

CS477 Formal Software Dev 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 19, 2020

First Order Logic vs Propositional Logic

First Order Logic extends Propositional Logic with

- Non-boolean constants
- Variables
- Functions and relations (or predicates, more generally)
- Quantification of variables

Sample first order formula:

$$\forall x. \exists y. x < y \wedge y \leq x + 1$$

Reference: Peled, *Software Reliability Methods*, Chapter 3

Start with **signature**:

$$\mathcal{G} = (V, F, af, R, ar)$$

- V a countably infinite set of *variables*
- F finite set of function symbols
- $af : F \rightarrow \mathbb{N}$ gives the *arity*, the number of arguments for each function Constant c is a function symbol of arity 0 ($af(c) = 0$)
- R finite set of relation symbols
- $ar : R \rightarrow \mathbb{N}$, the arity for each relation symbol
 - Assumes $= \in R$ and $ar(=) = 2$

Terms over Signature

Terms t are expressions built over a signature (V, F, af, R, ar)

$$\begin{array}{l} t ::= v \qquad v \in V \\ \quad | f(t_1, \dots, t_n) \quad f \in F \text{ and } n = af(f) \end{array}$$

- **Example:** $add(1, abs(x))$ where $add, abs, 1 \in F$; $x \in V$
- For constant c write c instead of $c()$
- Will write $s + t$ instead of $+(s, t)$
 - Similarly for other common infixes (e.g. $+$, $-$, $*$, \dots)

Meaning of terms starts with a **structure**:

$$S = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$$

where

- $\mathcal{G} = (V, F, af, R, ar)$ a signature,
- \mathcal{D} and *domain* on interpretation
- \mathcal{F} set of functions over \mathcal{D} ; $\mathcal{F} \subseteq \bigcup_{n \geq 0} \mathcal{D}^n \rightarrow \mathcal{D}$
 - **Note:** \mathcal{F} can contain elements of \mathcal{D} since $\mathcal{D} = (\mathcal{D}^0 \rightarrow \mathcal{D})$
- $\phi : F \rightarrow \mathcal{F}$ where if $\phi(f) \in (\mathcal{D}^n \rightarrow \mathcal{D})$ then $n = af(f)$
- \mathcal{R} set of relations over \mathcal{D} ; $\mathcal{R} \subseteq \bigcup_{n \geq 1} \mathcal{P}(\mathcal{D}^n)$
- $\rho : R \rightarrow \mathcal{R}$ where if $\rho(r) \subseteq \mathcal{D}^n$ then $n = ar(r)$

Assignments

V set of variables, \mathcal{D} domain of interpretation

An **assignment** is a function $a : V \rightarrow \mathcal{D}$

Example:

$$V = \{w, x, y, z\}$$

$$a = \{w \mapsto 3.14, x \mapsto -2.75, y \mapsto 13.9, z \mapsto -25.3\}$$

- Assignment is a fixed association of values to variables; not “update-able”

Interpretation of Terms

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{T}_a of a term t is defined by structural induction on terms:

- $\mathcal{T}_a(v) = a(v)$ for $v \in V$
- $\mathcal{T}_a(f(t_1, \dots, t_n)) = \phi(f)(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n))$

Example of Interpretation

- $V = \{w, x, y, z\}$, $\mathcal{D} = \mathbb{R}$
- $1, \text{add}, \text{abs} \in F$, constant 1 , and functions (in \mathcal{F}) for addition and absolute value respectively
- $a = \{w \mapsto 3.14, x \mapsto -2.75, y \mapsto 13.9, z \mapsto -25.3\}$

$$\begin{aligned}\mathcal{T}_a(\text{add}(1, \text{abs}(x))) &= (\mathcal{T}_a(1)) + (\mathcal{T}_a(\text{abs}(x))) \\ &= 1.0 + (\mathcal{T}_a(\text{abs}(x))) \\ &= 1.0 + |\mathcal{T}_a(x)| \\ &= 1.0 + |a(x)| \\ &= 1.0 + |-2.75| \\ &= 1.0 + 2.75 \\ &= 3.75\end{aligned}$$

First-Order Formulae

First-order formulae built from terms using relations, logical connectives, quantifiers:

$$\begin{aligned} \text{form} ::= & \text{true} \mid \text{false} \\ & \mid r(t_1, \dots, t_n) \quad r \in R, t_i \text{ terms, } n = ar(r) \\ & \mid (\text{form}) \mid \neg \text{form} \\ & \mid \text{form} \wedge \text{form} \\ & \mid \text{form} \vee \text{form} \\ & \mid \text{form} \Rightarrow \text{form} \\ & \mid \forall v. \text{form} \\ & \mid \exists v. \text{form} \end{aligned}$$

Note: Scope of quantifiers as far to right as possible

$$\begin{aligned} \forall x.(x > y) \wedge (2 > x) & \text{ same as } \forall x.((x > y) \wedge (2 > x)) \\ & \text{ not same as } (\forall x.(x > y)) \wedge (2 > x) \end{aligned}$$

Subformulae

- A **subformula** of formula ψ is a formula that occurs in ψ
 - More rigorous definition by structural induction on formulae
 - ψ subformula of ψ
 - Use **proper subformula** to exclude ψ
- Write $\bigwedge_{i=1,\dots,n} \psi_i$ for $\psi_1 \wedge \dots \wedge \psi_n$
 - ψ_i called a **conjunct**
- Write $\bigvee_{i=1,\dots,n} \psi_i$ for $\psi_1 \vee \dots \vee \psi_n$
 - ψ_i called a **disjunct**

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$
- $\mathcal{M}_a((\psi)) = \mathcal{M}_a(\psi)$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$
- $\mathcal{M}_a((\psi)) = \mathcal{M}_a(\psi)$
- $\mathcal{M}_a(\neg\psi) = \mathbf{T}$ if $\mathcal{M}_a(\psi) = \mathbf{F}$ and $\mathcal{M}_a(\neg\psi) = \mathbf{F}$ if $\mathcal{M}_a(\psi) = \mathbf{T}$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$
- $\mathcal{M}_a((\psi)) = \mathcal{M}_a(\psi)$
- $\mathcal{M}_a(\neg\psi) = \mathbf{T}$ if $\mathcal{M}_a(\psi) = \mathbf{F}$ and $\mathcal{M}_a(\neg\psi) = \mathbf{F}$ if $\mathcal{M}_a(\psi) = \mathbf{T}$
- $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{T}$ and $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{F}$ otherwise

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$
- $\mathcal{M}_a((\psi)) = \mathcal{M}_a(\psi)$
- $\mathcal{M}_a(\neg\psi) = \mathbf{T}$ if $\mathcal{M}_a(\psi) = \mathbf{F}$ and $\mathcal{M}_a(\neg\psi) = \mathbf{F}$ if $\mathcal{M}_a(\psi) = \mathbf{T}$
- $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{T}$ and $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{F}$ otherwise
- $\mathcal{M}_a(\psi_1 \vee \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{T}$ or $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \vee \psi_2) = \mathbf{F}$ otherwise

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

For given assignment $a : V \rightarrow \mathcal{D}$, the **interpretation** \mathcal{M}_a of a formula ψ assigning a value in $\{\mathbf{T}, \mathbf{F}\}$ is defined by structural induction on formulae:

- $\mathcal{M}_a(\text{true}) = \mathbf{T}$ $\mathcal{M}_a(\text{false}) = \mathbf{F}$
- $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{T}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \in \rho(r)$ and
 $\mathcal{M}_a(r(t_1, \dots, t_n)) = \mathbf{F}$ if $(\mathcal{T}_a(t_1), \dots, \mathcal{T}_a(t_n)) \notin \rho(r)$
- $\mathcal{M}_a((\psi)) = \mathcal{M}_a(\psi)$
- $\mathcal{M}_a(\neg\psi) = \mathbf{T}$ if $\mathcal{M}_a(\psi) = \mathbf{F}$ and $\mathcal{M}_a(\neg\psi) = \mathbf{F}$ if $\mathcal{M}_a(\psi) = \mathbf{T}$
- $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{T}$ and $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \wedge \psi_2) = \mathbf{F}$ otherwise
- $\mathcal{M}_a(\psi_1 \vee \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{T}$ or $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \vee \psi_2) = \mathbf{F}$ otherwise
- $\mathcal{M}_a(\psi_1 \Rightarrow \psi_2) = \mathbf{T}$ if $\mathcal{M}_a(\psi_1) = \mathbf{F}$ or $\mathcal{M}_a(\psi_2) = \mathbf{T}$, and
 $\mathcal{M}_a(\psi_1 \Rightarrow \psi_2) = \mathbf{F}$ otherwise

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

Let

$$a + [v \mapsto d] (w) = \begin{cases} d & \text{if } w = v \\ a(w) & \text{if } w \neq v \end{cases}$$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

Let

$$(a + [v \mapsto d])(w) = \begin{cases} d & \text{if } w = v \\ a(w) & \text{if } w \neq v \end{cases}$$

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

Let

$$a + [v \mapsto d](w) = \begin{cases} d & \text{if } w = v \\ a(w) & \text{if } w \neq v \end{cases}$$

- $\mathcal{M}_a(\forall v.\psi) = \mathbf{T}$ if for every $d \in \mathcal{D}$ we have $\mathcal{M}_{a+[v \mapsto d]}(\psi) = \mathbf{T}$, and $\mathcal{M}_a(\forall v.\psi) = \mathbf{F}$ otherwise

Interpretation of Formulae

Fix structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

Let

$$a + [v \mapsto d](w) = \begin{cases} d & \text{if } w = v \\ a(w) & \text{if } w \neq v \end{cases}$$

- $\mathcal{M}_a(\forall v.\psi) = \mathbf{T}$ if for every $d \in \mathcal{D}$ we have $\mathcal{M}_{a+[v \mapsto d]}(\psi) = \mathbf{T}$, and $\mathcal{M}_a(\forall v.\psi) = \mathbf{F}$ otherwise
- $\mathcal{M}_a(\exists v.\psi) = \mathbf{T}$ if there exists $d \in \mathcal{D}$ such that $\mathcal{M}_{a+[v \mapsto d]}(\psi) = \mathbf{T}$, and $\mathcal{M}_a(\exists v.\psi) = \mathbf{F}$ otherwise

Modeling First-order Formulae

Given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$ where $\mathcal{G} = (V, F, af, R, ar)$

- $(\mathcal{S}, \mathcal{M})$ **model** for first-order language over signature \mathcal{G}
- Truth of formulae in language over signature \mathcal{G} depends on structure \mathcal{S}
- Assignment a **models** ψ , or a **satisfies** ψ , or $a \models^{\mathcal{S}} \psi$ if $\mathcal{M}_a(\psi) = \mathbf{T}$
- ψ is **valid** for \mathcal{S} if $a \models^{\mathcal{S}} \psi$ for some a .
- \mathcal{S} is a **model** of ψ , written $\models^{\mathcal{S}} \psi$ if every assignment for \mathcal{S} satisfies ψ .
- ψ is **valid**, or a **tautology** if ψ valid for every mode. Write $\models \psi$
- ψ_1 **logically equivalent** to ψ_2 if for all structures \mathcal{S} and assignments a , $a \models^{\mathcal{S}} \psi_1$ iff $a \models^{\mathcal{S}} \psi_2$

Examples

- Assignment $\{x \mapsto 0\}$ satisfies $\exists y. x < y$ valid in interval $[0, 1]$; assignment $\{x \mapsto 1\}$ doesn't
- $\forall x. \exists y. x < y$ valid in \mathbb{N} and \mathbb{R} , but not interval $[0, 1]$
- $(\exists x. \forall y. (y \leq x)) \Rightarrow (\forall y. \exists x. (y \leq x))$ tautology
 - Why?

Sample Tautologies

All instances of propositional tautologies

Sample Tautologies

All instances of propositional tautologies

$$\models (\exists x.\forall y.(y \leq x)) \Rightarrow (\forall y.\exists x.(y \leq x))$$

Sample Tautologies

All instances of propositional tautologies

$$\models (\exists x.\forall y.(y \leq x)) \Rightarrow (\forall y.\exists x.(y \leq x))$$

$$\models ((\forall x.\forall y.\psi) \Leftrightarrow (\forall y.\forall x.\psi))$$

Sample Tautologies

All instances of propositional tautologies

$$\models (\exists x.\forall y.(y \leq x)) \Rightarrow (\forall y.\exists x.(y \leq x))$$

$$\models ((\forall x.\forall y.\psi) \Leftrightarrow (\forall y.\forall x.\psi))$$

$$\models ((\forall x.\psi) \Rightarrow (\exists x.\psi))$$

Sample Tautologies

All instances of propositional tautologies

$$\models (\exists x.\forall y.(y \leq x)) \Rightarrow (\forall y.\exists x.(y \leq x))$$

$$\models ((\forall x.\forall y.\psi) \Leftrightarrow (\forall y.\forall x.\psi))$$

$$\models ((\forall x.\psi) \Rightarrow (\exists x.\psi))$$

$$\models (\forall x.\psi_1 \wedge \psi_2) \Leftrightarrow ((\forall x.\psi_1) \wedge (\forall x.\psi_2))$$

Sample Tautologies

All instances of propositional tautologies

$$\models (\exists x.\forall y.(y \leq x)) \Rightarrow (\forall y.\exists x.(y \leq x))$$

$$\models ((\forall x.\forall y.\psi) \Leftrightarrow (\forall y.\forall x.\psi))$$

$$\models ((\forall x.\psi) \Rightarrow (\exists x.\psi))$$

$$\models (\forall x.\psi_1 \wedge \psi_2) \Leftrightarrow ((\forall x.\psi_1) \wedge (\forall x.\psi_2))$$

$$\models (\exists x.\psi_1 \wedge \psi_2) \Rightarrow ((\exists x.\psi_1) \wedge (\exists x.\psi_2))$$

Free Variables: Terms

Informally: **free variables** of an expression are variables that have an occurrence in an expression that is not bound. Written $fv(e)$ for expression e

Free variables of terms defined by structural induction over terms; written

- $fv(x) = \{x\}$
- $fv(f(t_1, \dots, t_n)) = \bigcup_{i=1, \dots, n} fv(t_i)$

Note:

- Free variables of term just variables occurring in term; no bound variables
- No free variables in constants
- **Example:** $fv(add(1, abs(x))) = \{x\}$

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, term t over \mathcal{G} , and a and b assignments. If for every $x \in \text{fv}(t)$ we have $a(x) = b(x)$ then $\mathcal{T}_a(t) = \mathcal{T}_b(a)$.

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, formula ψ over \mathcal{G} , and a and b assignments. If for every $x \in \text{fv}(\psi)$ we have $a(x) = b(x)$ then $\mathcal{M}_a(\psi) = \mathcal{M}_b(\psi)$.

Syntactic Substitution versus Assignment Update

- When interpreting universal quantification ($\forall x. \psi$), wanted to check interpretation of every instance of ψ where v was replaced by element of semantic domain \mathcal{D}
- How: semantically - interpret ψ with assignment updated by $v \mapsto d$ for every $d \in \mathcal{D}$
- Syntactically?
- Answer: substitution

Substitution in Terms

- Substitution of term t for variable x in term s (written $s[t/x]$) gotten by replacing every instance of x in s by t
 - x called **redex**; t called **residue**
- Yields *instance* of s

Formally defined by structural induction on terms:

- $x[t/x] = t$
- $y[t/x] = y$ for variable y where $y \neq x$
- $f(t_1, \dots, t_n)[t/x] = f(t_1[t/x], \dots, t_n[t/x])$

Example: $(add(1, abs(x)))[add(x, y)/x] = add(1, abs(add(x, y)))$

Substitution in Formulae: Problems

- Want to define by structural induction, similar to terms
- Quantifiers must be handled with care
 - Substitution only replaces **free** occurrences of variable

Example:

$$\begin{aligned} & (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[x + 2/z] = \\ & (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (x + 2 \geq y))) \end{aligned}$$

- Need to avoid *free variable capture*

Example Problem:

$$\begin{aligned} & (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[x + y/z] \neq \\ & (x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (x + y \geq y))) \end{aligned}$$

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, variable x , terms s and t over \mathcal{G} , and a assignment. Let $b = a[x \mapsto \mathcal{T}_a(t)]$. Then $\mathcal{T}_a(s[t/x]) = \mathcal{T}_b(s)$.

Substitution in Formulae: Two Approaches

- When quantifier would capture free variable of redex, can't substitute in formula as is
- Solution 1: Make substitution partial function – undefined in this case
- Solution 2: Define equivalence relation based on renaming bound variables; define substitution on equivalence classes
- Will take Solution 1 here
- Still need definition of equivalence up to renaming bound variables

Substitution in Formulae

- Defined by structural induction; uses substitution in terms
- Read equations below as saying left is not defined if any expression on right not defined
- $\text{true}[t/x] = \text{true}$ $\text{false}[t/x] = \text{false}$
- $r(t_1, \dots, t_n)[t/x] = r((t_1[t/x], \dots, t_n[t/x]))$
- $(\psi)[t/x] = (\psi[t/x])$ $(\neg\psi)[t/x] = \neg(\psi[t/x])$
- $(\psi_1 \otimes \psi_2)[t/x] = (\psi_1[t/x]) \otimes (\psi_2[t/x])$ for $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $(Qx. \psi)[t/x] = Qx. \psi$ for $Q \in \{\forall, \exists\}$
- $(Qy. \psi)[t/x] = Qy. (\psi[t/x])$ if $x \neq y$ and $y \notin \text{fv}(t)$ for $Q \in \{\forall, \exists\}$
- $(Qy. \psi)[t/x]$ not defined if $x \neq y$ and $y \in \text{fv}(t)$ for $Q \in \{\forall, \exists\}$

Examples

$(x > 3 \wedge (\exists y. (\forall z. z \geq (y - x)) \vee (z \geq y)))[x + y/z]$ not defined

$$\begin{aligned} & (x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee (z \geq w)))[x + y/z] = \\ & (x > 3 \wedge (\exists w. (\forall z. z \geq (w - x)) \vee ((x + y) \geq y))) \end{aligned}$$

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, formula ψ over \mathcal{G} , and a assignment. If $\psi[t/x]$ defined, then $a \models^{\mathcal{S}} \psi[t/x]$ if and only if $a[x \mapsto \mathcal{T}_a(t)] \models^{\mathcal{S}} \psi$

Renaming by Swapping: Terms

Define the **swapping** of two variables in a term $t[x \leftrightarrow y]$ by structural induction on terms:

- $x[x \leftrightarrow y] = y$ and $y[x \leftrightarrow y] = x$
- $z[x \leftrightarrow y] = z$ for z a variable, $z \neq x$, $z \neq y$
- $f(t_1, \dots, t_n)[x \leftrightarrow y] = f(t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y])$

Examples:

$$\begin{aligned} \text{add}(1, \text{abs}(\text{add}(x, y)))[x \leftrightarrow y] &= \text{add}(1, \text{abs}(\text{add}(y, x))) \\ \text{add}(1, \text{abs}(\text{add}(x, y)))[x \leftrightarrow z] &= \text{add}(1, \text{abs}(\text{add}(z, y))) \end{aligned}$$

Renaming by Swapping: Terms

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, variables x and y , term t over \mathcal{G} , and a assignment. Let $b = a[x \mapsto a(y)][y \mapsto a(x)]$. Then $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$

Renaming by Swapping: Terms

Proof.

By structural induction on terms, suffices to show theorem for the case where t variable, and case $t = f(t_1, \dots, t_n)$, assuming result for t_1, \dots, t_n

- Case: t variable

- Subcase: $t = x$. Then $\mathcal{T}_a(x[x \leftrightarrow y]) = \mathcal{T}_a(y) = a(y)$ and $\mathcal{T}_b(x) = b(x) = a[x \mapsto a(y)][y \mapsto a(x)](x) = a[x \mapsto \mathcal{T}_a(y)](x) = a(y)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$
- Subcase: $t = y$. Then $\mathcal{T}_a(y[x \leftrightarrow y]) = \mathcal{T}_a(x) = a(x)$ and $\mathcal{T}_b(y) = b(y) = a[x \mapsto a(y)][y \mapsto a(x)](x) = a(x)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$
- Subcase: $t = z$ variable, $z \neq x$ and $z \neq y$. Then $\mathcal{T}_a(z[x \leftrightarrow y]) = \mathcal{T}_a(z) = a(z)$ and $\mathcal{T}_b(z) = b(z) = a[x \mapsto a(y)][y \mapsto a(x)](z) = a[x \mapsto \mathcal{T}_a(y)](z) = a(z)$ so $\mathcal{T}_a(t[x \leftrightarrow y]) = \mathcal{T}_b(t)$

Renaming by Swapping: Terms

Proof.

- Case: $t = f(t_1, \dots, t_n)$. Assume $\mathcal{T}_a(t_i[x \leftrightarrow y]) = \mathcal{T}_b(t_i)$ for $i = 1, \dots, n$. Then

$$\begin{aligned}\mathcal{T}_a(t[x \leftrightarrow y]) &= \mathcal{T}_a(f(t_1, \dots, t_n)[x \leftrightarrow y]) \\ &= \mathcal{T}_a(f(t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y])) \\ &= \phi(f)(\mathcal{T}_a(t_1[x \leftrightarrow y]), \dots, \mathcal{T}_a(t_n[x \leftrightarrow y])) \\ &= \phi(f)(\mathcal{T}_b(t_1), \dots, \mathcal{T}_b(t_n)) \\ &\quad \text{since } \mathcal{T}_a(t_i[x \leftrightarrow y]) = \mathcal{T}_b(t_i) \text{ for } i = 1, \dots, n \\ &= \mathcal{T}_b(f(t_1, \dots, t_n)) \\ &= \mathcal{T}_b(t) \quad \square\end{aligned}$$

Renaming by Swapping: Formulae

Define the **swapping** of two variables in a formula $\psi[x \leftrightarrow y]$ by structural induction, using swapping on terms:

- $\text{true}[x \leftrightarrow y] = \text{true}$ $\text{false}[x \leftrightarrow y] = \text{false}$
- $r(t_1, \dots, t_n)[x \leftrightarrow y] = r((t_1[x \leftrightarrow y], \dots, t_n[x \leftrightarrow y]))$
- $(\psi)[x \leftrightarrow y] = (\psi[x \leftrightarrow y])$ $(\neg\psi)[x \leftrightarrow y] = \neg(\psi[x \leftrightarrow y])$
- $(\psi_1 \otimes \psi_2)[x \leftrightarrow y] = (\psi_1[x \leftrightarrow y]) \otimes (\psi_2[x \leftrightarrow y])$ for
 $\otimes \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $(\mathcal{Q}x. \psi)[x \leftrightarrow y] = \mathcal{Q}y. (\psi[x \leftrightarrow y])$ for $\mathcal{Q} \in \{\forall, \exists\}$
- $(\mathcal{Q}y. \psi)[x \leftrightarrow y] = \mathcal{Q}y. (\psi[x \leftrightarrow y])$ for $\mathcal{Q} \in \{\forall, \exists\}$
- $(\mathcal{Q}z. \psi)[x \leftrightarrow y] = \mathcal{Q}z. (\psi[x \leftrightarrow y])$ for z a variable with $z \neq x$,
 $z \neq y$, and $\mathcal{Q} \in \{\forall, \exists\}$

Renaming by Swapping: Formulae

Examples

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

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

Theorem

Assume given structure $\mathcal{S} = (\mathcal{G}, \mathcal{D}, \mathcal{F}, \phi, \mathcal{R}, \rho)$, variables x and y , formula ψ over \mathcal{G} , and a assignment. If $x \notin \text{fv}(t)$ and $y \notin \text{fv}(t)$ then $\psi[x \leftrightarrow y] \equiv \psi$

α -equivalence

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

α -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

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

Example

Show

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

Example

Show

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

Example

Show

$$\frac{\frac{\frac{}{\{\exists x. \forall y. x \leq y\} \vdash \exists x. \forall y. x \leq y} \text{Hyp}}{\{\exists x. \forall y. x \leq y\} \vdash \exists y. y \leq x} \text{Ex E}}{\{\exists x. \forall y. x \leq y\} \vdash \exists y. y \leq x} \text{All I}}{\{\exists x. \forall y. x \leq y\} \vdash \forall x. \exists y. y \leq x} \text{Imp I}}{\{\} \vdash (\exists x. \forall y. x \leq y) \Rightarrow (\forall x. \exists y. y \leq x)} \text{Imp I}$$

Example of Failure

Let's try to show 1

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

Example of Failure

Let's try to show 2

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

Example of Failure

Let's try to show 3

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

Example of Failure

Let's try to show 4

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

Example of Failure

Let's try to show

5

$$\overline{\{\forall x. \exists y. y \leq x\} \vdash \forall x. \exists y. y \leq x}$$

$$\overline{\left\{ \begin{array}{l} \forall x. \exists y. y \leq x; \\ \exists y. y \leq x \end{array} \right\} \vdash z \leq x}$$

All E

$$\overline{\{\forall x. \exists y. y \leq x\} \vdash z \leq x}$$

All I

$$\overline{\{\forall x. \exists y. y \leq x\} \vdash \forall y. z \leq y}$$

Ex I

$$\overline{\{\forall x. \exists y. y \leq x\} \vdash \exists x. \forall y. x \leq y}$$

Imp I

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

Floyd-Hoare Logic

- Also called **Axiomatic Semantics**
- Based on formal logic (first order predicate calculus)
- Logical system built from **axioms** and **inference rules**
- Mainly suited to simple imperative programming languages
- Ideas applicable quite broadly

- 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

- 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

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

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

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:

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

The Assignment 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 Assignment Rule – Your Turn

- What is the weakest precondition of

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

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

The Assignment Rule – Your Turn

- What is the weakest precondition of

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

$$\{ (x + y) + y = w(x + y) \}$$
$$x := x + y$$
$$\{ x + y = wx \}$$

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:

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

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

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

Which Inferences Are Correct?

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

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

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

Which Inferences Are Correct?

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

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

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

Which Inferences Are Correct?

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

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

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

Which Inferences Are Correct?

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

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

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