Program Verification using Hoare logic

ThanhVu Nguyen
CSCE 467
Adapted from Jonathan Aldrich's Program Analysis slides

November 19, 2019

Big-Step Operational Semantics

E-Assign
$$\frac{E \vdash a \Downarrow n}{E \vdash x := a \Downarrow E\{x \mapsto n\}}$$

E-Skip
$$\frac{}{E \vdash \text{skip} \Downarrow E}$$

E-Seq
$$\frac{E \vdash S1 \Downarrow E' \quad E' \vdash S1 \Downarrow E''}{E \vdash S1; S2 \Downarrow E''}$$

E-IfTrue
$$\frac{E \vdash b \Downarrow \mathsf{True} \quad E \vdash S1 \Downarrow E'}{E \vdash \text{if } b \text{ then } S1 \text{ else } S2 \Downarrow E'}$$

E-IfFalse
$$\frac{E \vdash b \Downarrow \mathsf{False} \quad E \vdash S2 \Downarrow E''}{E \vdash \text{if } b \text{ then } S1 \text{ else } S2 \Downarrow E''}$$

E-While
$$\frac{E \vdash c \Downarrow \text{True} \quad E \vdash S; \text{while } b \text{ do } S \Downarrow \text{E'}}{E \vdash \text{while } b \text{ do } S \Downarrow E'}$$

$$\frac{E \vdash c \Downarrow \text{False}}{E \vdash \text{while } b \text{ do } S \Downarrow E}$$

Axiomatic Semantics

- Big step semantics: relates intial state to final one,
  - e.g., if we start the program with the env/state $\{x \mapsto 3, y \mapsto 4\}$, we get the new env $\{x \mapsto 7, y \mapsto 2\}$.
- Axiomantic Semantics: instead of single state (e.g., $\{x \mapsto 3, y \mapsto 4\}$, work with a *set* of states, described by a formula
  - e.g., if we start the program with variables having values satisfying $x >= 0, y >= 0$, we get a new state that satisfy $x < 100, y = x^2$.

Hoare Tripple

$$\{P\}\ S\ \{Q\}$$

- By Tony Hoare
- Reasoning about partial program correctness using pre- and post-conditions
- Hoare Tripple
  - P: a formula representing the precondition
  - Q: a formula representing the postcondition
  - Read: assume P holds, if S successfully executes, then Q holds
  - P and Q: specifications of the program S
- Partial Correctness: assume S terminates
- Total Correctness: require S terminates

Examples of Hoare Tripples

1. $\{\,\text{True}\,\}$ x:=5 $\{\,x \equiv 5\,\}$
2. $\{\,x \equiv y\,\}$ x := x + 3 $\{\,x\equiv y+3\,\}$
3. $\{\,x>-1\,\}$ x:=2*x + 3 $\{\,x > 1\,\}$
4. $\{\,x \equiv a\,\}$ if x < 0 then x := -x $\{\,x \equiv |a|\,\}$
5. $\{\,\text{False}\,\}$ x:=3 $\{\,x \equiv 8\,\}$

Examples of Hoare Tripples

1. { True } x:=5 { x $\equiv$ 5 }
2. { x $\equiv$ y } x := x + 3 { x$\equiv$ y+3 }
3. { x>-1 } x:=2*x + 3 { x > 1 }
4. { x $\equiv$ a } if x < 0 then x := -x { x $\equiv$ $|a|$ }
5. { False } x:=3 { x $\equiv$ 8 }

In-class Questions:

- { x$\equiv$ y } ??? { x$\equiv$ y }
- { ??? } x:= y - 3 { x $\equiv$ 8 }
- { x<0 } while(x!=0) x:=x - 1 { ??? }

Examples of Hoare Tripples

1. $\{$ True $\}$ x:=5 $\{$ x $\equiv$ 5 $\}$
2. $\{$ x $\equiv$ y $\}$ x := x + 3 $\{$ x$\equiv$ y+3 $\}$
3. $\{$ x>-1 $\}$ x:=2*x + 3 $\{$ x > 1 $\}$
4. $\{$ x $\equiv$ a $\}$ if x < 0 then x := -x $\{$ x $\equiv$ $|a|$ $\}$
5. $\{$ False $\}$ x:=3 $\{$ x $\equiv$ 8 $\}$

In-class Questions:

- $\{$ x$\equiv$ y $\}$??? $\{$ x$\equiv$ y $\}$
- $\{$??? $\}$ x:= y - 3 $\{$ x $\equiv$ 8 $\}$
- $\{$ x<0 $\}$ while(x!=0) x:=x - 1 $\{$??? $\}$
    - Not valid for Total Correctess

Strongest Postconditions

Which are valid?

- $\{ x \equiv 5 \}$ x:=x*2 $\{ \text{true} \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x > 0 \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x \equiv 10 \lor x \equiv 5 \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x \equiv 10 \}$

Strongest Postconditions

Which are valid?

- $\{ x \equiv 5 \}$ x:=x*2 $\{ \text{true} \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x > 0 \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x \equiv 10 \lor x \equiv 5 \}$
- $\{ x \equiv 5 \}$ x:=x*2 $\{ x \equiv 10 \}$
- All are valid, but which one is the most useful?

Strongest Postconditions

Which are valid?

- { x ≡ 5 } x:=x*2 { true }
- { x ≡ 5 } x:=x*2 { x >0 }
- { x ≡ 5 } x:=x*2 { x ≡ 10 ∨ x≡ 5 }
- { x ≡ 5 } x:=x*2 { x ≡ 10 }
- All are valid, but which one is the most useful?
  - x ≡ 10 is the *strongest* postcondition
  - In general, we want strong postconditions

Definition

- In { P } S { Q } , Q is the strongest postcondition if $\forall Q'.${ P } S { Q' } , $Q \Rightarrow Q'$
- Ex: x ≡ 10 is the *strongest* postcondition
  - $x \equiv 10 \Rightarrow true$
  - $x \equiv 10 \Rightarrow x > 0$
  - $x \equiv 10 \Rightarrow (x \equiv 10 \lor x \equiv 5)$
  - $x \equiv 10 \Rightarrow x \equiv 10$

Weakest Preconditions

- $\{ x \equiv 5 \land y \equiv 10 \}$ z:=x/y { z<1 }
- $\{ x < y \land y > 0 \}$ z:=x/y { z<1 }
- $\{ y \neq 0 \land x/y < 1 \}$ z:=x/y { z<1 }
- All are true, but which one is the most useful?

Weakest Preconditions

- $\{ x \equiv 5 \wedge y \equiv 10 \}$ z:=x/y { z<1 }
- $\{ x < y \wedge y > 0 \}$ z:=x/y { z<1 }
- $\{ y \neq 0 \wedge x/y < 1 \}$ z:=x/y { z<1 }
- All are true, but which one is the most useful?
  - $y \neq 0 \wedge x/y < 1$ is the *weakest* precondition
  - In general, we want weak preconditions (allowing us to run the program with fewer assumptions or restrictions)

Definition

- In { P } S { Q } , P is the weakest precondition if $\forall P'.$ { P' } S { Q' } , $P' \Rightarrow P$

Program Verification

Verification using Hoare Triples and Weakest Preconditions

- To prove { P } S { Q } is valid, we check $P \Rightarrow \mathrm{wp}(S, Q)$
- wp: a function returning the weakest precondition allowing the execution of S to achieve Q
- Need to define wp for different statements in WHILE

WP for Assignment

Find the weakest precondition P

- $\{\, P \,\}\ x := 3\ \{\, x + y \equiv 10 \,\}$?

WP for Assignment

Find the weakest precondition P

- $\{ P \}$ x := 3 $\{ x + y \equiv 10 \}$?
  - A: $y \equiv 7$
  - Check $\{ y \equiv 7 \}$ x := 3 $\{ x + y \equiv 10 \}$

WP for Assignment

Find the weakest precondition P

- $\{ P \}$ x := 3 $\{ x + y \equiv 10 \}$?
  - A: $y \equiv 7$
  - Check $\{ y \equiv 7 \}$ x := 3 $\{ x + y \equiv 10 \}$
- $\{ P \}$ x := 3 $\{ x + y > 0 \}$

WP for Assignment

Find the weakest precondition P

- $\{ P \}$ x := 3 $\{ x + y \equiv 10 \}$?
  - A: $y \equiv 7$
  - Check $\{ y \equiv 7 \}$ x := 3 $\{ x + y \equiv 10 \}$
- $\{ P \}$ x := 3 $\{ x + y > 0 \}$
  - A: $3 + y > 0$, (or $y > -3$)
  - Check $\{ y > -3 \}$ x:=3 $\{ x + y > 0 \}$

WP for Assignment

Find the weakest precondition P

- $\{ P \}$ x := 3 $\{ x + y \equiv 10 \}$?
  - A: $y \equiv 7$
  - Check $\{ y \equiv 7 \}$ x := 3 $\{ x + y \equiv 10 \}$
- $\{ P \}$ x := 3 $\{ x + y > 0 \}$
  - A: $3 + y > 0$, (or $y > -3$)
  - Check $\{ y > -3 \}$ x:=3 $\{ x + y > 0 \}$

WP for Assignment

$$\text{wp}(\text{x:= E, Q}) = \text{Q}_x^E$$

WP for Assignment

Find the weakest precondition P

- { P } x := 3 { x + y ≡ 10 } ?
  - A: y ≡ 7
  - Check { y≡7 } x := 3 { x + y ≡ 10 }
- { P } x := 3 { x + y > 0 }
  - A: 3 + y > 0, (or y > -3)
  - Check { y > -3 } x:=3 { x + y > 0 }

WP for Assignment

$$\mathrm{wp}(\mathrm{x}{:=}\ \mathrm{E},\ \mathrm{Q}) = \mathrm{Q}_x^E$$

- $\mathrm{wp}(\mathrm{x}{:=}3, x + y \equiv 10) = (x + y \equiv 10)_x^3 = 3 + y = 10 = y = 7$
- $\mathrm{wp}(\mathrm{x}{:=}3, x + y > 0) = (x + y > 0)_x^3 = 3 + y > 0$

WP for While statements

Statement	S	wp(S,Q)
Assignment	x:= e	Q_x^e
Skip	skip	Q
Sequential	S1;S2	wp(S1, wp(S2,Q))
Conditional	if b then S1 else S2	$b \Rightarrow \mathrm{wp}(S1, Q) \ \wedge \ \neg b \Rightarrow \mathrm{wp}(S2, Q)$

WP for While statements

Statement	S	wp(S,Q)
Assignment	x:= e	Q_x^e
Skip	skip	Q
Sequential	S1;S2	wp(S1, wp(S2,Q))
Conditional	if b then S1 else S2	$b \Rightarrow \text{wp}(S1, Q) \ \wedge \ \neg b \Rightarrow \text{wp}(S2, Q)$

In-class Exercise

Find the weakest preconditions for

1. { ?? } x := x + 3 { x ≡ z }
2. { ?? } x := x + 1; y := y * x { y ≡ 2 * z }
3. { ?? } if (x > 0) then y := x else y := 0 { y > 0 }

Loops

- wp(while b do S) = ??
- Idea: use loop invariant
  - holds when the loop is entered
  - preserves after the loop body is executed

Loops

- wp(while b do S) = ??
- Idea: use loop invariant
  - holds when the loop is entered
  - preserves after the loop body is executed

Example

$\{N \geq 0\}$

```
i := 0;
while (i < N)
    i := N;
```

Which ones are loop invariants? For those that are not, explain why

1. $i \equiv 0$
2. $i \equiv N$
3. $N \geq 0$
4. $i \leq N$

WP for Loop

$$\text{wp(while b do S)} = (I) \;\wedge\; (I \wedge b \Rightarrow \text{wp}(S, I)) \;\wedge\; (I \wedge \neg b \Rightarrow Q)$$

Find/Guess a loop invariant I:

- $P \Rightarrow I$: initially I is true wrt P (base case)
- $I \wedge b \Rightarrow I$: I is preserved after each execution (inductive case)
- $I \wedge \neg B \Rightarrow Q$: if the loop terminates, the post condition holds (Partial correctness)

$\{N \geq 0\}$
```
i := 0;
while (i < N)
    i := N;
```
$\{i \equiv N\}$

- Which ones would be good invariant to find the wp?
  1. $N \geq 0$
  2. $i \leq N$

WP for Loop

$$\text{wp(while b do S)} = (I) \ \land \ (I \land b \Rightarrow \text{wp}(S, I)) \ \land \ (I \land \neg b \Rightarrow Q)$$

Find/Guess a loop invariant I:

- $P \Rightarrow I$: initially I is true wrt P (base case)
- $I \land b \Rightarrow I$: I is preserved after each execution (inductive case)
- $I \land \neg B \Rightarrow Q$: if the loop terminates, the post condition holds (Partial correctness)

$\{N \geq 0\}$
```
i := 0;
while (i < N)
    i := N;
```
$\{i \equiv N\}$

- Which ones would be good invariant to find the wp?
  1. N $\geq$ 0
  2. i $\leq$ N
- Find the wp for the loop
- Prove the program is correct (show that $P \Rightarrow$ wp)

In-class Exercise

$\{x \leq 10\}$

```
while x != 10
   x := x + 1
```

$\{x \equiv 10\}$

- Find an invariant I for the loop
- Find the wp of the loop
- Prove the program is correct