C03 – Hoare Logic (cont.)

Program Verification

FMI · Denisa Diaconescu · Spring 2025

Proof rules for Hoare logic

The assignment axiom: $\boxed{\{Q(\mathbb{E})\}\ \texttt{x := } \mathbb{E}\ \{Q(x)\}}$

Precondition Strengthening rule: $\boxed{\dfrac{P_s \rightarrow P_w \qquad \{P_w\}\ \mathbb{C}\ \{Q\}}{\{P_s\}\ \mathbb{C}\ \{Q\}}}$

Postcondition Weakening rule: $\boxed{\dfrac{\{P\}\ \mathbb{C}\ \{Q_s\} \qquad Q_s \rightarrow Q_w}{\{P\}\ \mathbb{C}\ \{Q_w\}}}$

Sequencing rule: $\boxed{\dfrac{\{P\}\mathbb{C}_1\{Q\} \qquad \{Q\}\mathbb{C}_2\{R\}}{\{P\}\mathbb{C}_1;\mathbb{C}_2\{R\}}}$

Conditional rule: $\boxed{\dfrac{\{P \wedge \mathbb{B}\}\ \mathbb{C}_1\ \{Q\} \qquad \{P \wedge \neg\mathbb{B}\}\ \mathbb{C}_2\ \{Q\}}{\{P\}\ \texttt{if}\ \mathbb{B}\ \texttt{then}\ \mathbb{C}_1\ \texttt{else}\ \mathbb{C}_2\ \{Q\}}}$

Proof rule for While Loops

Suppose we want to prove

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}$$

Proof rule for While Loops

Suppose we want to prove

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}$$

While rule:

$$\boxed{\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

- I is called loop invariant
- I is true before we encounter the while statement, and remains true after each iteration of the loop (although not necessarily midway during execution of the loop body).
- If the loop terminates the loop condition must be false, so $\neg\mathbb{B}$ appears in the postcondition.
- For the body of the loop $\mathbb{C}$ to execute, $\mathbb{B}$ needs to be true, so it appears in the precondition.

Proof rule for While Loops

Suppose we want to prove

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}$$

While rule:

$$\boxed{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

- I is called loop invariant
- I is true before we encounter the while statement, and remains true after each iteration of the loop (although not necessarily midway during execution of the loop body).
- If the loop terminates the loop condition must be false, so $\neg\mathbb{B}$ appears in the postcondition.
- For the body of the loop $\mathbb{C}$ to execute, $\mathbb{B}$ needs to be true, so it appears in the precondition.
- The most difficult part is to come up with the invariant.
- This requires intuition. There is no algorithm that will find the invariant.

Applying the while rule

How does the while rule helps to solve our problem?

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}$$

$$\boxed{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

Applying the while rule

How does the while rule helps to solve our problem?

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\} \qquad \boxed{\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

- The postcondition we get after applying our rule has the form $I \wedge \neg\mathbb{B}$. This might not be the same as the postcondition Q we want!

Applying the while rule

How does the while rule helps to solve our problem?

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\} \qquad \boxed{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

- The postcondition we get after applying our rule has the form $I \wedge \neg\mathbb{B}$. This might not be the same as the postcondition Q we want!
- If $(I \wedge \neg\mathbb{B}) \leftrightarrow Q$, we can replace $I \wedge \neg\mathbb{B}$ with Q.
- If $(I \wedge \neg\mathbb{B}) \rightarrow Q$ we can use the Postcondition weakening rule:

$$\dfrac{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}} \qquad I \wedge \neg\mathbb{B} \rightarrow Q}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}}$$

Applying the while rule

How does the while rule helps to solve our problem?

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\} \qquad \boxed{\dfrac{\{I \land \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \land \neg\mathbb{B}\}}}$$

- Similarly, P and I might be different formulas.

Applying the while rule

How does the while rule helps to solve our problem?

$$\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}$$

$$\boxed{\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

- Similarly, P and I might be different formulas.
- If $I \leftrightarrow P$, we can replace I with P to complete our proof.
- If $P \rightarrow I$ we can use the Precondition strengthening rule:

$$\frac{P \rightarrow I \qquad \{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}}{\{P\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{Q\}}$$

Proof rule for While Loops

Example
Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

Proof rule for While Loops

Example
Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We want a loop invariant I such that

- if I is true initially
- I remains true each time around the loop
- $I \wedge \neg(n > 0) \to (n = 0)$

Proof rule for While Loops

Example
Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We want a loop invariant I such that

- if I is true initially
- I remains true each time around the loop
- $I \land \neg(n > 0) \to (n = 0)$

$I \equiv n \geq 0$ looks like a reasonable loop invariant.

Proof rule for While Loops

While rule:

$$\boxed{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

Example (cont.)

Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We consider the loop invariant $I \equiv n \geq 0$. Let's try to find P.

1. $\{n - 1 \geq 0\}$ `n := n-1` $\{n \geq 0\}$ (Assignment rule)

Proof rule for While Loops

While rule:

$$\boxed{\dfrac{\{I \land \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \land \neg\mathbb{B}\}}}$$

Example (cont.)

Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We consider the loop invariant $I \equiv n \geq 0$. Let's try to find P.

1. $\{n - 1 \geq 0\}$ `n := n-1` $\{n \geq 0\}$ (Assignment rule)
2. $\{n \geq 0 \land n > 0\}$ `n := n-1` $\{n \geq 0\}$ (1, Precond. Equiv.)

Proof rule for While Loops

While rule:

$$\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}$$

Example (cont.)

Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We consider the loop invariant $I \equiv n \geq 0$. Let's try to find P.

1. $\{n - 1 \geq 0\}$ `n := n-1` $\{n \geq 0\}$ (Assignment rule)
2. $\{n \geq 0 \wedge n > 0\}$ `n := n-1` $\{n \geq 0\}$ (1, Precond. Equiv.)
3. $\{n \geq 0\}$ `while (n>0) do n := n-1` $\{n \geq 0 \wedge \neg(n > 0)\}$ (2, While rule)

Proof rule for While Loops

While rule:

$$\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}$$

Example (cont.)

Suppose we want to find a precondition P such that

$$\{P\}\ \texttt{while (n > 0) do n := n-1}\ \{n = 0\}$$

We consider the loop invariant $I \equiv n \geq 0$. Let's try to find P.

1. $\{n - 1 \geq 0\}$ `n := n-1` $\{n \geq 0\}$ (Assignment rule)
2. $\{n \geq 0 \wedge n > 0\}$ `n := n-1` $\{n \geq 0\}$ (1, Precond. Equiv.)
3. $\{n \geq 0\}$ `while (n>0) do n := n-1` $\{n \geq 0 \wedge \neg(n > 0)\}$ (2, While rule)
4. $\{n \geq 0\}$ `while (n>0) do n := n-1` $\{n = 0\}$ (3, Postcond. Equiv.)

So we take P to be $n \geq 0$.

Proof rules for Hoare logic

The assignment axiom: $\boxed{\{Q[x/\mathbb{E}]\}\ \texttt{x := } \mathbb{E}\ \{Q\}}$

Precondition Strengthening rule: $\boxed{\dfrac{P_s \to P_w \qquad \{P_w\}\ \mathbb{C}\ \{Q\}}{\{P_s\}\ \mathbb{C}\ \{Q\}}}$

Postcondition Weakening rule: $\boxed{\dfrac{\{P\}\ \mathbb{C}\ \{Q_s\} \qquad Q_s \to Q_w}{\{P\}\ \mathbb{C}\ \{Q_w\}}}$

Sequencing rule: $\boxed{\dfrac{\{P\}\mathbb{C}_1\{Q\} \qquad \{Q\}\mathbb{C}_2\{R\}}{\{P\}\mathbb{C}_1;\mathbb{C}_2\{R\}}}$

Conditional rule: $\boxed{\dfrac{\{P \wedge \mathbb{B}\}\ \mathbb{C}_1\ \{Q\} \qquad \{P \wedge \neg\mathbb{B}\}\ \mathbb{C}_2\ \{Q\}}{\{P\}\ \texttt{if}\ \mathbb{B}\ \texttt{then}\ \mathbb{C}_1\ \texttt{else}\ \mathbb{C}_2\ \{Q\}}}$

While rule: $\boxed{\dfrac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$

A simple program

Example

Consider the `Program`:

```
i := 0;
s := 0;
while (i != n) do
  i := i+1;
  s := s+(2*i-1)
```

Goal: prove $\{\top\}$ `Program` $\{s = n^2\}$

The sum of the first n odd numbers is n^2.

A simple program

Example (cont.)

Let us check some examples:

- $1 = 1 = 1^2$
- $1 + 3 = 4 = 2^2$
- $1 + 3 + 5 = 9 = 3^2$
- $1 + 3 + 5 + 7 = 16 = 4^2$

It looks OK. Let us see if we can prove it!

Goal: prove $\{\top\}$ `Program` $\{s = n^2\}$

A simple program

Example (cont.)

First we need a loop invariant I.

$$\boxed{\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}}$$

```
while (i != n) do
  i := i+1;
  s := s+(2*i-1)
```

$\{s = n^2\}$

From the while rule, we want $I \wedge (i = n) \to (s = n^2)$ in order to be able to apply Postcond. Weak.

In the loop body, each time, i increments and s moves on the next square number.

A simple program

Example (cont.)

First we need a loop invariant I.

$$\frac{\{I \wedge \mathbb{B}\}\ \mathbb{C}\ \{I\}}{\{I\}\ \texttt{while}\ \mathbb{B}\ \texttt{do}\ \mathbb{C}\ \{I \wedge \neg\mathbb{B}\}}$$

```
while (i != n) do
  i := i+1;
  s := s+(2*i-1)
```

$\{s = n^2\}$

From the while rule, we want $I \wedge (i = n) \rightarrow (s = n^2)$ in order to be able to apply Postcond. Weak.

In the loop body, each time, i increments and s moves on the next square number.

Loop invariant $I \equiv (s = i^2)$ seems plausible.

A simple program

Example (cont.)

We check that $I \equiv (s = i^2)$ is an invariant. Let us prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$.

$$\frac{\{s = i^2 \wedge i \neq n\}\mathtt{i} := \mathtt{i} + 1\{Q\} \qquad \{Q\}\mathtt{s} := \mathtt{s} + (2 * \mathtt{i} - 1)\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\mathtt{i} := \mathtt{i} + 1;\ \mathtt{s} := \mathtt{s} + (2 * \mathtt{i} - 1)\{s = i^2\}}$$

A simple program

Example (cont.)

We check that $I \equiv (s = i^2)$ is an invariant. Let us prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$.

$$\frac{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1\{Q\} \qquad \{Q\}\texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1;\ \texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}$$

1. $\{Q\}$ `s:=s+(2*i-1)` $\{s = i^2\}$
2.

3. $\{s = i^2 \wedge i \neq n\}$ `i:=i+1` $\{Q\}$
4. $\{s = i^2 \wedge i \neq n\}$ `i:=i+1; s:=s+(2*i-1)` $\{s = i^2\}$ (1,3, Seq.)

A simple program

Example (cont.)
Check $I \equiv (s = i^2)$ is an invariant: prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$

$$\frac{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1\{Q\} \qquad \{Q\}\texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1;\ \texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}$$

Q is $\{s + (2 * i - 1) = i^2\}$

1. $\{s + (2 * i - 1) = i^2\}$ `s := s+(2*i-1)` $\{s = i^2\}$ (Assignment)
2.
3. $\{s = i^2 \wedge i \neq n\}$ `i := i+1` $\{Q\}$
4. $\{s = i^2 \wedge i \neq n\}$ `i := i+1; s := s+(2*i-1)` $\{s = i^2\}$ (1,3, Seq.)

A simple program

Example (cont.)
Check $I \equiv (s = i^2)$ is an invariant: prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$

$$\frac{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + \texttt{1}\{Q\} \qquad \{Q\}\texttt{s} := \texttt{s} + (\texttt{2} * \texttt{i} - \texttt{1})\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + \texttt{1};\ \texttt{s} := \texttt{s} + (\texttt{2} * \texttt{i} - \texttt{1})\{s = i^2\}}$$

Q is $\{s + (2 * i - 1) = i^2\}$

1. $\{s + (2 * i - 1) = i^2\}$ `s := s+(2*i-1)` $\{s = i^2\}$ (Assignment)
2.
3. $\{s = i^2 \wedge i \neq n\}$ `i := i+1` $\{s + (2 * i - 1) = i^2\}$
4. $\{s = i^2 \wedge i \neq n\}$ `i := i+1; s := s+(2*i-1)` $\{s = i^2\}$ (1,3, Seq.)

A simple program

Example (cont.)
Check $I \equiv (s = i^2)$ is an invariant: prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$

$$\frac{\{s = i^2 \wedge i \neq n\}\texttt{i := i + 1}\{Q\} \qquad \{Q\}\texttt{s := s + (2 * i - 1)}\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\texttt{i := i + 1; s := s + (2 * i - 1)}\{s = i^2\}}$$

Q is $\{s + (2 * i - 1) = i^2\}$

1. $\{s + (2 * i - 1) = i^2\}$ `s := s+(2*i-1)` $\{s = i^2\}$ (Assignment)
2. $\{s + (2 * (i + 1) - 1) = (i + 1)^2\}$ `i := i+1` $\{s + (2 * i - 1) = i^2\}$ (Assignment)
3. $\{s = i^2 \wedge i \neq n\}$ `i := i+1` $\{s + (2 * i - 1) = i^2\}$
4. $\{s = i^2 \wedge i \neq n\}$ `i := i+1; s := s+(2*i-1)` $\{s = i^2\}$ (1,3, Seq.)

A simple program

Example (cont.)
Check $I \equiv (s = i^2)$ is an invariant: prove $\{I \wedge (i \neq n)\}\ \mathbb{C}\ \{I\}$

$$\frac{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1\{Q\} \qquad \{Q\}\texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}{\{s = i^2 \wedge i \neq n\}\texttt{i} := \texttt{i} + 1;\ \texttt{s} := \texttt{s} + (2 * \texttt{i} - 1)\{s = i^2\}}$$

Q is $\{s + (2 * i - 1) = i^2\}$

1. $\{s + (2 * i - 1) = i^2\}$ `s := s+(2*i-1)` $\{s = i^2\}$ (Assignment)
2. $\{s + (2 * (i + 1) - 1) = (i + 1)^2\}$ `i := i+1` $\{s + (2 * i - 1) = i^2\}$ (Assignment)
3. $\{s = i^2 \wedge i \neq n\}$ `i := i+1` $\{s + (2 * i - 1) = i^2\}$ (2, Strength. Precond.)
4. $\{s = i^2 \wedge i \neq n\}$ `i := i+1; s := s+(2*i-1)` $\{s = i^2\}$ (1,3, Seq.)

So far, so good.

A simple program

Example (cont.)

Completing the proof of $\{\top\}$ `Program` $\{s = n^2\}$

1. We have

$$\{(s = i^2) \wedge (i \neq n)\}\ \texttt{i := i+1; s := s+(2*i-1)}\ \{s = i^2\}$$

A simple program

Example (cont.)

Completing the proof of $\{\top\}$ `Program` $\{s = n^2\}$

1. We have

$$\{(s = i^2) \wedge (i \neq n)\}\ \texttt{i := i+1; s := s+(2*i-1)}\ \{s = i^2\}$$

2. Apply the While rule and Postcondition Weakening rule since $(s = i^2) \wedge (i = n) \rightarrow s = n^2$

$$\{s = i^2\}\ \texttt{while ... s:=s+(2*i-1)}\ \{s = n^2\}$$

A simple program

Example (cont.)

Completing the proof of $\{\top\}$ `Program` $\{s = n^2\}$

1. We have

$$\{(s = i^2) \wedge (i \neq n)\} \texttt{ i := i+1; s := s+(2*i-1) } \{s = i^2\}$$

2. Apply the While rule and Postcondition Weakening rule since $(s = i^2) \wedge (i = n) \rightarrow s = n^2$

$$\{s = i^2\} \texttt{ while ... s:=s+(2*i-1) } \{s = n^2\}$$

3. Check that the initialization establishes the invariant:

$$\frac{\{0 = 0^2\}\texttt{i} := 0\{0 = i^2\} \qquad \{0 = i^2\}\texttt{s} := 0\{s = i^2\}}{\{0 = 0^2\}\texttt{i} := 0; \texttt{s} := 0\{s = i^2\}}$$

A simple program

Example (cont.)

Completing the proof of $\{\top\}$ `Program` $\{s = n^2\}$

1. We have

$$\{(s = i^2) \wedge (i \neq n)\}\ \texttt{i := i+1; s := s+(2*i-1)}\ \{s = i^2\}$$

2. Apply the While rule and Postcondition Weakening rule since $(s = i^2) \wedge (i = n) \rightarrow s = n^2$

$$\{s = i^2\}\ \texttt{while ... s:=s+(2*i-1)}\ \{s = n^2\}$$

3. Check that the initialization establishes the invariant:

$$\frac{\{0 = 0^2\}\texttt{i} := 0\{0 = i^2\} \qquad \{0 = i^2\}\texttt{s} := 0\{s = i^2\}}{\{0 = 0^2\}\texttt{i} := 0; \texttt{s} := 0\{s = i^2\}}$$

4. $(0 = 0^2) \leftrightarrow \top$, so putting it all together with Sequencing we have

$$\{\top\}\ \texttt{i:=0; s:=0; while (i != n) do S}\ \{s = n^2\}$$

Verifying programs with Hoare Logic

Exercise:

Consider the program `Factorial`:

```
y := 1;
z := 0;
while (z != x) do
  z := z+1;
  y := y*z
```

Goal: prove $\{\top\}$ `Program` $\{y = x!\}$

Verifying programs with Hoare Logic

Exercise:

Consider the program `Factorial`:

```
y := 1;
z := 0;
while (z != x) do
  z := z+1;
  y := y*z
```

Goal: prove $\{\top\}$ `Program` $\{y = x!\}$

Hint! Use the loop invariant $I \equiv y = z!$

Quiz time!

https://tinyurl.com/FMI-PV2023-Quiz3