Theory of Computer Science

B5. Predicate Logic II

Gabriele Röger

University of Basel

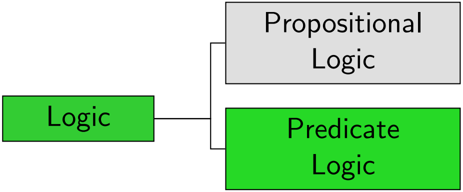
March 16, 2020

B5.1 Logical Consequences

B5.2 Further Topics

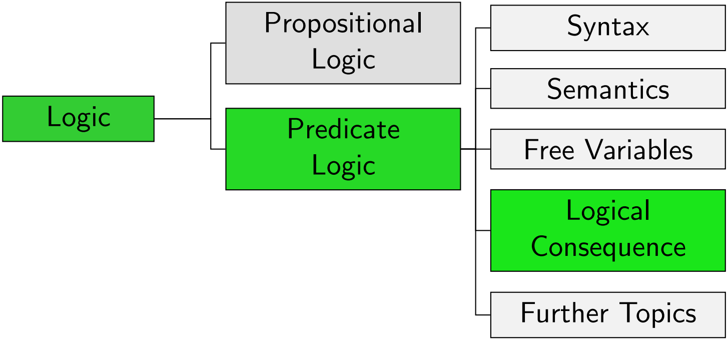
B5.3 Summary

Logic: Overview



B5.1 Logical Consequences

Logic: Overview



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 (Satisfied/True Sets of Formulas)

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

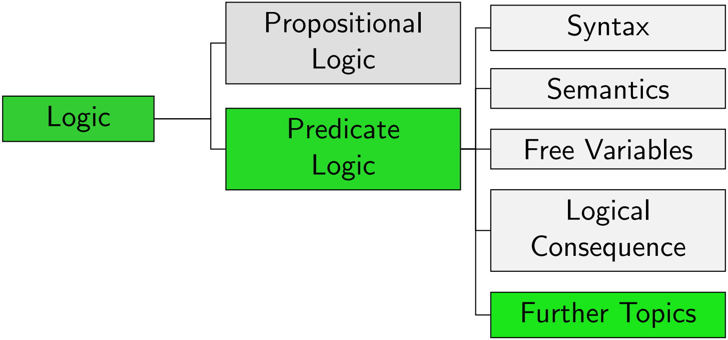
- Again, we use the same notations and concepts as in propositional logic.

  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.
- 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 $\psi \in \Psi$

B5.2 Further Topics

Logic: Overview



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

B5.3 Summary

Summary

- Predicate logic is more expressive than propositional logic and allows statements over objects and their properties.
- Objects are described by terms that are built from variable, constant and function symbols.
- Properties and relations are described by formulas that are built from predicates, quantifiers and the usual logical operators.
- 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

Summary

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:

A. background ✓
- ▷ mathematical foundations and proof techniques

B. logic ✓
- ▷ How can knowledge be represented?
  How can reasoning be automated?

C. automata theory and formal languages
- ▷ What is a computation?

D. Turing computability
- ▷ What can be computed at all?

E. complexity theory
- ▷ What can be computed efficiently?

F. more computability theory
- ▷ Other models of computability