Logical and Theoretical Foundations of Computer Science

LaTFoCS

Pamela Fleischmann

fpa@informatik.uni-kiel.de

Winter Semester 2019

Kiel University

Dependable Systems Group

SEIT 350 JAHREN GANZ WEIT OBEN

CAU

Interpretations in Predicate Logic

Models

Dealing with variables:

- for $\exists x \varphi$ we should only apply `true` iff we find at least one value for x such that φ is `true`

Models

Dealing with variables:

- for $\exists x \varphi$ we should only apply `true` iff we find at least one value for x such that φ is `true`
- for $\forall x \varphi$ we should only apply `true` iff φ is `true` for all values x may take

Models

Dealing with variables:

- for $\exists x \varphi$ we should only apply `true` iff we find at least one value for x such that φ is `true`
- for $\forall x \varphi$ we should only apply `true` iff φ is `true` for all values x may take
- for a good definition of a model we need to fix the values a variable may take in before (model depends on this universe)

Models

Dealing with variables:

- for $\exists x \varphi$ we should only apply `true` iff we find at least one value for x such that φ is `true`
- for $\forall x \varphi$ we should only apply `true` iff φ is `true` for all values x may take
- for a good definition of a model we need to fix the values a variable may take in before (model depends on this universe)
- the model needs to specify the functions and predicates

Models of Function and Predicate Symbols

$\mathscr{F}$ set of function symbols, $\mathscr{P}$ set of predicate symbols

Definition

$\mathscr{M}$ is model of $(\mathscr{F}, \mathscr{P})$ if $\mathscr{M}$ consists of

Models of Function and Predicate Symbols

$\mathscr{F}$ set of function symbols, $\mathscr{P}$ set of predicate symbols

Definition

$\mathscr{M}$ is model of $(\mathscr{F}, \mathscr{P})$ if $\mathscr{M}$ consists of

1. non-empty set A (Universe of concrete values)

Models of Function and Predicate Symbols

$\mathcal{F}$ set of function symbols, $\mathcal{P}$ set of predicate symbols

Definition

$\mathcal{M}$ is model of $(\mathcal{F}, \mathcal{P})$ if $\mathcal{M}$ consists of

1. non-empty set A (Universe of concrete values)
2. for each nullary symbol $f \in \mathcal{F}$ an element $f^{\mathcal{M}}$ in A

Models of Function and Predicate Symbols

$\mathcal{F}$ set of function symbols, $\mathcal{P}$ set of predicate symbols

Definition

$\mathcal{M}$ is model of $(\mathcal{F}, \mathcal{P})$ if $\mathcal{M}$ consists of

1. non-empty set A (Universe of concrete values)
2. for each nullary symbol $f \in \mathcal{F}$ an element $f^{\mathcal{M}}$ in A
3. for each $f \in \mathcal{F}$ a function $f^{\mathcal{M}} : A^n \to A$ if f is n-ary

Models of Function and Predicate Symbols

$\mathcal{F}$ set of function symbols, $\mathcal{P}$ set of predicate symbols

Definition

$\mathcal{M}$ is model of $(\mathcal{F}, \mathcal{P})$ if $\mathcal{M}$ consists of

1. non-empty set A (Universe of concrete values)
2. for each nullary symbol $f \in \mathcal{F}$ an element $f^{\mathcal{M}}$ in A
3. for each $f \in \mathcal{F}$ a function $f^{\mathcal{M}} : A^n \to A$ if f is n-ary
4. for each $P \in \mathcal{P}$ a subset $P^{\mathcal{M}} \subseteq A^n$ if P is n-ary

Difference between $f, f^{\mathscr{M}}$ resp. $P, P^{\mathscr{M}}$

- the elements of $\mathscr{F}, \mathscr{P}$ are mere symbols

Difference between $f, f^{\mathcal{M}}$ resp. $P, P^{\mathcal{M}}$

- the elements of $\mathcal{F}, \mathcal{P}$ are mere symbols
- the elements of $\mathcal{F}^{\mathcal{M}}, \mathcal{P}^{\mathcal{M}}$ are concrete functions resp. relations

Difference between $f, f^{\mathscr{M}}$ resp. $P, P^{\mathscr{M}}$

- the elements of $\mathscr{F}, \mathscr{P}$ are mere symbols
- the elements of $\mathscr{F}^{\mathscr{M}}, \mathscr{P}^{\mathscr{M}}$ are concrete functions resp. relations
- we usually associate something with a symbol $(+, =, \text{owner})$ **but** this is already the application of $\mathscr{M}$ in our heads

Difference between $f, f^{\mathscr{M}}$ resp. $P, P^{\mathscr{M}}$

- the elements of $\mathscr{F}, \mathscr{P}$ are mere symbols
- the elements of $\mathscr{F}^{\mathscr{M}}, \mathscr{P}^{\mathscr{M}}$ are concrete functions resp. relations
- we usually associate something with a symbol $(+, =, \text{owner})$ **but** this is already the application of $\mathscr{M}$ in our heads
- $=^{\mathscr{M}}: \mathbb{N}^2 \longrightarrow \mathbb{N}; (n_1, n_2) \mapsto n_1 +_{\mathbb{N}} n_2$ **is possible**

Example

Consider $\mathscr{F} = \{+\}, \mathscr{P} = \{=\}$

- $A_1 = \mathbb{N}$

Example

Consider $\mathscr{F} = \{+\}, \mathscr{P} = \{=\}$

- $A_1 = \mathbb{N}$
- $A_2 = \mathfrak{M}$ (set of all matrices)

Example

Consider $\mathcal{F} = \{+\}, \mathcal{P} = \{=\}$

- $A_1 = \mathbb{N}$
- $A_2 = \mathfrak{M}$ (set of all matrices)
- think about the differences of $+^{\mathcal{M}_1}, +^{\mathcal{M}_2}, =^{\mathcal{M}_1}, =^{\mathcal{M}_2}$

Example

Consider $\mathscr{F} = \{+\}, \mathscr{P} = \{=\}$

- $A_1 = \mathbb{N}$
- $A_2 = \mathfrak{M}$ (set of all matrices)
- think about the differences of $+^{\mathcal{M}_1}, +^{\mathcal{M}_2}, =^{\mathcal{M}_1}, =^{\mathcal{M}_2}$

Formulae are always interpreted relative to an environment.

Look-up Tables, Environments

Definition

A function $\ell : \mathcal{V} \longrightarrow A$ for set of variables $\mathcal{V}$ and universe A is called an assignement (look-up table, environment).

Look-up Tables, Environments

Definition

A function $\ell : \mathscr{V} \to A$ for set of variables $\mathscr{V}$ and universe A is called an assignement (look-up table, environment).

Notation: $\ell[x \mapsto a]$ for $\ell(x) = a$

Look-up Tables, Environments

Definition

A function $\ell : \mathscr{V} \longrightarrow A$ for set of variables $\mathscr{V}$ and universe A is called an assignement (look-up table, environment).

Notation: $\ell[x \mapsto a]$ for $\ell(x) = a$

Definition

For a term t and an environment ℓ let $\ell(t)$ denote the element of the universe obtained by replacing every variable v in t by $\ell(v)$

Satisfaction Relation $\models_\ell$

Definition

$\mathcal{M}$ model for $(\mathcal{F}, \mathcal{P})$, ℓ environment, φ predicate logic formula

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \ldots, t_n)$ iff $(\ell(t_1), \ldots, \ell(t_n)) \in P^{\mathscr{M}}$

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \ldots, t_n)$ iff $(\ell(t_1), \ldots, \ell(t_n)) \in P^{\mathscr{M}}$
- $\mathscr{M} \models_\ell \forall x \psi$ iff $\forall a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \ldots, t_n)$ iff $(\ell(t_1), \ldots, \ell(t_n)) \in P^{\mathscr{M}}$
- $\mathscr{M} \models_\ell \forall x \psi$ iff $\forall a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \exists x \psi$ iff $\exists a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \ldots, t_n)$ iff $(\ell(t_1), \ldots, \ell(t_n)) \in P^{\mathscr{M}}$
- $\mathscr{M} \models_\ell \forall x \psi$ iff $\forall a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \exists x \psi$ iff $\exists a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \neg \psi$ iff not $\mathscr{M} \models_\ell \psi$

Satisfaction Relation $\models_\ell$

Definition

$\mathcal{M}$ model for $(\mathcal{F}, \mathcal{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \dots, t_n)$ iff $(\ell(t_1), \dots, \ell(t_n)) \in P^{\mathcal{M}}$
- $\mathcal{M} \models_\ell \forall x \psi$ iff $\forall a \in A :\ \mathcal{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathcal{M} \models_\ell \exists x \psi$ iff $\exists a \in A :\ \mathcal{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathcal{M} \models_\ell \neg \psi$ iff not $\mathcal{M} \models_\ell \psi$
- $\mathcal{M} \models_\ell \psi_1 \vee \psi_2$ iff $\mathcal{M} \models_\ell \psi_1$ or $\mathcal{M} \models_\ell \psi_2$

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \ldots, t_n)$ iff $(\ell(t_1), \ldots, \ell(t_n)) \in P^{\mathscr{M}}$
- $\mathscr{M} \models_\ell \forall x \psi$ iff $\forall a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \exists x \psi$ iff $\exists a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \neg \psi$ iff not $\mathscr{M} \models_\ell \psi$
- $\mathscr{M} \models_\ell \psi_1 \vee \psi_2$ iff $\mathscr{M} \models_\ell \psi_1$ or $\mathscr{M} \models_\ell \psi_2$
- $\mathscr{M} \models_\ell \psi_1 \wedge \psi_2$ iff $\mathscr{M} \models_\ell \psi_1$ and $\mathscr{M} \models_\ell \psi_2$

Satisfaction Relation $\models_\ell$

Definition

$\mathscr{M}$ model for $(\mathscr{F}, \mathscr{P})$, ℓ environment, φ predicate logic formula

- $M \models_\ell P(t_1, \dots, t_n)$ iff $(\ell(t_1), \dots, \ell(t_n)) \in P^{\mathscr{M}}$
- $\mathscr{M} \models_\ell \forall x \psi$ iff $\forall a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \exists x \psi$ iff $\exists a \in A : \mathscr{M} \models_{\ell[x \mapsto a]} \psi$
- $\mathscr{M} \models_\ell \neg\psi$ iff not $\mathscr{M} \models_\ell \psi$
- $\mathscr{M} \models_\ell \psi_1 \vee \psi_2$ iff $\mathscr{M} \models_\ell \psi_1$ or $\mathscr{M} \models_\ell \psi_2$
- $\mathscr{M} \models_\ell \psi_1 \wedge \psi_2$ iff $\mathscr{M} \models_\ell \psi_1$ and $\mathscr{M} \models_\ell \psi_2$
- $\mathscr{M} \models_\ell \psi_1 \rightarrow \psi_2$ iff $\mathscr{M} \models_\ell \psi_1$ implies $\mathscr{M} \models_\ell \psi_2$

Sentences

Definition

A closed formulae are also called a sentence.

Sentences

Definition

A closed formulae are also called a sentence.

Theorem

Given two environment ℓ, ℓ' being identical on the free variable of φ implies

$$\mathcal{M} \models_{\ell} \varphi \text{ iff } \mathcal{M} \models_{\ell'} \varphi$$

Sentences

Definition

A closed formulae are also called a sentence.

Theorem

Given two environment ℓ, ℓ' being identical on the free variable of φ implies

$$\mathcal{M} \models_{\ell} \varphi \text{ iff } \mathcal{M} \models_{\ell'} \varphi$$

- sentences are either true or false independent of the environment

Sentences

Definition

A closed formulae are also called a sentence.

Theorem

Given two environment ℓ, ℓ' being identical on the free variable of φ implies

$$\mathcal{M} \models_{\ell} \varphi \text{ iff } \mathcal{M} \models_{\ell'} \varphi$$

- sentences are either true or false independent of the environment
- write for sentences $\mathcal{M} \models \varphi$

Quantifiers and Models

Theorem

Let φ be a predicate logic formula with exactly the free variables $x_1, \ldots, x_n \in \mathcal{V}$. Let $\mathcal{M}$ be a model and ℓ an environment. Then

- $\mathcal{M} \models_\ell \varphi$ *iff* $\mathcal{M} \models \exists x_1 \ldots \exists x_n \varphi$
- $\mathcal{M} \models \varphi$ *iff* $\mathcal{M} \models \forall x_1 \ldots \exists x_n \varphi$

Example

The enemy of my enemy is not my enemy.

Example

The enemy of my enemy is not my enemy.

or formalised

$$\forall x \forall y (\text{enemyOf}(x, I) \wedge \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

Example Cont

$$\forall x \forall y (\text{enemyOf}(x, I) \wedge \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

Example Cont

$$\forall x \forall y (\text{enemyOf}(x, I) \wedge \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

$\mathscr{M}$ given by

- $A = \{i, e_1, e_2\}, \mathscr{F} = \{I\}$ with $I^{\mathscr{M}} = i$
- $P = \{\text{enemyOf}\}$ with $\text{enemyOf}^{\mathscr{M}} = \{(i, i), (e_1, i), (e_2, i)\}$

Example Cont

$$\forall x \forall y (\text{enemyOf}(x, I) \wedge \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

$\mathcal{M}$ given by

- $A = \{i, e_1, e_2\}, \mathcal{F} = \{I\}$ with $I^{\mathcal{M}} = i$
- $P = \{\text{enemyOf}\}$ with $\text{enemyOf}^{\mathcal{M}} = \{(i, i), (e_1, i), (e_2, i)\}$

the formula is **not** true since for $\ell(x) = i$ and $\ell(y) = e_1$ we get

$$\text{enemyOf}(i, I) \wedge \text{enemyOf}(e_1, i) \rightarrow \neg\, \text{enemyOf}(e_1, I)$$

Example Cont

$$\forall x \forall y (\text{enemyOf}(x, I) \wedge \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

Example Cont

$$\forall x \forall y(\text{enemyOf}(x, I) \land \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

$\mathscr{M}$ given by

- $A = \{i, e_1, e_2\}, \mathscr{F} = \{I\}$ with $I^{\mathscr{M}} = i$
- $P = \{\text{enemyOf}\}$ with $\text{enemyOf}^{\mathscr{M}} = \{(e_1, i), (e_2, e_1)\}$

Example Cont

$$\forall x \forall y (\text{enemyOf}(x, I) \land \text{enemyOf}(y, x) \rightarrow \neg\, \text{enemyOf}(y, I))$$

$\mathscr{M}$ given by

- $A = \{i, e_1, e_2\}, \mathscr{F} = \{I\}$ with $I^{\mathscr{M}} = i$
- $P = \{\text{enemyOf}\}$ with $\text{enemyOf}^{\mathscr{M}} = \{(e_1, i), (e_2, e_1)\}$

the formula is true since

Proof

$$\mathcal{M} \models \forall x \forall y(\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x) \rightarrow \neg\, \mathrm{eO}(y, I))$$

iff for all $a \in A(\mathcal{M} \models_{\ell[x \mapsto a]} \forall y(\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x) \rightarrow \neg\, \mathrm{eO}(y, I)))$

iff for all $a \in A$ for all $b \in A(\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]}$
$(\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x) \rightarrow \neg\, \mathrm{eO}(y, I)))$

iff for all $a \in A$ for all $b \in A(\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]}$
$(\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x))$ implies $\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]} \neg\, \mathrm{eO}(y, I))$

iff for all $a \in A$ for all $b \in A$
$((\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]} \mathrm{eO}(x, I))$ and $\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]} \mathrm{eO}(y, x))$
implies not $\mathcal{M} \models_{\ell[x \mapsto a, y \mapsto b]} \mathrm{eO}(y, I))$

Proof Cont

$$\mathscr{M} \models \forall x \forall y (\mathrm{eO}(x, I) \land \mathrm{eO}(y, x) \rightarrow \neg\, \mathrm{eO}(y, I))$$

$$\ldots$$

$$\text{iff for all } a \in A \text{ for all } b \in A (\mathscr{M} \models_{\ell[x \mapsto a, y \mapsto b]}$$

$$(\mathrm{eO}(x, I) \land \mathrm{eO}(y, x)) \text{ implies } \mathscr{M} \models_{\ell[x \mapsto a, y \mapsto b]} \neg\, \mathrm{eO}(y, I))$$

$$\text{iff for all } a \in A \text{ for all } b \in A$$

$$(((a, i) \in \mathrm{eO}^{\mathscr{M}} \text{ and } (b, a) \in \mathrm{eO}^{\mathscr{M}}) \text{ implies } (b, i) \notin \mathrm{eO}^{\mathscr{M}})$$

Proof Cont

$$\mathscr{M} \models \forall x \forall y (\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x) \rightarrow \neg\, \mathrm{eO}(y, I))$$

$$\ldots$$

iff for all $a \in A$ for all $b \in A(\mathscr{M} \models_{\ell[x \mapsto a, y \mapsto b]}$

$(\mathrm{eO}(x, I) \wedge \mathrm{eO}(y, x))$ implies $\mathscr{M} \models_{\ell[x \mapsto a, y \mapsto b]} \neg\, \mathrm{eO}(y, I))$

iff for all $a \in A$ for all $b \in A$

$(((a, i) \in \mathrm{eO}^{\mathscr{M}}$ and $(b, a) \in \mathrm{eO}^{\mathscr{M}})$ implies $(b, i) \notin \mathrm{eO}^{\mathscr{M}})$

the premise is only true for $a = e_1$ and $b = e_2$ but (e_2, i) is indeed not in $\mathrm{eO}^{\mathscr{M}}$