Deductive Verification of Programs

Laura Kovács

TU Wien

How to Prove that: an IMP Program Satisfies its Requirements?

Example of an IMP program *p*

```
$s := 0; n := 1;$
while $\neg(n = 101)$ do
  $s := s + n; n := n + 1$
od
```

How to prove that, upon termination of p, the value of s is $\sum_{i=1}^{100} i$?

- Take arbitrary σ and compute $\langle p, \sigma \rangle \rightarrow \sigma'$
- "Check" what is $\sigma'(s)$?

How to Prove that: an IMP Program Satisfies its Requirements?

Another example of an IMP program p

```
$s := 0; n := 1;$
while $\neg(n = m + 1)$ do
   $s := s + n; n := n + 1$
od
```

How to prove that, upon termination of p, the value of s is $\sum_{i=1}^{m} i$?

- Note: m can take infinitely many values.

How to Prove that: an IMP Program Satisfies its Requirements?

Another example of an IMP program p

```
$s := 0; n := 1;$
while $\neg(n = m + 1)$ do
    $s := s + n; n := n + 1$
od
```

How to prove that, upon termination of p, the value of s is $\sum_{i=1}^{m} i$?

- Note: m can take infinitely many values.
- We need some logic to reason about programs.
- We will rely on the **axiomatic semantics** of IMP:
  - making assertions about IMP programs;
  - providing proof rules for proving assertions;
  - using Hoare logic.

Outline

Axiomatic Semantics of IMP

Hoare Logic

- Basis of all deductive verification techniques;
- Named after Tony Hoare:
  - inventor of quick sort
  - father of formal verification
  - Turing award winner 1980
  - ...

Tony Hoare(1971): *An axiomatic basis for computer programming*

Hoare Logic

- Basis of all deductive verification techniques;
- Named after Tony Hoare:
  - inventor of quick sort
  - father of formal verification
  - Turing award winner 1980
  - . . .

  Tony Hoare(1971): *An axiomatic basis for computer programming*

- Also known as Floyd-Hoare logic

  Robert Floyd (1967): *Assigning meanings to programs*

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

$$\{A\}\ p\ \{B\}$$

For all states σ that satisfy A,
if $\langle p, \sigma \rangle \rightarrow \sigma'$, *for some* σ', **then** σ' *satisfies* B.

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

$$\{A\}\ p\ \{B\}$$

For all states σ that satisfy A,
if $\langle p, \sigma \rangle \to \sigma'$*, for some* σ', **then** σ' *satisfies* B*.*

- Total correctness assertion, written as a Hoare triple:

$$[A]\ p\ [B]$$

For all states σ that satisfy A,
then $\langle p, \sigma \rangle \to \sigma'$*, for some* σ'*, and* σ' *satisfies* B*.*

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

  $$\{A\}\ p\ \{B\}$$

  For all states σ that satisfy A,
  if $\langle p, \sigma\rangle \to \sigma'$*, for some* σ', **then** σ' *satisfies B.*

  The partial correctness assertion does not require p to terminate.

- Total correctness assertion, written as a Hoare triple:

  $$[A]\ p\ [B]$$

  For all states σ that satisfy A,
  then $\langle p, \sigma\rangle \to \sigma'$*, for some* σ'*, and* σ' *satisfies B.*

  The total correctness assertion requires p to terminate.

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

  $$\{A\}\ p\ \{B\}$$

  For all states σ that satisfy A,
  if $\langle p, \sigma\rangle \to \sigma'$*, for some* σ', **then** σ' *satisfies* B.

- Total correctness assertion, written as a Hoare triple:

  $$[A]\ p\ [B]$$

  For all states σ that satisfy A,
  then $\langle p, \sigma\rangle \to \sigma'$*, for some* σ'*, and* σ' *satisfies* B.

A is called precondition and B is called post-condition of p.

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

  $$\{A\}\ p\ \{B\}$$

  For all states σ that satisfy A,
  if $\langle p, \sigma \rangle \to \sigma'$*, for some* σ', **then** σ' *satisfies B.*

- Total correctness assertion, written as a Hoare triple:

  $$[A]\ p\ [B]$$

  For all states σ that satisfy A,
  then $\langle p, \sigma \rangle \to \sigma'$*, for some* σ'*, and* σ' *satisfies B.*

- Is $\{x = 0\}\ x := x + 1\ \{x = 1\}$ valid?
- Is $\{x = 0\}\ \textbf{while}\ \mathrm{true}\ \textbf{do}\ x := 1\ \{x = 1\}$ valid?
- Is $[x = 0]\ \textbf{while}\ \mathrm{true}\ \textbf{do}\ x := 1\ [x = 1]$ valid?

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

  $$\{A\}\ p\ \{B\}$$

  For all states σ that satisfy A,
  if $\langle p, \sigma \rangle \to \sigma'$, *for some* σ', **then** σ' *satisfies* B.

- Total correctness assertion, written as a Hoare triple:

  $$[A]\ p\ [B]$$

  For all states σ that satisfy A,
  then $\langle p, \sigma \rangle \to \sigma'$, *for some* σ', *and* σ' *satisfies* B.

What is validity of Hoare triples?

What is "state σ satisfies assertion A"?

Correctness Assertions as Hoare triples

- Partial correctness assertion, written as a Hoare triple:

$$\{A\}\ p\ \{B\}$$

For all states σ that satisfy A,
if $\langle p, \sigma\rangle \to \sigma'$, *for some* σ', **then** σ' *satisfies B.*

- Total correctness assertion, written as a Hoare triple:

$$[A]\ p\ [B]$$

For all states σ that satisfy A,
then $\langle p, \sigma\rangle \to \sigma'$, *for some* σ', *and* σ' *satisfies B.*

What is validity of Hoare triples?

What is "state σ satisfies assertion A"?

What kind of assertions about IMP **we consider?**

The Assertion Language Assn

- includes all boolean expressions/formulas from BExp;
- extends BExp and AExp, allowing *quantified first-order formulas* about IMP (e.g. $\exists i : x = i * y$)

The Assertion Language Assn

Syntax of Assn

$$
\begin{array}{lcll}
A & ::= & \mathrm{true} & \\
 & \mid & \mathrm{false} & \\
 & \mid & a_1\ \mathcal{AOP}\ a_2 & \text{for } a_1, a_2 \in \mathrm{AExp} \text{ and } \mathcal{AOP} \in \{=, <, >, \leq, \geq\} \\
 & \mid & \neg A & \text{for } A \in \mathrm{Assn} \\
 & \mid & A_1\ \mathcal{BOP}\ A_2 & \text{for } A_1, A_2 \in \mathrm{Assn} \text{ and } \mathcal{BOP} \in \{\wedge, \vee, \Rightarrow\} \\
 & \mid & \forall i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable} \\
 & \mid & \exists i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable}
\end{array}
$$

The Assertion Language Assn

Syntax of Assn: AExp and First-order logic

$$
\begin{array}{lcll}
A & ::= & \text{true} & \\
 & \mid & \text{false} & \\
 & \mid & a_1 \ \mathcal{AOP} \ a_2 & \text{for } a_1, a_2 \in \mathrm{AExp} \text{ and } \mathcal{AOP} \in \{=, <, >, \leq, \geq\} \\
 & \mid & \neg A & \text{for } A \in \mathrm{Assn} \\
 & \mid & A_1 \ \mathcal{BOP} \ A_2 & \text{for } A_1, A_2 \in \mathrm{Assn} \text{ and } \mathcal{BOP} \in \{\wedge, \vee, \Rightarrow\} \\
 & \mid & \forall i . A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable} \\
 & \mid & \exists i . A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable}
\end{array}
$$

Example of assertions: preconditions, post-conditions

The Assertion Language Assn

Syntax of Assn: AExp and First-order logic

$$
\begin{array}{lcll}
A & ::= & \mathrm{true} & \\
 & \mid & \mathrm{false} & \\
 & \mid & a_1 \ \mathcal{AOP}\ a_2 & \text{for } a_1, a_2 \in \mathrm{AExp} \text{ and } \mathcal{AOP} \in \{=, <, >, \leq, \geq\} \\
 & \mid & \neg A & \text{for } A \in \mathrm{Assn} \\
 & \mid & A_1 \ \mathcal{BOP}\ A_2 & \text{for } A_1, A_2 \in \mathrm{Assn} \text{ and } \mathcal{BOP} \in \{\wedge, \vee, \Rightarrow\} \\
 & \mid & \forall i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable} \\
 & \mid & \exists i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable}
\end{array}
$$

Bexp are assertions.

Example of assertions: preconditions, post-conditions

The Assertion Language Assn

Syntax of Assn: AExp and First-order logic

$$
\begin{array}{lcll}
A & ::= & \mathrm{true} & \\
 & \mid & \mathrm{false} & \\
 & \mid & a_1\ \mathcal{AOP}\ a_2 & \text{for } a_1, a_2 \in \mathrm{AExp} \text{ and } \mathcal{AOP} \in \{=, <, >, \leq, \geq\} \\
 & \mid & \neg A & \text{for } A \in \mathrm{Assn} \\
 & \mid & A_1\ \mathcal{BOP}\ A_2 & \text{for } A_1, A_2 \in \mathrm{Assn} \text{ and } \mathcal{BOP} \in \{\wedge, \vee, \Rightarrow\} \\
 & \mid & \forall i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable} \\
 & \mid & \exists i.A & \text{for } A \in \mathrm{Assn} \text{ and } i \text{ integer-valued (logical) variable}
\end{array}
$$

Bexp are assertions.

Example of assertions: preconditions, post-conditions

Partial/Total correctness assertions are not in Assn.

The Assertion Language Assn

Notation: We write $\sigma \models A$ to denote σ satisfies assertion A.
We write $\sigma \not\models A$ to denote not $\sigma \models A$.

The Assertion Language Assn

Semantics of Assn

$$\sigma \models \mathrm{true}$$

$$\sigma \models a_1 = a_2 \quad \text{iff } \langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2, \text{ and } n_1 = n_2$$

$$\sigma \models a_1 < a_2 \quad \text{iff } \langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2, \text{ and } n_1 < n_2$$

$$\sigma \models a_1 > a_2 \quad \text{iff } \langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2, \text{ and } n_1 > n_2$$

$$\sigma \models a_1 \leq a_2 \quad \text{iff } \langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2, \text{ and } n_1 \leq n_2$$

$$\sigma \models a_1 \geq a_2 \quad \text{iff } \langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2, \text{ and } n_1 \geq n_2$$

The Assertion Language Assn

Semantics of Assn

$\sigma \models \mathrm{true}$

$\sigma \models a_1 = a_2$ iff $\langle a_1, \sigma\rangle \to n_1, \langle a_2, \sigma\rangle \to n_2,$ and $n_1 = n_2$

$\sigma \models a_1 < a_2$ iff $\langle a_1, \sigma\rangle \to n_1, \langle a_2, \sigma\rangle \to n_2,$ and $n_1 < n_2$

$\sigma \models a_1 > a_2$ iff $\langle a_1, \sigma\rangle \to n_1, \langle a_2, \sigma\rangle \to n_2,$ and $n_1 > n_2$

$\sigma \models a_1 \leq a_2$ iff $\langle a_1, \sigma\rangle \to n_1, \langle a_2, \sigma\rangle \to n_2,$ and $n_1 \leq n_2$

$\sigma \models a_1 \geq a_2$ iff $\langle a_1, \sigma\rangle \to n_1, \langle a_2, \sigma\rangle \to n_2,$ and $n_1 \geq n_2$

$\sigma \models \neg A$ iff $\sigma \not\models A$

$\sigma \models A_1 \wedge A_2$ iff $\sigma \models A_1$ and $\sigma \models A_2$

$\sigma \models A_1 \vee A_2$ iff $\sigma \models A_1$ or $\sigma \models A_2$

$\sigma \models A_1 \Rightarrow A_2$ iff $\sigma \not\models A_1$ or $\sigma \models A_2$

The Assertion Language Assn

Semantics of Assn

$\sigma \models \mathrm{true}$

$\sigma \models a_1 = a_2$ iff $\langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2,$ and $n_1 = n_2$

$\sigma \models a_1 < a_2$ iff $\langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2,$ and $n_1 < n_2$

$\sigma \models a_1 > a_2$ iff $\langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2,$ and $n_1 > n_2$

$\sigma \models a_1 \leq a_2$ iff $\langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2,$ and $n_1 \leq n_2$

$\sigma \models a_1 \geq a_2$ iff $\langle a_1, \sigma \rangle \to n_1, \langle a_2, \sigma \rangle \to n_2,$ and $n_1 \geq n_2$

$\sigma \models \neg A$ iff $\sigma \not\models A$

$\sigma \models A_1 \wedge A_2$ iff $\sigma \models A_1$ and $\sigma \models A_2$

$\sigma \models A_1 \vee A_2$ iff $\sigma \models A_1$ or $\sigma \models A_2$

$\sigma \models A_1 \Rightarrow A_2$ iff $\sigma \not\models A_1$ or $\sigma \models A_2$

$\sigma \models \forall i.A$ iff for **all** $n \in \mathbf{Z} : \sigma[i/n] \models A$

$\sigma \models \exists i.A$ iff for **some** $n \in \mathbf{Z} : \sigma[i/n] \models A$

Semantics of Correctness Assertions

Let $A, B \in \mathrm{Assn}, p \in \mathrm{P}$.

FOR SIMPLICITY, WE CONSIDER ONLY PARTIAL CORRECTNESS.

Semantics of Partial Correctness

- We write $\sigma \vDash \{A\}\ p\ \{B\}$ to denote that σ satisfies the partial correctness assertion $\{A\}\ p\ \{B\}$.

Semantics of Correctness Assertions

Let $A, B \in \mathrm{Assn}, p \in \mathrm{P}$.

FOR SIMPLICITY, WE CONSIDER ONLY PARTIAL CORRECTNESS.

Semantics of Partial Correctness

- We write $\sigma \vDash \{A\}\ p\ \{B\}$ to denote that σ satisfies the partial correctness assertion $\{A\}\ p\ \{B\}$.

  Then,
  $\sigma \vDash \{A\}\ p\ \{B\}$ iff $\big(\sigma \vDash A \Rightarrow (\forall \sigma'. \langle p, \sigma\rangle \to \sigma' \Rightarrow \sigma' \vDash B)\big)$

Semantics of Correctness Assertions

Let $A, B \in \mathrm{Assn}, p \in \mathrm{P}$.

FOR SIMPLICITY, WE CONSIDER ONLY PARTIAL CORRECTNESS.

Semantics of Partial Correctness

- We write $\sigma \vDash \{A\}\ p\ \{B\}$ to denote that σ satisfies the partial correctness assertion $\{A\}\ p\ \{B\}$.

  Then,
  $\sigma \vDash \{A\}\ p\ \{B\} \quad$ iff $\quad \big(\sigma \vDash A \Rightarrow (\forall \sigma'. \langle p, \sigma \rangle \to \sigma' \Rightarrow \sigma' \vDash B)\big)$

- $\{A\}\ p\ \{B\}$ is valid iff $\sigma \vDash \{A\}\ p\ \{B\}$ for all σ.

  We write $\vDash \{A\}\ p\ \{B\}$ to denote that $\{A\}\ p\ \{B\}$ is valid.

Semantics of Correctness Assertions

Let $A, B \in \mathrm{Assn}, p \in \mathrm{P}$.

FOR SIMPLICITY, WE CONSIDER ONLY PARTIAL CORRECTNESS.

Semantics of Partial Correctness

- We write $\sigma \vDash \{A\}\ p\ \{B\}$ to denote that σ satisfies the partial correctness assertion $\{A\}\ p\ \{B\}$.

  Then,
  $\sigma \vDash \{A\}\ p\ \{B\} \quad \text{iff} \quad \big(\sigma \vDash A \Rightarrow (\forall \sigma'. \langle p, \sigma\rangle \to \sigma' \Rightarrow \sigma' \vDash B)\big)$

- $\{A\}\ p\ \{B\}$ is valid iff $\sigma \vDash \{A\}\ p\ \{B\}$ for all σ.

  We write $\vDash \{A\}\ p\ \{B\}$ to denote that $\{A\}\ p\ \{B\}$ is valid.

  When $\vDash \{A\}\ p\ \{B\}$, we also say that $\{A\}\ p\ \{B\}$ is true/valid regarding partial correctness (is "partially correct").

Correctness Assertions - Examples

Which Hoare triples given below are valid?

- $\{x = 0\}\ x := x + 1\ \{x = 1\}$
- $\{x = 0 \wedge y = 1\}\ x := x + 1\ \{x = 1 \wedge y = 2\}$
- $\{x = 0\}\ x := x + 1\ \{x = 1 \vee y = 2\}$
- $\{x = 0\}$ **while** true **do** $x := 0$ **od** $\{x = 1\}$

Correctness Assertions - Examples

Which Hoare triples given below are valid?

Why? Give a formal proof of validity of the below Hoare triples.

- $\{x = 0\}\ x := x + 1\ \{x = 1\}$
- $\{x = 0 \wedge y = 1\}\ x := x + 1\ \{x = 1 \wedge y = 2\}$
- $\{x = 0\}\ x := x + 1\ \{x = 1 \vee y = 2\}$
- $\{x = 0\}$ **while** true **do** $x := 0$ **od** $\{x = 1\}$

Proving Correctness Assertions

Key issue: How to prove validity of Hoare triples?

- $\models \{A\}\ p\ \{B\}$ and $\models [A]\ p\ [B]$ are tedious to use
- Defined in terms of the operational semantics

Proving Correctness Assertions

Key issue: How to prove validity of Hoare triples?

- $\models \{A\}\ p\ \{B\}$ and $\models [A]\ p\ [B]$ are tedious to use
- Defined in terms of the operational semantics
- Need a symbolic technique for proving valid triples $\{A\}\ p\ \{B\}$

Proving Correctness Assertions

Key issue: How to prove validity of Hoare triples?

- $\models \{A\}\ p\ \{B\}$ and $\models [A]\ p\ [B]$ are tedious to use
- Defined in terms of the operational semantics
- Need a symbolic technique for proving valid triples $\{A\}\ p\ \{B\}$
  - Need of a proof system to prove $\models \{A\}\ p\ \{B\}$
  - Write $\vdash \{A\}\ p\ \{B\}$ to denote that we can prove validity of $\{A\}\ p\ \{B\}$ using the proof rules of our proof system

Proving Correctness Assertions – Hoare Logic

- Hoare gave a proof system for proving correctness assertions
  - This proof system is called Hoare logic
  - It consists of a collection of proof rules, called Hoare rules.

Proving Correctness Assertions – Hoare Logic

- Hoare gave a proof system for proving correctness assertions
  - This proof system is called Hoare logic
  - It consists of a collection of proof rules, called Hoare rules.

- Hoare logic is sound and (relative-)complete
  - Soundness: if $\vdash \{A\}\ p\ \{B\}$ using Hoare rules then $\vDash \{A\}\ p\ \{B\}$

    "if a Hoare triple is proved to be valid using Hoare rules, then it is a valid Hoare triple."

Proving Correctness Assertions – Hoare Logic

- Hoare gave a proof system for proving correctness assertions
  - This proof system is called Hoare logic
  - It consists of a collection of proof rules, called Hoare rules.

- Hoare logic is sound and (relative-)complete
  - Soundness: if $\vdash \{A\}\ p\ \{B\}$ using Hoare rules then $\vDash \{A\}\ p\ \{B\}$

    "if a Hoare triple is proved to be valid using Hoare rules, then it is a valid Hoare triple."
  - Completeness: if $\vDash \{A\}\ p\ \{B\}$ then $\vdash \{A\}\ p\ \{B\}$ using Hoare rules.

    "any valid Hoare triple can be proved to be valid using Hoare rules"

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

- each rule is sound (admissible): if the assumptions in the rule's premise are valid, so is its conclusion.

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Soundness of the rule of consequence:

Assume: $\models A \Rightarrow A', \quad \models \{A'\}\ p\ \{B'\}, \quad \models B' \Rightarrow B.$
Prove: $\models \{A\}\ p\ \{B\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Soundness of the rule of consequence:

Assume: $\models A \Rightarrow A'$, $\quad \models \{A'\}\ p\ \{B'\}$, $\quad \models B' \Rightarrow B$.

Prove: $\sigma \models \{A\}\ p\ \{B\}$ for arbitrary σ.

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Soundness of the rule of consequence:

Assume: $\models A \Rightarrow A', \quad \models \{A'\}\ p\ \{B'\}, \quad \models B' \Rightarrow B$.

Prove: $\sigma \models \{A\}\ p\ \{B\}$ for arbitrary σ.

Assume $\sigma \models A$.

Since $\models A \Rightarrow A'$, we have $\sigma \models A'$. (why?)

Take any σ' s.t. $\langle p, \sigma \rangle \to \sigma'$. From $\sigma \models A'$ and $\models \{A'\}\ p\ \{B'\}$, we have $\sigma' \models B'$. (why?)

From $\models B' \Rightarrow B$, we get $\sigma' \models B$ (why?)

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

- one rule for each IMP program construct (command)
- and the rule of consequence:

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Soundness of the rule of consequence:

Assume: $\models A \Rightarrow A'$, $\quad \models \{A'\}\ p\ \{B'\}$, $\quad \models B' \Rightarrow B$.
Prove: $\sigma \models \{A\}\ p\ \{B\}$ for arbitrary σ.

Assume $\sigma \models A$.
Since $\models A \Rightarrow A'$, we have $\sigma \models A'$. (why?)
Take any σ' s.t. $\langle p, \sigma\rangle \to \sigma'$. From $\sigma \models A'$ and $\models \{A'\}\ p\ \{B'\}$, we have $\sigma' \models B'$. (why?)
From $\models B' \Rightarrow B$, we get $\sigma' \models B$ (why?)
Then, $\sigma \models \{A\}\ p\ \{B\}$.

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Understanding the Rule for Assignment

- Consider the assignment $x := y$ and post-condition $x > 0$.
- What do we need to know before the assignment so that $x > 0$ holds afterwards?

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Understanding the Rule for Assignment

- Consider the assignment $x := y$ and post-condition $x > 0$.
- What do we need to know before the assignment so that $x > 0$ holds afterwards?

- Consider the assignment $x := x + 1$ and post-condition $x > 5$.
- What do we need to know before the assignment so that $x > 5$ holds afterwards?

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment.

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall y. x = x\}\ y := x\ \{\forall y. y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall y. x = x\}\ y := x\ \{\forall y. y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment. Only substitute free occurrences of x in B!

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall y.x = x\}\ y := x\ \{\forall y.y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment. Only substitute free occurrences of x in B!

  Assertions are equivalent up to renaming of bound variables: $\forall x.x = y$ is the same as $\forall z.z = y$

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall y.x = x\}\ y := x\ \{\forall y.y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment. Only substitute free occurrences of x in B!

  If B is a quantified assertions, then rename bound variables of B if these bound variables occur in $x := a$.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall y.x = x\}\ y := x\ \{\forall y.y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment. Only substitute free occurrences of x in B!

  If B is a quantified assertions, then rename bound variables of B if these bound variables occur in $x := a$.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall z. z = x\}\ y := x\ \{\forall z. z = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

Rule for Assignment

$$\vdash \{B[x/a]\}\ x := a\ \{B\}$$

- To prove B holds after assignment $x := a$, sufficient to show that B with a substituted for x, that is $B[x/a]$, holds before the assignment. Only substitute free occurrences of x in B!

  If B is a quantified assertions, then rename bound variables of B if these bound variables occur in $x := a$.

- Using this rule, which Hoare triples are provable?
  - $\{y = 4\}\ x := 4\ \{y = x\}$
  - $\{x + 1 = y\}\ x := x + 1\ \{x = y\}$
  - $\{y = x\}\ y := 0\ \{y = x\}$
  - $\{z = x\}\ y := x\ \{z = x\}$
  - $\{\forall z.z = x\}\ y := x\ \{\forall z.z = x\}$
  - $\{\forall y.y = 1\}\ x := 1\ \{\forall y.y = x\}$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}}$$

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\quad\ \}\ \textbf{abort}\ \{\mathbf{B}\}}$$

recall $\{A\}\ p\ \{B\}$:
if $\sigma \vDash A$ and if $\langle p, \sigma\rangle \rightarrow \sigma',$ then $\sigma' \vDash B$

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\quad\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

recall $\{A\}\ p\ \{B\}$:
if $\sigma \models A$ and if $\langle p, \sigma\rangle \to \sigma'$, then $\sigma' \models B$
but $\langle \textbf{abort}, \sigma\rangle \to \text{undefined}$ for any σ

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\; p\; \{B\}$

$$\frac{}{\{B[x/a]\}\; x := a\; \{B\}} \qquad \frac{}{\{A\}\; \textbf{skip}\; \{\textbf{A}\}} \qquad \frac{}{\{\quad\}\; \textbf{abort}\; \{\textbf{B}\}}$$

recall $\{A\}\; p\; \{B\}$:
if $\sigma \vDash A$ and if $\langle p, \sigma\rangle \to \sigma'$, then $\sigma' \vDash B$
but $\langle \textbf{abort}, \sigma\rangle \to \mathrm{undefined}$ for any σ
so, the above implication always holds

$$\frac{A \Rightarrow A' \quad \{A'\}\; p\; \{B'\} \quad B' \Rightarrow B}{\{A\}\; p\; \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\mathrm{true}\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

recall $\{A\}\ p\ \{B\}$:
if $\sigma \vDash A$ and if $\langle p, \sigma\rangle \rightarrow \sigma'$, then $\sigma' \vDash B$
but $\langle \textbf{abort}, \sigma\rangle \rightarrow \mathrm{undefined}$ for any σ
so, the above implication always holds

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\mathrm{true}\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

$$\frac{\{A\}\ p_1\ \{C\} \quad \{C\}\ p_2\ \{B\}}{\{A\}\ p_1 ; p_2\ \{B\}}$$

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\mathrm{true}\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

$$\frac{\{A\}\ p_1\ \{C\} \quad \{C\}\ p_2\ \{B\}}{\{A\}\ p_1 ; p_2\ \{B\}} \qquad \frac{\{A \wedge b\}\ p_1\ \{B\} \quad \{A \wedge \neg b\}\ p_2\ \{B\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p_1\ \textbf{else}\ p_2\ \{B\}}$$

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\mathrm{true}\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

$$\frac{\{A\}\ p_1\ \{C\} \quad \{C\}\ p_2\ \{B\}}{\{A\}\ p_1 ; p_2\ \{B\}} \qquad \frac{\{A \wedge b\}\ p_1\ \{B\} \quad \{A \wedge \neg b\}\ p_2\ \{B\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p_1\ \textbf{else}\ p_2\ \{B\}}$$

$$\frac{???}{\{A\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Hoare Rules for While Loops

$$\frac{}{\{A\}\qquad\qquad \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- A satisfies states before the first iteration of the loop;

Hoare Rules for While Loops

Recall the operational semantics of loops:

$$\frac{\qquad}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- A satisfies states before the first iteration of the loop;

Hoare Rules for While Loops

Recall the operational semantics of loops:

$$\dfrac{\dfrac{}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{C\}} \qquad \dfrac{}{\{C\} \qquad\qquad\qquad \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p;\textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- *A* satisfies states before the first iteration of the loop;
- *C* satisfies states after the first iteration of the loop;
- *C* satisfies states before the second iteration of the loop;

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{A \wedge b\}\ p\ \{C\}}{\{A\}\ \mathbf{if}\ b\ \mathbf{then}\ p\ \{C\}} \qquad \dfrac{}{\{C\}\ \qquad\qquad \mathbf{while}\ b\ \mathbf{do}\ p\ \mathbf{od}\ \{B\}}}{\{A\}\ \mathbf{if}\ b\ \mathbf{then}\ p;\ \mathbf{while}\ b\ \mathbf{do}\ p\ \mathbf{od}\ \{B\}}$$

- *A* satisfies states before the first iteration of the loop;
- *C* satisfies states after the first iteration of the loop;
- *C* satisfies states before the second iteration of the loop;

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{A \wedge b\}\ p\ \{C\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{C\}} \qquad \dfrac{}{\{C\}\ \textbf{if}\ b\ \textbf{then}\ p; \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p; \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- *A* satisfies states before the first iteration of the loop;
- *C* satisfies states after the first iteration of the loop;
- *C* satisfies states before the second iteration of the loop;

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{A \wedge b\}\ p\ \{C\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{C\}} \qquad \dfrac{\{C \wedge b\}\ p\ \{D\} \quad \dfrac{\vdots}{\{D\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{C\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- A satisfies states before the first iteration of the loop;
- C satisfies states after the first iteration of the loop;
- C satisfies states before the second iteration of the loop;
- D satisfies states after the second iteration of the loop;
- D satisfies states before the third iteration of the loop;
- …

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{A \wedge b\}\ p\ \{C\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{C\}} \qquad \dfrac{\{C \wedge b\}\ p\ \{D\} \qquad \dfrac{\vdots}{\{D\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{C\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{B\}}$$

- *A* satisfies states before the first iteration of the loop;
- *C* satisfies states after the first iteration of the loop;
- *C* satisfies states before the second iteration of the loop;
- *D* satisfies states after the second iteration of the loop;
- *D* satisfies states before the third iteration of the loop;
- …

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{I \land b\}\ p\ \{I\}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{I\}} \qquad \dfrac{\{I \land b\}\ p\ \{I\} \quad \dfrac{\vdots}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}$$

Inductive Loop Invariant *I*

- *I* satisfies states before the first iteration of the loop;
- *I* satisfies states before and after each iteration of the loop;

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{I \wedge b\}\ p\ \{I\}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{I\}} \qquad \dfrac{\{I \wedge b\}\ p\ \{I\} \quad \dfrac{\vdots}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}$$

Inductive Loop Invariant *I*

- *I* satisfies states before the first iteration of the loop;
- *I* satisfies states before and after each iteration of the loop;
- If *I* is an inductive loop invariant,
  – does *I* also hold after the loop terminates?

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{I \wedge b\}\ p\ \{I\}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{I\}} \qquad \dfrac{\{I \wedge b\}\ p\ \{I\} \quad \dfrac{\vdots}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}$$

Inductive Loop Invariant *I*

- *I* satisfies states before the first iteration of the loop;
- *I* satisfies states before and after each iteration of the loop;

- If *I* is an inductive loop invariant,
  – does *I* also hold after the loop terminates? Yes, by definition!

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{I \wedge b\}\ p\ \{I\}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{I\}} \qquad \dfrac{\{I \wedge b\}\ p\ \{I\} \quad \dfrac{\vdots}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{\qquad\}}$$

Inductive Loop Invariant *I*

- *I* satisfies states before the first iteration of the loop;
- *I* satisfies states before and after each iteration of the loop;
- If *I* is an inductive loop invariant,
  – does *I* also hold after the loop terminates? Yes, by definition!
  – what is guaranteed to hold after the loop terminates?

Hoare Rules for While Loops

$$\dfrac{\dfrac{\{I \wedge b\}\ p\ \{I\}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p\ \{I\}} \qquad \dfrac{\{I \wedge b\}\ p\ \{I\} \quad \dfrac{\vdots}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{I \wedge \neg b\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{I \wedge \neg b\}}}{\{I\}\ \textbf{if}\ b\ \textbf{then}\ p;\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{I \wedge \neg b\}}$$

Inductive Loop Invariant I

- I satisfies states before the first iteration of the loop;
- I satisfies states before and after each iteration of the loop;

- If I is an inductive loop invariant,
  – does I also hold after the loop terminates? Yes, by definition!
  – what is guaranteed to hold after the loop terminates?

$I \wedge \neg b$

Hoare Rules for While Loops

Putting everything together, the Hoare rule for while-loops is:

$$\frac{\{I \wedge b\}\ p\ \{I\}}{\{I\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{I \wedge \neg b\}}$$

Inductive Loop Invariant I

- I satisfies states before the first iteration of the loop;
- I satisfies states before and after each iteration of the loop;
- If I is an inductive loop invariant,
  – does I also hold after the loop terminates? Yes, by definition!
  – what is guaranteed to hold after the loop terminates?

  $I \wedge \neg b$

Hoare Rules for $\vdash \{A\}\ p\ \{B\}$

$$\frac{}{\{B[x/a]\}\ x := a\ \{B\}} \qquad \frac{}{\{A\}\ \textbf{skip}\ \{\mathbf{A}\}} \qquad \frac{}{\{\mathrm{true}\}\ \textbf{abort}\ \{\mathbf{B}\}}$$

$$\frac{\{A\}\ p_1\ \{C\} \quad \{C\}\ p_2\ \{B\}}{\{A\}\ p_1 ; p_2\ \{B\}} \qquad \frac{\{A \wedge b\}\ p_1\ \{B\} \quad \{A \wedge \neg b\}\ p_2\ \{B\}}{\{A\}\ \textbf{if}\ b\ \textbf{then}\ p_1\ \textbf{else}\ p_2\ \{B\}}$$

$$\frac{\{A \wedge b\}\ p\ \{A\}}{\{A\}\ \textbf{while}\ b\ \textbf{do}\ p\ \textbf{od}\ \{A \wedge \neg b\}}\ ,$$

where A is an **inductive loop invariant**

- A holds before and after each loop iteration

$$\frac{A \Rightarrow A' \quad \{A'\}\ p\ \{B'\} \quad B' \Rightarrow B}{\{A\}\ p\ \{B\}}$$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop. That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
    $x := x + 1; y := y + x$
od
```

Which assertions below are loop invariants/inductive invariants?

- $x \leq n$
- $x < n$
- $y \geq 0$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
x := 0; y := 0; n := 10;
while x < n do
      x := x + 1; y := y + x
od
```

Which assertions below are loop invariants/inductive invariants?

- $x \leq n$ inductive invariant (hence, also an invariant)
- $x < n$ not an invariant (hence, not inductive invariant either)
- $y \geq 0$ invariant, but not inductive invariant

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
x := 0; y := 0; n := 10;
while x < n do
      x := x + 1; y := y + x
od
```

Which assertions below are loop invariants/inductive invariants? **Why?**

- $x \leq n$ inductive invariant (hence, also an invariant)
- $x < n$ not an invariant (hence, not inductive invariant either)
- $y \geq 0$ invariant, but not inductive invariant

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
x := 0; y := 0; n := 10;
while x < n do
      x := x + 1; y := y + x
od
```

$x \leq n$ is inductive invariant iff $\vDash \{x \leq n \wedge x < n\}\ x := x + 1; y := y + x\ \{x \leq n\}$

$$\dfrac{\dfrac{\dfrac{}{\{\qquad\}\ x := x + 1\ \{\qquad\}} \qquad \dfrac{}{\{\qquad\}\ y := y + x\ \{x \leq n\}}}{\{\qquad\}\ x := x + 1; y := y + x\ \{x \leq n\}}}{}$$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
      $x := x + 1; y := y + x$
od
```

$x \leq n$ is inductive invariant iff $\models \{x \leq n \wedge x < n\}\ x := x + 1; y := y + x\ \{x \leq n\}$

$$\dfrac{\dfrac{\dfrac{}{\{\qquad\qquad\}\ x := x + 1\ \{x \leq n\}} \qquad \dfrac{}{\{x \leq n\}\ y := y + x\ \{x \leq n\}}}{\{\qquad\qquad\}\ x := x + 1; y := y + x\ \{x \leq n\}}}{}$$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop. That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
      $x := x + 1; y := y + x$
od
```

$x \leq n$ is inductive invariant iff $\vDash \{x \leq n \wedge x < n\}\ x := x + 1; y := y + x\ \{x \leq n\}$

$$\dfrac{\dfrac{\dfrac{}{\{x+1 \leq n\}\ x := x+1\ \{x \leq n\}} \qquad \dfrac{}{\{x \leq n\}\ y := y+x\ \{x \leq n\}}}{\{x+1 \leq n\}\ x := x+1; y := y+x\ \{x \leq n\}}}{}$$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop. That is: $\{A \land b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
      $x := x + 1; y := y + x$
od
```

$x \le n$ is inductive invariant iff $\models \{x \le n \land x < n\}\ x := x + 1; y := y + x\ \{x \le n\}$

$$\dfrac{x \le n \land x < n \Rightarrow x + 1 \le n \qquad \dfrac{\dfrac{}{\{x + 1 \le n\}\ x := x + 1\ \{x \le n\}} \quad \dfrac{}{\{x \le n\}\ y := y + x\ \{x \le n\}}}{\{x + 1 \le n\}\ x := x + 1; y := y + x\ \{x \le n\}}}{\{x \le n \land x < n\}\ x := x + 1; y := y + x\ \{x \le n\}}\ \mathit{conseq}$$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
    $x := x + 1; y := y + x$
od
```

$x \le n$ is inductive invariant iff $\models \{x \le n \wedge x < n\}\ x := x + 1; y := y + x\ \{x \le n\}$

$$\frac{x \le n \wedge x < n \Rightarrow x + 1 \le n \qquad \dfrac{\dfrac{}{\{x + 1 \le n\}\ x := x + 1\ \{x \le n\}} \quad \dfrac{}{\{x \le n\}\ y := y + x\ \{x \le n\}}}{\{x + 1 \le n\}\ x := x + 1; y := y + x\ \{x \le n\}}}{\{x \le n \wedge x < n\}\ x := x + 1; y := y + x\ \{x \le n\}}\ \mathit{conseq}$$

So, $x \le n$ is inductive invariant.

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \land b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
    $x := x + 1; y := y + x$
od
```

$y \geq 0$ is not an inductive invariant since:

$\{y \geq 0 \land x < n\}\ x := x + 1; y := y + x\ \{y \geq 0\}$ is not valid.

Counterexample (e.g. state σ that does not satisfy the Hoare triple):
$\sigma(x) = -3, \sigma(y) = 0, \sigma(n) = 10$.

.

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop. That is: $\{A \wedge b\}\ p\ \{A\}$.

Example: Consider the following IMP program:

```
$x := 0; y := 0; n := 10;$
while $x < n$ do
      $x := x + 1; y := y + x$
od
```

$y \geq 0$ is not an inductive invariant since:

$\{y \geq 0 \wedge x < n\}\ x := x + 1; y := y + x\ \{y \geq 0\}$ is not valid.

Counterexample (e.g. state σ that does not satisfy the Hoare triple):
$\sigma(x) = -3, \sigma(y) = 0, \sigma(n) = 10$.

Strengthened invariant $y \geq 0 \wedge x \geq 0$ is inductive.

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

- Key challenge in automated verification is finding inductive loop invariants
- Inductive loop invariants are the only invariants we can prove.

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

- Key challenge in automated verification is finding inductive loop invariants
- Inductive loop invariants are the only invariants we can prove.
- Assume A is an inductive loop invariant of the considered loop.

  What about $\vdash \{P\}$ **while** b **do** p **od** $\{Q\}$?

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

- Key challenge in automated verification is finding inductive loop invariants
- Inductive loop invariants are the only invariants we can prove.
- Assume A is an inductive loop invariant of the considered loop.

  What about $\vdash \{P\}$ **while** b **do** p **od** $\{Q\}$?

  - Use rule of consequence
  - Prove $P \Rightarrow A$ and $A \wedge \neg b \Rightarrow Q$

Loop Invariants vs Inductive Loop Invariant

Consider the loop **while** b **do** p **od**.

A loop invariant A:

- holds after each iteration of the loop.

An inductive loop invariant A:

- holds before and after each iteration of the loop.
  That is: $\{A \wedge b\}\ p\ \{A\}$.

- Key challenge in automated verification is finding inductive loop invariants that are "good".
- Inductive loop invariants are the only invariants we can prove.
- Assume A is an inductive loop invariant of the considered loop.

  What about $\vdash \{P\}$ **while** b **do** p **od** $\{Q\}$?

  - Use rule of consequence
  - Prove $P \Rightarrow A$ and $A \wedge \neg b \Rightarrow Q$
- A *"good"* invariant depends on your correctness assertion.

Learning Objectives

- Operational semantics of IMP
- Reasoning using operational semantics of IMP
- Partial vs total correctness of IMP programs
- Validity of Hoare triples