  $@L' : F'$



### Example: BinarySearch

@pre  $H?$

@post  $\top$

```
bool BinarySearch(int[] a, int l, int u, int e) {  
    if (l > u) return false;  
    else {  
        @ 2 ≠ 0;           ... basic fact  
        int m := (l + u) div 2;  
        @ 0 ≤ m < |a|;     ... basic fact  
        if (a[m] = e) return true;  
        else if (a[m] < e) return BinarySearch(a, m + 1, u, e);  
        else return BinarySearch(a, l, m - 1, e);  
    }  
}
```

---

(.)

---

@pre  $H : ?$

$S_1 : \text{assume } l \leq u;$

$S_2 : m := (l + u) \text{ div } 2;$

@  $F : 0 \leq m < |a|$

---

Compute

$\text{wp}(F, S_1; S_2)$

$\Leftrightarrow \text{wp}(\text{wp}(F, m := (l + u) \text{ div } 2), S_1)$

$\Leftrightarrow \text{wp}(F\{m \mapsto (l + u) \text{ div } 2\}, S_1)$

$\Leftrightarrow \text{wp}(F\{m \mapsto (l + u) \text{ div } 2\}, \text{assume } l \leq u)$

$\Leftrightarrow l \leq u \rightarrow F\{m \mapsto (l + u) \text{ div } 2\}$

$\Leftrightarrow l \leq u \rightarrow 0 \leq (l + u) \text{ div } 2 < |a|$

$\Leftarrow 0 \leq l \wedge u < |a|$

$$\textcircled{\text{pre } H : 0 \leq \ell \wedge u < |a|}$$

guaranteed

$$0 \leq \ell \wedge u < |a| \rightarrow \text{wp}(F, S_1; S_2)$$

is  $T_{\mathbb{Z}}$ -valid. The runtime assertion

$$0 \leq m < |a|$$

holds in every execution of BinarySearch in which the precondition

$$\textcircled{\text{pre } 0 \leq \ell \wedge u < |a|}$$

is satisfied.