IMP: A Simple Imperative Language

An IMP program:

```
p := 0;
x := 0;
while x < n do
        x := x + 1;
        p := p + m;
```

SOS: Inference Rules for *Aexp*

- In general, we have one rule per language construct:

$$\langle n, q\rangle \Downarrow n \qquad \langle x, q\rangle \Downarrow q(x)$$

$$\frac{\langle e_1, q\rangle \Downarrow n_1 \quad \langle e_2, q\rangle \Downarrow n_2}{\langle e_1 + e_2, q\rangle \Downarrow (n_1 + n_2)} \qquad \frac{\langle e_1, q\rangle \Downarrow n_1 \quad \langle e_2, q\rangle \Downarrow n_2}{\langle e_1 - e_2, q\rangle \Downarrow (n_1 - n_2)}$$

$$\frac{\langle e_1, q\rangle \Downarrow n_1 \quad \langle e_2, q\rangle \Downarrow n_2}{\langle e_1 * e_2, q\rangle \Downarrow (n_1 \cdot n_2)}$$

- This is called structural operational semantics.
  - rules are defined based on the structure of the expressions.

SOS: Inference Rules for Commands

$$q \xrightarrow{\texttt{skip}} q$$

$$\frac{\langle e, q\rangle \Downarrow n}{q \xrightarrow{x := e} q[x \mapsto n]}$$

$$\frac{q \xrightarrow{c_1} q' \quad q' \xrightarrow{c_2} q''}{q \xrightarrow{c_1 ; c_2} q''}$$

$$\frac{\langle b, q\rangle \Downarrow \text{true} \quad q \xrightarrow{c_1} q'}{q \xrightarrow{\texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2} q'}$$

$$\frac{\langle b, q\rangle \Downarrow \text{false} \quad q \xrightarrow{c_2} q'}{q \xrightarrow{\texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2} q'}$$

$$\frac{\langle b, q\rangle \Downarrow \text{false}}{q \xrightarrow{\texttt{while } b \texttt{ do } c} q}$$

$$\frac{\langle b, q\rangle \Downarrow \text{true} \quad q \xrightarrow{c} q' \quad q' \xrightarrow{\texttt{while } b \texttt{ do } c} q''}{q \xrightarrow{\texttt{while } b \texttt{ do } c} q''}$$

Axiomatic Semantics

- An axiomatic semantics consists of:
  - a language for stating assertions about programs;
  - rules for establishing the truth of assertions.
- Some typical kinds of assertions:
  - This program terminates.
  - If this program terminates, the variables x and y have the same value throughout the execution of the program.
  - The array accesses are within the array bounds.
- Some typical languages of assertions
  - First-order logic
  - Other logics (temporal, linear)
  - Special-purpose specification languages (Z, Larch, JML)

Assertions for IMP

- The assertions we make about IMP programs are of the form:

  $$\{A\}\ c\ \{B\}$$

  with the meaning that:
  - If A holds in state q and $q \xrightarrow{c} q'$
  - then B holds in q'
- A is the pre-condition and B is the post-condition
- For example:

  $$\{\ y \leq x\ \}\ z := x;\ z := z + 1\ \{\ y < z\ \}$$

  is a valid assertion
- These are called Hoare triples or Hoare assertions

Assertions for IMP

- {A} c {B} is a partial correctness assertion. It does not imply termination of c.
- [A] c [B] is a total correctness assertion meaning that
  - If A holds in state q
  - then there exists q' such that $q \xrightarrow{c} q'$ and B holds in state q'
- Now let's be more formal
  - Formalize the language of assertions, A and B
  - Say when an assertion holds in a state
  - Give rules for deriving valid Hoare triples

The Assertion Language

- We use first-order predicate logic with IMP expressions

$$A ::= \text{true} \mid \text{false} \mid e_1 = e_2 \mid e_1 \geq e_2 \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid A_1 \Rightarrow A_2 \mid \forall x.A \mid \exists x.A$$

- Note that we are somewhat sloppy and mix the logical variables and the program variables.
- Implicitly, all IMP variables range over integers.
- All IMP Boolean expressions are also assertions.

Semantics of Assertions

- We introduced a language of assertions, we need to assign meanings to assertions.
- Notation $q \models A$ says that assertion A holds in a given state q.
  - This is well-defined when q is defined on all variables occurring in A.
- The $\models$ judgment is defined inductively on the structure of assertions.
- It relies on the semantics of arithmetic expressions from IMP.

Semantics of Assertions

- $q \models \text{true}$ always
- $q \models e_1 = e_2$ iff $\langle e_1, q\rangle\Downarrow = \langle e_2, q\rangle\Downarrow$
- $q \models e_1 \geq e_2$ iff $\langle e_1, q\rangle\Downarrow \geq \langle e_2, q\rangle\Downarrow$
- $q \models A_1 \wedge A_2$ iff $q \models A_1$ and $q \models A_2$
- $q \models A_1 \vee A_2$ iff $q \models A_1$ or $q \models A_2$
- $q \models A_1 \Rightarrow A_2$ iff $q \models A_1$ implies $q \models A_2$
- $q \models \forall x.A$ iff $\forall n \in \mathbb{Z}.\ q[x \mapsto n] \models A$
- $q \models \exists x.A$ iff $\exists n \in \mathbb{Z}.\ q[x \mapsto n] \models A$

Semantics of Hoare Triples

- Now we can define formally the meaning of a partial correctness assertion:

  $\models \{A\}\ c\ \{B\}$ iff

  $\forall q \in Q.\ \forall q' \in Q.\ q \models A \wedge q \xrightarrow{c} q' \Rightarrow q' \models B$
- and the meaning of a total correctness assertion:

  $\models [A]\ c\ [B]$ iff

  $\forall q \in Q.\ q \models A \Rightarrow \exists q' \in Q.\ q \xrightarrow{c} q' \wedge q' \models B$

  or even better:

  $\forall q \in Q.\ \forall q' \in Q.\ q \models A \wedge q \xrightarrow{c} q' \Rightarrow q' \models B$

  $\wedge$

  $\forall q \in Q.\ q \models A \Rightarrow \exists q' \in Q.\ q \xrightarrow{c} q' \wedge q' \models B$

Inferring Validity of Assertions

- Now we have the formal mechanism to decide when $\{A\}\ c\ \{B\}$
  - But it is not satisfactory,
  - because $\vDash \{A\}\ c\ \{B\}$ is defined in terms of the operational semantics.
  - We practically have to run the program to verify an assertion.
  - Also it is impossible to effectively verify the truth of a $\forall x.\ A$ assertion (by using the definition of validity)
- So we define a symbolic technique for deriving valid assertions from others that are known to be valid
  - We start with validity of first-order formulas

Inference Rules

- We write $\vdash A$ when A can be inferred from basic axioms.
- The inference rules for $\vdash A$ are the usual ones from first-order logic with arithmetic (more on this later).
- Natural deduction style rules:

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B} \qquad \frac{\vdash A[a/x]}{\vdash \forall\, x.\, A} \text{ where } a \text{ is fresh} \qquad \frac{\vdash \forall\, x.\, A}{\vdash A[e/x]}$$

$$\frac{\vdash A}{\vdash A \vee B} \qquad \frac{\vdash B}{\vdash A \vee B}$$

$$\frac{\vdash A[e/x]}{\vdash \exists\, x.\, A} \qquad \frac{\vdash \exists\, x.\, A \qquad \begin{array}{c}\vdash A[a/x]\\ \cdots\end{array}\ \vdash B}{\vdash B} \text{ where } a \text{ is fresh} \qquad \frac{\vdash A \Rightarrow B \qquad \vdash A}{\vdash B} \qquad \frac{\begin{array}{c}\vdash A\\ \cdots\end{array}\ \vdash B}{\vdash A \Rightarrow B}$$

Inference Rules for Hoare Triples

- Similarly we write $\vdash \{A\}\ c\ \{B\}$ when we can derive the triple using inference rules
- There is one inference rule for each command in the language.
- Plus, the rule of consequence

$$\frac{\vdash A' \Rightarrow A \qquad \vdash \{A\}\ c\ \{B\} \qquad \vdash B \Rightarrow B'}{\vdash \{A'\}\ c\ \{B'\}}$$

Inference Rules for Hoare Logic

- One rule for each syntactic construct:

$$\vdash \{A\}\ \texttt{skip}\ \{A\} \qquad\qquad \vdash \{A[e/x]\}\ x{:=}e\ \{A\}$$

$$\frac{\vdash \{A\}\ c_1\ \{B\} \quad \vdash \{B\}\ c_2\ \{C\}}{\vdash \{A\}\ c_1; c_2\ \{C\}}$$

Inference Rules for Hoare Logic

- One rule for each syntactic construct:

$$\vdash \{A\}\ \texttt{skip}\ \{A\} \qquad\qquad \vdash \{A[e/x]\}\ x{:=}e\ \{A\}$$

$$\frac{\vdash \{A\}\ c_1\ \{B\} \quad \vdash \{B\}\ c_2\ \{C\}}{\vdash \{A\}\ c_1; c_2\ \{C\}}$$

$$\frac{\vdash \{A \wedge b\}\ c_1\ \{B\} \quad \vdash \{A \wedge \neg b\}\ c_2\ \{B\}}{\vdash \{A\}\ \texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \{B\}}$$

$$\frac{\vdash \{I \wedge b\}\ c\ \{I\}}{\vdash \{I\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{I \wedge \neg b\}}$$

Exercise: Hoare Rules

- Is the following alternative rule for assignment still correct?

$$\vdash \{\text{true}\}\ x := e\ \{x = e\}$$

Hoare Rules

- For some constructs, multiple rules are possible

  alternative "forward axiom" for assignment:

  $$\vdash \{A\}\ x := e\ \{\exists x_0.\ x_0 = e \wedge A[x_0/x]\}$$

  alternative rule for `while` loops:

  $$\frac{\vdash I \wedge b \Rightarrow C \qquad \vdash \{C\}\ c\ \{I\} \qquad \vdash I \wedge \neg b \Rightarrow B}{\vdash \{I\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}$$

- These alternative rules are derivable from the previous rules, plus the rule of consequence.

Example: Conditional

$$\frac{}{\vdash \{\text{true}\}\ \texttt{if}\ y \le 0\ \texttt{then}\ x := 1\ \texttt{else}\ x := y\ \{x > 0\}}$$

Example: Conditional

$$\frac{D1 :: \vdash \{\text{true} \wedge y \leq 0\}\ x := 1\ \{x > 0\} \qquad D2 :: \vdash \{\text{true} \wedge y > 0\}\ x := y\ \{x > 0\}}{\vdash \{\text{true}\}\ \texttt{if}\ y \leq 0\ \texttt{then}\ x := 1\ \texttt{else}\ x := y\ \{x > 0\}}$$

Example: Conditional

$$\frac{\begin{array}{c}D1 :: \vdash \{true \wedge y \leq 0\}\ x := 1\ \{x > 0\} \\ D2 :: \vdash \{true \wedge y > 0\}\ x := y\ \{x > 0\}\end{array}}{\vdash \{true\}\ \texttt{if}\ y \leq 0\ \texttt{then}\ x := 1\ \texttt{else}\ x := y\ \{x > 0\}}$$

- D1 is obtained by consequence and assignment

$$\frac{\begin{array}{c}\vdash \{1 > 0\}\ x := 1\ \{x > 0\} \\ \vdash true \wedge y \leq 0 \Rightarrow 1 > 0\end{array}}{\vdash \{true \wedge y \leq 0\}\ x := 1\ \{x \geq 0\}}$$

Example: Conditional

$$\frac{D1 :: \vdash \{true \wedge y \leq 0\}\ x := 1\ \{x > 0\} \qquad D2 :: \vdash \{true \wedge y > 0\}\ x := y\ \{x > 0\}}{\vdash \{true\}\ \texttt{if}\ y \leq 0\ \texttt{then}\ x := 1\ \texttt{else}\ x := y\ \{x > 0\}}$$

- D1 is obtained by consequence and assignment

$$\frac{\vdash \{1 > 0\}\ x := 1\ \{x > 0\} \qquad \vdash true \wedge y \leq 0 \Rightarrow 1 > 0}{\vdash \{true \wedge y \leq 0\}\ x := 1\ \{x \geq 0\}}$$

- D2 is also obtained by consequence and assignment

$$\frac{\vdash \{y > 0\}\ x := y\ \{x > 0\} \qquad \vdash true \wedge y > 0 \Rightarrow y > 0}{\vdash \{true \wedge y > 0\}\ x := y\ \{x > 0\}}$$

Example: A Proof in Hoare Logic

We want to derive that

$\{n \geq 0\}$
p := 0;
x := 0;
`while` *x* < *n* `do`
    x := *x* + 1;
    p := *p* + *m*
$\{p = n * m\}$

is a valid Hoare triple.

Example: A Proof in Hoare Logic

$\vdash \{n \geq 0\}$ p:=0; x:=0; while x < n do (x:=x+1; p:=p+m) $\{p = n * m\}$

Example: A Proof in Hoare Logic

Only applicable rule (except for rule of consequence):

$$\frac{\vdash \{A\}\ c_1\{C\} \quad \vdash \{C\}\ c_2\ \{B\}}{\vdash \{A\}\ c_1;\ c_2\ \{B\}}$$

$\vdash$ $\underbrace{\{n \geq 0\}}_{A}$ $\underbrace{p:=0;\ x:=0;}_{c_1}$ $\underbrace{\text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)}_{c_2}$ $\underbrace{\{p = n * m\}}_{B}$

Example: A Proof in Hoare Logic

Only applicable rule (except for rule of consequence):

$$\frac{\vdash \{A\}\ c_1\{C\} \quad \vdash \{C\}\ c_2\ \{B\}}{\vdash \{A\}\ c_1;\ c_2\ \{B\}}$$

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{C\} \quad \vdash \{C\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}{\vdash \underbrace{\{n \geq 0\}}_{A}\ \underbrace{p:=0;\ x:=0;}_{c_1}\ \underbrace{\text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)}_{c_2}\ \underbrace{\{p = n * m\}}_{B}}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{C\} \qquad \vdash \{C\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}{\vdash \underbrace{\{n \geq 0\}}_{A}\ \underbrace{p:=0;\ x:=0;}_{c_1}\ \underbrace{\text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)}_{c_2}\ \underbrace{\{p = n * m\}}_{B}}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

Only applicable rule (except for rule of consequence):

$$\frac{\vdash \{I \wedge b\}\ c\ \{I\}}{\vdash \{I\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{I \wedge \neg b\}}$$

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{C\} \quad \vdash \{C\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

Only applicable rule (except for rule of consequence):

$$\frac{\vdash \{I \wedge b\}\ c\ \{I\}}{\vdash \{I\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{I \wedge \neg b\}}$$

We can match {I} with {C} but we cannot match {I ∧ ¬b} and {p = n * m} directly. Need to apply the rule of consequence first!

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{C\} \quad \vdash \{C\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

Only applicable rule (except for rule of consequence):

$$\frac{\vdash \{I \wedge b\}\ c\ \{I\}}{\vdash \underbrace{\{I\}}_{A}\ \underbrace{\texttt{while}\ b\ \texttt{do}\ c}_{c'}\ \underbrace{\{I \wedge \neg b\}}_{B}}$$

$I = A = A' = C$

Rule of consequence:

$$\frac{\vdash A' \Rightarrow A \quad \vdash \{A\}\ c'\ \{B\} \quad \vdash B \Rightarrow B'}{\vdash \{A'\}\ c'\ \{B'\}}$$

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{C\} \quad \vdash \{C\}\ \underbrace{\text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)}_{c'}\ \underbrace{\{p = n * m\}}_{B'}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

(A' = $\{C\}$)

Example: A Proof in Hoare Logic

What is I? Let's keep it as a placeholder for now!

$$\dfrac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \qquad \dfrac{\dfrac{\vdash \{I \wedge x<n\}\ x := x+1;\ p:=p+m\ \{I\}}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \qquad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

What is I? Let's keep it as a placeholder for now!

Next applicable rule:

$$\frac{\vdash \{A\}\ c_1\ \{C\} \quad \vdash \{C\}\ c_2\ \{B\}}{\vdash \{A\}\ c_1;\ c_2\ \{B\}}$$

$$\frac{\dfrac{\vdash \{\overbrace{I \wedge x<n}^{A}\}\ \overbrace{x := x+1}^{c_1};\ \overbrace{p:=p+m}^{c_2}\ \{\overbrace{I}^{B}\}}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \quad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \quad \vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

$$\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}$$

Example: A Proof in Hoare Logic

$$\dfrac{\dfrac{\dfrac{\overbrace{\vdash\{I \wedge x<n\}}^{A}\ \overbrace{x := x+1}^{c_1}\ \{C\} \qquad \overbrace{\vdash\{C\}}^{c_2}\ p:=p+m\ \overbrace{\{I\}}^{B}}{\vdash\{I \wedge x<n\}\ x := x+1;\ p:=p+m\ \{I\}}}{\vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \qquad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \qquad \vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

$$\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

Only applicable rule (except for rule of consequence):

$\vdash \{A[e/x]\}\ x{:=}e\ \{A\}$

$$\dfrac{\dfrac{\dfrac{\vdash \overbrace{\{I \wedge x<n\}}^{A}\ \overbrace{x := x+1}^{c_1}\ \{C\} \qquad \vdash \overbrace{\{C\}}^{c_2}\ p:=p+m\ \overbrace{\{I\}}^{B}}{\vdash \{I \wedge x<n\}\ x := x+1;\ p:=p+m\ \{I\}}}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \qquad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \qquad \vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

$$\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}$$

Example: A Proof in Hoare Logic

What is C? Look at the next possible matching rules for c_2!

Only applicable rule (except for rule of consequence):

$\vdash \{A[e/x]\}\ x:=e\ \{A\}$

$$\dfrac{\dfrac{\dfrac{\vdash\{I \wedge x<n\}\ x:=x+1\ \{I[p+m/p]\} \quad \vdash\{I[p+m/p\}\ p:=p+m\ \{I\}}{\vdash\{I \wedge x<n\}\ x:=x+1;\ p:=p+m\ \{I\}}}{\vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \quad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \quad \vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \{I \wedge x<n\}\ x:=x+1\ \{I[p+m/p]\} \quad \vdash \{I[p+m/p\}\ p:=p+m\ \{I\}}{\vdash \{I \wedge x<n\}\ x:=x+1;\ p:=p+m\ \{I\}}}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \quad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \quad \vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

Only applicable rule (except for rule of consequence):

$\vdash \{A[e/x]\}\ x{:=}e\ \{A\}$

Need rule of consequence to match $\{I \wedge x<n\}$ and $\{I[x+1/x,\ p+m/p]\}$

$$\dfrac{\dfrac{\dfrac{\vdash\{I \wedge x<n\}\ x:=x+1\ \{I[p+m/p]\} \quad \vdash\{I[p+m/p]\}\ p:=p+m\ \{I\}}{\vdash\{I \wedge x<n\}\ x:=x+1;\ p:=p+m\ \{I\}}}{\vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \quad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \quad \vdash\{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

$$\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}$$

Example: A Proof in Hoare Logic

Let's just remember the open proof obligations!

$$\dfrac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{I\} \qquad \dfrac{\dfrac{\vdash \{I[x+1/x,\ p+m/p]\}\ x:=x+1\ \{I[p+m/p]\} \qquad \vdash I \wedge x < n \Rightarrow I[x+1/x,\ p+m/p]}{\vdash \{I \wedge x<n\}\ x:=x+1\ \{I[p+m/p]\} \qquad \vdash \{I[p+m/p]\}\ p:=p+m\ \{I\}}}{\vdash \{I \wedge x<n\}\ x:=x+1;\ p:=p+m\ \{I\}}}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{I \wedge x \geq n\}} \qquad \vdash I \wedge x \geq n \Rightarrow p = n * m}{\vdash \{I\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

Let's just remember the open proof obligations!

$$\vdash \mathtt{I} \wedge x < n \Rightarrow \mathtt{I}[x+1/x, p+m/p]$$

$$\vdash \mathtt{I} \wedge x \geq n \Rightarrow p = n * m$$

$$\frac{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{\mathtt{I}\} \qquad \dfrac{\vdots}{\vdash \{\mathtt{I}\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

Let's just remember the open proof obligations!

$$\vdash \mathtt{I} \wedge x < n \Rightarrow \mathtt{I}[x+1/x,\ p+m/p]$$

$$\vdash \mathtt{I} \wedge x \geq n \Rightarrow p = n * m$$

Continue with the remaining part of the proof tree, as before.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash n \geq 0 \Rightarrow \mathtt{I}[0/p,\ 0/x] \quad \vdash \{\mathtt{I}[0/p,\ 0/x]\}\ p:=0\ \{\mathtt{I}[0/x]\}}{\vdash \{n \geq 0\}\ p:=0\ \{\mathtt{I}[0/x]\}} \quad \vdash \{\mathtt{I}[0/x]\}\ x:=0\ \{\mathtt{I}\}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{\mathtt{I}\}} \quad \dfrac{\vdots}{\vdash \{\mathtt{I}\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Example: A Proof in Hoare Logic

Let's just remember the open proof obligations!

$\vdash \mathtt{I} \wedge x < n \Rightarrow \mathtt{I}[x+1/x, p+m/p]$

$\vdash \mathtt{I} \wedge x \geq n \Rightarrow p = n * m$

Continue with the remaining part of the proof tree, as before.

$$\dfrac{\dfrac{\dfrac{\vdash n \geq 0 \Rightarrow \mathtt{I}[0/p, 0/x] \quad \vdash \{\mathtt{I}[0/p, 0/x]\}\ p:=0\ \{\mathtt{I}[0/x]\}}{\vdash \{n \geq 0\}\ p:=0\ \{\mathtt{I}[0/x]\}} \quad \vdash \{\mathtt{I}[0/x]\}\ x:=0\ \{\mathtt{I}\}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0\ \{\mathtt{I}\}} \quad \dfrac{\vdots}{\vdash \{\mathtt{I}\}\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}}{\vdash \{n \geq 0\}\ p:=0;\ x:=0;\ \text{while } x < n \text{ do } (x:=x+1;\ p:=p+m)\ \{p = n * m\}}$$

Now we only need to solve the remaining constraints!

Example: A Proof in Hoare Logic

Find $\mathtt{I}$ such that all constraints are simultaneously valid:

$\vdash n \geq 0 \Rightarrow \mathtt{I}[0/p, 0/x]$

$\vdash \mathtt{I} \wedge x < n \Rightarrow \mathtt{I}[x+1/x, p+m/p]$

$\vdash \mathtt{I} \wedge x \geq n \Rightarrow p = n * m$

Example: A Proof in Hoare Logic

Find I such that all constraints are simultaneously valid:

$\vdash n \geq 0 \Rightarrow I[0/p, 0/x]$

$\vdash I \wedge x < n \Rightarrow I[x+1/x, p+m/p]$

$\vdash I \wedge x \geq n \Rightarrow p = n * m$

$I \equiv p = x * m \wedge x \leq n$

$\vdash n \geq 0 \Rightarrow 0 = 0 * m \wedge 0 \leq n$

$\vdash p = x * m \wedge x \leq n \wedge x < n \Rightarrow p+m = (x+1) * m \wedge x+1 \leq n$

$\vdash p = x * m \wedge x \leq n \wedge x \geq n \Rightarrow p = n * m$

All constraints are valid!

Using Hoare Rules

- Hoare rules are mostly syntax directed
- There are three obstacles to automation of Hoare logic proofs:
  - When to apply the rule of consequence?
  - What invariant to use for `while`?
  - How do you prove the implications involved in the rule of consequence?
- The last one is how theorem proving gets in the picture
  - This turns out to be doable!
  - The loop invariants turn out to be the hardest problem!
  - Should the programmer give them?

Hoare Logic: Summary

- We have a language for asserting properties of programs.
- We know when such an assertion is true.
- We also have a symbolic method for deriving assertions.