Logical Foundations of Computer Science

Lecture 15: Program verification (part 2)
Tiago Cogumbreiro
Equiv.v

Due Thursday October 25, 11:59pm EST
Imp.v

Due Friday October 26, 11:59pm EST
Hoare.v

Due Thursday November 1, 11:59pm EST
Why are we learning this?

In this class we are learning about three techniques:

- **formalize the PL semantics** (eg, formalize an imperative PL)
- **prove PL properties** (eg, composing Hoare triples)
- **verify programs** (eg, proving that an algorithm follows a given specification)
Summary

- Consequence Theorem
- Conditional Theorem
- While-Loop Theorem
- Axiomatic Hoare Logic
Exercise

Does \( \{ x = 2[x \mapsto x + 1][x \mapsto 1] \} \ x ::= 1 ; ; x ::= x + 1 \ \{ x = 2 \} \) hold?

Goal \{\{ \text{fun } st : \text{state} \Rightarrow \text{st } X = 2 \} [X \mapsto X + 1] [X \mapsto 1] \}\n
\[\begin{align*}
& X ::= 1 ; ; X ::= X + 1 \\
& \{\{ \text{fun } st \Rightarrow \text{st } X = 2 \} \}.
\end{align*}\]
Exercise

Does \( \{ x = 2[x \mapsto x + 1][x \mapsto 1] \} x ::= 1 ;; x ::= x + 1 \{ x = 2 \} \) hold?

Goal \{ (\text{fun } st : \text{state } \Rightarrow st X = 2) \ [X |\rightarrow X + 1] \ [ X |\rightarrow 1] \}\n
\[ X ::= 1 ;; X ::= X + 1 \]

\{ (\text{fun } st \Rightarrow st X = 2) \}.

Yes. Does \( \{ \top \} x ::= 1 ;; x ::= x + 1 \{ x = 2 \} \) hold? And, can we prove it T-seq and T-asgn?

Goal \{ (\text{fun } st \Rightarrow \text{True}) \}\n
\[ X ::= 1 ;; X ::= X + 1 \]

\{ (\text{fun } st \Rightarrow st X = 2) \}.
Exercise

Does \( \{ x = 2[x \mapsto x + 1][x \mapsto 1] \} x ::= 1;; x ::= x + 1 \{ x = 2 \} \) hold?

**Goal** \{ (fun st : state \Rightarrow st X = 2) [X |\rightarrow X + 1] [ X |\rightarrow 1] \}\{ X ::= 1;; X ::= X + 1 \}\{ fun st \Rightarrow st X = 2 \}\}.

Yes. Does \( \{\top\} x ::= 1;; x ::= x + 1 \{ x = 2 \} \) hold? And, can we prove it T-seq and T-\(\text{asgn}\)?

**Goal** \{ fun st \Rightarrow True \}\{ X ::= 1;; X ::= X + 1 \}\{ fun st \Rightarrow st X = 2 \}\}.

No. The pre-condition has to match what we stated H-\(\text{asgn}\). But we know that the above statement holds. Let us write a new theorem that handles such cases.
Assertion implication

We say that assertion $A$ implies assertion $B$, notation $A \rightarrow B$, if, and only if, for any state $s$, $A(s) \implies B(s)$. Similarly, we say that two assertions are equivalent, notation $A \iff B$, if, and only if, $A(s) \iff B(s)$ for any state $s$.

1. $\{x = 3\} \rightarrow \{x = 3 \lor x \leq y\}$
2. $\{x \neq x\} \rightarrow \{x = 3\}$
3. $\{x \leq y\} \iff \{x < y \lor x = y\}$
4. $\{x = 2[x \mapsto x + 1][x \mapsto 1]\} \iff \{\top\}$
Assertion implication

We say that assertion $A$ implies assertion $B$, notation $A \implies B$, if, and only if, for any state $s$, $A(s) \implies B(s)$. Similarly, we say that two assertions are equivalent, notation $A \iff B$, if, and only if, $A(s) \iff B(s)$ for any state $s$.

1. $\{x = 3\} \implies \{x = 3 \lor x \leq y\}$
2. $\{x \neq x\} \implies \{x = 3\}$
3. $\{x \leq y\} \iff \{x < y \lor x = y\}$
4. $\{x = 2[x \mapsto x + 1][x \mapsto 1]\} \iff \{\top\}$

Goal ((fun st ⇒ st X = 2)[X |→ X + 1][X |→ 1]) \iff (fun st ⇒ True).
Proof.
  unfold assn_sub, assert_implies; auto.
Qed.
Weakening and strengthening pre-/post conditions

We showed that $\{\top\} \ x ::= 1 ; \ x ::= x + 1 \ {x = 2}$.

Which of the following hold?

1. $\{y = 1\} \ x ::= 1 ; \ x ::= x + 1 \ {x = 2}$
Weakening and strengthening pre-/post conditions

We showed that \( \{ \top \} \ x ::= 1 ; \ x ::= x + 1 \{ x = 2 \} \).

Which of the following hold?

1. \( \{ y = 1 \} \ x ::= 1 ; \ x ::= x + 1 \{ x = 2 \} \) Holds.
2. \( \{ x = 10 \} \ x ::= 1 ; \ x ::= x + 1 \{ x = 2 \} \)
Weakening and strengthening pre-/post conditions

We showed that \( \{\top\} \ x ::= 1; \ x ::= x + 1 \ {x = 2}\).

Which of the following hold?

1. \( \{y = 1\} \ x ::= 1; \ x ::= x + 1 \ {x = 2}\) Holds.
2. \( \{x = 10\} \ x ::= 1; \ x ::= x + 1 \ {x = 2}\) Holds.
3. \( \{\top\} \ x ::= 1; \ x ::= x + 1 \ {x = 2 \land y = 1}\)
Weakening and strengthening pre-/post conditions

We showed that \( \{\top\} \; x ::= 1; \; x ::= x + 1 \{x = 2\} \).

Which of the following hold?

1. \( \{y = 1\} \; x ::= 1; \; x ::= x + 1 \{x = 2\} \) Holds.
2. \( \{x = 10\} \; x ::= 1; \; x ::= x + 1 \{x = 2\} \) Holds.
3. \( \{\top\} \; x ::= 1; \; x ::= x + 1 \{x = 2 \land y = 1\} \) Does NOT hold.
4. \( \{\top\} \; x ::= 1; \; x ::= x + 1 \{\top\} \)
Weakening and strengthening pre-/post conditions

We showed that \( \{\top\} \ x ::= 1; \ x ::= x + 1 \{x = 2\} \).

Which of the following hold?

1. \( \{y = 1\} \ x ::= 1; \ x ::= x + 1 \{x = 2\} \) Holds.
2. \( \{x = 10\} \ x ::= 1; \ x ::= x + 1 \{x = 2\} \) Holds.
3. \( \{\top\} \ x ::= 1; \ x ::= x + 1 \{x = 2 \land y = 1\} \) Does NOT hold.
4. \( \{\top\} \ x ::= 1; \ x ::= x + 1 \{\top\} \) Holds.
5. \( \{\top\} \ x ::= 1; \ x ::= x + 1 \{\bot\} \)
Weakening and strengthening pre-/post conditions

We showed that \( \{ \top \} \ x ::= 1; \ x ::= x + 1 \ {x = 2} \).

Which of the following hold?

1. \( \{ y = 1 \} \ x ::= 1; \ x ::= x + 1 \ {x = 2} \) Holds.
2. \( \{ x = 10 \} \ x ::= 1; \ x ::= x + 1 \ {x = 2} \) Holds.
3. \( \{ \top \} \ x ::= 1; \ x ::= x + 1 \ {x = 2 \land y = 1} \) Does NOT hold.
4. \( \{ \top \} \ x ::= 1; \ x ::= x + 1 \ {\top} \) Holds.
5. \( \{ \top \} \ x ::= 1; \ x ::= x + 1 \ {\bot} \) Does NOT hold.

Thus,
Weakening and strengthening pre-/post conditions

We showed that \( \{ \top \} x ::= 1; ; x ::= x + 1 \{ x = 2 \} \).

Which of the following hold?

1. \( \{ y = 1 \} x ::= 1; ; x ::= x + 1 \{ x = 2 \} \) Holds.
2. \( \{ x = 10 \} x ::= 1; ; x ::= x + 1 \{ x = 2 \} \) Holds.
3. \( \{ \top \} x ::= 1; ; x ::= x + 1 \{ x = 2 \land y = 1 \} \) Does NOT hold.
4. \( \{ \top \} x ::= 1; ; x ::= x + 1 \{ \top \} \) Holds.
5. \( \{ \top \} x ::= 1; ; x ::= x + 1 \{ \bot \} \) Does NOT hold.

Thus,

**Theorem (H-cons):** If \( \{ P' \} c \{ Q' \} \), \( P \rightarrow P' \), and \( Q' \rightarrow Q \), then \( \{ P \} c \{ Q \} \).
Theorem hoare_consequence_pre : \forall (P P' Q : Assertion) c,
    \{\{P'\}\} c \{\{Q\}\} \rightarrow
    P \rightarrow P' \rightarrow
    \{\{P\}\} c \{\{Q\}\}.

Theorem hoare_consequence_post : \forall (P Q Q' : Assertion) c,
    \{\{P\}\} c \{\{Q'\}\} \rightarrow
    Q' \rightarrow Q \rightarrow
    \{\{P\}\} c \{\{Q\}\}.

Theorem hoare_consequence : \forall (P P' Q Q' : Assertion) c,
    \{\{P'\}\} c \{\{Q'\}\} \rightarrow
    P \rightarrow P' \rightarrow
    Q' \rightarrow Q \rightarrow
    \{\{P\}\} c \{\{Q\}\}.
Exercise

Goal \{\text{fun st } \Rightarrow \text{True}\}
\{\text{fun st } \Rightarrow \text{True}\} (X ::= 1;; X ::= X + 1)
\{\text{fun st } \Rightarrow \text{st X = 2}\}.
Conditionals

**Theorem (H-cond):** If \(\{P\} \ c_1 \ \{Q\}\) and \(\{P\} \ c_2 \ \{Q\}\), then \(\{P\} \ \text{IFB} \ b \ \text{THEN} \ c_1 \ \text{ELSE} \ c_2 \ \text{FI} \ \{Q\}\).

**Theorem** hoare_cond: forall \(P\) \(Q\) \(b\) \(c_1\) \(c_2\),

\[
\begin{align*}
\{P\}\ c_1 \ \{Q\} \rightarrow \\
\{P\}\ c_2 \ \{Q\} \rightarrow \\
\{P\}\ \text{IFB} \ b \ \text{THEN} \ c_1 \ \text{ELSE} \ c_2 \ \text{FI} \ \{Q\}.
\end{align*}
\]

Prove that

\[
\begin{align*}
\{\top\} \ y := 2 \ \{x \leq y\} & \quad \{\top\} y := x + 1 \ \{x \leq y\} \\
\{\top\} \text{IFB} \ x = 0 \ \text{THEN} \ y := 2 \ \text{ELSE} \ y := x + 1 \ \text{FI} \ \{x \leq y\}\end{align*}
\]
Conditionals

Proving ELSE:

\[
\begin{align*}
\{\top\} & \Rightarrow \{x \leq y[y \mapsto x + 1]\} \\
\{x \leq y[y \mapsto x + 1]\}y & := x + 1 \{x \leq y\} \\
\{\top\}y & := x + 1\{x \leq y\} \\
\{\top\}\text{IFB } x = 0 \text{ THEN } y & := 2 \text{ ELSE } y := x + 1 \text{ FI } \{x \leq y\}
\end{align*}
\]
Conditionals

Proving **ELSE**:

\[
\{\top\} \implies \{x \leq y[y \mapsto x + 1]\} \quad \{x \leq y[y \mapsto x + 1]\} y := x + 1\{x \leq y\} \quad \text{H-asgn} \\
\{\top\} y := x + 1\{x \leq y\} \quad \text{H-cons-pre} \\
\{\top\} \text{IFB } x = 0 \text{ THEN } y := 2 \text{ ELSE} y := x + 1 \text{ FI } \{x \leq y\} \quad \text{H-cond}
\]

Proving **THEN**:

\[
\{\top\} \text{IFB } x = 0 \text{ THEN } y := 2 \text{ ELSE} y := x + 1 \text{ FI } \{x \leq y\}
\]

???

\[
\{\top\} y := 2 \{x \leq y\} \\
\{\top\} \text{IFB } x = 0 \text{ THEN } y := 2 \text{ ELSE} y := x + 1 \text{ FI } \{x \leq y\} \quad \text{H-cond}
\]
Conditionals

Proving **ELSE:**

\[
\begin{align*}
\{\top\} \rightarrow \{x \leq y[y \mapsto x + 1]\} & \quad \{x \leq y[y \mapsto x + 1]\} y := x + 1\{x \leq y\} \quad \text{H-asgn} \\
\{\top\} y := x + 1\{x \leq y\} & \quad \text{H-cons-pre} \\
\{\top\}\text{IFB } x = 0 \text{ THEN } y := 2 \text{ ELSE } y := x + 1\text{ FI } \{x \leq y\} \quad \text{H-cond}
\end{align*}
\]

Proving **THEN:**

\[
\begin{align*}
??\
\{\top\} y := 2\{x \leq y\} & \quad \text{H-cond} \\
\{\top\}\text{IFB } x = 0 \text{ THEN } y := 2 \text{ ELSE } y := x + 1\text{ FI } \{x \leq y\}
\end{align*}
\]

We are missing that \(x = 0\), which would help us prove this result!
The Hoare theorem for If

Theorem (H-if): If \( \{P \land b\} \ c_1 \ \{Q\}\) and \( \{P \land \neg b\} \ c_2 \ \{Q\}\), then
\( \{P\} \ \text{IFB} \ b \ \text{THEN} \ c_1 \ \text{ELSE} \ c_2 \ \text{FI} \ \{Q\}\).
The Hoare theorem for If in Coq

**Definition** bassn b : Assertion := fun st ⇒ (beval st b = true).

**Theorem** hoare_if : forall P Q b c1 c2,
{{fun st ⇒ P st \ bassn b st}} c1 {{Q}} →
{{fun st ⇒ P st \ ~(bassn b st)}} c2 {{Q}} →
{{P}} (IFB b THEN c1 ELSE c2 FI) {{Q}}.

**Proof.**
intros.
Example

Goal

```{fun st ⇒ True}``
IFB X = 0
THEN Y ::= 2
ELSE Y ::= X + 1
FI
```

```{fun st ⇒ st X ≤ st Y}```
The Hoare theorem for While

1. \( \{P\} \text{WHILE} \ b \ \text{DO} \ c \ \text{END} \ \{P\} \)
The Hoare theorem for While

1. \( \{P\} \text{WHILE } b \text{ DO } c \text{ END } \{P\} \)

2. \( \{P\} \text{WHILE } b \text{ DO } c \text{ END } \{P \land \neg b\} \)

We know that \( b \) is false after the loop. Can we state something about the body of the loop?
The Hoare theorem for While

1. \( \{P\} \text{WHILE } b \text{ DO } c \text{ END } \{P\} \)

2. \( \{P\} \text{WHILE } b \text{ DO } c \text{ END } \{P \land \neg b\} \)
   
   We know that \( b \) is false after the loop. Can we state something about the body of the loop?

3. If \( \{P\} \text{ c } \{P\} \), then \( \{P\} \text{WHILE } b \text{ DO } c \text{ END } \{P \land \neg b\} \)
   
   We know that the loop body must at least preserve \( \{P\} \). Why? Can we do better?
The Hoare theorem for While

1. \{P\} WHILE b DO c END \{P\}

2. \{P\} WHILE b DO c END \{P \land \neg b\}
   
   We know that \(b\) is false after the loop. Can we state something about the body of the loop?

3. If \{P\} c \{P\}, then \{P\} WHILE b DO c END \{P \land \neg b\}
   
   We know that the loop body must at least preserve \{P\}. Why? Can we do better?

Theorem (H-while): If \{P \land b\} c \{P\}, then \{P\} WHILE b DO c END \{P \land \neg b\}.

Theorem hoare_while : forall P b c,
\{\{fun st \Rightarrow P st \land \neg \text{bassn b st}\}\} c \{\{P\}\} \rightarrow
\{\{P\}\} WHILE b DO c END \{\{fun st \Rightarrow P st \land \neg \text{bassn b st}\}\}.

Proof.
unfold hoare_triple; intros.
While Example:

```
Example while_example :
    {{fun st ⇒ st X ≤ 3}}
    WHILE X ≤ 2
    DO X ::= X + 1 END
    {{fun st ⇒ st X = 3}}.
```

Proof.
Recap

- We introduced Hoare triples $\{P\} c \{Q\}$ as a framework to specify programs.
- We introduced a set of theorems (syntax-oriented) to help us prove results on Hoare triples.
Hoare Logic Theory

\{P\} \text{SKIP} \{P\} \quad (\text{H-skip}) \\
\{P[x \mapsto a]\} \ x ::= a \ {P} \quad (\text{H-asgn})

\frac{\{P\} \; c_1 \ {Q} \quad \{Q\} \; c_2 \ {R}}{\{P\} \; c_1; ; \ c_2 \ {R}} \quad (\text{H-seq})

\frac{P \rightarrow P' \quad \{P'\} \; c \ {Q'} \quad Q' \rightarrow Q}{\{P\} \; c \ {Q}} \quad (\text{H-cons})

\frac{\{P \land b\} \ c_1 \ {Q} \quad \{P \land \neg b\} \ c_2 \ {Q}}{\{P\} \text{ IFB } b \text{ THEN } c_1 \text{ ELSE } c_2 \text{ FI } \ {Q}} \quad (\text{H-if})

\frac{\{P \land b\} \ c \ {P}}{\{P\} \text{ WHILE } b \text{ DO } c \text{ END } \{P \land \neg b\}} \quad (\text{H-while})
Hoare Logic as an Axiomatic Logic

- The set of theorems in slide 12 can describe Hoare's Logic axiomatically
- **Necessary** condition (sound): \( \text{hoare\_proof}(P, c, Q) \rightarrow \{P\} c \{Q\} \)
- **Sufficient** condition (complete): \( \{P\} c \{Q\} \rightarrow \text{hoare\_proof}(P, c, Q) \)

\[
\text{Inductive hoare\_proof : Assertion} \rightarrow \text{com} \rightarrow \text{Assertion} \rightarrow \text{Type} := \\
\mid \text{H\_Skip} : \forall P, \text{hoare\_proof } P (\text{SKIP}) P \\
\mid \text{H\_Asgn} : \forall Q V a, \text{hoare\_proof } (\text{assn\_sub } V a Q) (V := a) Q \\
\mid \text{H\_Seq} : \forall P c Q d R, \text{hoare\_proof } P c Q \rightarrow \text{hoare\_proof } Q d R \rightarrow \text{hoare\_proof } P (c;;d) R \\
\mid \text{H\_If} : \forall P Q b c1 c2, \\
\quad \text{hoare\_proof } (\text{fun } st \Rightarrow P st /\ \text{bassn } b \text{ st}) c1 Q \rightarrow \\
\quad \text{hoare\_proof } (\text{fun } st \Rightarrow P st /\ \sim(\text{bassn } b \text{ st})) c2 Q \rightarrow \\
\quad \text{hoare\_proof } P (\text{IFB } b \text{ THEN } c1 \text{ ELSE } c2 \text{ FI}) Q \\
\mid \text{H\_While} : \forall P b c, \\
\quad \text{hoare\_proof } (\text{fun } st \Rightarrow P st /\ \text{bassn } b \text{ st}) c P \rightarrow \\
\quad \text{hoare\_proof } P (\text{WHILE } b \text{ DO } c \text{ END}) (\text{fun } st \Rightarrow P st /\ \sim(\text{bassn } b \text{ st})) \\
\mid \text{H\_Consequence} : \forall (P Q P' Q' : \text{Assertion}) c, \\
\quad \text{hoare\_proof } P' c Q' \rightarrow (\text{forall } st, P st \Rightarrow P' st) \rightarrow (\text{forall } st, Q' st \Rightarrow Q st) \rightarrow \text{hoare\_proof } P c Q.
\]
Summary

- Consequence Theorem
- Conditional Theorem
- While-Loop Theorem
- Axiomatic Hoare Logic