

## CS477 Formal Software Development Methods

Elsa L Gunter  
2112 SC, UIUC  
egunter@illinois.edu

<http://courses.engr.illinois.edu/cs477>

Slides based in part on previous lectures by Mahesh Vishwanathan, and by Gul Agha

February 26, 2014

## $\alpha$ -equivalence

- $\psi \stackrel{\alpha}{\equiv} \psi$
- If  $\psi_1 \stackrel{\alpha}{\equiv} \psi_2$  then  $\psi_2 \stackrel{\alpha}{\equiv} \psi_1$ .
- If  $\psi_1 \stackrel{\alpha}{\equiv} \psi_2$  and  $\psi_2 \stackrel{\alpha}{\equiv} \psi_3$  then  $\psi_1 \stackrel{\alpha}{\equiv} \psi_3$
- If  $x \notin \text{fv}(\psi)$  and  $y \notin \text{fv}(\psi)$  then  $\psi \stackrel{\alpha}{\equiv} \psi[x \leftrightarrow y]$ .
- If  $\psi_i \stackrel{\alpha}{\equiv} \psi'_i$  for  $i = 1, 2$  then
  - $(\psi_1) \stackrel{\alpha}{\equiv} (\psi'_1) \quad \neg\psi_1 \stackrel{\alpha}{\equiv} \neg\psi'_1$
  - $\psi_1 \otimes \psi_2 \stackrel{\alpha}{\equiv} \psi'_1 \otimes \psi'_2$  for  $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
  - $Qz. \psi_1 \stackrel{\alpha}{\equiv} Qz. \psi'_1$  for  $Q \in \{\forall, \exists\}$

## $\alpha$ -equivalence: Example

$$(x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))$$

$$\stackrel{\alpha}{\equiv} (x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee (z \geq w)))$$

$$(x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))$$

$$\stackrel{\alpha}{\equiv} (x > 3 \wedge (\exists w. (\forall y. y \geq (w - x)) \vee (z \geq w)))$$

## Proof Rules

Will give Sequent version of Natural Deduction rules  
All rules from Propositional Logic included

$$\frac{\Gamma \vdash \psi'[t/x]}{\Gamma \vdash \exists x. \psi} \text{ Ex I}$$

provided  $\psi \stackrel{\alpha}{\equiv} \psi'$

$$\frac{\Gamma \vdash \exists x. \psi \quad \Gamma \cup \{\psi[y/x]\} \vdash \varphi}{\Gamma \vdash \varphi} \text{ Ex E}$$

provided  
 $y \notin \text{fv}(\varphi) \cup (\text{fv}(\psi) \setminus \{x\}) \cup \bigcup_{\psi' \in \Gamma} \text{fv}(\psi')$

$$\frac{\Gamma \vdash \psi[y/x]}{\Gamma \vdash \forall x. \psi} \text{ All I}$$

provided  
 $y \notin (\text{fv}(\psi) \setminus \{x\}) \cup \bigcup_{\psi' \in \Gamma} \text{fv}(\psi')$

$$\frac{\Gamma \vdash \forall x. \psi \quad \Gamma \cup \{\psi'[t/x]\} \vdash \varphi}{\Gamma \vdash \varphi} \text{ All E}$$

provided  $\psi \stackrel{\alpha}{\equiv} \psi'$

## Example

Show

$$\frac{}{\{\} \vdash (\exists x. \forall y. x \leq y) \Rightarrow (\forall x. \exists y. y \leq x)}$$

## Example

Show

$$\frac{\frac{}{\{\} \vdash (\exists x. \forall y. x \leq y)} \quad \frac{}{\{\} \vdash \forall x. \exists y. y \leq x}}{\{\} \vdash (\exists x. \forall y. x \leq y) \Rightarrow (\forall x. \exists y. y \leq x)} \text{ Imp I}$$







## Floyd-Hoare Logic

- Used to formally prove a property (**post-condition**) of the **state** (the values of the program variables) after the execution of program, assuming another property (**pre-condition**) of the state holds before execution

## Floyd-Hoare Logic

- Goal: Derive statements of form

$$\{P\} C \{Q\}$$

- $P, Q$  logical statements about state,  $P$  precondition,  $Q$  postcondition,  $C$  program
- Example:

$$\{x = 1\} x := x + 1 \{x = 2\}$$

## Floyd-Hoare Logic

- Approach:** For each type of language statement, give an axiom or inference rule stating how to derive assertions of form

$$\{P\} C \{Q\}$$

where  $C$  is a statement of that type

- Compose axioms and inference rules to build proofs for complex programs

## Partial vs Total Correctness

- An expression  $\{P\} C \{Q\}$  is a **partial correctness** statement
- For **total correctness** must also prove that  $C$  terminates (i.e. doesn't run forever)
  - Written:  $[P] C [Q]$
- Will only consider partial correctness here

## Simple Imperative Language

- We will give rules for simple imperative language

```
 $\langle \text{command} \rangle ::= \langle \text{variable} \rangle := \langle \text{term} \rangle$   
|  $\langle \text{command} \rangle; \dots; \langle \text{command} \rangle$   
|  $\text{if } \langle \text{statement} \rangle \text{ then } \langle \text{command} \rangle \text{ else } \langle \text{command} \rangle$   
|  $\text{while } \langle \text{statement} \rangle \text{ do } \langle \text{command} \rangle$ 
```

- Could add more features, like for-loops

## Substitution

- Notation:  $P[e/v]$  (sometimes  $P[v \rightarrow e]$ )
- Meaning: Replace every  $v$  in  $P$  by  $e$
- Example:

$$(x + 2)[y - 1/x] = ((y - 1) + 2)$$

## The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{ \quad ? \quad \} x := y \{x = 2\}}$$

## The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{\square = 2\} x := y \{\square = 2\}}$$

## The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Example:

$$\frac{}{\{\square = 2\} x := y \{\square = 2\}}$$

## The Assingment Rule

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

Examples:

$$\frac{}{\{y = 2\} x := y \{x = 2\}}$$

$$\frac{}{\{y = 2\} x := 2 \{y = x\}}$$

$$\frac{}{\{x + 1 = n + 1\} x := x + 1 \{x = n + 1\}}$$

$$\frac{}{\{2 = 2\} x := 2 \{x = 2\}}$$

## The Assingment Rule – Your Turn

- What is the weakest precondition of

$$x := x + y \{x + y = wx\}?$$

$$\left\{ \begin{array}{c} \quad ? \quad \\ x := x + y \\ \{x + y = wx\} \end{array} \right\}$$

## The Assingment Rule – Your Turn

- What is the weakest precondition of

$$x := x + y \{x + y = wx\}?$$

$$\left\{ \begin{array}{c} (x + y) + y = w(x + y) \\ x := x + y \\ \{x + y = wx\} \end{array} \right\}$$

## Precondition Strengthening

$$\frac{(P \Rightarrow P') \{P'\} C \{Q\}}{\{P\} C \{Q\}}$$

- Meaning: If we can show that  $P$  implies  $P'$  (i.e.  $(P \Rightarrow P')$  and we can show that  $\{P'\} C \{Q\}$ , then we know that  $\{P\} C \{Q\}$
- $P$  is **stronger** than  $P'$  means  $P \Rightarrow P'$

## Precondition Strengthening

- Examples:

$$\frac{x = 3 \Rightarrow x < 7 \quad \{x < 7\} x := x + 3 \quad \{x < 10\}}{\{x = 3\} x := x + 3 \quad \{x < 10\}}$$

$$\frac{\text{True} \Rightarrow (2 = 2) \quad \{2 = 2\} x := 2 \quad \{x = 2\}}{\{\text{True}\} x := 2 \quad \{x = 2\}}$$

$$\frac{x = n \Rightarrow x + 1 = n + 1 \quad \{x + 1 = n + 1\} x := x + 1 \quad \{x = n + 1\}}{\{x = n\} x := x + 1 \quad \{x = n + 1\}}$$

## Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}{\{x = 3\} x := x * x \quad \{x < 25\}}$$

$$\frac{\{x = 3\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}$$

$$\frac{\{x * x < 25\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}$$

## Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}{\{x = 3\} x := x * x \quad \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}$$

$$\frac{\{x * x < 25\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}$$

## Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}{\{x = 3\} x := x * x \quad \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}} \text{ NO}$$

$$\frac{\{x * x < 25\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}$$

## Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}}{\{x = 3\} x := x * x \quad \{x < 25\}} \text{ YES}$$

$$\frac{\{x = 3\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}} \text{ NO}$$

$$\frac{\{x * x < 25\} x := x * x \quad \{x < 25\}}{\{x > 0 \wedge x < 5\} x := x * x \quad \{x < 25\}} \text{ YES}$$