CSCI 740 - Programming Language Theory

Lecture 24

Verifying Programs with Arrays

Instructor: Hossein Hojjat

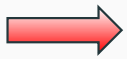
October 30, 2017

Weakest Precondition Rules: Summary

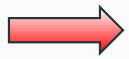
c	$\mathsf{wp}(c, Q)$
$x := e$	$Q[x \mapsto e]$
$\mathsf{assume}(b)$	$b \to Q$
$\mathsf{assert}(b)$	$\mathsf{wp}(b \wedge Q)$
$\mathsf{havoc}(x)$	$\forall y. Q[x \mapsto y]$
$c_1; c_2$	$\mathsf{wp}(c_1, \mathsf{wp}(c_2, Q))$
$\texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2$	$b \to \mathsf{wp}(c_1, Q) \wedge \neg b \to \mathsf{wp}(c_2, Q)$
$\texttt{while}\ b\ \texttt{do}\ c$	$I \wedge \forall \vec{y}. \Big((I \wedge b \to \mathsf{wp}(c, I)) \wedge (I \wedge \neg b \to Q) \Big) [\vec{x} \mapsto \vec{y}]$ ($\vec{x}$ are variables modified in c and I is the loop invariant)

Problem with Arrays

```
a[k]=1;
a[j]=2;
x=a[k]+a[j];
{x=3}
```



```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

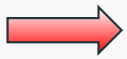


Problem with Arrays

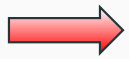
- Now what? Can we use the standard rule for assignment?

$$\mathsf{wp}(x := e, C) = C[x \mapsto e]$$

```
a[k]=1;
a[j]=2;
x=a[k]+a[j];
{x=3}
```



```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

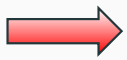


Problem with Arrays

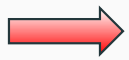
- Now what? Can we use the standard rule for assignment?

$$\mathsf{wp}(x := e, C) = C[x \mapsto e]$$

```
a[k]=1;
a[j]=2;
x=a[k]+a[j];
{x=3}
```



```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```



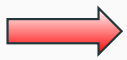
```
{1+2=3} = {true}
a[k]=1;
{a[k]+2=3}
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

Problem with Arrays

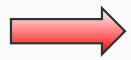
- Now what? Can we use the standard rule for assignment?

$$\mathsf{wp}(x := e, C) = C[x \mapsto e]$$

```
a[k]=1;
a[j]=2;
x=a[k]+a[j];
{x=3}
```



```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```



```
{1+2=3} = {true}
a[k]=1;
{a[k]+2=3}
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

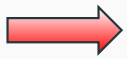
What if $k = j$?

Problem with Arrays

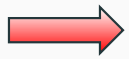
- Now what? Can we use the standard rule for assignment?

$$\mathsf{wp}(x := e, C) = C[x \mapsto e]$$

```
a[k]=1;
a[j]=2;
x=a[k]+a[j];
{x=3}
```



```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```



```
{1+2=3} = {true}
a[k]=1;
{a[k]+2=3}
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

What if $k = j$?

Problem with Arrays

- Naïve array assignment axiom does not work

$$\{Q[A[e_1] \mapsto e_2]\} \quad A[e_1] := e_2 \quad \{Q\}$$

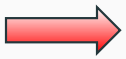
- Changes to $A[i]$ may also change $A[j]$, $A[k]$, ...
  - (since i might equal j, k, ...)
- **Solution:** enrich the assertion language with expressions $A\{e_1 \mapsto e_2\}$
- Meaning: the array equal to A except that index e_1 maps to value e_2

$$A\{e_1 \mapsto e_2\}[i] = \begin{cases} A[i] & \text{if} \quad i \neq e_1 \\ e_2 & \text{if} \quad i = e_1 \end{cases}$$

Assignment Rule with Theory of Arrays

$$\frac{}{\vdash \{Q[A \mapsto A\{i \mapsto e\}]\}\ A[i] := e\ \{Q\}}$$

```
a[k]=1;
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```



```
{k≠j}
{a{k↦1}{j↦2}[k]+a{k↦1}{j↦2}[j]=3}
a[k]=1;
{a{j↦2}[k]+a{j↦2}[j]=3}
a[j]=2;
{a[k]+a[j]=3}
x=a[k]+a[j];
{x=3}
```

Exercise

Prove the array sum is correct

```
{n≥0}
j = 0;
s = 0;
while (j<n) do{
  s = s + a[j];
  j = j + 1;
}
{ s = ∑_{0≤i<n} a[i] }
```

$$\frac{A \Rightarrow I \qquad \vdash \{b \wedge I\}\ c\ \{I\} \qquad I \wedge \neg b \Rightarrow B}{\vdash\ \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}$$

Exercise

Prove the array sum is correct

```
{n≥0}
j = 0;
s = 0;
while (j<n) do{
   s = s + a[j];
   j = j + 1;
}
{ s = ∑_{0≤i<n} a[i] }
```

$$\frac{A \Rightarrow I \qquad \vdash \{b \wedge I\}\ c\ \{I\} \qquad I \wedge \neg b \Rightarrow B}{\vdash\ \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}$$

Choose invariant $(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n$

Step 1. Prove invariant is maintained throughout the loop

$$\{j < n \wedge (s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\}$$

```
s = s + a[j]; j = j + 1
```

$$\{(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\}$$

Exercise

Prove the array sum is correct

```
{n≥0}
j = 0;
s = 0;
while (j<n) do{
   s = s + a[j];
   j = j + 1;
}
{ s = ∑_{0≤i<n} a[i] }
```

$$\frac{A \Rightarrow I \qquad \vdash \{b \wedge I\}\ c\ \{I\} \qquad I \wedge \neg b \Rightarrow B}{\vdash\ \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}$$

Choose invariant $(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n$

Step 2. Prove invariant is initially *true*

$$\{n \ge 0\}$$

$$\texttt{j = 0; s = 0}$$

$$\{(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\}$$

Exercise

Prove the array sum is correct

```
{n≥0}
j = 0;
s = 0;
while (j<n) do{
  s = s + a[j];
  j = j + 1;
}
{ s = ∑_{0≤i<n} a[i] }
```

$$\frac{A \Rightarrow I \qquad \vdash \{b \wedge I\}\ c\ \{I\} \qquad I \wedge \neg b \Rightarrow B}{\vdash\ \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}$$

Choose invariant $(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n$

Step 3. Prove invariant and exit condition implies postcondition

$$\left((s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n \wedge j \ge n\right) \Rightarrow s = \sum_{0 \le i < n} a[i]$$

Proof Obligations

Step 1. Prove invariant is maintained throughout the loop

$$\{(s + a[j] = \sum_{0 \le i < j+1} a[i]) \wedge 0 \le j + 1 \le n\} \quad \text{(by assignment rule)}$$

```
s = s + a[j]
```

$$\{(s = \sum_{0 \le i < j+1} a[i]) \wedge 0 \le j + 1 \le n\} \quad \text{(by assignment rule)}$$

```
j = j + 1
```

$$\{(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\}$$

Need to show:

$$(0 \le j \le n \wedge (s = \textstyle\sum_{0 \le i < j} a[i]) \wedge j < n) \Rightarrow$$
$$(0 \le j + 1 \le n \wedge (s + a[j] = \textstyle\sum_{0 \le i < j+1} a[i]))$$

Proof Obligations

Step 2. Prove invariant is initially *true*

$$\{(0 = \sum_{0 \le i < 0} a[i]) \wedge 0 \le 0 \le n\} \qquad \text{(by assignment rule)}$$

```
j = 0
```

$$\{(0 = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\} \qquad \text{(by assignment rule)}$$

```
s = 0
```

$$\{(s = \sum_{0 \le i < j} a[i]) \wedge 0 \le j \le n\}$$

Need to show:

$$(n \ge 0) \Rightarrow (0 = \textstyle\sum_{0 \le i < 0} a[i]) \wedge 0 \le 0 \le n$$

Proof Obligations

Step 3. Prove invariant and exit condition implies postcondition

$$((s = \sum_{0 \leq i < j} a[i]) \wedge 0 \leq j \leq n \wedge j \geq n) \Rightarrow (s = \sum_{0 \leq i < n} a[i])$$

Exercise

Consider the following program:

```
{0≤i<n}
j = i+1 ;
while (j<n) {
     a[i] = max(a[i],a[j]);
     j = j+1;
}
```
{ $\forall_{i \leq k < n}\ a_0[k] \leq a[i]$ }

Is the following a loop invariant?

$$\{\forall_{i \leq k < j}\ a_0[k] \leq a[i] \wedge 0 \leq j \leq n\}$$

(a_0 is the initial array)

Invariant Proof

Prove invariant is maintained throughout the loop

$\{\forall_{i \le k < j+1}\ a_0[k] \le \mathsf{max}(a[i], a[j]) \land 0 \le j+1 \le n\}$

$\{\forall_{i \le k < j+1}\ a_0[k] \le a\{i \mapsto \mathsf{max}(a[i], a[j])\}[i] \land 0 \le j+1 \le n\}$ (by array assignment)

```
a[i] = max(a[i],a[j])
```

$\{\forall_{i \le k < j+1}\ a_0[k] \le a[i] \land 0 \le j+1 \le n\}$ (by assignment)

```
j = j + 1
```

$\{\forall_{i \le k < j}\ a_0[k] \le a[i] \land 0 \le j \le n\}$

Need to show:

$$(\forall_{i \le k < j}\ a_0[k] \le a[i] \land j < n) \Rightarrow$$
$$(\forall_{i \le k < j+1}\ a_0[k] \le \mathsf{max}(a[i], a[j]) \land 0 \le j+1 \le n)$$

We don't know that $a_0[j] \le \mathsf{max}(a[i], a[j])$!

Conjoin a new constraint $(\forall_{i < k < n}\ a[k] = a_0[k]) \land i < j$