Testing, Debugging, and Verification

Formal Verification, Part II

Srinivas Pinisetty[1]

4 December 2017

[1] Lecture slides based on material from Wolfgang Aherndt,..

Recap

```
method MyMethod(. . .)
  requires Q
  ensures R
  {
   S: program statements
  }
```

Hoare Triple: {Q} S {R}

If execution of program S starts in a state satisfying pre-condition Q, the is is guaranteed to terminate in a state satisfying the post-condition R.

Recap

```
method MyMethod(. . .)
  requires Q
  ensures R
  {
   S: program statements
  }
```

Proving {Q} S {R}:

- Extract Weakest Precondition $wp(S, R)$:
  - Logical formula specifying set of initial states s.t.
  - if program S terminates,
  - end up in state satisfying postcondition R.
- Extract $wp(S, R)$ by reasoning backwards.
- Show that precondition Q implies wp:
  $Q \rightarrow wp(S, R)$

A small imperative language

Assignment: `x := e`

Sequential: `S1; S2`

Assertions: `assert B`

If-statements: `if B then S1 else S2`

While-loops: `while B S` $\Longleftarrow$ todays topic

Semantics

The weakest precondition calculus provide a semantic for each language construct.

Recap: The Weakest Precondition Calculus

Assignment: $wp(x := e, R) = R[x \mapsto e]$

Recap: The Weakest Precondition Calculus

Assignment: $wp(x := e, R) = R[x \mapsto e]$

Sequential: $wp(S1; S2, R) = wp(S1, wp(S2, R))$

Recap: The Weakest Precondition Calculus

Assignment: $wp(x := e, R) = R[x \mapsto e]$

Sequential: $wp(S1; S2, R) = wp(S1, wp(S2, R))$

Assertions: $wp(\mathit{assert}\ B,\ R) = B \wedge R$

Recap: The Weakest Precondition Calculus

Assignment: $wp(x := e, R) = R[x \mapsto e]$

Sequential: $wp(S1; S2, R) = wp(S1, wp(S2, R))$

Assertions: $wp(\textit{assert } B,\ R) = B \wedge R$

Conditional: $wp(\textit{if } B \textit{ then } S1 \textit{ else } S2,\ R) =$
$(B \rightarrow wp(S1, R)) \wedge (\neg B \rightarrow wp(S2, R))$

Recap: The Weakest Precondition Calculus

Assignment: $wp(x := e, R) = R[x \mapsto e]$

Sequential: $wp(S1; S2, R) = wp(S1, wp(S2, R))$

Assertions: $wp(\mathit{assert}\ B,\ R) = B \wedge R$

Conditional: $wp(\mathit{if}\ B\ \mathit{then}\ S1\ \mathit{else}\ S2,\ R) =$
$(B \rightarrow wp(S1, R)) \wedge (\neg B \rightarrow wp(S2, R))$

Conditional 2: (empty else branch)
$wp(\mathit{if}\ B\ \mathit{then}\ S1,\ R) = (B \rightarrow wp(S1, R)) \wedge (\neg B \rightarrow R)$

Recall: Reasoning Backwards

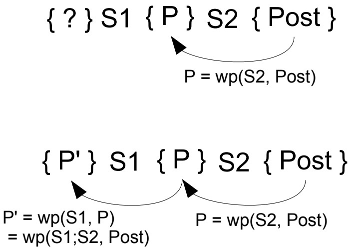
- Note: Verification Proofs by Backwards Reasoning.
- Start from post-condition.
- "Execute" program backwards by computing weakest pre-conditions.
- The *wp* of a statement become the "post-condition" for the previous statement.

Recall: Reasoning Backwards

- Note: Verification Proofs by Backwards Reasoning.
- Start from post-condition.
- "Execute" program backwards by computing weakest pre-conditions.
- The *wp* of a statement become the "post-condition" for the previous statement.

Example: Weakest Precondition of a sequential:

{ ? } S1; S2 { Post }



Let's Prove ManyReturns Correct!

Recall

To prove correct a program S with precondition Q and postcondition R we need to show that $Q \to wp(S, R)$.

```
method ManyReturns(x:int, y:int) returns (more:int, less:
 int)
 requires 0 < y;
 ensures less < x < more;
 {
   more := x+y;
   less := x-y;
 }
```

Show that

$0 < y \to wp(more := x + y; less := x - y,\ less < x < more)$

Let's Prove ManyReturns Correct!

Show that $Pre \rightarrow wp(S, Post)$.
First compute wp:

$$wp(more := x + y; less := x - y,\ less < x < more)$$

Let's Prove ManyReturns Correct!

Show that *Pre* $\rightarrow$ *wp(S, Post)*.
First compute wp:

$$wp(more := x + y; less := x - y,\ less < x < more)$$

Seq. rule

$$wp(more := x + y, wp(less := x - y,\ less < x < more))$$

Let's Prove ManyReturns Correct!

Show that $Pre \rightarrow wp(S, Post)$.

First compute wp:

$wp(more := x + y; less := x - y,\ less < x < more)$

Seq. rule

$wp(more := x + y, wp(less := x - y,\ less < x < more))$

Assignment rule

$wp(more := x + y, x - y < x < more)$

Let's Prove ManyReturns Correct!

Show that $Pre \rightarrow wp(S, Post)$.

First compute wp:

$wp(more := x + y; less := x - y,\ less < x < more)$

Seq. rule

$wp(more := x + y, wp(less := x - y,\ less < x < more))$

Assignment rule

$wp(more := x + y, x - y < x < more)$

Assignment rule

$x - y < x < x + y$

Let's Prove ManyReturns Correct!

Show that *Pre* $\rightarrow$ *wp(S, Post)*.
First compute wp:

$wp(more := x + y; less := x - y,\ less < x < more)$
Seq. rule
$wp(more := x + y, wp(less := x - y,\ less < x < more))$
Assignment rule
$wp(more := x + y, x - y < x < more)$
Assignment rule
$x - y < x < x + y$

Show that this follows from the precondition $0 < y$:
$0 < y \rightarrow x - y < x < x + y$

Let's Prove ManyReturns Correct!

Show that $Pre \to wp(S, Post)$.
First compute wp:

$wp(more := x + y; less := x - y,\ less < x < more)$
Seq. rule
$wp(more := x + y, wp(less := x - y,\ less < x < more))$
Assignment rule
$wp(more := x + y, x - y < x < more)$
Assignment rule
$x - y < x < x + y$

Show that this follows from the precondition $0 < y$:
$0 < y \to x - y < x < x + y$
which follows from the precondition by simple arithmetic.

Another Example

```
method f ( x : int) returns (y : int)
requires x > 8
ensures  y > 10
        {
                y := x + 1;
                if (y mod 2 == 0) { y := 100; }
                else { y := y + 2; }
        }
```

Exercise: Prove f correct

Show that

$x > 8 \rightarrow wp(y := x + 1; if \cdots, y > 10).$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

If rule

$= ((y\%2 == 0) \rightarrow wp(y := 100, y > 10))$
$\quad \wedge(\neg(y\%2 == 0) \rightarrow wp(y := y + 2, y > 10))$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

If rule

$= ((y\%2 == 0) \rightarrow wp(y := 100, y > 10))$
$\quad \wedge(\neg(y\%2 == 0) \rightarrow wp(y := y + 2, y > 10))$

Assignment rule (2x)

$= ((y\%2 == 0) \rightarrow 100 > 10) \wedge (\neg(y\%2 == 0) \rightarrow y + 2 > 10)$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

If rule

$= ((y\%2 == 0) \rightarrow wp(y := 100, y > 10))$
$\quad \wedge(\neg(y\%2 == 0) \rightarrow wp(y := y + 2, y > 10))$

Assignment rule (2x)

$= ((y\%2 == 0) \rightarrow 100 > 10) \wedge (\neg(y\%2 == 0) \rightarrow y + 2 > 10)$

Simplify

$= ((y\%2 == 0) \rightarrow true) \wedge (\neg(y\%2 == 0) \rightarrow y > 8))$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

If rule

$= ((y\%2 == 0) \to wp(y := 100, y > 10))$
$\quad \wedge(\neg(y\%2 == 0) \to wp(y := y + 2, y > 10))$

Assignment rule (2x)

$= ((y\%2 == 0) \to 100 > 10) \wedge (\neg(y\%2 == 0) \to y + 2 > 10)$

Simplify

$= ((y\%2 == 0) \to \mathit{true}) \wedge (\neg(y\%2 == 0) \to y > 8))$

By $a \to \mathit{true} = \mathit{true}$

$= \mathit{true} \wedge (\neg(y\%2 == 0) \to y > 8)$

Solution

First compute wp:

$wp(y := x + 1; \text{if } y\%2 == 0 \cdots, y > 10)$

Seq. rule

$= wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

Compute $wp(\text{if } y\%2 == 0 \cdots, y > 10)$

If rule

$= ((y\%2 == 0) \to wp(y := 100, y > 10))$
$\quad \wedge(\neg(y\%2 == 0) \to wp(y := y + 2, y > 10))$

Assignment rule (2x)

$= ((y\%2 == 0) \to 100 > 10) \wedge (\neg(y\%2 == 0) \to y + 2 > 10)$

Simplify

$= ((y\%2 == 0) \to \mathit{true}) \wedge (\neg(y\%2 == 0) \to y > 8))$

By $a \to \mathit{true} = \mathit{true}$

$= \mathit{true} \wedge (\neg(y\%2 == 0) \to y > 8)$

By $\mathit{true} \wedge a = a$

$= (\neg(y\%2 == 0) \to y > 8)$

Another Example

$wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$
By $wp(\text{if } y\%2 == 0 \cdots, y > 10) = (\neg(y\%2 == 0) \rightarrow y > 8)$
$= wp(y := x + 1; (\neg(y\%2 == 0) \rightarrow y > 8))$

Another Example

$wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

By $wp(\text{if } y\%2 == 0 \cdots, y > 10) = (\neg(y\%2 == 0) \rightarrow y > 8)$

$= wp(y := x + 1; (\neg(y\%2 == 0) \rightarrow y > 8))$

By Assignment Rule

$= (\neg((x + 1)\%2 == 0) \rightarrow x + 1 > 8)$

Another Example

$wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

By $wp(\text{if } y\%2 == 0 \cdots, y > 10) = (\neg(y\%2 == 0) \rightarrow y > 8)$

$= wp(y := x + 1; (\neg(y\%2 == 0) \rightarrow y > 8))$

By Assignment Rule

$= (\neg((x + 1)\%2 == 0) \rightarrow x + 1 > 8)$

Simplify

$= (\neg((x + 1)\%2 == 0) \rightarrow x > 7)$

Another Example

$wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

By $wp(\text{if } y\%2 == 0 \cdots, y > 10) = (\neg(y\%2 == 0) \rightarrow y > 8)$

$= wp(y := x + 1; (\neg(y\%2 == 0) \rightarrow y > 8))$

By Assignment Rule

$= (\neg((x + 1)\%2 == 0) \rightarrow x + 1 > 8)$

Simplify

$= (\neg((x + 1)\%2 == 0) \rightarrow x > 7)$

To prove: $x > 8 \rightarrow wp(y := x + 1; if \cdots, y > 10)$

$x > 8 \rightarrow (\neg((x + 1)\%2 == 0) \rightarrow x > 7)$

Another Example

$wp(y := x + 1;\ wp(\text{if } y\%2 == 0 \cdots, y > 10))$

By $wp(\text{if } y\%2 == 0 \cdots, y > 10) = (\neg(y\%2 == 0) \rightarrow y > 8)$

$= wp(y := x + 1; (\neg(y\%2 == 0) \rightarrow y > 8))$

By Assignment Rule

$= (\neg((x + 1)\%2 == 0) \rightarrow x + 1 > 8)$

Simplify

$= (\neg((x + 1)\%2 == 0) \rightarrow x > 7)$

To prove: $x > 8 \rightarrow wp(y := x + 1; if \cdots, y > 10)$

$x > 8 \rightarrow (\neg((x + 1)\%2 == 0) \rightarrow x > 7)$

Simplify using $x > 8$ in RHS

$= x > 8 \rightarrow (\neg((x + 1)\%2 == 0) \rightarrow true)$

By $a \rightarrow true = true$

$= x > 8 \rightarrow true$

By $a \rightarrow true = true$

$=$ true