

Lecture 11: Reasoning about Programs using Hoare Logic II

Yu Feng
Spring 2022

Summary of previous lecture

- Reasoning about (partial) correctness with Hoare Logic

Simple Imperative Programming Language

Expression E

- $Z \mid V \mid E_1 + E_2 \mid E_1 * E_2$

Conditional C

True | False | $E_1 = E_2$ | $E_1 \leq E_2$

A minimalist programming language for demonstrating key features of Hoare logic.

Statement S

- skip (Skip)
- abort. (Abort)
- $V := E$ (Assignment)
- $S_1; S_2$. (Composition)
- **if** C **then** S_1 **else** S_2 (If)
- **while** C **do** S (While)

Hoare logic rules

$$\frac{}{\vdash \{P\} \text{Skip} \{P\}}$$

$$\vdash \{\text{true}\} \text{abort} \{\text{false}\}$$

$$\vdash \{Q[E/x]\} x := E \{Q\}$$

$$\frac{\vdash \{P_1\} S \{Q_1\} \quad P \Rightarrow P_1 \quad Q_1 \Rightarrow Q}{\vdash \{P\} S \{Q\}}$$

$$\frac{\vdash \{P\} S_1 \{R\} \quad \vdash \{R\} S_2 \{Q\}}{\vdash \{P\} S_1; S_2 \{Q\}}$$

$$\frac{\vdash \{P \wedge C\} S_1 \{Q\} \quad \vdash \{P \wedge \neg C\} S_2 \{Q\}}{\vdash \{P\} \text{if } C \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

$$\frac{\vdash \{I \wedge C\} S \{I\}}{\vdash \{I\} \text{while } C \text{ do } S \{I \wedge \neg C\}}$$

Proof rule for assignment

$$\frac{}{\vdash \{Q[E/x]\} x := E \{Q\}}$$

- To prove Q holds after assignment $x := E$, sufficient to show that Q with E substituted for x holds before the assignment. $\boxed{?}$
- Using this rule, which of these are provable?

- $\{y=4\} x:=4 \{y=x\}$



- $\{x+1=n\} x:=x+1 \{x=n\}$



- $\{y=x\} y:=2 \{y=x\}$



- $\{z = 2\} y := x \{y = x\}$



Precondition strengthening

- Is the Hoare triple $\{z = 2\} y := x \{y = x\}$ valid?
- Is it provable using our assignment rule?

$$\frac{\vdash \{P_1\} S \{Q\} \quad P \Rightarrow P_1}{\vdash \{P\} S \{Q\}}$$

Precondition
strengthening

$$\frac{\frac{\vdash \{y = x[x/y]\} y = x \{y = x\}}{\vdash \{true\} y := x \{y = x\}} \quad z = 2 \Rightarrow true}{\vdash \{z = 2\} y := x \{y = x\}}$$

Postcondition weakening

$$\frac{\vdash \{P\} S \{Q_1\} \quad Q_1 \Rightarrow Q}{\vdash \{P\} S \{Q\}}$$

Postcondition weakening

- Suppose we can prove $\{\text{true}\} S \{x = y \wedge z = 2\}$.
- Which of these can be proved?
 - $\{\text{true}\} S \{x=y\}$
 - $\{\text{true}\} S \{z = 2\}$
 - $\{\text{true}\} S \{z > 0\}$
 - $\{\text{true}\} S \{y > 2\}$

Proof rule for If statement

$$\frac{\begin{array}{l} \vdash \{P \wedge C\} S_1 \{Q\} \\ \vdash \{P \wedge \neg C\} S_2 \{Q\} \end{array}}{\vdash \{P\} \text{ if } C \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

- Prove the correctness of this Hoare triple
 - $\{\text{true}\} \text{ if } x > 0 \text{ then } y := x \text{ else } y := -x \{y \geq 0\}$

Proof rule for loop

$$\frac{\vdash \{I \wedge C\} S \{I\}}{\vdash \{I\} \textbf{while } C \textbf{ do } S \{I \wedge \neg C\}}$$

- A loop invariant I has following properties:
 - I holds before the loop
 - I holds after each iteration of the loop
- Suppose I is a loop invariant for this loop. What is guaranteed to hold after loop terminates?
- This rule simply says “If I is a loop invariant, then $I \wedge \neg C$ must hold after loop terminates”

Proof rule for loop

$$\frac{\vdash \{I \wedge C\} S \{I\}}{\vdash \{I\} \textbf{while } C \textbf{ do } S \{I \wedge \neg C\}}$$

Consider the statement $S = \text{while } x < n \text{ do } x = x + 1$

Let's prove validity of $\{x \leq n\} S \{x \geq n\}$

What is the appropriate loop invariant?

First, let's prove $x \leq n$ is loop invariant. What do we need to show?

$$\frac{\frac{\vdash \{x + 1 \leq n\} x = x + 1 \{x \leq n\}}{x \leq n \wedge x < n \Rightarrow x + 1 < n + 1}}{\vdash \{x \leq n \wedge x < n\} x = x + 1 \{x \leq n\}} \quad x \leq n \wedge \neg(x < n) \Rightarrow x \geq n$$

$$\frac{\vdash \{x \leq n\} S \{x \leq n \wedge \neg(x < n)\}}{\{x \leq n\} S \{x \geq n\}}$$

Invariant vs. Inductive Invariant

- Suppose I is a loop invariant for “while C do S ”
- Does it always satisfy $\{I \wedge C\} S \{I\}$?
- Consider $I = j \geq 1$ and the code:

$i:=1; j:=1; \text{ while } i < n \text{ do } \{j:=j+i; i:=i+1\}$

- Strengthened invariant $j \geq 1 \wedge i \geq 1$
- Key challenge in verification is finding inductive loop invariants

Manual proof construction is tedious

$\{x \leq n\}$ // precondition

while ($x < n$) **do**

$\{x \leq n \wedge x < n\}$ // loop invariant

$x := x + 1$

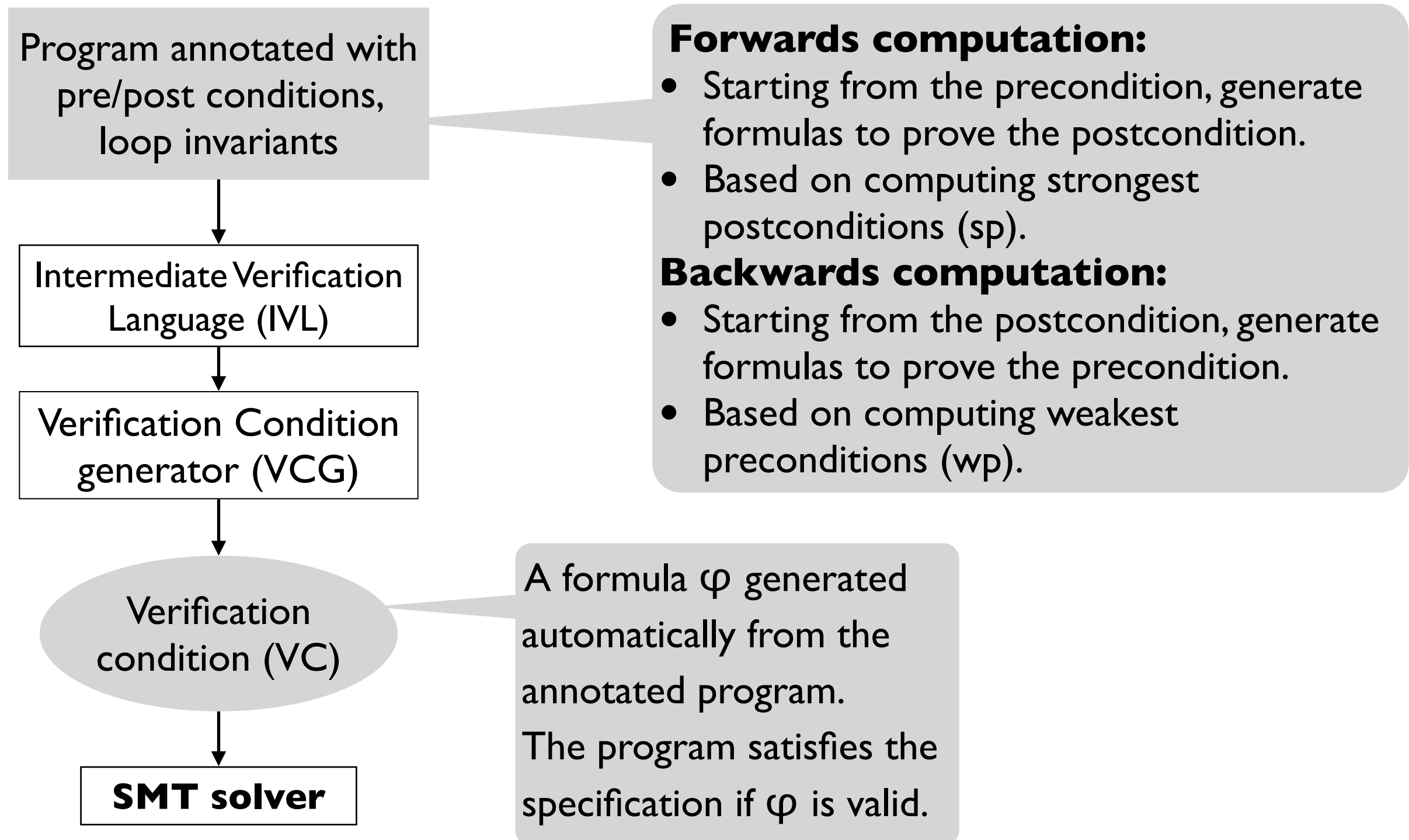
$\{x = n\}$ // postcondition

Hoare Logic proofs are highly manual:

- When to apply the rule of consequence?
- What loop invariants to use?

We can automate much of the proof process with **verification condition generation!**
But loop invariants still need to be provided...

Automating Hoare Logic via VC generation



VC generation with WP and SP

- **sp (S, P)**

- The strongest predicate that holds for states produced by executing S on a state satisfying P.

Symbolic execution, covered in next lecture, computes SPs for finite programs (no unbounded loops).

- **wp (S, Q)**

- The weakest predicate that guarantees Q will hold for states produced by executing S on a state satisfying that predicate.

Today, we'll see how to compute weakest preconditions (VWP) for IMP. This lets us verify partial correctness properties.

- **{P} S {Q} is valid if**

- $P \Rightarrow wp(S, Q)$ or $sp(S, P) \Rightarrow Q$

VC generation with WP

wp (S, Q)

- $\text{wp}(\text{skip}, Q) = Q$
- $\text{wp}(\mathbf{abort}, Q) = \text{true}$
- $\text{wp}(\mathbf{assert} \ C, Q) = C \wedge Q$
- $\text{wp}(\mathbf{assume} \ C, Q) = C \rightarrow Q$
- $\text{wp}(\mathbf{havoc} \ x, Q) = \forall x. Q$
- $\text{wp}(x := E, Q) = Q[E / x]$
- $\text{wp}(S_1; S_2, Q) = \text{wp}(S_1, \text{wp}(S_2, Q))$
- $\text{wp}(\mathbf{if} \ C \ \mathbf{then} \ S_1 \ \mathbf{else} \ S_2, Q) = (C \rightarrow \text{wp}(S_1, Q)) \wedge (\neg C \rightarrow \text{wp}(S_2, Q))$
- $\text{wp}(\mathbf{while} \ C \ \{\mathbf{I}\} \ \mathbf{do} \ S, Q) = ?$

What about loops?

VC generation for loops

- $VC(x := E, Q) = \text{true}$
- $VC(S_1; S_2, Q) = VC(S_2, Q) \wedge VC(S_1, \text{awp}(S_2, Q))$ □
- $VC(\text{if } C \text{ then } S_1 \text{ else } S_2, Q) = VC(S_1, Q) \wedge VC(S_2, Q)$
- To show I is preserved in loop, need:
 - $I \wedge C \Rightarrow \text{awp}(S, I) \wedge VC(S, I)$
- To show I is strong enough to establish Q , need:
 - $I \wedge \neg C \Rightarrow Q$
- Putting this together, verification condition for a while loop $S = \text{while } C \text{ do } \{I\}$ is:
 - $VC(S, Q) = (I \wedge C \Rightarrow \text{awp}(S, I) \wedge VC(S, I)) \wedge (I \wedge \neg C \Rightarrow Q)$

Verifying a Hoare triple

Theorem: $\{P\} S \{Q\}$ is valid if the following formula is valid:

$$P \rightarrow \text{wp}(S_{\text{IVL}}, Q)$$

TODOs by next lecture

- Start to work on your final report/project! (50%)