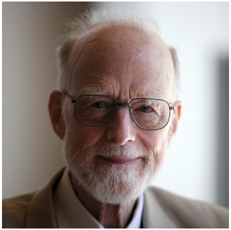
COMP2111 Week 8/9
Term 1, 2024
Hoare Logic

Sir Tony Hoare

- Pioneer in formal verification
- Invented: Quicksort,
- the null reference (called it his “billion dollar mistake”)
- CSP (formal specification language), and
- Hoare Logic



Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Imperative Programming

imperō

Definition

Imperative programming is where programs are described as a series of *statements* or commands to manipulate mutable *state* or cause externally observable *effects*.

States may take the form of a *mapping* from variable names to their values, or even a model of a CPU state with a memory model (for example, in an *assembly language*).

$\mathcal{L}$: A simple imperative programming language

Consider the vocabulary of basic arithmetic:

- Constant symbols: $0, 1, 2, \ldots$
- Function symbols: $+, *, \ldots$
- Predicate symbols: $<, \leq, \geq, |, \ldots$

$\mathcal{L}$: A simple imperative programming language

Consider the vocabulary of basic arithmetic:

- Constant symbols: $0, 1, 2, \ldots$
- Function symbols: $+, *, \ldots$
- Predicate symbols: $<, \leq, \geq, |, \ldots$
- An **(arithmetic) expression** is a term over this vocabulary.

$\mathcal{L}$: A simple imperative programming language

Consider the vocabulary of basic arithmetic:

- Constant symbols: $0, 1, 2, \ldots$
- Function symbols: $+, *, \ldots$
- Predicate symbols: $<, \leq, \geq, |, \ldots$

- An **(arithmetic) expression** is a term over this vocabulary.
- A **boolean expression** is a predicate formula over this vocabulary.

The language $\mathcal{L}$

The language $\mathcal{L}$ is a simple imperative programming language made up of four statements:

Assignment: $x := e$

where x is a variable and e is an arithmetic expression.

The language $\mathcal{L}$

The language $\mathcal{L}$ is a simple imperative programming language made up of four statements:

Assignment: $x{:=}e$

where x is a variable and e is an arithmetic expression.

Sequencing: $P;Q$

The language $\mathcal{L}$

The language $\mathcal{L}$ is a simple imperative programming language made up of four statements:

Assignment: $x{:=}e$

where x is a variable and e is an arithmetic expression.

Sequencing: $P;Q$

Conditional: if g then P else Q fi

where g is a boolean expression.

The language $\mathcal{L}$

The language $\mathcal{L}$ is a simple imperative programming language made up of four statements:

Assignment: $x{:=}e$
where x is a variable and e is an arithmetic expression.

Sequencing: $P;Q$

Conditional: if g then P else Q fi
where g is a boolean expression.

While: while g do P od

Factorial in $\mathcal{L}$

Example

$i := 0;$
$m := 1;$
while $i < N$ do
    $i := i + 1;$
    $m := m * i$
od

Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Hoare Logic

We are going to define what's called a *Hoare Logic* for $\mathcal{L}$ to allow us to prove properties of our program.
We write a *Hoare triple* judgement as:

$$\{\varphi\}\ P\ \{\psi\}$$

Where φ and ψ are logical formulae about states, called *assertions*, and P is a program. This triple states that if the program P terminates and it successfully evaluates from a starting state satisfying the *precondition* φ, then the result state will satisfy the *postcondition* ψ.

Hoare triple: Examples

Example

$$\{(x = 0)\}\, x := 1\, \{(x = 1)\}$$

Hoare triple: Examples

Example

$$\{(x = 0)\}\, x := 1\, \{(x = 1)\}$$

$$\{(x = 499)\}\, x := x + 1\, \{(x = 500)\}$$

Hoare triple: Examples

Example

$$\{(x = 0)\}\, x := 1\, \{(x = 1)\}$$

$$\{(x = 499)\}\, x := x + 1\, \{(x = 500)\}$$

$$\{(x > 0)\}\, y := 0 - x\, \{(y < 0) \wedge (x \neq y)\}$$

Hoare triple: Factorial Examples

Example

$$\{N \geq 0\}$$
$$i := 0;$$
$$m := 1;$$
$$\text{while } i < N \text{ do}$$
$$\quad i := i + 1;$$
$$\quad m := m * i$$
$$\text{od}$$
$$\{m = N!\}$$

Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Motivation

Question

We know what we want informally; how do we establish when a triple is valid?

Motivation

Question

We know what we want informally; how do we establish when a triple is valid?

- Develop a semantics, OR

Hoare logic consists of one axiom and four inference rules for deriving Hoare triples.

Motivation

Question

We know what we want informally; how do we establish when a triple is valid?

- Develop a semantics, OR
- Derive the triple in a syntactic manner (i.e. Hoare proof)

Hoare logic consists of one axiom and four inference rules for deriving Hoare triples.

Assignment

$$\frac{}{\{\varphi[e/x]\}\; x := e\; \{\varphi\}} \quad \text{(assign)}$$

Intuition:
If x has property φ *after* executing the assignment; then e must have property φ *before* executing the assignment

Assignment: Example

Example

$$\{(y = 0)\}\, x := y\, \{(x = 0)\}$$

Assignment: Example

Example

$$\{(y = 0)\}\, x := y\, \{(x = 0)\}$$

$$\{ \qquad \qquad \}\, x := y\, \{(x = y)\}$$

Assignment: Example

Example

$$\{(y = 0)\}\, x := y\, \{(x = 0)\}$$

$$\{(y = y)\}\, x := y\, \{(x = y)\}$$

Assignment: Example

Example

$$\{(y = 0)\}\, x := y\, \{(x = 0)\}$$

$$\{(y = y)\}\, x := y\, \{(x = y)\}$$

$$\{\qquad\qquad\}\, x := 1\, \{(x < 2)\}$$

Assignment: Example

Example

$$\{(y = 0)\}\; x := y\; \{(x = 0)\}$$

$$\{(y = y)\}\; x := y\; \{(x = y)\}$$

$$\{(1 < 2)\}\; x := 1\; \{(x < 2)\}$$

$$\{(y = 3)\}\; x := y\; \{(x > 2)\}$$

Assignment: Example

Example

$$\{(y = 0)\}\, x := y\, \{(x = 0)\}$$

$$\{(y = y)\}\, x := y\, \{(x = y)\}$$

$$\{(1 < 2)\}\, x := 1\, \{(x < 2)\}$$

$$\{(y = 3)\}\, x := y\, \{(x > 2)\}$$ *Problem*!

Sequence

$$\frac{\{\varphi\}\ P\ \{\psi\} \qquad \{\psi\}\ Q\ \{\rho\}}{\{\varphi\}\ P; Q\ \{\rho\}} \quad \text{(seq)}$$

Intuition:
If the postcondition of P matches the precondition of Q we can sequentially combine the two program fragments

Sequence: Example

Example

$$\frac{\{\qquad\}\, x := 0\, \{\qquad\} \qquad \{\qquad\}\, y := 0\, \{(x = y)\}}{\{\qquad\}\, x := 0;\, y := 0\, \{(x = y)\}} \quad (\text{seq})$$

Sequence: Example

Example

$$\frac{\{\qquad\qquad\}\, x := 0 \,\{(x = 0)\} \qquad \{(x = 0)\}\, y := 0 \,\{(x = y)\}}{\{\qquad\qquad\}\, x := 0; y := 0 \,\{(x = y)\}} \ (\text{seq})$$

Sequence: Example

Example

$$\frac{\{(0=0)\}\; x := 0 \;\{(x=0)\} \qquad \{(x=0)\}\; y := 0 \;\{(x=y)\}}{\{(0=0)\}\; x := 0;\, y := 0 \;\{(x=y)\}} \quad (\text{seq})$$

Conditional

$$\frac{\{\varphi \wedge g\}\, P\, \{\psi\} \qquad \{\varphi \wedge \neg g\}\, Q\, \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Intuition:

- When a conditional is executed, either P or Q will be executed.
- If ψ is a postcondition of the conditional, then it must be a postcondition of *both* branches
- Likewise, if φ is a precondition of the conditional, then it must be a precondition of both branches
- Which branch gets executed depends on g, so we can assume g to be a precondition of P and $\neg g$ to be a precondition of Q.

While

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Intuition:

- φ is a **loop invariant**. It must be both a pre- and postcondition of P, so that sequences of Ps can be run together.
- If the while loop terminates, g cannot hold.

Consequence

There is one more rule, called the *rule of consequence*, that we need to insert ordinary logical reasoning into our Hoare logic proofs:

$$\frac{\varphi' \to \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \to \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Consequence

There is one more rule, called the *rule of consequence*, that we need to insert ordinary logical reasoning into our Hoare logic proofs:

$$\frac{\varphi' \rightarrow \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \rightarrow \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Intuition:

- Adding assertions to the precondition makes it more likely the postcondition will be reached
- Removing assertions from the postcondition makes it more likely the postcondition will be reached
- If you can reach the postcondition initially, then you can reach it in the more likely scenario

Back to Assignment Example

Example

$$\{(y = 3)\}\; x := y \;\{(x > 2)\} \qquad \textit{Problem!}$$

Back to Assignment Example

Example

$\{(y = 3)\}\; x := y\; \{(x > 2)\}$ *Problem*!

$$\{(y > 2)\}x := y\{(x > 2)\}(assign)$$

Back to Assignment Example

Example

$\{(y = 3)\}\, x := y\, \{(x > 2)\}$ *Problem*!

$$\{(y = 3)\}x := y\{(x > 2)\}(assign, cons)$$
$$\{(y > 2)\}x := y\{(x > 2)\}(assign)$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$

$i := 0;$

$m := 1;$

while $i \neq N$ do

$i := i + 1;$

$m := m \times i$

od

$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \qquad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$

$i := 0;$

$m := 1;$

while $i \neq N$ do

$i := i + 1;$

$m := m \times i$

od $\{m = i! \wedge N \geq 0 \wedge i = N\}$

$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0;$
$m := 1;$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do

$i := i + 1;$

$m := m \times i$

od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \qquad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$

$i := 0;$

$m := 1;$

$\{m = i! \land N \geq 0\}$

while $i \neq N$ do

$i := i + 1;$

$m := m \times i$

$\{m = i! \land N \geq 0\}$

od $\{m = i! \land N \geq 0 \land i = N\}$

$\{m = N!\}$

$$\frac{\{\varphi \land g\}\ P\ \{\psi\} \quad \{\varphi \land \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \land g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \land \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0;$
$m := 1;$
$\{m = i! \land N \geq 0\}$
while $i \neq N$ do $\{m = i! \land N \geq 0 \land iN\}$

$i := i + 1;$

$m := m \times i$
$\{m = i! \land N \geq 0\}$
od $\{m = i! \land N \geq 0 \land i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \land g\}\ P\ \{\psi\} \quad \{\varphi \land \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \land g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \land \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \qquad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0;$
$m := 1;$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$

$i := i + 1;$
$\{m \times i = i! \wedge N \geq 0\}$
$m := m \times i$
$\{m = i! \wedge N \geq 0\}$
od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0;$
$m := 1;$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$
$\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$
$i := i + 1;$
$\{m \times i = i! \wedge N \geq 0\}$
$m := m \times i$
$\{m = i! \wedge N \geq 0\}$
od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0;$
$m := 1;$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$
$\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$
$i := i + 1;$
$\{m \times i = i! \wedge N \geq 0\}$
$m := m \times i$
$\{m = i! \wedge N \geq 0\}$
od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

note: $(i+1)! = i! \times (i+1)$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$

$i := 0;$

$m := 1; \{m = i! \wedge N \geq 0\}$

$\{m = i! \wedge N \geq 0\}$

while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$

$\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$

$i := i + 1;$

$\{m \times i = i! \wedge N \geq 0\}$

$m := m \times i$

$\{m = i! \wedge N \geq 0\}$

od $\{m = i! \wedge N \geq 0 \wedge i = N\}$

$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \mathtt{if}\ g\ \mathtt{then}\ P\ \mathtt{else}\ Q\ \mathtt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \mathtt{while}\ g\ \mathtt{do}\ P\ \mathtt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

note: $(i+1)! = i! \times (i+1)$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$

$i := 0;$

$\{1 = i! \wedge N \geq 0\}$ $m := 1;$ $\{m = i! \wedge N \geq 0\}$

$\{m = i! \wedge N \geq 0\}$

while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$

   $\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$

   $i := i + 1;$

   $\{m \times i = i! \wedge N \geq 0\}$

   $m := m \times i$

   $\{m = i! \wedge N \geq 0\}$

od $\{m = i! \wedge N \geq 0 \wedge i = N\}$

$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \qquad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

note: $(i+1)! = i! \times (i+1)$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$i := 0; \{1 = i! \wedge N \geq 0\}$
$\{1 = i! \wedge N \geq 0\}\ m := 1; \{m = i! \wedge N \geq 0\}$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$
  $\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$
  $i := i + 1;$
  $\{m \times i = i! \wedge N \geq 0\}$
  $m := m \times i$
  $\{m = i! \wedge N \geq 0\}$
od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \qquad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

note: $(i+1)! = i! \times (i+1)$

Factorial Example

Let's verify the Factorial program using our Hoare rules:

$\{N \geq 0\}$
$\{1 = 0! \wedge N \geq 0\}\ i := 0; \{1 = i! \wedge N \geq 0\}$
$\{1 = i! \wedge N \geq 0\}\ m := 1; \{m = i! \wedge N \geq 0\}$
$\{m = i! \wedge N \geq 0\}$
while $i \neq N$ do $\{m = i! \wedge N \geq 0 \wedge iN\}$
  $\{m \times (i+1) = (i+1)! \wedge N \geq 0\}$
  $i := i + 1;$
  $\{m \times i = i! \wedge N \geq 0\}$
  $m := m \times i$
  $\{m = i! \wedge N \geq 0\}$
od $\{m = i! \wedge N \geq 0 \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \quad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ g\ \texttt{then}\ P\ \texttt{else}\ Q\ \texttt{fi}\ \{\psi\}}$$

$$\frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ g\ \texttt{do}\ P\ \texttt{od}\ \{\varphi \wedge \neg g\}}$$

$$\frac{\{\varphi\}\ P\ \{\alpha\} \quad \{\alpha\}\ Q\ \{\psi\}}{\{\varphi\}\ P; Q\ \{\psi\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

note: $(i+1)! = i! \times (i+1)$

Practice Exercise

Example

$m := 1;$
$n := 1;$
$i := 1;$
while $i < N$ do
    $t := m;$
    $m := n;$
    $n := m + t;$
    $i := i + 1$
od

Practice Exercise

Example

$$
\begin{array}{l}
m := 1; \\
n := 1; \\
i := 1; \\
\text{while } i < N \text{ do} \\
\quad t := m; \\
\quad m := n; \\
\quad n := m + t; \\
\quad i := i + 1 \\
\text{od}
\end{array}
$$

- What does this $\mathcal{L}$ program P compute?
- What is a valid Hoare triple $\{\varphi\}P\{\psi\}$ of this program?
- Prove using the inference rules and consequence axiom that this Hoare triple is valid.

Summary

- $\mathcal{L}$: A simple imperative programming language
- Hoare triples (SYNTAX)
- Hoare logic (PROOF)
- Semantics for Hoare logic

Recall

If R and S are binary relations, then the **relational composition** of R and S, $R;S$ is the relation:

$$R;S := \{(a,c) \ : \ \exists b \text{ such that } (a,b) \in R \text{ and } (b,c) \in S\}$$

If $R \subseteq A \times B$ is a relation, and $X \subseteq A$, then the **image of X under R**, $R(X)$ is the subset of B defined as:

$$R(X) := \{b \in B \ : \ \exists a\, inX \text{ such that } (a,b) \in R\}.$$

Informal semantics

Hoare logic gives a proof of $\{\varphi\}\, P\, \{\psi\}$, that is: $\vdash \{\varphi\}\, P\, \{\psi\}$ (axiomatic semantics)

How do we determine when $\{\varphi\}\, P\, \{\psi\}$ is **valid**, that is: $\models \{\varphi\}\, P\, \{\psi\}$?

Informal semantics

Hoare logic gives a proof of $\{\varphi\}\, P\, \{\psi\}$, that is: $\vdash \{\varphi\}\, P\, \{\psi\}$ (axiomatic semantics)

How do we determine when $\{\varphi\}\, P\, \{\psi\}$ is **valid**, that is: $\models \{\varphi\}\, P\, \{\psi\}$?

If φ holds in a state of some computational model
then ψ holds in the state reached after a successful execution of P.

Informal semantics: Programs

What is a program?

Informal semantics: Programs

What is a program?

A function mapping system states to system states

Informal semantics: Programs

What is a program?

A partial function mapping system states to system states

Informal semantics: Programs

What is a program?

A relation between system states

Informal semantics: States

What is a state of a computational model?

Informal semantics: States

What is a state of a computational model?

Two approaches:

- Concrete: from a physical perspective

- Abstract: from a mathematical perspective

Informal semantics: States

What is a state of a computational model?

Two approaches:

- Concrete: from a physical perspective
  - States are memory configurations, register contents, etc.
  - Store of variables and the values associated with them
- Abstract: from a mathematical perspective

Informal semantics: States

What is a state of a computational model?

Two approaches:

- Concrete: from a physical perspective
  - States are memory configurations, register contents, etc.
  - Store of variables and the values associated with them
- Abstract: from a mathematical perspective
  - The pre-/postcondition predicates *hold* in a state
  - $\Rightarrow$ States are **logical interpretations** (Model + Environment)

Informal semantics: States

What is a state of a computational model?

Two approaches:

- Concrete: from a physical perspective
  - States are memory configurations, register contents, etc.
  - Store of variables and the values associated with them
- Abstract: from a mathematical perspective
  - The pre-/postcondition predicates *hold* in a state
  - ⇒ States are **logical interpretations** (Model + Environment)
  - There is only one model of interest: standard interpretations of arithmetical symbols

Informal semantics: States

What is a state of a computational model?

Two approaches:

- Concrete: from a physical perspective
  - States are memory configurations, register contents, etc.
  - Store of variables and the values associated with them
- Abstract: from a mathematical perspective
  - The pre-/postcondition predicates *hold* in a state
  - $\Rightarrow$ States are **logical interpretations** (Model + Environment)
  - There is only one model of interest: standard interpretations of arithmetical symbols
  - $\Rightarrow$ States are fully determined by **environments**
  - $\Rightarrow$ States are functions that map variables to values

Informal semantics: States

State space (ENV)

$x \leftarrow 0$
$y \leftarrow 0$
$z \leftarrow 0$

$x \leftarrow 3$
$y \leftarrow 2$
$z \leftarrow 1$

$x \leftarrow 1$
$y \leftarrow 1$
$z \leftarrow 1$

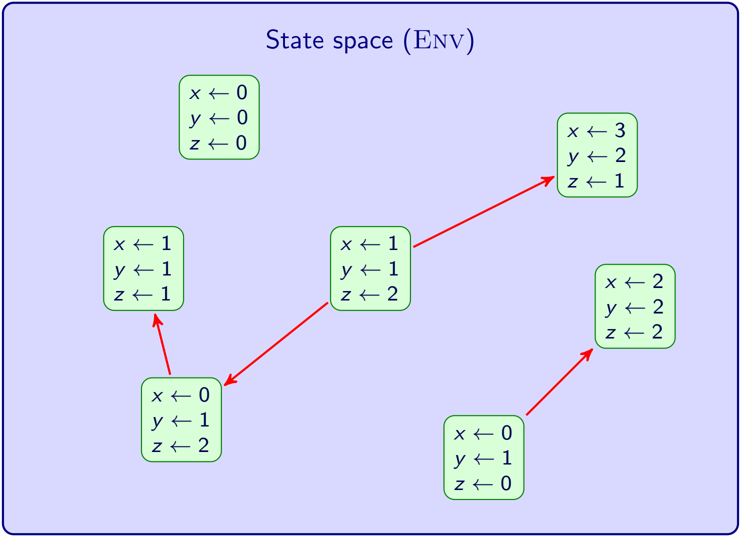
$x \leftarrow 1$
$y \leftarrow 1$
$z \leftarrow 2$

$x \leftarrow 2$
$y \leftarrow 2$
$z \leftarrow 2$

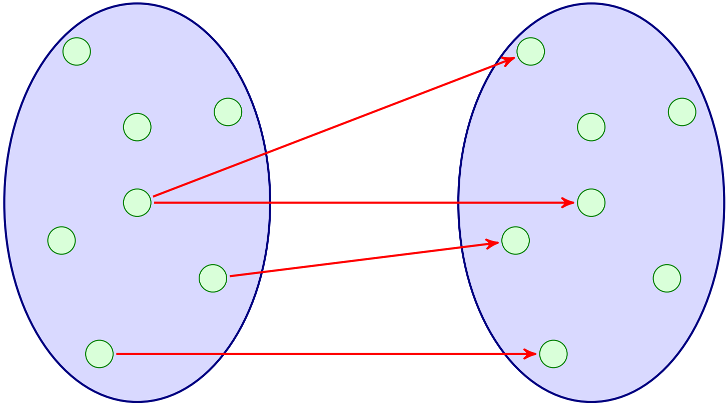
$x \leftarrow 0$
$y \leftarrow 1$
$z \leftarrow 2$

$x \leftarrow 0$
$y \leftarrow 1$
$z \leftarrow 0$

Informal semantics: States and Programs



Informal semantics: States and Programs



Semantics for $\mathcal{L}$

An **environment** or **state** is a function from variables to numeric values. We denote by $\textsc{Env}$ the set of all environments.

NB

An environment, η, assigns a numeric value $[\![e]\!]^\eta$ to all expressions e, and a boolean value $[\![b]\!]^\eta$ to all boolean expressions b.

Semantics for $\mathcal{L}$

An **environment** or **state** is a function from variables to numeric values. We denote by $\textsc{Env}$ the set of all environments.

NB

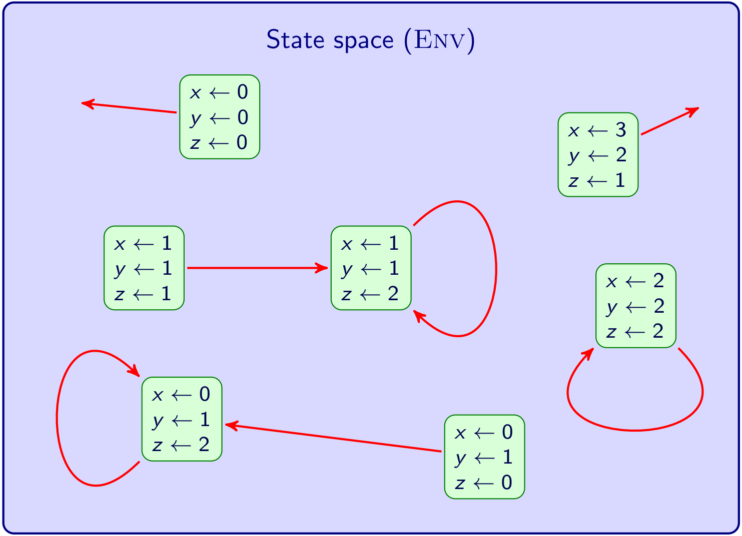
An environment, η, assigns a numeric value $[\![e]\!]^{\eta}$ to all expressions e, and a boolean value $[\![b]\!]^{\eta}$ to all boolean expressions b.

Given a program P of $\mathcal{L}$, we define $[\![P]\!]$ to be a **binary relation** on $\textsc{Env}$ in the following manner...

Assignment

$$(\eta, \eta') \in [\![x := e]\!] \quad \text{if, and only if} \quad \eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$$

Assignment: $[\![z := 2]\!]$



Sequencing

$$[\![P; Q]\!] = [\![P]\!] ; [\![Q]\!]$$

where, on the RHS, ; is relational composition.

Conditional, first attempt

$$[\![\mathsf{if}\ b\ \mathsf{then}\ P\ \mathsf{else}\ Q\ \mathsf{fi}]\!] = \begin{cases} [\![P]\!] & \mathsf{if}\ [\![b]\!]^{\eta} = \mathtt{true} \\ [\![Q]\!] & \mathsf{otherwise.} \end{cases}$$

Detour: Predicates as programs

A boolean expression b defines a subset (or unary relation) of $\textsc{Env}$:

$$\langle b \rangle = \{\eta \,:\, [\![b]\!]^{\eta} = \mathtt{true}\}$$

This can be extended to a binary relation (i.e. a program):

$$[\![b]\!] = \{(\eta, \eta) \,:\, \eta \in \langle b \rangle\}$$

Detour: Predicates as programs

A boolean expression b defines a subset (or unary relation) of $\textsc{Env}$:

$$\langle b \rangle = \{\eta \ : \ [\![b]\!]^{\eta} = \mathtt{true}\}$$

This can be extended to a binary relation (i.e. a program):

$$[\![b]\!] = \{(\eta, \eta) \ : \ \eta \in \langle b \rangle\}$$

Intuitively, b corresponds to the program

$$\text{if } b \text{ then skip else } \perp \text{ fi}$$

Conditional, better attempt

$$[\![\text{if } b \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![b; P]\!] \cup [\![\neg b; Q]\!]$$

While

while b do P od

- Do 0 or more executions of P while b holds
- Terminate when b does not hold

While

while b do P od

- Do 0 or more executions of $(b; P)$
- Terminate with an execution of $\neg b$

While

$$\text{while } b \text{ do } P \text{ od}$$

- Do 0 or more executions of $(b; P)$
- Terminate with an execution of $\neg b$

How to do “0 or more” executions of $(b; P)$?

Transitive closure

Given a binary relation $R \subseteq E \times E$, the *transitive closure of* R, R^* is defined to be the limit of the sequence

$$R^0 \cup R^1 \cup R^2 \cdots$$

where

- $R^0 = \Delta$, the diagonal relation
- $R^{n+1} = R^n; R$

NB

- *R^* is the smallest transitive relation which contains R*
- *Related to the Kleene star operation seen in languages: Σ^**

Transitive closure

Given a binary relation $R \subseteq E \times E$, the *transitive closure of* R, R^* is defined to be the limit of the sequence

$$R^0 \cup R^1 \cup R^2 \cdots$$

where

- $R^0 = \Delta$, the diagonal relation
- $R^{n+1} = R^n; R$

NB

- *R^* is the smallest transitive relation which contains R*
- *Related to the Kleene star operation seen in languages: Σ^**

Technically, R^* is the **least-fixed point** of $f(X) = \Delta \cup X; R$

While

$$[\![\text{while } b \text{ do } P \text{ od}]\!] = [\![b; P]\!]^*; [\![\neg b]\!]$$

- Do 0 or more executions of $(b; P)$
- Conclude with an execution of $\neg b$

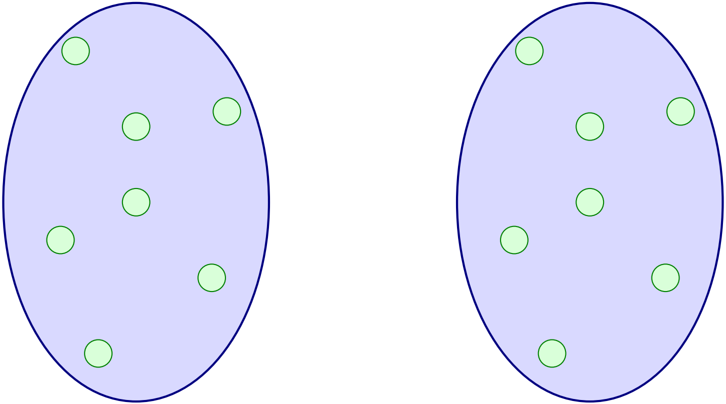
Validity

A Hoare triple is **valid**, written $\models \{\varphi\}\, P\, \{\psi\}$ if

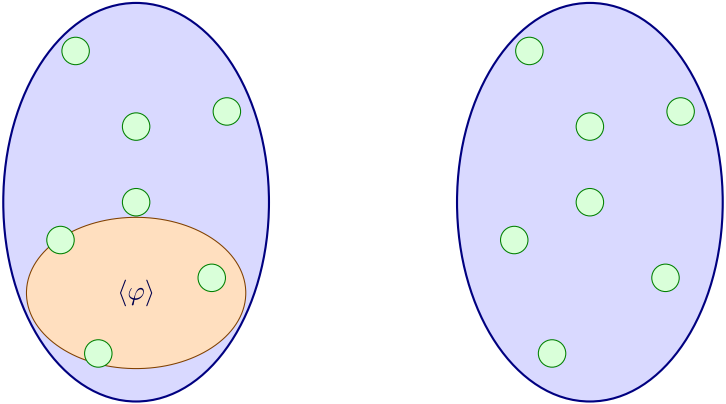
$$[\![P]\!](\langle\varphi\rangle) \subseteq \langle\psi\rangle.$$

That is, the relational image under $[\![P]\!]$ of the set of states where φ holds is contained in the set of states where ψ holds.

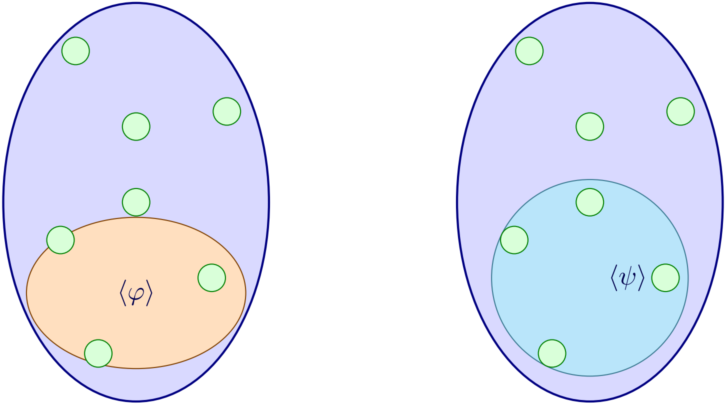
Validity



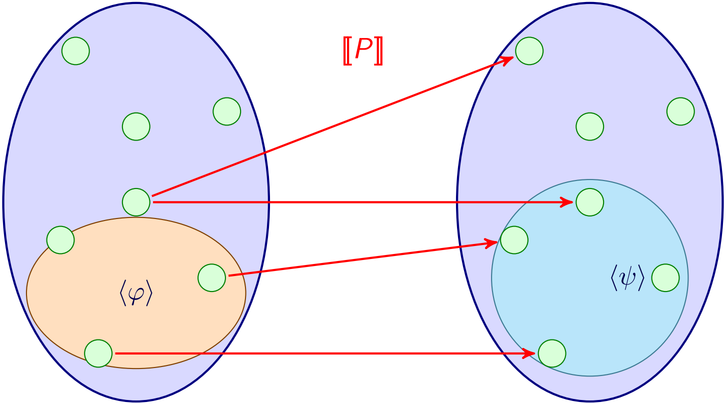
Validity



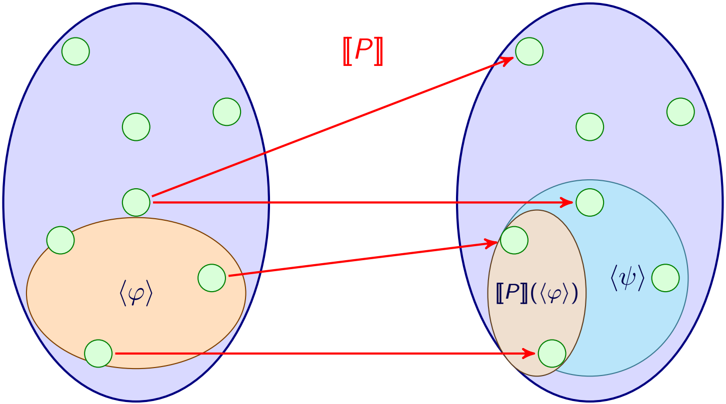
Validity



Validity



Validity



$[\![P]\!]$
$\langle\varphi\rangle$
$[\![P]\!](\langle\varphi\rangle)$
$\langle\psi\rangle$

Soundness of Hoare Logic

Hoare Logic is **sound** with respect to the semantics given. That is,

Theorem

If $\vdash \{\varphi\}\, P\, \{\psi\}$ *then* $\models \{\varphi\}\, P\, \{\psi\}$

Summary

- Set theory revisited
- Soundness of Hoare Logic
- Completeness of Hoare Logic

Summary

- Set theory revisited
- Soundness of Hoare Logic
- Completeness of Hoare Logic

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ *and subsets* $A, B \subseteq X$*:*

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ *and subsets* $A, B \subseteq X$:

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Proof (a):

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ and subsets $A, B \subseteq X$:

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Proof (a):

$$
\begin{aligned}
y \in R(A) &\Leftrightarrow \exists x \in A \text{ such that } (x, y) \in R \\
&\Rightarrow \exists x \in B \text{ such that } (x, y) \in R \\
&\Leftrightarrow y \in R(B)
\end{aligned}
$$

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ *and subsets* $A, B \subseteq X$:

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Proof (b):

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ and subsets $A, B \subseteq X$:

- (a) *If $A \subseteq B$ then $R(A) \subseteq R(B)$*
- (b) $R(A) \cup S(A) = (R \cup S)(A)$
- (c) $R(S(A)) = (S; R)(A)$

Proof (b):

$$
\begin{aligned}
y \in R(A) \cup S(A) \quad & \Leftrightarrow \quad y \in R(A) \text{ or } y \in S(A) \\
& \Leftrightarrow \quad \exists x \in A \text{ s.t. } (x, y) \in R \text{ or } \exists x \in A \text{ s.t. } (x, y) \in S \\
& \Leftrightarrow \quad \exists x \in A \text{ s.t. } (x, y) \in R \text{ or } (x, y) \in S \\
& \Leftrightarrow \quad \exists x \in A \text{ s.t. } (x, y) \in (R \cup S) \\
& \Leftrightarrow \quad y \in (R \cup S)(A)
\end{aligned}
$$

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ *and subsets* $A, B \subseteq X$:

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Proof (c):

Some results on relational images

Lemma

For any binary relations $R, S \subseteq X \times Y$ *and subsets* $A, B \subseteq X$*:*

(a) *If* $A \subseteq B$ *then* $R(A) \subseteq R(B)$

(b) $R(A) \cup S(A) = (R \cup S)(A)$

(c) $R(S(A)) = (S; R)(A)$

Proof (c):

$$\begin{aligned} z \in R(S(A)) \quad &\Leftrightarrow \quad \exists y \in S(A) \text{ s.t. } (y,z) \in R \\ &\Leftrightarrow \quad \exists x \in A,\ y \in S(A) \text{ s.t. } (x,y) \in S \text{ and } (y,z) \in R \\ &\Leftrightarrow \quad \exists x \in A \text{ s.t. } (x,z) \in (S;R) \\ &\Leftrightarrow \quad z \in (S;R)(A) \end{aligned}$$

Some results on relational images

Corollary

If $R(A) \subseteq A$ *then* $R^*(A) \subseteq A$

Some results on relational images

Corollary

If $R(A) \subseteq A$ *then* $R^*(A) \subseteq A$

Proof:

Some results on relational images

Corollary

If $R(A) \subseteq A$ *then* $R^*(A) \subseteq A$

Proof:

$$
\begin{aligned}
R(A) \subseteq A \quad &\Rightarrow \quad R^{i+1}(A) = R^i(R(A)) \subseteq R^i(A) \\
&\Rightarrow \quad R^{i+1}(A) \subseteq R(A) \subseteq A \\
\text{So } R^*(A) \quad &= \quad \left(\bigcup_{i=0}^{\infty} R^i\right)(A) \\
&= \quad \bigcup_{i=0}^{\infty} R^i(A) \\
&\subseteq \quad A
\end{aligned}
$$

Summary

- Set theory revisited
- Soundness of Hoare Logic
- Completeness of Hoare Logic

Soundness of Hoare Logic

Theorem

If $\vdash \{\varphi\}\, P\, \{\psi\}$ *then* $\models \{\varphi\}\, P\, \{\psi\}$

Soundness of Hoare Logic

Theorem

If $\vdash \{\varphi\}\, P\, \{\psi\}$ *then* $\models \{\varphi\}\, P\, \{\psi\}$

Proof:

Soundness of Hoare Logic

Theorem

If $\vdash \{\varphi\}\, P\, \{\psi\}$ *then* $\models \{\varphi\}\, P\, \{\psi\}$

Proof:
By induction on the structure of the proof.

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\; x := e \;\{\varphi\}} \quad \text{(ass)}$$

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Observation: $[\![\varphi[e/x]]\!]^{\eta} = [\![\varphi]\!]^{\eta'}$ where $\eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Observation: $[\![\varphi[e/x]]\!]^{\eta} = [\![\varphi]\!]^{\eta'}$ where $\eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$

So if $\eta \in \langle\varphi[e/x]\rangle$ then $\eta' \in \langle\varphi\rangle$

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Observation: $[\![\varphi[e/x]]\!]^{\eta} = [\![\varphi]\!]^{\eta'}$ where $\eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$

So if $\eta \in \langle\varphi[e/x]\rangle$ then $\eta' \in \langle\varphi\rangle$

Recall: $(\eta, \eta'') \in [\![x := e]\!]$ if and only if $\eta'' = \eta[x \mapsto [\![e]\!]^{\eta}]$,

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Observation: $[\![\varphi[e/x]]\!]^{\eta} = [\![\varphi]\!]^{\eta'}$ where $\eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$

So if $\eta \in \langle\varphi[e/x]\rangle$ then $\eta' \in \langle\varphi\rangle$

Recall: $(\eta, \eta'') \in [\![x := e]\!]$ if and only if $\eta'' = \eta[x \mapsto [\![e]\!]^{\eta}]$,

So $[\![x := e]\!](\eta) \in \langle\varphi\rangle$ for all $\eta \in \langle\varphi[e/x]\rangle$

Base case: Assignment rule

$$\frac{}{\{\varphi[e/x]\}\, x := e\, \{\varphi\}} \quad \text{(ass)}$$

Need to show $\{\varphi[e/x]\}\, x := e\, \{\varphi\}$ is always valid. That is,

$$[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle.$$

Observation: $[\![\varphi[e/x]]\!]^{\eta} = [\![\varphi]\!]^{\eta'}$ where $\eta' = \eta[x \mapsto [\![e]\!]^{\eta}]$

So if $\eta \in \langle\varphi[e/x]\rangle$ then $\eta' \in \langle\varphi\rangle$

Recall: $(\eta, \eta'') \in [\![x := e]\!]$ if and only if $\eta'' = \eta[x \mapsto [\![e]\!]^{\eta}]$,

So $[\![x := e]\!](\eta) \in \langle\varphi\rangle$ for all $\eta \in \langle\varphi[e/x]\rangle$

So $[\![x := e]\!](\langle\varphi[e/x]\rangle) \subseteq \langle\varphi\rangle$

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\ P\ \{\psi\} \qquad \{\psi\}\ Q\ \{\rho\}}{\{\varphi\}\ P; Q\ \{\rho\}} \quad \text{(seq)}$$

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\, P\, \{\psi\} \qquad \{\psi\}\, Q\, \{\rho\}}{\{\varphi\}\, P; Q\, \{\rho\}} \quad \text{(seq)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ and $\{\psi\}\, Q\, \{\rho\}$ are valid. Need to show that $\{\varphi\}\, P; Q\, \{\rho\}$ is valid.

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\, P\, \{\psi\} \qquad \{\psi\}\, Q\, \{\rho\}}{\{\varphi\}\, P; Q\, \{\rho\}} \quad \text{(seq)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ and $\{\psi\}\, Q\, \{\rho\}$ are valid. Need to show that $\{\varphi\}\, P; Q\, \{\rho\}$ is valid.

Recall: $[\![P; Q]\!] = [\![P]\!]; [\![Q]\!]$

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\, P\, \{\psi\} \qquad \{\psi\}\, Q\, \{\rho\}}{\{\varphi\}\, P; Q\, \{\rho\}} \quad \text{(seq)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ and $\{\psi\}\, Q\, \{\rho\}$ are valid. Need to show that $\{\varphi\}\, P; Q\, \{\rho\}$ is valid.

Recall: $[\![P; Q]\!] = [\![P]\!]; [\![Q]\!]$

So: $[\![P; Q]\!](\langle\varphi\rangle) = [\![Q]\!]\big([\![P]\!](\langle\varphi\rangle)\big)$ (see Lemma 1(c))

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\,P\,\{\psi\} \qquad \{\psi\}\,Q\,\{\rho\}}{\{\varphi\}\,P;Q\,\{\rho\}} \quad \text{(seq)}$$

Assume $\{\varphi\}\,P\,\{\psi\}$ and $\{\psi\}\,Q\,\{\rho\}$ are valid. Need to show that $\{\varphi\}\,P;Q\,\{\rho\}$ is valid.

Recall: $[\![P;Q]\!] = [\![P]\!];[\![Q]\!]$

So: $[\![P;Q]\!](\langle\varphi\rangle) = [\![Q]\!]\big([\![P]\!](\langle\varphi\rangle)\big)$ (see Lemma 1(c))

By IH: $[\![P]\!](\langle\varphi\rangle) \subseteq \langle\psi\rangle$ and $[\![Q]\!](\langle\psi\rangle) \subseteq \langle\rho\rangle$

Inductive case 1: Sequence rule

$$\frac{\{\varphi\}\,P\,\{\psi\} \qquad \{\psi\}\,Q\,\{\rho\}}{\{\varphi\}\,P;Q\,\{\rho\}} \quad \text{(seq)}$$

Assume $\{\varphi\}\,P\,\{\psi\}$ and $\{\psi\}\,Q\,\{\rho\}$ are valid. Need to show that $\{\varphi\}\,P;Q\,\{\rho\}$ is valid.

Recall: $[\![P;Q]\!] = [\![P]\!];[\![Q]\!]$

So: $[\![P;Q]\!](\langle\varphi\rangle) = [\![Q]\!]\big([\![P]\!](\langle\varphi\rangle)\big)$ (see Lemma 1(c))

By IH: $[\![P]\!](\langle\varphi\rangle) \subseteq \langle\psi\rangle$ and $[\![Q]\!](\langle\psi\rangle) \subseteq \langle\rho\rangle$

So: $[\![Q]\!]\big([\![P]\!](\langle\varphi\rangle)\big) \subseteq [\![Q]\!]\big(\langle\psi\rangle\big) \subseteq \langle\rho\rangle$ (see Lemma 1(a))

Two more useful results

Lemma

For $R \subseteq \textsc{Env} \times \textsc{Env}$, *predicates* φ *and* ψ, *and* $X \subseteq \textsc{Env}$*:*

(a) $[\![\varphi]\!](X) = \langle\varphi\rangle \cap X$

(b) $R(\langle\varphi \wedge \psi\rangle) = ([\![\varphi]\!]; R)(\langle\psi\rangle)$

Two more useful results

Lemma

For $R \subseteq \textsc{Env} \times \textsc{Env}$*, predicates* φ *and* ψ*, and* $X \subseteq \textsc{Env}$*:*

- (a) $[\![\varphi]\!](X) = \langle\varphi\rangle \cap X$
- (b) $R(\langle\varphi \wedge \psi\rangle) = ([\![\varphi]\!]; R)(\langle\psi\rangle)$

Proof (a):

Two more useful results

Lemma

For $R \subseteq \textsc{Env} \times \textsc{Env}$, *predicates* φ *and* ψ, *and* $X \subseteq \textsc{Env}$*:*

(a) $[\![\varphi]\!](X) = \langle\varphi\rangle \cap X$

(b) $R(\langle\varphi \wedge \psi\rangle) = ([\![\varphi]\!]; R)(\langle\psi\rangle))$

Proof (a):

$$\begin{aligned} \eta' \in [\![\varphi]\!](X) &\Leftrightarrow \exists \eta \in X \text{ s.t. } (\eta, \eta') \in [\![\varphi]\!] \\ &\Leftrightarrow \exists \eta \in X \text{ s.t. } \eta = \eta' \text{ and } \eta \in \langle\varphi\rangle \\ &\Leftrightarrow \eta' \in X \cap \langle\varphi\rangle \end{aligned}$$

Two more useful results

Lemma

For $R \subseteq \textsc{Env} \times \textsc{Env}$, predicates φ and ψ, and $X \subseteq \textsc{Env}$:

(a) $[\![\varphi]\!](X) = \langle\varphi\rangle \cap X$

(b) $R(\langle\varphi \wedge \psi\rangle) = ([\![\varphi]\!]; R)(\langle\psi\rangle)$

Proof (b):

$$\begin{aligned} \langle\varphi \wedge \psi\rangle &= \langle\varphi\rangle \cap \langle\psi\rangle = [\![\varphi]\!](\langle\psi\rangle) \\ \text{So } R(\langle\varphi \wedge \psi\rangle) &= R\big([\![\varphi]\!](\langle\psi\rangle)\big) \\ &= ([\![\varphi]\!]; R)(\langle\psi\rangle) \qquad \text{(see Lemma 1(b))} \end{aligned}$$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \land g\}\ P\ \{\psi\} \qquad \{\varphi \land \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}$ if g then P else Q fi $\{\psi\}$ is valid.

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}$ if g then P else Q fi $\{\psi\}$ is valid.

Recall: $[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![g; P]\!] \cup [\![\neg g; Q]\!]$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}$ if g then P else Q fi $\{\psi\}$ is valid.

Recall: $[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![g; P]\!] \cup [\![\neg g; Q]\!]$

$$[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!](\langle\varphi\rangle)$$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}$ if g then P else Q fi $\{\psi\}$ is valid.

Recall: $[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![g; P]\!] \cup [\![\neg g; Q]\!]$

$$[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!](\langle\varphi\rangle)$$

$$= [\![g; P]\!](\langle\varphi\rangle) \cup [\![\neg g; Q]\!](\langle\varphi\rangle) \qquad \text{(see Lemma 1(b))}$$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}$ if g then P else Q fi $\{\psi\}$ is valid.

Recall: $[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![g; P]\!] \cup [\![\neg g; Q]\!]$

$$\begin{aligned}
&[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!](\langle\varphi\rangle) \\
&\quad = [\![g; P]\!](\langle\varphi\rangle) \cup [\![\neg g; Q]\!](\langle\varphi\rangle) && \text{(see Lemma 1(b))} \\
&\quad = [\![P]\!](\langle g \wedge \varphi\rangle) \cup [\![Q]\!](\langle \neg g \wedge \varphi\rangle) && \text{(see Lemma 2(b))}
\end{aligned}$$

Inductive case 2: Conditional rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg g\}\ Q\ \{\psi\}}{\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}} \quad \text{(if)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\psi\}$ and $\{\varphi \wedge \neg g\}\ Q\ \{\psi\}$ are valid. Need to show that $\{\varphi\}\ \text{if } g \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}$ is valid.

Recall: $[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!] = [\![g; P]\!] \cup [\![\neg g; Q]\!]$

$$
\begin{aligned}
&[\![\text{if } g \text{ then } P \text{ else } Q \text{ fi}]\!](\langle\varphi\rangle) \\
&\quad = [\![g; P]\!](\langle\varphi\rangle) \cup [\![\neg g; Q]\!](\langle\varphi\rangle) && \text{(see Lemma 1(b))} \\
&\quad = [\![P]\!](\langle g \wedge \varphi\rangle) \cup [\![Q]\!](\langle \neg g \wedge \varphi\rangle) && \text{(see Lemma 2(b))} \\
&\quad \subseteq \langle\psi\rangle && \text{(by IH)}
\end{aligned}
$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}}\quad \text{(loop)}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}$ is valid.

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}$ is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}$ is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$[\![g; P]\!](\langle\varphi\rangle) = [\![P]\!](\langle g \wedge \varphi\rangle) \qquad \text{(see Lemma 2(b))}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}$ is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$\begin{aligned} [\![g; P]\!](\langle\varphi\rangle) &= [\![P]\!](\langle g \wedge \varphi\rangle) && \text{(see Lemma 2(b))} \\ &\subseteq \langle\varphi\rangle && \text{(IH)} \end{aligned}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}$ is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$\begin{aligned} [\![g; P]\!](\langle\varphi\rangle) &= [\![P]\!](\langle g \wedge \varphi\rangle) && \text{(see Lemma 2(b))} \\ &\subseteq \langle\varphi\rangle && \text{(IH)} \\ \text{So } [\![g; P]\!]^*(\langle\varphi\rangle) &\subseteq \langle\varphi\rangle && \text{(see Corollary)} \end{aligned}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}\ \text{while}\ g\ \text{do}\ P\ \text{od}\ \{\varphi \wedge \neg g\}$ is valid.

Recall: $[\![\text{while}\ g\ \text{do}\ P\ \text{od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$\begin{aligned}
[\![g; P]\!](\langle\varphi\rangle) &= [\![P]\!](\langle g \wedge \varphi\rangle) && \text{(see Lemma 2(b))}\\
&\subseteq \langle\varphi\rangle && \text{(IH)}\\
\text{So } [\![g; P]\!]^*(\langle\varphi\rangle) &\subseteq \langle\varphi\rangle && \text{(see Corollary)}\\
\text{So } [\![g; P]\!]^*; [\![\neg g]\!](\langle\varphi\rangle) &= [\![\neg g]\!]\big([\![g; P]\!]^*(\langle\varphi\rangle)\big) && \text{(see Lemma 1(c))}
\end{aligned}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}$ while g do P od $\{\varphi \wedge \neg g\}$is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$\begin{aligned}
[\![g; P]\!](\langle\varphi\rangle) &= [\![P]\!](\langle g \wedge \varphi\rangle) && \text{(see Lemma 2(b))}\\
&\subseteq \langle\varphi\rangle && \text{(IH)}\\
\text{So } [\![g; P]\!]^*(\langle\varphi\rangle) &\subseteq \langle\varphi\rangle && \text{(see Corollary)}\\
\text{So } [\![g; P]\!]^*; [\![\neg g]\!](\langle\varphi\rangle) &= [\![\neg g]\!]\left([\![g; P]\!]^*(\langle\varphi\rangle)\right) && \text{(see Lemma 1(c))}\\
&\subseteq [\![\neg g]\!](\langle\varphi\rangle) && \text{(see Lemma 1(a))}
\end{aligned}$$

Inductive case 3: While rule

$$\frac{\{\varphi \wedge g\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } g \text{ do } P \text{ od}\ \{\varphi \wedge \neg g\}} \quad \text{(loop)}$$

Assume $\{\varphi \wedge g\}\ P\ \{\varphi\}$ is valid. Need to show that $\{\varphi\}$ while g do P od $\{\varphi \wedge \neg g\}$is valid.

Recall: $[\![\text{while } g \text{ do } P \text{ od}]\!] = [\![g; P]\!]^*; [\![\neg g]\!]$

$$\begin{aligned}
[\![g; P]\!](\langle\varphi\rangle) &= [\![P]\!](\langle g \wedge \varphi\rangle) && \text{(see Lemma 2(b))}\\
&\subseteq \langle\varphi\rangle && \text{(IH)}\\
\text{So } [\![g; P]\!]^*(\langle\varphi\rangle) &\subseteq \langle\varphi\rangle && \text{(see Corollary)}\\
\text{So } [\![g; P]\!]^*; [\![\neg g]\!](\langle\varphi\rangle) &= [\![\neg g]\!]\big([\![g; P]\!]^*(\langle\varphi\rangle)\big) && \text{(see Lemma 1(c))}\\
&\subseteq [\![\neg g]\!](\langle\varphi\rangle) && \text{(see Lemma 1(a))}\\
&= \langle\neg g \wedge \varphi\rangle && \text{(see Lemma 2(a))}
\end{aligned}$$

Inductive case 4: Consequence rule

$$\frac{\varphi' \to \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \to \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Inductive case 4: Consequence rule

$$\frac{\varphi' \to \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \to \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ is valid and $\varphi' \to \varphi$ and $\psi \to \psi'$. Need to show that $\{\varphi'\}\, P\, \{\psi'\}$ is valid.

Inductive case 4: Consequence rule

$$\frac{\varphi' \rightarrow \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \rightarrow \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ is valid and $\varphi' \rightarrow \varphi$ and $\psi \rightarrow \psi'$. Need to show that $\{\varphi'\}\, P\, \{\psi'\}$ is valid.

Observe: If $\varphi' \rightarrow \varphi$ then $\langle \varphi' \rangle \subseteq \langle \varphi \rangle$

Inductive case 4: Consequence rule

$$\frac{\varphi' \to \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \to \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ is valid and $\varphi' \to \varphi$ and $\psi \to \psi'$. Need to show that $\{\varphi'\}\, P\, \{\psi'\}$ is valid.

Observe: If $\varphi' \to \varphi$ then $\langle \varphi' \rangle \subseteq \langle \varphi \rangle$

$$[\![P]\!](\langle \varphi' \rangle) \quad \subseteq \quad [\![P]\!](\langle \varphi \rangle) \quad \text{(see Lemma 1(a))}$$

Inductive case 4: Consequence rule

$$\frac{\varphi' \to \varphi \qquad \{\varphi\}\, P\, \{\psi\} \qquad \psi \to \psi'}{\{\varphi'\}\, P\, \{\psi'\}} \quad \text{(cons)}$$

Assume $\{\varphi\}\, P\, \{\psi\}$ is valid and $\varphi' \to \varphi$ and $\psi \to \psi'$. Need to show that $\{\varphi'\}\, P\, \{\psi'\}$ is valid.

Observe: If $\varphi' \to \varphi$ then $\langle \varphi' \rangle \subseteq \langle \varphi \rangle$

$$\begin{aligned}
[\![P]\!](\langle \varphi' \rangle) &\subseteq [\![P]\!](\langle \varphi \rangle) && \text{(see Lemma 1(a))} \\
&\subseteq \langle \psi \rangle && \text{(IH)} \\
&\subseteq \langle \psi' \rangle
\end{aligned}$$

Soundness of Hoare Logic

Theorem

If $\vdash \{\varphi\}\, P\, \{\psi\}$ *then* $\models \{\varphi\}\, P\, \{\psi\}$

Summary

- Set theory revisited
- Soundness of Hoare Logic
- Completeness of Hoare Logic

Incompleteness

Theorem (Gödel's Incompleteness Theorem)

There is no proof system that can prove every valid first-order sentence about arithmetic over the natural numbers.

Incompleteness

Theorem (Gödel's Incompleteness Theorem)

There is no proof system that can prove every valid first-order sentence about arithmetic over the natural numbers.

$\Rightarrow$ There are true statements that do not have a proof.

Incompleteness

Theorem (Gödel's Incompleteness Theorem)

There is no proof system that can prove every valid first-order sentence about arithmetic over the natural numbers.

$\Rightarrow$ There are true statements that do not have a proof.

$\Rightarrow$ Because of (cons) there are valid triples that result from valid, but unprovable, consequences.

Incompleteness

Theorem (Gödel's Incompleteness Theorem)

There is no proof system that can prove every valid first-order sentence about arithmetic over the natural numbers.

$\Rightarrow$ There are true statements that do not have a proof.

$\Rightarrow$ Because of (cons) there are valid triples that result from valid, but unprovable, consequences.

$\Rightarrow$ Hoare Logic is not complete.

Relative completeness of Hoare Logic

Theorem (Relative completeness of Hoare Logic)

With an oracle that decides the validity of predicates,

$$\textit{if } \models \{\varphi\}\, P\, \{\psi\} \;\; \textit{then } \vdash \{\varphi\}\, P\, \{\psi\}\,.$$

Need to know for this course

- Write programs in $\mathcal{L}$.
- Give proofs using the Hoare logic rules (full and outline)
- Definition of $[\![\cdot]\!]$
- Definition of composition and transitive closure