Theory of Computer Science

B4. Predicate Logic II

Malte Helmert

University of Basel

March 9, 2016

Free and Bound Variables

Free and Bound Variables: Motivation

Question:

- Consider a signature with variable symbols $\{x_1, x_2, x_3, \dots\}$ and an interpretation $\mathcal{I}$.
- Which parts of the definition of α are relevant to decide whether $\mathcal{I}, \alpha \models (\forall x_4 (R(x_4, x_2) \lor (f(x_3) = x_4)) \lor \exists x_3 S(x_3, x_2))$?

Free and Bound Variables: Motivation

Question:

- Consider a signature with variable symbols $\{x_1, x_2, x_3, \dots\}$ and an interpretation $\mathcal{I}$.
- Which parts of the definition of α are relevant to decide whether $\mathcal{I}, \alpha \models (\forall x_4(\mathsf{R}(x_4, x_2) \lor (\mathsf{f}(x_3) = x_4)) \lor \exists x_3 \mathsf{S}(x_3, x_2))$?
- $\alpha(x_1)$, $\alpha(x_5)$, $\alpha(x_6)$, $\alpha(x_7)$, ... are irrelevant since those variable symbols occur in no formula.

Free and Bound Variables: Motivation

Question:

- Consider a signature with variable symbols $\{x_1, x_2, x_3, \dots\}$ and an interpretation $\mathcal{I}$.
- Which parts of the definition of α are relevant to decide whether $\mathcal{I}, \alpha \models (\forall x_4 (R(x_4, x_2) \lor (f(x_3) = x_4)) \lor \exists x_3 S(x_3, x_2))$?
- $\alpha(x_1)$, $\alpha(x_5)$, $\alpha(x_6)$, $\alpha(x_7)$, $\dots$ are irrelevant since those variable symbols occur in no formula.
- $\alpha(x_4)$ also is irrelevant: the variable occurs in the formula, but all occurrences are bound by a surrounding quantifier.

Free and Bound Variables: Motivation

Question:

- Consider a signature with variable symbols $\{x_1, x_2, x_3, \dots\}$ and an interpretation $\mathcal{I}$.
- Which parts of the definition of α are relevant to decide whether $\mathcal{I}, \alpha \models (\forall x_4(\mathsf{R}(x_4, x_2) \lor (\mathsf{f}(x_3) = x_4)) \lor \exists x_3 \mathsf{S}(x_3, x_2))$?
- $\alpha(x_1)$, $\alpha(x_5)$, $\alpha(x_6)$, $\alpha(x_7)$, ... are irrelevant since those variable symbols occur in no formula.
- $\alpha(x_4)$ also is irrelevant: the variable occurs in the formula, but all occurrences are bound by a surrounding quantifier.
- $\rightsquigarrow$ only assignments for free variables x_2 and x_3 relevant

German: gebundene und freie Variablen

Variables of a Term

Definition (variables of a term)

Let t be a term. The set of variables that occur in t, written as $var(t)$, is defined as follows:

- $var(x) = \{x\}$
  for variable symbols x
- $var(\mathrm{c}) = \emptyset$
  for constant symbols c
- $var(\mathrm{f}(t_1, \dots, t_l)) = var(t_1) \cup \dots \cup var(t_l)$
  for function terms

terminology: A term t with $var(t) = \emptyset$ is called ground term.
German: Grundterm

example: $var(\mathrm{product}(x, \mathrm{sum}(\mathrm{k}, y))) =$

Free and Bound Variables of a Formula

Definition (free variables)

Let φ be a predicate logic formula. The set of free variables of φ, written as $\mathit{free}(\varphi)$, is defined as follows:

- $\mathit{free}(\mathsf{P}(t_1, \dots, t_k)) = \mathit{var}(t_1) \cup \dots \cup \mathit{var}(t_k)$
- $\mathit{free}((t_1 = t_2)) = \mathit{var}(t_1) \cup \mathit{var}(t_2)$
- $\mathit{free}(\neg\varphi) = \mathit{free}(\varphi)$
- $\mathit{free}((\varphi \land \psi)) = \mathit{free}((\varphi \lor \psi)) = \mathit{free}(\varphi) \cup \mathit{free}(\psi)$
- $\mathit{free}(\forall x\, \varphi) = \mathit{free}(\exists x\, \varphi) = \mathit{free}(\varphi) \setminus \{x\}$

Example: $\mathit{free}((\forall x_4(\mathsf{R}(x_4, x_2) \lor (\mathsf{f}(x_3) = x_4)) \lor \exists x_3 \mathsf{S}(x_3, x_2)))$

$=$

Closed Formulas/Sentences

Note: Let φ be a formula and let α and β variable assignments with $\alpha(x) = \beta(x)$ for all free variables x of φ.

Then $\mathcal{I}, \alpha \models \varphi$ iff $\mathcal{I}, \beta \models \varphi$.

Closed Formulas/Sentences

Note: Let φ be a formula and let α and β variable assignments with $\alpha(x) = \beta(x)$ for all free variables x of φ.

Then $\mathcal{I}, \alpha \models \varphi$ iff $\mathcal{I}, \beta \models \varphi$.

In particular, α is completely irrelevant if $\mathit{free}(\varphi) = \emptyset$.

Closed Formulas/Sentences

Note: Let φ be a formula and let α and β variable assignments with $\alpha(x) = \beta(x)$ for all free variables x of φ.

Then $\mathcal{I}, \alpha \models \varphi$ iff $\mathcal{I}, \beta \models \varphi$.

In particular, α is completely irrelevant if $\mathit{free}(\varphi) = \emptyset$.

Definition (closed formulas/sentences)

A formula φ without free variables (i. e., $\mathit{free}(\varphi) = \emptyset$) is called closed formula or sentence.

If φ is a sentence, then we often write $\mathcal{I} \models \varphi$ instead of $\mathcal{I}, \alpha \models \varphi$, since the definition of α does not influence whether φ is true under $\mathcal{I}$ and α or not.

Formulas with at least one free variable are called open.

German: geschlossene Formel/Satz, offene Formel

Closed Formulas/Sentences: Examples

Question: Which of the following formulas are sentences?

- $(\text{Block}(\text{b}) \vee \neg\text{Block}(\text{b}))$
- $(\text{Block}(x) \rightarrow (\text{Block}(x) \vee \neg\text{Block}(y)))$
- $(\text{Block}(\text{a}) \wedge \text{Block}(\text{b}))$
- $\forall x(\text{Block}(x) \rightarrow \text{Red}(x))$

Questions

Questions?

Logical Consequences

Terminology for Formulas

The terminology we introduced for propositional logic similarly applies to predicate logic:

- Interpretation $\mathcal{I}$ and variable assignment α form a model of the formula φ if $\mathcal{I}, \alpha \models \varphi$.
- Formula φ is satisfiable if $\mathcal{I}, \alpha \models \varphi$ for at least one $\mathcal{I}, \alpha$.
- Formula φ is falsifiable if $\mathcal{I}, \alpha \not\models \varphi$. for at least one $\mathcal{I}, \alpha$
- Formula φ is valid if $\mathcal{I}, \alpha \models \varphi$ for all $\mathcal{I}, \alpha$.
- Formula φ is unsatisfiable if $\mathcal{I}, \alpha \not\models \varphi$ for all $\mathcal{I}, \alpha$.
- Formulas φ and ψ are logically equivalent, written as $\varphi \equiv \psi$, if they have the same models.

German: Modell, erfüllbar, falsifizierbar, gültig, unerfüllbar, logisch äquivalent

Sets of Formulas: Semantics

Definition (set of formulas is satisfied or true)

Let $\mathcal{S}$ be a signature, Φ a set of formulas over $\mathcal{S}$, $\mathcal{I}$ an interpretation for $\mathcal{S}$ and α a variable assignment for $\mathcal{S}$ and the universe of $\mathcal{I}$.

We say that $\mathcal{I}$ and α satisfy the formulas Φ (also: Φ is true under $\mathcal{I}$ and α), written as: $\mathcal{I}, \alpha \models \Phi$, if $\mathcal{I}, \alpha \models \varphi$ for all $\varphi \in \Phi$.

German: $\mathcal{I}$ und α erfüllen Φ, Φ ist wahr unter $\mathcal{I}$ und α

Terminology for Sets of Formulas and Sentences

- Again, we use the same notations and concepts as in propositional logic.

  Example:

  - A set of formulas Φ is satisfiable if $\mathcal{I}, \alpha \models \Phi$ for at least one $\mathcal{I}, \alpha$.
  - A set of formulas Φ (logically) implies formula ψ, written as $\Phi \models \psi$, if all models of Φ are models of ψ.

Terminology for Sets of Formulas and Sentences

- Again, we use the same notations and concepts as in propositional logic.

  Example:

  - A set of formulas Φ is satisfiable if $\mathcal{I}, \alpha \models \Phi$ for at least one $\mathcal{I}, \alpha$.
  - A set of formulas Φ (logically) implies formula ψ, written as $\Phi \models \psi$, if all models of Φ are models of ψ.

- All concepts can be used for the special case of sentences (or sets of sentences). In this case we usually omit α.

  Examples:

  - Interpretation $\mathcal{I}$ is a model of a sentence φ if $\mathcal{I} \models \varphi$.
  - Sentence φ is unsatisfiable if $\mathcal{I} \not\models \varphi$ for all $\mathcal{I}$.

Terminology for Sets of Formulas and Sentences

- Again, we use the same notations and concepts as in propositional logic.

  Example:

  - A set of formulas Φ is satisfiable if $\mathcal{I}, \alpha \models \Phi$ for at least one $\mathcal{I}, \alpha$.
  - A set of formulas Φ (logically) implies formula ψ, written as $\Phi \models \psi$, if all models of Φ are models of ψ.

- All concepts can be used for the special case of sentences (or sets of sentences). In this case we usually omit α.

  Examples:

  - Interpretation $\mathcal{I}$ is a model of a sentence φ if $\mathcal{I} \models \varphi$.
  - Sentence φ is unsatisfiable if $\mathcal{I} \not\models \varphi$ for all $\mathcal{I}$.

- similarly:
  - $\varphi \models \psi$ if $\{\varphi\} \models \psi$
  - $\Phi \models \Psi$ if $\Phi \models \psi$ for all $\psi \in \Psi$

Questions

Questions?

Further Topics

Further Topics

Based on these definitions we could cover the same topics as in propositional logic:

- important logical equivalences
- normal forms
- theorems about reasoning (deduction theorem etc.)

We briefly discuss some general results on those topics but will not go into detail.

Logical Equivalences

- All logical equivalences of propositional logic also hold in predicate logic (e. g., $(\varphi \lor \psi) \equiv (\psi \lor \varphi)$). (Why?)
- Additionally the following equivalences and implications hold:

$(\forall x\varphi \land \forall x\psi) \equiv \forall x(\varphi \land \psi)$		
$(\forall x\varphi \lor \forall x\psi) \models \forall x(\varphi \lor \psi)$	but not vice versa	
$(\forall x\varphi \land \psi) \equiv \forall x(\varphi \land \psi)$	if $x \notin \mathit{free}(\psi)$	
$(\forall x\varphi \lor \psi) \equiv \forall x(\varphi \lor \psi)$	if $x \notin \mathit{free}(\psi)$	
$\neg\forall x\varphi \equiv \exists x\neg\varphi$		
$\exists x(\varphi \lor \psi) \equiv (\exists x\varphi \lor \exists x\psi)$		
$\exists x(\varphi \land \psi) \models (\exists x\varphi \land \exists x\psi)$	but not vice versa	
$(\exists x\varphi \lor \psi) \equiv \exists x(\varphi \lor \psi)$	if $x \notin \mathit{free}(\psi)$	
$(\exists x\varphi \land \psi) \equiv \exists x(\varphi \land \psi)$	if $x \notin \mathit{free}(\psi)$	
$\neg\exists x\varphi \equiv \forall x\neg\varphi$		

Normal Forms

Analogously to DNF and CNF for propositional logic there are several normal forms for predicate logic, such as

- negation normal form (NNF):
  negation symbols ($\neg$) are only allowed in front of atoms
- prenex normal form:
  quantifiers must form the outermost part of the formula
- Skolem normal form:
  prenex normal form without existential quantifiers

German: Negationsnormalform, Pränexnormalform, Skolemnormalform

Normal Forms (ctd.)

Efficient methods transform formula φ

- into an equivalent formula in negation normal form,
- into an equivalent formula in prenex normal form, or
- into an equisatisfiable formula in Skolem normal form.

German: erfüllbarkeitsäquivalent

Questions

Questions?

Summary

Summary

bound vs. free variables:

- bound vs. free variables: to decide if $\mathcal{I}, \alpha \models \varphi$, only free variables in α matter
- sentences (closed formulas): formulas without free variables

Once the basic definitions are in place, predicate logic can be developed in the same way as propositional logic:

- logical consequences
- logical equivalences
- normal forms
- deduction theorem etc.

Other Logics

- We considered first-order predicate logic.
- Second-order predicate logic allows quantifying over predicate symbols.
- There are intermediate steps, e. g. monadic second-order logic (all quantified predicates are unary).
- Modal logics have new operators $\Box$ and $\Diamond$.
    - classical meaning: $\Box\varphi$ for "φ is necessary",
      $\Diamond\varphi$ for "φ is possible".
    - temporal logic: $\Box\varphi$ for "φ is always true in the future",
      $\Diamond\varphi$ for "φ is true at some point in the future"
    - deontic logic: $\Box\varphi$ for "φ is obligatory",
      $\Diamond\varphi$ for "φ is permitted"
    - ...
- In fuzzy logic, formulas are not true or false but have values between 0 and 1.

What's Next?

contents of this course:

- logic
  - ▷ How can knowledge be represented?
    How can reasoning be automated?
- automata theory and formal languages
  - ▷ What is a computation?
- computability theory
  - ▷ What can be computed at all?
- complexity theory
  - ▷ What can be computed efficiently?

What's Next?

contents of this course:

- logic ✓
  - ▷ How can knowledge be represented?
    How can reasoning be automated?
- automata theory and formal languages
  - ▷ What is a computation?
- computability theory
  - ▷ What can be computed at all?
- complexity theory
  - ▷ What can be computed efficiently?