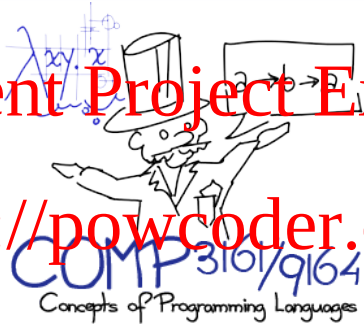


Imperative Programming Languages

Dr. Liam O'Connor
University of Edinburgh LFCS
UNSW, Term 3 2020

Imperative Programming

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*).

The Old Days

Early microcomputer languages used a line numbering system with GO TO statements used to arrange control flow.

Factorial Example in BASIC (1964)

Dijkstra (1968)


Letters to the Editor

"Go To Statement Considered Harmful"

Key Words and Phrases: go to statement, jump instruction, branch instruction, conditional clause, alternative clause, repetitive clause, program intelligibility, program sequencing

CR Categories: 4.22, 5.23, 5.24

The *structured programming* movement brought in *control structures* to mainstream use, such as conditionals and loops.

Factorial Example in Pascal (1970)

```
program factorial;
var n : integer;
    m : integer;
    i : integer;
begin
n := 5;
m := 1;
i := 0;
while (i < n) do
begin
  i := i + 1;
  m := m * i;
end;
println(m);
```

Syntax

We're going to specify a language **TinyImp**, based on structured programming. The syntax consists of statements and expressions.

Grammar

Stmt	::=	`skip`	*Do nothing*
	|	`x := Expr`	*Assignment*
	|	`var y ; Stmt`	*Declaration*
	|	`if` **Expr** `then` **Stmt** `else` **Stmt** `fi`	*Conditional*
	|	`while` **Expr** `do` **Stmt** `od`	*Loop*
	|	**Stmt** `;` **Stmt**	*Sequencing*
Expr	::=	⟨*Arithmetic expressions*⟩	

We already know how to make unambiguous abstract syntax, so we will use concrete syntax in the rules for readability.

Examples

Example (Factorial and Fibonacci)

```
var i ·
var m ·
i := 0;
m := 1;
while i < N do
  i := i + 1;
  m := m × i
od
```

```
var m · var n · var i ·
m := 1; n := 1;
i := 1;
while i < N do
  var t · t := m;
  m := n;
  n := n + t;
  i := i + 1
od
```

Static Semantics

Types?

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes?

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else?

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!

$$U, V \vdash s \Rightarrow \mathbf{ok} \rightsquigarrow W$$

Note: $V \subseteq U$

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!

$$U / V \vdash s \Rightarrow \mathbf{ok} \rightsquigarrow W$$

Set of declared free variables

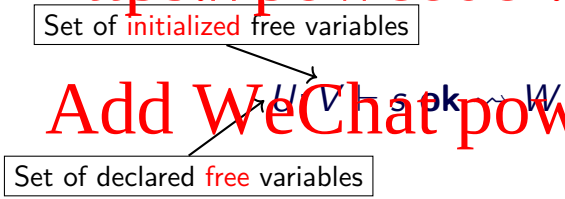
Note: $V \subseteq U$

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!



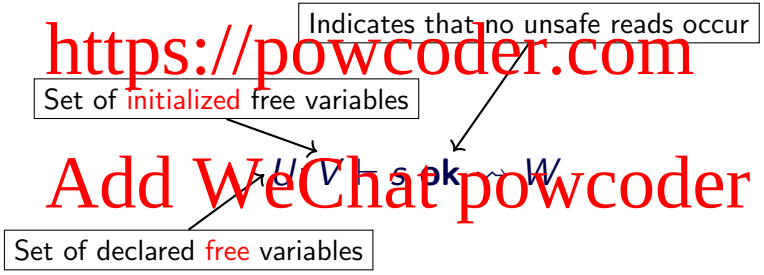
Note: $V \subseteq U$

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!



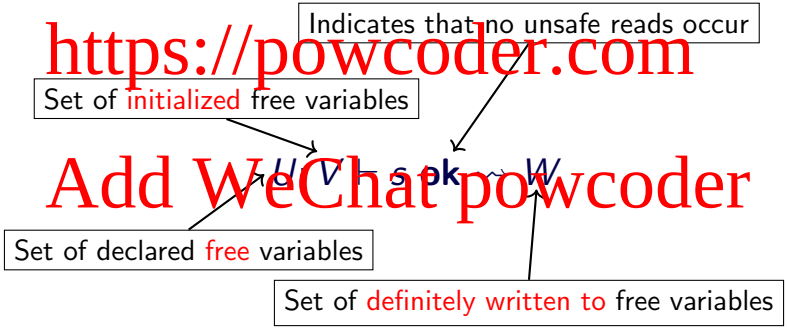
Note: $V \subseteq U$

Static Semantics

Types? We only have one type (`int`), so type checking is a wash.

Scopes? We have to check that variables are declared before use.

Anything Else? We have to check that variables are *initialized* before they are used!



Note: $V \subseteq U$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad FV(e) \subseteq V}{U; V \vdash x := e\ \textbf{ok} \rightsquigarrow \{x\}}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x := e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{}{U; V \vdash \texttt{var}\ x;\ s\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad FV(e) \subseteq V}{U; V \vdash x := e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y \cdot s\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad FV(e) \subseteq V}{U; V \vdash x := e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y \cdot s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad FV(e) \subseteq V}{U; V \vdash x := e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x := e \texttt{ ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s \texttt{ ok} \rightsquigarrow W}{U; V \vdash \texttt{var } y \texttt{ ; } s \texttt{ ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1 \texttt{ ok} \rightsquigarrow W_1}{U; V \vdash \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow \emptyset}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{:}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow \emptyset}$$

$$\frac{}{U; V \vdash s_1; s_2\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow \emptyset}$$

$$\frac{U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1}{U; V \vdash s_1; s_2\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow \emptyset}$$

$$\frac{U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; (V \cup W_1) \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash s_1; s_2\ \textbf{ok} \rightsquigarrow}$$

Static Semantics Rules

$$\frac{}{U; V \vdash \texttt{skip}\ \textbf{ok} \rightsquigarrow \emptyset} \qquad \frac{x \in U \qquad \mathrm{FV}(e) \subseteq V}{U; V \vdash x\ \texttt{:=}\ e\ \textbf{ok} \rightsquigarrow \{x\}}$$

$$\frac{U \cup \{y\}; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{var}\ y\ \texttt{;}\ s\ \textbf{ok} \rightsquigarrow W \setminus \{y\}}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; V \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \textbf{ok} \rightsquigarrow W_1 \cap W_2}$$

$$\frac{\mathrm{FV}(e) \subseteq V \qquad U; V \vdash s\ \textbf{ok} \rightsquigarrow W}{U; V \vdash \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \textbf{ok} \rightsquigarrow \emptyset}$$

$$\frac{U; V \vdash s_1\ \textbf{ok} \rightsquigarrow W_1 \qquad U; (V \cup W_1) \vdash s_2\ \textbf{ok} \rightsquigarrow W_2}{U; V \vdash s_1; s_2\ \textbf{ok} \rightsquigarrow W_1 \cup W_2}$$

Dynamic Semantics

We will use big-step operational semantics. What are the sets of evaluable expressions and values here?

Dynamic Semantics

We will use big-step operational semantics. What are the sets of evaluable expressions and values here?

Evaluable Expressions:

Dynamic Semantics

We will use big-step operational semantics. What are the sets of evaluable expressions and values here?

Evaluable Expressions: A pair containing a statement to execute **and** a *state* σ.

Values:

Dynamic Semantics

We will use big-step operational semantics. What are the sets of evaluable expressions and values here?

Evaluable Expressions: A pair containing a statement to execute **and** a *state* σ.

Values: The final state that results from executing the statement.

Dynamic Semantics

We will use big-step operational semantics. What are the sets of evaluable expressions and values here?

Evaluable Expressions: A pair containing a statement to execute **and** a *state* σ.

Values: The final state that results from executing the statement.

States

A *state* is a mutable mapping from variables to their values. We use the following notation:

- To read a variable x from the state σ, we write $\sigma(x)$.
- To update an existing variable x to have value v inside the state σ, we write $(\sigma : x \mapsto v)$.
- To extend a state σ with a new, previously undeclared variable x, we write $\sigma \cdot x$. In such a state, x has undefined value.

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{(\sigma_1, s_1; s_2) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, s_1; s_2) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad (\sigma_1, \texttt{var}\ x \cdot s) \Downarrow$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto 0, s) \Downarrow (\sigma_2 : x \mapsto v)}{(\sigma_1, \texttt{var}\ x \cdot s) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto 0, s) \Downarrow (\sigma_2 : x \mapsto v)}{(\sigma_1, \texttt{var}\ x \cdot s) \Downarrow \sigma_2}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma_1, x, s) \Downarrow (\sigma_2, x)}{(\sigma_1, \texttt{var}\ x \cdot s) \Downarrow \sigma_2}$$

$$\frac{}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow}$$

$$\frac{}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma, x := x.s) \Downarrow (\sigma_2 : x)}{(\sigma_1, \texttt{var}\ x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow}$$

$$\frac{}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma, x := s) \Downarrow (\sigma_2 \; x)}{(\sigma_1, \texttt{var}\; x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if}\; e\; \texttt{then}\; s_1\; \texttt{else}\; s_2\; \texttt{fi}) \Downarrow}$$

$$\frac{}{(\sigma_1, \texttt{if}\; e\; \texttt{then}\; s_1\; \texttt{else}\; s_2\; \texttt{fi}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto v, s) \Downarrow (\sigma_2 : x \mapsto v_2)}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto x, s) \Downarrow (\sigma_2 : x \mapsto y)}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto v, s) \Downarrow (\sigma_2 : x \mapsto v_2) \qquad (\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto v, s) \Downarrow (\sigma_2 : x \mapsto v_2) \qquad (\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow} \qquad \frac{}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x, s) \Downarrow (\sigma_2 : x)}{(\sigma_1, \texttt{var}\ x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}) \Downarrow} \qquad \frac{}{(\sigma_1, \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x, s) \Downarrow (\sigma_2 : x)}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_1} \qquad \frac{}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto v, s) \Downarrow (\sigma_2 : x \mapsto v_2) \qquad (\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_1} \qquad \frac{\sigma_1 \vdash e \Downarrow v \quad v \neq 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x \mapsto v, s) \Downarrow (\sigma_2 : x \mapsto v')}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_1} \qquad \frac{\sigma_1 \vdash e \Downarrow v \quad v \neq 0 \qquad (\sigma_1, s) \Downarrow \sigma_2}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma, x := s) \Downarrow (\sigma_2 : x \mapsto \_)}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_1} \qquad \frac{\sigma_1 \vdash e \Downarrow v \quad v \neq 0 \qquad (\sigma_1, s) \Downarrow \sigma_2 \quad (\sigma_2, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_3}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow}$$

Evaluation Rules

We will assume we have defined a relation $\sigma \vdash e \Downarrow v$ for arithmetic expressions, much like in the previous lecture.

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{(\sigma_1, s_1) \Downarrow \sigma_2 \quad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{(\sigma : x, s) \Downarrow (\sigma_2 : x)}{(\sigma_1, \texttt{var } x \cdot s) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow v \qquad v \neq 0 \qquad (\sigma_1, s_1) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0 \qquad (\sigma_1, s_2) \Downarrow \sigma_2}{(\sigma_1, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}) \Downarrow \sigma_2}$$

$$\frac{\sigma_1 \vdash e \Downarrow 0}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_1} \qquad \frac{\sigma_1 \vdash e \Downarrow v \quad v \neq 0 \quad (\sigma_1, s) \Downarrow \sigma_2 \quad (\sigma_2, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_3}{(\sigma_1, \texttt{while } e \texttt{ do } s \texttt{ od}) \Downarrow \sigma_3}$$

Hoare Logic

To give you a taste of *axiomatic semantics*, and also how *formal verification* works, we are going to define what's called a *Hoare Logic* for TinyImp to allow us to prove properties of our program.

We write a *Hoare triple* judgement as:

$$\{\varphi\}\ s\ \{\psi\}$$

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

$$\varphi(\sigma) \wedge (\sigma, s) \Downarrow \sigma' \Rightarrow \psi(\sigma')$$

Proving Hoare Triples

To prove a Hoare triple like:

$$\{\text{true}\}$$
$$\textbf{var } i \cdot \textbf{var } m \cdot$$
$$i := 0;$$
$$m := 1;$$

$$\textbf{while } i < N \textbf{ do}$$
$$\quad i := i + 1;$$
$$\quad m := m \times i$$
$$\textbf{od}$$
$$\{m = N!\}$$

It is undesirable to look at the operational semantics derivations of this whole program to compute what the possible final states are for a given input state. Instead we shall define a set of rules to prove Hoare triples directly (called *a proof calculus*).

Hoare Rules

$$\frac{}{(\sigma, \mathsf{skip}) \Downarrow \sigma} \qquad \frac{}{\{\sigma\}\ \mathsf{skip}\ \{\sigma\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)}$$

Hoare Rules

$$\frac{}{(\sigma, \mathsf{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \mathsf{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)}$$

Hoare Rules

$$\frac{}{(\sigma, \mathsf{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \mathsf{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{}{\{\varphi\}\ s_1; s_2}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)}$$

Hoare Rules

$$\frac{}{(\sigma, \mathsf{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \mathsf{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)}$$

Hoare Rules

$$\frac{}{(\sigma, \mathsf{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \mathsf{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{x := e\ \{\varphi\}}$$

Hoare Rules

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \texttt{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

Continuing on, we can get rules for `if`, and `while` with a *loop invariant*:

$$\frac{}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}} \qquad \frac{}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}}$$

Hoare Rules

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \texttt{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

Continuing on, we can get rules for `if`, and `while` with a *loop invariant*:

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}} \qquad \frac{}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}}$$

Hoare Rules

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \texttt{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

Continuing on, we can get rules for `if`, and `while` with a *loop invariant*:

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \quad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}} \qquad \frac{}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}}$$

Hoare Rules

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \texttt{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

Continuing on, we can get rules for `if`, and `while` with a *loop invariant*:

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \quad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}} \qquad \frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}}$$

Hoare Rules

$$\frac{}{(\sigma, \texttt{skip}) \Downarrow \sigma} \qquad \frac{}{\{\varphi\}\ \texttt{skip}\ \{\varphi\}}$$

$$\frac{(\sigma_1, s_1) \Downarrow \sigma_2 \qquad (\sigma_2, s_2) \Downarrow \sigma_3}{(\sigma_1, s_1; s_2) \Downarrow \sigma_3} \qquad \frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\sigma \vdash e \Downarrow v}{(\sigma, x := e) \Downarrow (\sigma : x \mapsto v)} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}}$$

Continuing on, we can get rules for `if`, and `while` with a *loop invariant*:

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \quad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}} \qquad \frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

Consequence

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

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Consequence

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

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

This is the only rule that is not directed entirely by syntax. This means a Hoare logic proof need not look like a derivation tree. Instead, we can sprinkle assertions through our program and specially note uses of the consequence rule.

Factorial Example

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

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

$\{n \geq 0\}$

var i · **var** m ·

$i := 0;$

$m := 1;$

while $i < N$ **do**

$i := i + 1;$

$m := m \times i$

od

$\{m = N!\}$

Factorial Example

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

$$\{1 = 0! \wedge 0 \leq N\}$$
var $i \cdot$ **var** $m \cdot$
$\quad i := 0;$
$\quad m := 1;$
$$\{m = i! \wedge i \leq N\}$$
while $i < N$ **do**
$$\{m = i! \wedge i < N\}$$
$\quad i := i + 1;$
$$\{m = (i-1)! \wedge i \leq N\}$$
$\quad m := m \times i$
$$\{m = i! \wedge i \leq N\}$$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Factorial Example

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

$$\{\top\}$$
var $i \cdot$ **var** $m \cdot$
$\quad i := 0;$
$\quad m := 1;$
$\{m = i!\}$
while $i < N$ **do**
$\quad i := i + 1;$
$\quad m := m \times i$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\varphi \wedge e \; s_1 \; \{\psi\} \qquad \{\varphi \wedge \neg e\} \; s_2 \; \{\psi\}}{\{\varphi\} \; \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi} \; \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\} \; s \; \{\varphi\}}{\{\varphi\} \; \texttt{while } e \texttt{ do } s \texttt{ od} \; \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\} \; s_1 \; \{\alpha\} \qquad \{\alpha\} \; s_2 \; \{\psi\}}{\{\varphi\} \; s_1; s_2 \; \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\} \; s \; \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\} \; s \; \{\psi\}}$$

Factorial Example

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

$\{1=0!\}$
var $i \cdot$ **var** $m \cdot$
$i := 0;$
$m := 1;$
$\{m = i!\}$
while $i < N$ **do**
$i := i + 1;$
$m := m \times i;$
$\{m = i!\}$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Factorial Example

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

$$\{1 = 0! \wedge 0 \leq N\}$$
var $i \cdot$ **var** $m \cdot$
$$i := 0;$$
$$m := 1;$$
$$\{m = i! \wedge i \leq N\}$$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$$i := i + 1;$$
$$m := m \times i$$
$$\{m = i!\}$$
od $\{m = i! \wedge i = N\}$
$$\{m = N!\}$$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Factorial Example

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

$$\{\top\}$$
var $i \cdot$ **var** $m \cdot$
$\quad i := 0;$
$\quad m := 1;$
$\{m = i! \wedge i \leq N\}$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$\quad i := i + 1;$
$\quad \{m \times i = i!\}$
$\quad m := m \times i;$
$\quad \{m = i!\}$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Factorial Example

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

$$\{1=0!\}$$
var $i \cdot$ **var** $m \cdot$
$i := 0;$
$m := 1;$
$\{m = i!\}$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$\{m \times (i+1) = (i+1)!\}$
$i := i + 1;$
$\{m \times i = i!\}$
$m := m \times i$
$\{m = i!\}$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

Factorial Example

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

$$\{n \geq 0\}$$
var $i \cdot$ **var** $m \cdot$
$i := 0;$
$m := 1;$
$\{m = i! \wedge i \leq N\}$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$\{m \times (i+1) = (i+1)!\}$
$i := i + 1;$
$\{m \times i = i!\}$
$m := m \times i$
$\{m = i!\}$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

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

Factorial Example

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

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

$\{\top\}$
var $i \cdot$ **var** $m \cdot$
$i := 0;$
$m := 1; \{m = i!\}$
$\{m = i!\}$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$\{m \times (i+1) = (i+1)!\}$
$i := i + 1;$
$\{m \times i = i!\}$
$m := m \times i$
$\{m = i!\}$
od $\{m = i! \wedge i = N\}$
$\{m = N!\}$

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

Factorial Example

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

$$\{1 = i!\}\ m := 1;\ \{m = i!\}$$

$$\{m = i!\}\ i := i + 1;\ \{m = (i-1)!\}$$

$$\{m = i!\}\ m := m \times i;\ \{m = (i-1)!\}$$

$$\{m = i! \wedge i < N\}\ i := i + 1;\ m := m \times i;\ \{m = (i-1)!\}$$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \texttt{ fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while } e \texttt{ do } s \texttt{ od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

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

Factorial Example

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

$$\{1 = i!\}$$
$$\{1 = i! \wedge 1 = 1\}$$
$$\{m = i! \wedge 1 = 1\}$$
var $i \cdot$ **var** $m \cdot$
$$i := 0; \{1 = i!\}$$
$$\{1 = i!\}\ m := 1; \{m = i!\}$$
$$\{m = i!\}$$
while $i < N$ **do** $\{m = i! \wedge i < N\}$
$$\{m \times (i+1) = (i+1)!\}$$
$$i := i + 1;$$
$$\{m \times i = i!\}$$
$$m := m \times i$$
$$\{m = i!\}$$
od $\{m = i! \wedge i = N\}$
$$\{m = N!\}$$

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

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

Factorial Example

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

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

$$\frac{\{\varphi \wedge e\}\ s_1\ \{\psi\} \qquad \{\varphi \wedge \neg e\}\ s_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{fi}\ \{\psi\}}$$

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

$$\frac{\{\varphi \wedge e\}\ s\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ e\ \texttt{do}\ s\ \texttt{od}\ \{\varphi \wedge \neg e\}}$$

$$\frac{\{\varphi\}\ s_1\ \{\alpha\} \qquad \{\alpha\}\ s_2\ \{\psi\}}{\{\varphi\}\ s_1; s_2\ \{\psi\}}$$

$$\frac{\varphi \Rightarrow \alpha \qquad \{\alpha\}\ s\ \{\beta\} \qquad \beta \Rightarrow \psi}{\{\varphi\}\ s\ \{\psi\}}$$

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