Logica (I&E)

najaar 2018

`http://liacs.leidenuniv.nl/~vlietrvan1/logica/`

Rudy van Vliet
kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 12, maandag 26 november 2018

2. Predicate logic
2.4. Semantics of predicate logic

Er moet op elke plaats in het veld verdedigd worden, dat kost het minste energie, want dan moet je niet helemaal terug lopen om een doelpunt te maken.

2.4. Semantics of predicate logic

In propositional logic:

A slide from lecture 6:

Corollary 1.39. (Soundness and Completeness)
Let $\phi_1, \phi_2, \ldots, \phi_n$ and ψ be formulas of propositional logic. Then

$$\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$$

holds, iff the sequent

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$$

is valid.

Truth values for

$$(p \vee \neg q) \to (q \to p)$$

Truth values for

$$\forall x \exists y ((P(x) \vee \neg Q(y)) \to (Q(x) \to P(y)))$$

?

Or for

$$P(t_1, t_2, \ldots, t_n)$$

?

Definition 2.14.
Let $\mathcal{F}$ be a set of function symbols and $\mathcal{P}$ a set of predicate symbols, each symbol with a fixed arity.
A model $\mathcal{M}$ of the pair $(\mathcal{F}, \mathcal{P})$ consists of the following set of data:

1. A non-empty set A, the universe of concrete values (one set);

2. for each nullary symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}}$ of A;

3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : A^n \to A$ from A^n, the set of n-tuples over A, to A;

4. for each $P \in \mathcal{P}$ with arity $n > 0$, a subset $P^{\mathcal{M}} \subseteq A^n$ of n-tuples over A;

5. $=^{\mathcal{M}}$ is equality on A

For all students x: ϕ

There exists a student x: ϕ

Use predicate Student(x)...

Example 2.15.

$\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ (nullary)

$\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$ (binary, unary)

Example 2.15.
$\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ (nullary)
$\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$ (binary, unary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b, c\}$ (states in computer program)
$i^{\mathcal{M}} \stackrel{\text{def}}{=} a$, $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a,a),(a,b),(a,c),(b,c),(c,c)\}$ $F^{\mathcal{M}} \stackrel{\text{def}}{=} \{b, c\}$

1. Informal model check of formula

$$\exists y R(i, y)$$

Example 2.15.
$\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ (nullary)
$\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$ (binary, unary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b, c\}$ (states in computer program)
$i^{\mathcal{M}} \stackrel{\text{def}}{=} a$, $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a, a), (a, b), (a, c), (b, c), (c, c)\}$ $F^{\mathcal{M}} \stackrel{\text{def}}{=} \{b, c\}$

2. Informal model check of formula

$$\neg F(i)$$

Example 2.15.
$\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ (nullary)
$\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$ (binary, unary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b, c\}$ (states in computer program)
$i^{\mathcal{M}} \stackrel{\text{def}}{=} a$, $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a, a), (a, b), (a, c), (b, c), (c, c)\}$ $F^{\mathcal{M}} \stackrel{\text{def}}{=} \{b, c\}$

3. Informal model check of formula

$$\forall x \forall y \forall z (R(x, y) \land R(x, z) \to y = z)$$

Example 2.15.
$\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ (nullary)
$\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$ (binary, unary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b, c\}$ (states in computer program)
$i^{\mathcal{M}} \stackrel{\text{def}}{=} a$, $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a,a),(a,b),(a,c),(b,c),(c,c)\}$ $F^{\mathcal{M}} \stackrel{\text{def}}{=} \{b, c\}$

4. Informal model check of formula

$$\forall x \exists y R(x, y)$$

Example 2.16.

$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)

$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)

Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Example 2.16.
$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)
$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)
Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{$(finite) binary strings (including empty string ϵ)$\}$
$e^{\mathcal{M}} \stackrel{\text{def}}{=} \epsilon$
$.^{\mathcal{M}} \stackrel{\text{def}}{=}$ ‘concatenation’
$\leq \stackrel{\text{def}}{=}$ ‘is prefix’

1. Informal model check of formula

$$\forall x((x \leq x \cdot e) \wedge (x \cdot e \leq x))$$

Example 2.16.
$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)
$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)
Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{$(finite) binary strings (including empty string ϵ)$\}$
$e^{\mathcal{M}} \stackrel{\text{def}}{=} \epsilon$
$.^{\mathcal{M}} \stackrel{\text{def}}{=}$ 'concatenation'
$\leq \stackrel{\text{def}}{=}$ 'is prefix'

2. Informal model check of formula

$$\exists y \forall x (y \leq x)$$

Example 2.16.
$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)
$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)
Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{$(finite) binary strings (including empty string ϵ)$\}$
$e^{\mathcal{M}} \stackrel{\text{def}}{=} \epsilon$
$\cdot^{\mathcal{M}} \stackrel{\text{def}}{=}$ 'concatenation'
$\leq \stackrel{\text{def}}{=}$ 'is prefix'

3. Informal model check of formula

$$\forall x \exists y (y \leq x)$$

Example 2.16.
$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)
$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)
Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{$(finite) binary strings (including empty string ϵ)$\}$
$e^{\mathcal{M}} \stackrel{\text{def}}{=} \epsilon$
$.^{\mathcal{M}} \stackrel{\text{def}}{=}$ 'concatenation'
$\leq \stackrel{\text{def}}{=}$ 'is prefix'

4. Informal model check of formula

$$\forall x \forall y \forall z ((x \leq y) \to (x \cdot z \leq y \cdot z))$$

Example 2.16.

$\mathcal{F} \stackrel{\text{def}}{=} \{e, \cdot\}$ (nullary, binary)

$\mathcal{P} \stackrel{\text{def}}{=} \{\leq\}$ (binary)

Infix: $t_1 \cdot t_2 \leq (t \cdot t)$

Model $\mathcal{M}$:

$A \stackrel{\text{def}}{=} \{$(finite) binary strings (including empty string ϵ)$\}$

$e^{\mathcal{M}} \stackrel{\text{def}}{=} \epsilon$

$.^{\mathcal{M}} \stackrel{\text{def}}{=}$ 'concatenation'

$\leq \stackrel{\text{def}}{=}$ 'is prefix'

5. Informal model check of formula

$$\neg\exists x \forall y((x \leq y) \rightarrow (y \leq x))$$

Example.
$\mathcal{F} \stackrel{\text{def}}{=} \emptyset$
$\mathcal{P} \stackrel{\text{def}}{=} \{P, Q, R\}$ (unary, unary, binary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b\}$
$P^{\mathcal{M}} \stackrel{\text{def}}{=} \{a, b\}$ $\quad Q^{\mathcal{M}} \stackrel{\text{def}}{=} \{a\}$ $\quad R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a, a), (a, b)\}$

Informal check of formula

$$\forall x \forall y (P(x) \wedge \exists x (Q(x) \wedge R(x, y)))$$

Mild requirements on model...

Choice of model...

$\phi[t/x]$ vs. $\phi[a/x]$

Definition 2.17.
A look-up table or environment for a universe A of concrete values is a function l : **var** $\to A$ from the set of variables **var** to A.
For such an l, we denote by $l[x \mapsto a]$ the look-up table which maps x to a and any other variable y to $l(y)$.

Example.

look-up table l	
x	b
y	b
z	a

updated look-up table $l[x \mapsto a]$	
x	a
y	b
z	a

updated look-up table $l[x \mapsto b]$	
x	b
y	b
z	a

updated look-up table $l[x \mapsto b][x \mapsto a][z \mapsto b]$	
x	a
y	b
z	b

Example.
$\mathcal{F} \stackrel{\text{def}}{=} \emptyset$
$\mathcal{P} \stackrel{\text{def}}{=} \{P, Q, R\}$ (unary, unary, binary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b\}$
$P^{\mathcal{M}} \stackrel{\text{def}}{=} \{a, b\}$ $\qquad Q^{\mathcal{M}} \stackrel{\text{def}}{=} \{a\}$ $\qquad R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a, a), (a, b)\}$

What happens to formula

$$\forall x \forall y (P(x) \wedge \exists x (Q(x) \wedge R(x, y)))$$

with

look-up table l	
x	b
y	b

That is: $l(x) = b,\ l(y) = b$

Definition 2.18.
Given a model $\mathcal{M}$ for a pair $(\mathcal{F}, \mathcal{P})$ and given a look-up table l, we define the satisfaction relation $\mathcal{M} \vDash_l \phi$ for each logical formula ϕ over the pair $(\mathcal{F}, \mathcal{P})$ and look-up table l by structural induction on ϕ.

If $\mathcal{M} \vDash_l \phi$ holds, we say that ϕ computes to $\top$ in the model $\mathcal{M}$ with respect to the look-up table l.

Definition 2.18. (continued)
P: If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set A by replacing all variables with their values according to l. In this way we compute concrete values $a_1, a_2, \ldots, a_n$ from A for each of these terms, where we interpret any function symbol $f \in \mathcal{F}$ by $f^{\mathcal{M}}$.
Now $\mathcal{M} \vDash_l P(t_1, t_2, \ldots, t_n)$ holds, iff $(a_1, a_2, \ldots, a_n)$ is in the set $P^{\mathcal{M}}$.

Exercise. Let
$A \stackrel{\text{def}}{=} \{a, b, c\}$
$R^{\mathcal{M}} = \{(b, a), (b, b), (b, c)\}$
$l(x) = b, \; l(y) = c$

(a) Is $\mathcal{M} \vDash_l R(x, y)$?
(b) Is $\mathcal{M} \vDash_l R(y, x)$?

Definition 2.18. (continued)
P: If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set A by replacing all variables with their values according to l. In this way we compute concrete values $a_1, a_2, \ldots, a_n$ from A for each of these terms, where we interpret any function symbol $f \in \mathcal{F}$ by $f^{\mathcal{M}}$.
Now $\mathcal{M} \vDash_l P(t_1, t_2, \ldots, t_n)$ holds, iff $(a_1, a_2, \ldots, a_n)$ is in the set $P^{\mathcal{M}}$.

Exercise. Let
$A \stackrel{\text{def}}{=} \{a, b, c\}$
$f^{\mathcal{M}}(a) = f^{\mathcal{M}}(b) = c, \ f^{\mathcal{M}}(c) = b$
$R^{\mathcal{M}} = \{(b, a), (b, b), (b, c)\}$
$l(x) = a, \ l(y) = c$

(a) Is $\mathcal{M} \vDash_l R(f(x), y)$?
(b) Is $\mathcal{M} \vDash_l R(f(y), x)$?

Definition 2.18. (continued)
$\forall x$: The relation $\mathcal{M} \vDash_l \forall x \psi$ holds, iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for all $a \in A$.

Exercise. Let
$A \stackrel{\text{def}}{=} \{a, b, c\}$
$R^{\mathcal{M}} = \{(b, a), (b, b), (b, c)\}$
$l(x) = b, \; l(y) = c$

(a) Is $\mathcal{M} \vDash_l \forall x R(x, y)$?
(b) Is $\mathcal{M} \vDash_l \forall y R(x, y)$?

Definition 2.18. (continued)
$\exists x$: The relation $\mathcal{M} \vDash_l \exists x \psi$ holds, iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for some $a \in A$.

Exercise. Let
$A \stackrel{\text{def}}{=} \{a, b, c\}$
$R^{\mathcal{M}} = \{(b, a), (b, b), (b, c)\}$
$l(x) = a,\ l(y) = c$

(a) Is $\mathcal{M} \vDash_l \exists x R(x, y)$?
(b) Is $\mathcal{M} \vDash_l \exists x R(y, x)$?

Definition 2.18. (continued)

$\neg$: The relation $\mathcal{M} \vDash_l \neg\psi$ holds, iff $\mathcal{M} \vDash_l \psi$ does not hold.

$\vee$: The relation $\mathcal{M} \vDash_l \psi_1 \vee \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_1$ or $\mathcal{M} \vDash_l \psi_2$ holds.

$\wedge$: The relation $\mathcal{M} \vDash_l \psi_1 \wedge \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_1$ and $\mathcal{M} \vDash_l \psi_2$ holds.

$\rightarrow$: The relation $\mathcal{M} \vDash_l \psi_1 \rightarrow \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_2$ holds whenever $\mathcal{M} \vDash_l \psi_1$ holds.

Example.
$\mathcal{F} \stackrel{\text{def}}{=} \emptyset$
$\mathcal{P} \stackrel{\text{def}}{=} \{P, Q, R\}$ (unary, unary, binary)

Model $\mathcal{M}$:
$A \stackrel{\text{def}}{=} \{a, b\}$
$P^{\mathcal{M}} \stackrel{\text{def}}{=} \{a, b\}$ $\quad Q^{\mathcal{M}} \stackrel{\text{def}}{=} \{a\}$ $\quad R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a, a), (a, b)\}$

Is

$$M \vDash_l \forall x \forall y (P(x) \land \exists x (Q(x) \land R(x, y)))$$

with

look-up table l	
x	b
y	b

That is: $l(x) = b$, $l(y) = b$

If l and l' are identical on all free variables in ϕ, then . . .

If ϕ has *no* free variables, then . . .
Notation $\mathcal{M} \models \phi$
Sentence ϕ

Shorter notation for formal model check. . .

Shorter notation for formal model check. . .