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

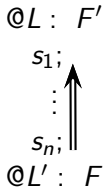
```
for
  @L : ℓ ≤ i ≤ u + 1
  (int i := ℓ; i ≤ u; i := i + 1) {
  if (a[i] = e) return true;
}
```


Example: BubbleSort

```
for
  @L1 : −1 ≤ i < |a|
  (int i := |a| − 1; i > 0; i := i − 1) {
  for
    @L2 : 0 < i < |a| ∧ 0 ≤ j ≤ 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` $(\ell > u)$ `return false`;
  `else` {
    @ $2 \neq 0$; ... basic fact
    `int` $m := (\ell + u)$ `div` 2;
    @ $0 \leq m < |a|$; ... basic fact
    `if` $(a[m] = e)$ `return true`;
    `else if` $(a[m] < e)$ `return` BinarySearch$(a, m+1, u, e)$;
    `else return` BinarySearch$(a, \ell, m-1, e)$;
  }
}

(·)

@pre H : ?
S_1 : `assume` $\ell \leq u$;
S_2 : $m := (\ell + u)$ `div` 2;
@ F : $0 \leq m < |a|$

Compute

$$\begin{aligned}
&\mathrm{wp}(F,\ S_1; S_2)\\
&\Leftrightarrow\ \mathrm{wp}(\mathrm{wp}(F,\ m := (\ell+u)\ \texttt{div}\ 2),\ S_1)\\
&\Leftrightarrow\ \mathrm{wp}(F\{m \mapsto (\ell+u)\ \texttt{div}\ 2\},\ S_1)\\
&\Leftrightarrow\ \mathrm{wp}(F\{m \mapsto (\ell+u)\ \texttt{div}\ 2\},\ \texttt{assume}\ \ell \leq u)\\
&\Leftrightarrow\ \ell \leq u \ \rightarrow\ F\{m \mapsto (\ell+u)\ \texttt{div}\ 2\}\\
&\Leftrightarrow\ \ell \leq u \ \rightarrow\ 0 \leq (\ell+u)\ \texttt{div}\ 2 < |a|\\
&\Leftarrow\ 0 \leq \ell \ \wedge\ u < |a|
\end{aligned}$$

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

guaranteed

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

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

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

holds in every execution of BinarySearch in which the precondition

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

is satisfied.