

EECS3311 Software Design  
Fall 2019  
Exercise: Proving Correctness of Loops

Chen-Wei Wang

Consider the following query which involves the use of a loop to find the maximum value from an integer array:

```
1 find_max (a: ARRAY [INTEGER]): INTEGER
2   require
3     not_empty: a.count > 0
4   local
5     i: INTEGER
6   do
7     from
8       i := a.lower
9       Result := a [i]
10    invariant
11      -- Predicate Equivalent:  $\forall j \mid a.lower \leq j < i \bullet \mathbf{Result} \geq a[j]$ 
12    across
13      a.lower |..| (i - 1) as j
14    all
15      Result >= a [j.item]
16    end
17  until
18    i > a.upper
19  loop
20    if a [i] > Result then
21      Result := a [i]
22    end
23    i := i + 1
24  variant
25    a.upper - i + 1
26  end
27 ensure
28   -- Predicate Equivalent:  $\forall j \mid a.lower \leq j \leq a.upper \bullet \mathbf{Result} \geq a[j]$ 
29 across
30   a.lower |..| a.upper as j
31 all
32   Result >= a [j.item]
33 end
34 end
```

## Your Tasks

Prove or disprove that the above program is *totally* correct, which involves the following steps:

1. State formally, in terms of Hoare Triples, the obligations for proving that:
  - The loop is *partially* correct (without considering termination); and
  - The loop terminates.
2. For each of the Hoare triple  $\{ Q \} S \{ R \}$  in Step 1, calculate the corresponding weakest precondition (i.e.,  $wp(S, R)$ ).
3. Prove or disprove that the calculated  $wp$  is equal to or weaker than the corresponding precondition (i.e., prove or disprove that  $Q \Rightarrow wp(S, R)$ ).

# 1 Partial Correctness

## 1.1 Establishing the Loop Invariant

**Proof Obligation:**

$$\begin{aligned} & \{ a.count > 0 \} \\ & \quad i := a.lower; \text{Result} := a[i] \\ & \{ \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] \} \end{aligned}$$

Notice that the augmented constraint  $\boxed{a.lower \leq j \wedge j \leq a.upper}$  is due to the array indexing expression  $a[j]$ . Similar augmentation is performed for each occurrence of an array indexing expression.

## 1.2 Maintaining the Loop Invariant

**Proof Obligation:**

$$\begin{aligned} & \{ \neg(i > a.upper) \wedge ( \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] ) \} \\ & \quad \text{if } a[i] > \text{Result} \text{ then } \text{Result} := a[i] \text{ end; } i := i + 1 \\ & \{ \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] \} \end{aligned}$$

## 1.3 Establishing the Postcondition

**Proof Obligation:**

$$\begin{aligned} & (i > a.upper) \wedge ( \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] ) \\ & \Rightarrow ( \forall j | a.lower \leq j \leq a.upper \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] ) \end{aligned}$$

# 2 Termination

## 2.1 Loop Variant Stays Positive

**Proof Obligation:**

$$\begin{aligned} & \{ \neg(i > a.upper) \wedge ( \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] ) \} \\ & \quad \text{if } a[i] > \text{Result} \text{ then } \text{Result} := a[i] \text{ end; } i := i + 1 \\ & \{ a.upper - i + 1 \geq 0 \} \end{aligned}$$

## 2.2 Loop Variant Decreases

**Proof Obligation:**

$$\begin{aligned} & \{ \neg(i > a.upper) \wedge ( \forall j | a.lower \leq j \leq i - 1 \bullet \boxed{a.lower \leq j \wedge j \leq a.upper} \wedge \mathbf{Result} \geq a[j] ) \} \\ & \quad \text{if } a[i] > \text{Result} \text{ then } \text{Result} := a[i] \text{ end; } i := i + 1 \\ & \{ a.upper - i + 1 < a.upper_0 - i_0 + 1 \} \end{aligned}$$

### Solution to Proving (1.2)

We first calculate the  $wp$  for the loop body to maintain the LI:

$$\begin{aligned}
& wp(\text{if } a[i] > \text{Result then Result} := a[i] \text{ end}; i := i + 1, \boxed{\forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]}) \\
&= \{wp \text{ rule for seq. comp.}\} \\
&= wp(\text{if } a[i] > \text{Result then Result} := a[i] \text{ end}, \boxed{wp(i := i + 1, \forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j])}) \\
&= \{wp \text{ rule for assignment}\} \\
&= wp(\text{if } a[i] > \text{Result then Result} := a[i] \text{ end}, \boxed{\forall j \mid a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]}) \\
&= \{wp \text{ rule for conditional}\} \\
&= a[i] > \text{Result} \implies wp(\text{Result} := a[i], \boxed{\forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]}) \\
&\quad \wedge \\
&= a[i] \leq \text{Result} \implies wp(\text{Result} := \text{Result}, \boxed{\forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]}) \\
&= \{wp \text{ rule for assignment, twice}\} \\
&= a[i] > \text{Result} \implies \forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge a[i] \geq a[j] \\
&\quad \wedge \\
&= a[i] \leq \text{Result} \implies \forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]
\end{aligned}$$

We then prove that the precondition (i.e.,  $\neg$ (exit condition) and LI) is no weaker than the above calculated  $wp$ :

- To prove:
$$\begin{aligned}
& \neg(i > a.upper) \wedge (\forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]) \\
& \implies a[i] > \text{Result} \implies \boxed{\forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge a[i] \geq a[j]}
\end{aligned}$$

**Proof:**

$$\begin{aligned}
& \boxed{\forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge a[i] \geq a[j]} \\
& \equiv \{\text{split range: } \forall j \mid a.lower \leq j \leq i \bullet P(j) \equiv (\forall j \mid a.lower \leq j \leq i - 1 \bullet P(j)) \wedge P(i)\} \\
& \quad (\forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge a[i] \geq a[j]) \wedge (a.lower \leq i \wedge i \leq a.upper \wedge a[i] \geq a[i]) \\
& \equiv \{\text{antecedent: } a[i] > \text{Result}; \text{ and RHS of precondition: } \forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]\} \\
& \quad \text{true} \wedge (a.lower \leq i \wedge i \leq a.upper \wedge a[i] \geq a[i]) \\
& \equiv \{\text{LHS of precondition: } \neg(i > a.upper) \text{ and } a[i] \geq a[i] \equiv \text{true}\} \\
& \quad \text{true}
\end{aligned}$$

- To prove:
$$\begin{aligned}
& \neg(i > a.upper) \wedge (\forall j \mid a.lower \leq j \leq i - 1 \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]) \\
& \implies a[i] \leq \text{Result} \implies \forall j \mid a.lower \leq j \leq i \bullet a.lower \leq j \wedge j \leq a.upper \wedge \text{Result} \geq a[j]
\end{aligned}$$

(Exercise)