Logic LVA 703600 VU3

http://cl-informatik.uibk.ac.at/teaching/ws05/logic/

Georg Moser (VU)[1] Christian Vogt (VU)[2]

[1]georg.moser@uibk.ac.at
office hours: Thursday 1pm–3pm

[2]christian.vogt@uibk.ac.at
office hours: Tuesday 9am–11am

Autumn 2005

Definition first-order language

fixed part	connectives	as for propositional logic	
	quantifiers	$\forall$	for all
		$\exists$	there exists
	auxiliary symbols	')','(', and ','	
	variables	$v_1, v_2, \ldots$ informally: $x, y, z, \ldots$	
variable part	relation symbols	$\mathbf{R}$	
	function symbols	$\mathbf{F}$	
	constant symbols	$\mathbf{C}$	

notation: $\mathrm{L}(\mathbf{R}, \mathbf{F}, \mathbf{C})$ describes the first-order language determined by $\mathbf{R}, \mathbf{F}, \mathbf{C}$

Definition inductive definition of terms & formulas

terms, $\mathbf{T}$	any variable is in $\mathbf{T}$	
	any constant is in $\mathbf{T}$	
	$t_1, \ldots, t_n \in \mathbf{T}$ ➡	
	➡ $f(t_1, \ldots, t_n) \in \mathbf{T}$	$f \in \mathbf{F}$, n-ary
atomic formulas	$R(t_1, \ldots, t_n)$	$R \in \mathbf{R}$, n-ary
	$\top$, $\bot$	$t_1, \ldots, t_n \in \mathbf{T}$
formulas, **Frm**	any atomic formula is in **Frm**	
	$A \in \mathbf{Frm}$ ➡ $\neg A \in \mathbf{Frm}$	
	$A, B \in \mathbf{Frm}$ ➡ $(A \circ B) \in \mathbf{Frm}$	$\circ$ binary
	$A \in \mathbf{Frm}$ ➡ $(\forall x)A \in \mathbf{Frm}$	x a variable
	$A \in \mathbf{Frm}$ ➡ $(\exists x)A \in \mathbf{Frm}$	

Example formulas

$$(\forall x)(\forall y)(< (x, y) \rightarrow (\exists z)(< (x, z) \wedge < (z, y)))$$
$$(\forall x)(\forall y)[x < y \rightarrow (\exists z)(x < z \wedge z < y)]$$
$$\forall x, y \; x < y \rightarrow \exists z \; (x < z \wedge z < y)$$

NB: we employ structural induction on terms (and formulas) and structural recursion on terms (and formulas) as proof-principles

Example free-variable occurrences, $\mathrm{fvar}(X)$

- ➡ if A atomic, then $\mathrm{fvar}(A)$ is the set of variables occurring in A
- ➡ $\mathrm{fvar}(\neg A) := \mathrm{fvar}(A)$
- ➡ $\mathrm{fvar}((A \circ B)) := \mathrm{fvar}(A) \cup \mathrm{fvar}(B)$
- ➡ $\mathrm{fvar}((\forall x)A) = \mathrm{fvar}((\exists x)A) := \mathrm{fvar}(A) - \{x\}$

Definition substitutions

- a substitution σ is a mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from the set of variables to the set of terms $\mathbf{T}$
- extend σ to terms:

$$c\sigma := c \qquad c \in \mathbf{C}$$
$$f(t_1, \dots, t_n)\sigma := f(t_1\sigma, \dots, t_n\sigma) \qquad f \in \mathbf{F}$$

example: set σ: $x \mapsto f(x, y), y \mapsto h(a), z \mapsto g(c, h(x))$

$$j(k(x), y)\sigma = j(k(x)\sigma, y\sigma) = j(k(x\sigma), y\sigma) = j(k(f(x, y)), h(a))$$

- σ, τ substitutions; the composition $\sigma\tau$ of σ and τ is defined as

$$x(\sigma\tau) := (x\sigma)\tau$$

- the domain of σ is $\{x \mid x\sigma \neq x\}$

Lemma

- for any term t: $t(\sigma\tau) = (t\sigma)\tau$ by structural induction
- composition is associative (i.e. $(\sigma_1\sigma_2)\sigma_3 = \sigma_1(\sigma_2\sigma_3)$)

Definition

- if the domain of σ is $\{x_1, \dots, x_n\}$ and $x_1\sigma = t_1, \dots, x_n\sigma = t_n$, then we write $\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ to denote σ

Theorem set $\sigma_1 = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$, $\sigma_2 = \{y_1 \mapsto s_1, \dots, y_k \mapsto s_k\}$; then the composition $\sigma_1\sigma_2$ can be written as

$$\{x_1 \mapsto t_1\sigma_2, \dots, x_n \mapsto t_n\sigma_2, z_1 \mapsto z_1\sigma_2, \dots, z_m \mapsto z_m\sigma_2\}$$

$$\{z_1, \dots, z_m\} = \{y_1, \dots, y_k\} - \{x_1, \dots, x_n\}$$

Definition substitutions on formulas

let σ be a substitution

- define σ_x:

$$y\sigma_x := \begin{cases} y\sigma & \text{if } y \neq x \\ x & \text{otherwise} \end{cases}$$

- $P(t_1, \dots, t_n)\sigma := P(t_1\sigma, \dots, t_n\sigma) \qquad P \in \mathbf{R}$, P n-ary

$\top\sigma := \top \qquad \bot\,\sigma := \bot$

$(\neg A)\sigma := \neg(A\sigma)$

$(A \circ B)\sigma := (A\sigma \circ B\sigma) \qquad \circ$ binary

$((\forall x)A)\sigma := (\forall x)(A\sigma_x)$

$((\exists x)A)\sigma := (\exists x)(A\sigma_x)$

Example $\sigma = \{x \mapsto a, y \mapsto b\}$

$$\begin{aligned}(\forall x R(x, y) \to \exists y R(x, y))\sigma &= (\forall x R(x, y))\sigma \to (\exists y R(x, y))\sigma \\ &= \forall x(R(x, y)\sigma_x) \to \exists y(R(x, y)\sigma_y) \\ &= \forall x R(x, b) \to \exists y R(a, y)\end{aligned}$$

Definition substitution σ is free for a formula:

- if A atomic, σ is free for A
- σ is free for A ➡ σ is free for $\neg A$
- σ is free for A and B ➡ σ is free for $(A \circ B)$
- σ_x is free for A and if $y \in \mathrm{fvar}(A)$, $y \neq x$, then $y\sigma$ does not contain x ➡ σ is free for $(\exists x)A$ and free for $(\forall x)A$

Theorem Suppose σ is free for A and τ is free for $A\sigma$, then $(A\sigma)\tau = A(\sigma\tau)$

Proof by structural induction on A

- Base: let $A = P(t_1, \ldots, t_n)$, hence

$$(P(t_1, \ldots, t_n)\sigma)\tau = P((t_1\sigma)\tau, \ldots, (t_n\sigma)\tau) =$$
$$= P(t_1(\sigma\tau), \ldots, t_n(\sigma\tau)) = P(\sigma\tau)$$

- Step: we only consider $A = (\forall x)A_1$

  assumptions σ_x is free for A_1

  τ_x is free for $A_1\sigma_x$ as $((\forall x)A_1)\sigma = (\forall x)A_1\sigma_x$

$(A_1\sigma_x)\tau_x = A_1(\sigma_x\tau_x)$ by IH

$A_1(\sigma_x\tau_x) = A_1(\sigma\tau)_x$ easy

$$(((\forall x)A_1)\sigma)\tau = ((\forall x)(A_1\sigma_x))\tau = (\forall x)((A_1\sigma_x)\tau_x) =$$
$$= (\forall x)(A_1(\sigma_x\tau_x)) = (\forall x)(A_1(\sigma\tau)_x) = ((\forall x)A_1)(\sigma\tau) \quad \square$$

Definition domain, models & assignments

model a model of $\mathrm{L}(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is a pair $\mathbf{M} = (\mathbf{D}, \mathbf{I})$, s.t.

$\mathbf{D} \neq \emptyset$ a set, called domain of $\mathbf{M}$

$\mathbf{I}$ a mapping, called interpretation associating

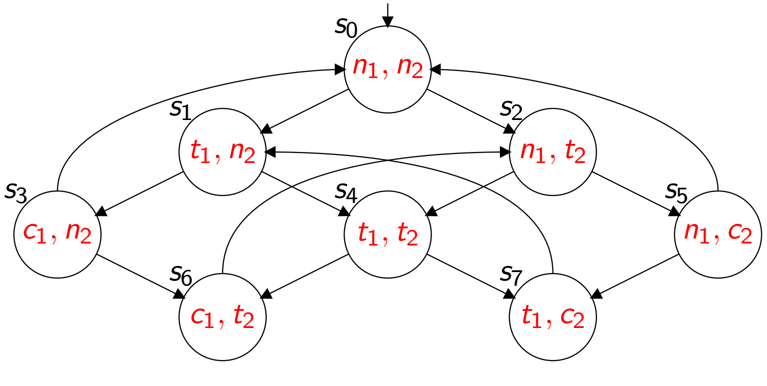
- to every $c \in \mathbf{C}$, some $c^{\mathbf{I}} \in \mathbf{D}$
- to every $f \in \mathbf{F}$, some function $f^{\mathbf{I}} \colon \mathbf{D}^n \to \mathbf{D}$
- to every $P \in \mathbf{R}$, some relation $P^{\mathbf{I}} \subseteq \mathbf{D}^n$

assignment an assignment in $\mathbf{M}$ is a mapping $\mathbf{A} \colon \mathbf{V} \to \mathbf{D}$; we write $v^{\mathbf{A}}$ instead of $A(v)$

value each term $t \in \mathrm{L}$ is associated a value $t^{\mathbf{I},\mathbf{A}}$

- for $c \in \mathbf{C}$: $c^{\mathbf{I},\mathbf{A}} := c^{\mathbf{I}}$
- for $v \in \mathbf{V}$: $v^{\mathbf{I},\mathbf{A}} := v^{\mathbf{A}}$
- for $f \in \mathbf{F}$: $[f(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} := f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}})$

Example: Modelling 'mutual exclusion'



define a first-order language $\mathrm{L}(\mathbf{R}, \mathbf{F}, \mathbf{C})$ for 'mutual exclusion'

relation symbols	R binary
	C_i, N_i, T_i unary for $i \in [1, 2]$
function symbols	none
constant symbols	$k_0, k_1, \ldots, k_7$

represent the protocol by a first-order model $\mathbf{M} = (\mathbf{D}, \mathbf{I})$

domain: $\mathbf{D} = \{s_0, s_1, \ldots, s_7\}$

interpretation $\mathbf{I}$:

- $\mathbf{I}$ interprets the symbol R by the relation $R^{\mathbf{I}}$ where

$$\{(s_0, s_1), (s_0, s_2), (s_1, s_3), (s_1, s_4), (s_2, s_4), (s_2, s_5), (s_3, s_6),$$
$$(s_4, s_6), (s_4, s_7), (s_5, s_7), (s_6, s_2), (s_7, s_1), (s_3, s_0), (s_5, s_0)\} = R^{\mathbf{I}}$$

- $\mathbf{I}$ interprets C_i by $C_i^{\mathbf{I}}$ where $s_3, s_6 \in C_1^{\mathbf{I}}$ and $s_5, s_7 \in C_2^{\mathbf{I}}$
- $\mathbf{I}$ interprets N_i by $N_i^{\mathbf{I}}$ where $s_0, s_2, s_5 \in N_1^{\mathbf{I}}$ and $s_0, s_1, s_3 \in N_2^{\mathbf{I}}$
- $\mathbf{I}$ interprets T_i by $T_i^{\mathbf{I}}$ where $s_1, s_4, s_7 \in T_1^{\mathbf{I}}$ and $s_2, s_4, s_6 \in T_2^{\mathbf{I}}$
- finally $\mathbf{I}$ associates a state s_i with each c_i: $c_i^{\mathbf{I}} = s_i$

assignment $\mathbf{A}$: arbitrary

Definition truth value for formulas

- an assignment **B** in a model **M** is an *x-variant* of an assignment **A** if the values differ only for x
- let $\mathbf{M} = (\mathbf{D}, \mathbf{I})$ be a model, **A** an assignment in **M**; define the truth-value $[X]^{\mathbf{I},\mathbf{A}}$ of a formula X
  - $[P(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ iff $(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}}) \in P^{\mathbf{I}}$
    $\top^{\mathbf{I},\mathbf{A}} := \mathbf{t}$
    $\bot^{\mathbf{I},\mathbf{A}} := \mathbf{f}$
  - $[\neg A]^{\mathbf{I},\mathbf{A}} := \neg(A^{\mathbf{I},\mathbf{A}})$
  - $[(A \circ B)]^{\mathbf{I},\mathbf{A}} := (A^{\mathbf{I},\mathbf{A}} \circ B^{\mathbf{I},\mathbf{A}})$
  - $[(\forall x)A]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ iff $A^{\mathbf{I},\mathbf{B}} = \mathbf{t}$ for every **B**, **B** x-variant of **A**
  - $[(\exists x)A]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ iff $A^{\mathbf{I},\mathbf{B}} = \mathbf{t}$ for some **B**, **B** x-variant of **A**

Definition validity & satisfiability

- X is true in **M**, if $X^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ for all assignments **A**
- X is valid, if X is true in all models for the language
- as set S of formulas is satisfiable in **M**, if there is some **A** such that $X^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ for all $X \in S$
- S is satisfiable, if satisfiable in some **M**

Example let **M** be the model of the 'mutual exclusion' protocol; we show that $\forall x \neg(C_1(x) \wedge C_2(x))$ in true in **M**

$$[\forall x \neg(C_1(x) \wedge C_2(x))]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$$
iff $[\neg(C_1(x) \wedge C_2(x))]^{\mathbf{I},\mathbf{B}} = \mathbf{t}$ for any x-variant **B** of **A**
iff $\neg([C_1(x)]^{\mathbf{I},\mathbf{B}} \wedge [C_2(x)]^{\mathbf{I},\mathbf{B}}) = \mathbf{t}$ for any x-variant **B** of **A**
iff $\neg(x^{\mathbf{B}} \in C_1^{\mathbf{I}} \wedge x^{\mathbf{B}} \in C_2^{\mathbf{I}}) = \mathbf{t}$ for any x-variant **B** of **A**
iff $\neg(s \in C_1^{\mathbf{I}} \wedge s \in C_2^{\mathbf{I}}) = \mathbf{t}$ for any $s \in \mathbf{D}$

Summary

- syntax of first-order logic
- substitutions
- semantics of first-order logic