

Theory of Computer Science

B4. Predicate Logic I

Gabriele Röger

University of Basel

March 4, 2019

Theory of Computer Science

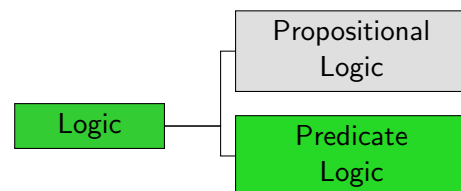
March 4, 2019 — B4. Predicate Logic I

B4.1 Motivation

B4.2 Syntax of Predicate Logic

B4.3 Summary

Logic: Overview



B4.1 Motivation

Limits of Propositional Logic

Cannot well be expressed in propositional logic:

- ▶ “Everyone who does the exercises passes the exam.”
- ▶ “If someone with administrator privileges presses ‘delete’, all data is gone.”
- ▶ “Everyone has a mother.”
- ▶ “If someone is the father of some person, the person is his child.”

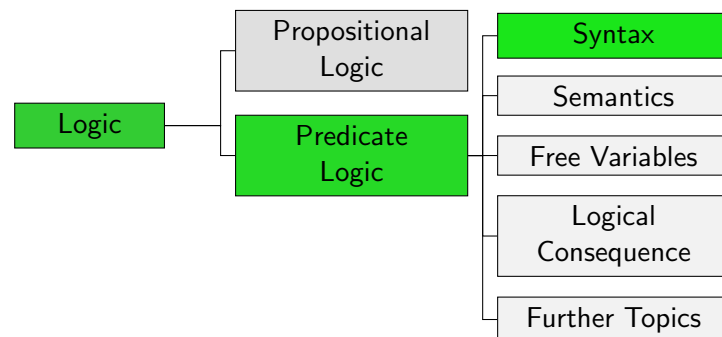
▷ need more expressive logic

↔ **predicate logic**

German: Prädikatenlogik

B4.2 Syntax of Predicate Logic

Logic: Overview



Syntax: Building Blocks

- ▶ **Signatures** define allowed symbols.
analogy: variable set A in propositional logic
- ▶ **Terms** are associated with objects by the semantics.
no analogy in propositional logic
- ▶ **Formulas** are associated with truth values (**true** or **false**) by the semantics.
analogy: formulas in propositional logic

German: Signatur, Term, Formel

Signatures: Definition

Definition (Signature)

A **signature** (of predicate logic) is a 4-tuple $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ consisting of the following four disjoint sets:

- ▶ a finite or countable set \mathcal{V} of **variable symbols**
- ▶ a finite or countable set \mathcal{C} of **constant symbols**
- ▶ a finite or countable set \mathcal{F} of **function symbols**
- ▶ a finite or countable set \mathcal{P} of **predicate symbols** (or **relation symbols**)

Every function symbol $f \in \mathcal{F}$ and predicate symbol $P \in \mathcal{P}$ has an associated **arity** $ar(f), ar(P) \in \mathbb{N}_0$ (number of arguments).

German: Variablen-, Konstanten-, Funktions-, Prädikat- und Relationssymbole; Stelligkeit

Signatures: Terminology and Conventions

terminology:

- ▶ ***k*-ary** (function or predicate) symbol: symbol s with arity $ar(s) = k$.
- ▶ also: **unary, binary, ternary**

German: *k*-stellig, unär, binär, ternär

conventions (in this lecture):

- ▶ variable symbols written in *italics*, other symbols upright.
- ▶ predicate symbols begin with capital letter, other symbols with lower-case letters

Signatures: Examples

Example: Arithmetic

- ▶ $\mathcal{V} = \{x, y, z, x_1, x_2, x_3, \dots\}$
- ▶ $\mathcal{C} = \{\text{zero, one}\}$
- ▶ $\mathcal{F} = \{\text{sum, product}\}$
- ▶ $\mathcal{P} = \{\text{Positive, SquareNumber}\}$

$ar(\text{sum}) = ar(\text{product}) = 2$, $ar(\text{Positive}) = ar(\text{SquareNumber}) = 1$

Signatures: Examples

Example: Genealogy

- ▶ $\mathcal{V} = \{x, y, z, x_1, x_2, x_3, \dots\}$
- ▶ $\mathcal{C} = \{\text{roger-federer, lisa-simpson}\}$
- ▶ $\mathcal{F} = \emptyset$
- ▶ $\mathcal{P} = \{\text{Female, Male, Parent}\}$

$ar(\text{Female}) = ar(\text{Male}) = 1$, $ar(\text{Parent}) = 2$

Terms: Definition

Definition (Term)

Let $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ be a signature.

A **term** (over \mathcal{S}) is inductively constructed according to the following rules:

- ▶ Every variable symbol $v \in \mathcal{V}$ is a term.
- ▶ Every constant symbol $c \in \mathcal{C}$ is a term.
- ▶ If t_1, \dots, t_k are terms and $f \in \mathcal{F}$ is a function symbol with arity k , then $f(t_1, \dots, t_k)$ is a term.

German: Term

examples:

- ▶ x_4
- ▶ lisa-simpson
- ▶ $\text{sum}(x_3, \text{product}(\text{one}, x_5))$

Formulas: Definition

Definition (Formula)

For a signature $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ the set of predicate logic formulas (over \mathcal{S}) is inductively defined as follows:

- ▶ If t_1, \dots, t_k are terms (over \mathcal{S}) and $P \in \mathcal{P}$ is a k -ary predicate symbol, then the **atomic formula** (or the **atom**) $P(t_1, \dots, t_k)$ is a formula over \mathcal{S} .
- ▶ If t_1 and t_2 are terms (over \mathcal{S}), then the **identity** $(t_1 = t_2)$ is a formula over \mathcal{S} .
- ▶ If $x \in \mathcal{V}$ is a variable symbol and φ a formula over \mathcal{S} , then the **universal quantification** $\forall x \varphi$ and the **existential quantification** $\exists x \varphi$ are formulas over \mathcal{S} .

...

German: atomare Formel, Atom, Identität, Allquantifizierung, Existenzquantifizierung

Formulas: Definition

Definition (Formula)

For a signature $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ the set of predicate logic formulas (over \mathcal{S}) is inductively defined as follows:

...

- ▶ If φ is a formula over \mathcal{S} , then so is its **negation** $\neg \varphi$.
- ▶ If φ and ψ are formulas over \mathcal{S} , then so are the **conjunction** $(\varphi \wedge \psi)$ and the **disjunction** $(\varphi \vee \psi)$.

German: Negation, Konjunktion, Disjunktion

Formulas: Examples

Examples: Arithmetic and Genealogy

- ▶ $\text{Positive}(x_2)$
- ▶ $\forall x (\neg \text{SquareNumber}(x) \vee \text{Positive}(x))$
- ▶ $\exists x_3 (\text{SquareNumber}(x_3) \wedge \neg \text{Positive}(x_3))$
- ▶ $\forall x (x = y)$
- ▶ $\forall x (\text{sum}(x, x) = \text{product}(x, \text{one}))$
- ▶ $\forall x \exists y (\text{sum}(x, y) = \text{zero})$
- ▶ $\forall x \exists y (\text{Parent}(y, x) \wedge \text{Female}(y))$

Terminology: The symbols \forall and \exists are called **quantifiers**.

German: Quantoren

Abbreviations and Placement of Parentheses by Convention

abbreviations:

- ▶ $(\varphi \rightarrow \psi)$ is an abbreviation for $(\neg\varphi \vee \psi)$.
- ▶ $(\varphi \leftrightarrow \psi)$ is an abbreviation for $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$.
- ▶ Sequences of the same quantifier can be abbreviated.
For example:
 - ▶ $\forall x\forall y\forall z \varphi \rightsquigarrow \forall xyz \varphi$
 - ▶ $\exists x\exists y\exists z \varphi \rightsquigarrow \exists xyz \varphi$
 - ▶ $\forall w\exists x\exists y\forall z \varphi \rightsquigarrow \forall w\exists xy\forall z \varphi$

placement of parentheses by convention:

- ▶ analogous to propositional logic
- ▶ quantifiers \forall and \exists bind more strongly than anything else.
- ▶ example: $\forall x P(x) \rightarrow Q(x)$ corresponds to $(\forall x P(x) \rightarrow Q(x))$,
not $\forall x (P(x) \rightarrow Q(x))$.

B4.3 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.