

Propositional and Predicate Logic - VI

Petr Gregor

KTIML MFF UK

WS 2015/2016

Predicate logic

Deals with statements about objects, their properties and relations.

“She is intelligent and her father knows the rector.”

$$I(x) \wedge K(f(x), r)$$

- x is a **variable**, representing an object,
- r is a **constant symbol**, representing a concrete object,
- f is a **function symbol**, representing a function,
- I, K are **relation (predicate) symbols**, representing relations (the property of “being intelligent” and the relation “to know”).

“Everybody has a father.”

$$(\forall x)(\exists y)(y = f(x))$$

- $(\forall x)$ is the **universal quantifier** (*for every x*),
- $(\exists y)$ is the **existential quantifier** (*there exists y*),
- $=$ is a (binary) **relation symbol**, representing the identity relation.

Language

A first-order language consists of

- **variables** $x, y, z, \dots, x_0, x_1, \dots$ (countable many),
the set of all variables is denoted by **Var**,
- **function symbols** f, g, h, \dots , including **constant symbols** c, d, \dots ,
which are nullary function symbols,
- **relation (predicate) symbols** P, Q, R, \dots , eventually the symbol $=$
(**equality**) as a special relation symbol,
- **quantifiers** $(\forall x), (\exists x)$ for every variable $x \in \text{Var}$,
- **logical connectives** $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- **parentheses** $(,)$

Every function and relation symbol S has an associated **arity** $\text{ar}(S) \in \mathbb{N}$.

***Remark** Compared to propositional logic we have no (explicit) propositional variables, but they can be introduced as nullary relation symbols.*

Signatures

- *Symbols of logic* are variables, quantifiers, connectives and parentheses.
- *Non-logical symbols* are function and relation symbols except the equality symbol. The equality is (usually) considered separately.
- A *signature* is a pair $\langle \mathcal{R}, \mathcal{F} \rangle$ of disjoint sets of relation and function symbols with associated arities, whereas none of them is the equality symbol. A signature lists all non-logical symbols.
- A *language* is determined by a signature $L = \langle \mathcal{R}, \mathcal{F} \rangle$ and by specifying whether it is a language with equality or not. A language must contain at least one relation symbol (non-logical or the equality).

Remark The meaning of symbols in a language is not assigned, e.g. the symbol $+$ does not have to represent the standard addition.

Examples of languages

We describe a language by a list of all non-logical symbols with eventual clarification of arity and whether they are relation or function symbols.

The following examples of languages are all with **equality**.

- $L = \langle \rangle$ is the language of **pure** equality,
- $L = \langle c_i \rangle_{i \in \mathbb{N}}$ is the language of countable many constants,
- $L = \langle \leq \rangle$ is the language of **orderings**,
- $L = \langle E \rangle$ is the language of the **graph** theory,
- $L = \langle +, -, 0 \rangle$ is the language of the **group** theory,
- $L = \langle +, -, \cdot, 0, 1 \rangle$ is the language of the **field** theory,
- $L = \langle -, \wedge, \vee, 0, 1 \rangle$ is the language of **Boolean algebras**,
- $L = \langle S, +, \cdot, 0, \leq \rangle$ is the language of **arithmetic**,

where $c_i, 0, 1$ are constant symbols, $S, -$ are unary function symbols, $+, \cdot, \wedge, \vee$ are binary function symbols, E, \leq are binary relation symbols.

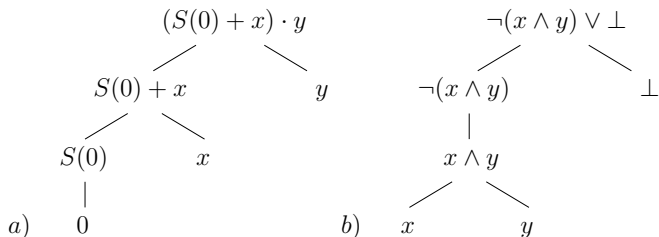
Terms

Are expressions representing values of (composed) functions.

Terms of a language L are defined inductively by

- (i) every variable or constant symbol in L is a term,
 - (ii) if f is a function symbol in L of arity $n > 0$ and t_0, \dots, t_{n-1} are terms, then also the expression $f(t_0, \dots, t_{n-1})$ is a term,
 - (iii) every term is formed by a **finite** number of steps (i), (ii).
- A **ground term** is a term with no variables.
 - The set of all terms of a language L is denoted by Term_L .
 - A term that is a part of another term t is called a **subterm** of t .
 - The structure of terms can be represented by their **formation trees**.
 - For binary function symbols we often use **infix** notation, e.g. we write $(x + y)$ instead of $+(x, y)$.

Examples of terms



- a) The formation tree of the term $(S(0) + x) \cdot y$ of the language of arithmetic.
- b) Propositional formulas only with connectives \neg , \wedge , \vee , eventually with constants \top , \perp can be viewed as terms of the language of Boolean algebras.

Atomic formulas

Are the simplest formulas.

- An *atomic formula* of a language L is an expression $R(t_0, \dots, t_{n-1})$ where R is an n -ary relation symbol in L and t_0, \dots, t_{n-1} are terms of L .
- The set of all atomic formulas of a language L is denoted by AFm_L .
- The structure of an atomic formula can be represented by a **formation tree** from the formation subtrees of its terms.
- For binary relation symbols we often use **infix** notation, e.g. $t_1 = t_2$ instead of $=(t_1, t_2)$ or $t_1 \leq t_2$ instead of $\leq(t_1, t_2)$.
- *Examples of atomic formulas*

$$K(f(x), r), \quad x \cdot y \leq (S(0) + x) \cdot y, \quad \neg(x \wedge y) \vee \perp = \perp.$$

Formula

Formulas of a language L are defined inductively by

- (i) every atomic formula is a formula,
 - (ii) if φ, ψ are formulas, then also the following expressions are formulas
 $(\neg\varphi), (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi),$
 - (iii) if φ is a formula and x is a variable, then also the expressions $((\forall x)\varphi)$
and $((\exists x)\varphi)$ are formulas.
 - (iv) every formula is formed by a **finite** number of steps (i), (ii), (iii).
- The set of all formulas of a language L is denoted by \mathbf{Fm}_L .
 - A formula that is a part of another formula φ is called a *subformula* of φ .
 - The structure of formulas can be represented by their **formation trees**.

Conventions

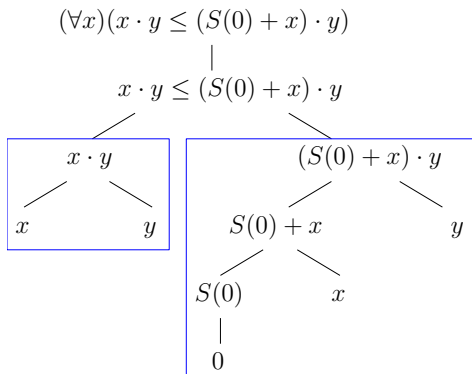
- After introducing *priorities* for binary function symbols e.g. $+$, \cdot we are in *infix* notation allowed to omit parentheses that are around a subterm formed by a symbol of *higher* priority, e.g. $x \cdot y + z$ instead of $(x \cdot y) + z$.
- After introducing *priorities* for connectives and quantifiers we are allowed to omit parentheses that are around subformulas formed by connectives of *higher* priority.

$$(1) \rightarrow, \leftrightarrow \quad (2) \wedge, \vee \quad (3) \neg, (\forall x), (\exists x)$$

- They can be always omitted around subformulas formed by $\neg, (\forall x), (\exists x)$.
- We may also omit parentheses in $(\forall x)$ and $(\exists x)$ for every $x \in \text{Var}$.
- The outer parentheses may be omitted as well.

$$\begin{aligned} & (((\neg((\forall x)R(x))) \wedge ((\exists y)P(y))) \rightarrow (\neg(((\forall x)R(x)) \vee (\neg((\exists y)P(y)))))) \\ & \neg\forall xR(x) \wedge \exists yP(y) \rightarrow \neg(\forall xR(x) \vee \neg\exists yP(y)) \end{aligned}$$

An example of a formula



The formation tree of the formula $(\forall x)(x \cdot y \leq (S(0) + x) \cdot y)$.

Occurrences of variables

Let φ be a formula and x be a variable.

- An **occurrence** of x in φ is a leaf labeled by x in the formation tree of φ .
- An occurrence of x in φ is **bound** if it is in some subformula ψ that starts with $(\forall x)$ or $(\exists x)$. An occurrence of x in φ is **free** if it is not bound.
- A variable x is **free** in φ if it has at least one free occurrence in φ . It is **bound** in φ if it has at least one bound occurrence in φ .
- A variable x can be both free and bound in φ . For example in

$$(\forall x)(\exists y)(x \leq y) \vee x \leq z.$$

- We write $\varphi(x_1, \dots, x_n)$ to denote that x_1, \dots, x_n are all free variables in the formula φ . (φ states something about these variables.)

Remark We will see that the truth value of a formula (in a given interpretation of symbols) depends only on the assignment of free variables.

Open and closed formulas

- A formula is *open* if it is without quantifiers. For the set OFm_L of all open formulas in a language L it holds that $\text{AFm}_L \subsetneq \text{OFm}_L \subsetneq \text{Fm}_L$.
- A formula is *closed* (a *sentence*) if it has no free variable; that is, all occurrences of variables are bound.
- A formula can be both open and closed. In this case, all its terms are ground terms.

$x + y \leq 0$	<i>open</i> , $\varphi(x, y)$
$(\forall x)(\forall y)(x + y \leq 0)$	<i>a sentence</i> ,
$(\forall x)(x + y \leq 0)$	<i>neither open nor a sentence</i> , $\varphi(y)$
$1 + 0 \leq 0$	<i>open sentence</i>

Remark We will see that in a fixed interpretation of symbols a sentence has a fixed truth value; that is, it does not depend on the assignment of variables.

Instances

After *substituting* a term t for a free variable x in a formula φ , we would expect that the new formula (newly) says about t “the same” as φ did about x .

$\varphi(x)$	$(\exists y)(x + y = 1)$	“there is an element $1 - x$ ”
for $t = 1$ we can $\varphi(x/t)$	$(\exists y)(1 + y = 1)$	“there is an element $1 - 1$ ”
for $t = y$ we cannot	$(\exists y)(y + y = 1)$	“1 is divisible by 2”

- A term t is **substitutable** for a variable x in a formula φ if substituting t for all free occurrences of x in φ does not introduce a new bound occurrence of a variable from t .
- Then we denote the obtained formula $\varphi(x/t)$ and we call it an **instance** of the formula φ after a **substitution** of a term t for a variable x .
- t is not substitutable for x in φ if and only if x has a free occurrence in some subformula that starts with $(\forall y)$ or $(\exists y)$ for some variable y in t .
- **Ground** terms are always substitutable.

Variants

Quantified variables can be (under *certain* conditions) renamed so that we obtain an equivalent formula.

Let $(Qx)\psi$ be a subformula of φ where Q means \forall or \exists and y is a variable such that the following conditions hold.

- 1) y is **substitutable** for x in ψ , and
- 2) y does not have a **free** occurrence in ψ .

Then by replacing the subformula $(Qx)\psi$ with $(Qy)\psi(x/y)$ we obtain a **variant** of φ **in subformula** $(Qx)\psi$. After variation of one or more subformulas in φ we obtain a **variant** of φ . For example,

$(\exists x)(\forall y)(x \leq y)$	is a formula φ ,
$(\exists u)(\forall v)(u \leq v)$	is a variant of φ ,
$(\exists y)(\forall y)(y \leq y)$	is not a variant of φ , 1) does not hold,
$(\exists x)(\forall x)(x \leq x)$	is not a variant of φ , 2) does not hold.

Structures

- $\underline{S} = \langle S, \leq \rangle$ is an **ordered** set where \leq is reflexive, antisymmetric, transitive binary relation on S ,
- $G = \langle V, E \rangle$ is an undirected **graph** without loops where V is the set of *vertices* and E is irreflexive, symmetric binary relation on V (*adjacency*),
- $\underline{\mathbb{Z}}_p = \langle \mathbb{Z}_p, +, -, 0 \rangle$ is the additive **group** of integers modulo p ,
- $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, -, \cdot, 0, 1 \rangle$ is the **field** of rational numbers,
- $\underline{\mathcal{P}(X)} = \langle \mathcal{P}(X), -, \cap, \cup, \emptyset, X \rangle$ is the **set algebra** over X ,
- $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ is the standard model of **arithmetic**,
- finite automata and other models of computation,
- relational databases, . . .

A structure for a language

Let $L = \langle \mathcal{R}, \mathcal{F} \rangle$ be a signature of a language and A be a nonempty set.

- A **realization** (*interpretation*) of a **relation symbol** $R \in \mathcal{R}$ on A is any relation $R^A \subseteq A^{\text{ar}(R)}$. A realization of $=$ on A is the relation Id_A (identity).
- A **realization** (*interpretation*) of a **function symbol** $f \in \mathcal{F}$ on A is any function $f^A: A^{\text{ar}(f)} \rightarrow A$. Thus a realization of a **constant symbol** is some element of A .

A **structure** for the language L (***L-structure***) is a triple $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$, where

- A is nonempty set, called the **domain** of the structure \mathcal{A} ,
- $\mathcal{R}^A = \langle R^A \mid R \in \mathcal{R} \rangle$ is a **collection** of realizations of relation symbols,
- $\mathcal{F}^A = \langle f^A \mid f \in \mathcal{F} \rangle$ is a **collection** of realizations of function symbols.

A structure for the language L is also called a **model of the language** L . The class of all models of L is denoted by $M(L)$. *Examples for $L = \langle \leq \rangle$ are*

$$\langle \mathbb{N}, \leq \rangle, \langle \mathbb{Q}, > \rangle, \langle X, E \rangle, \langle \mathcal{P}(X), \subseteq \rangle.$$

Value of terms

Let t be a term of $L = \langle \mathcal{R}, \mathcal{F} \rangle$ and $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be an L -structure.

- A *variable assignment* over the domain A is a function $e: \text{Var} \rightarrow A$.
- The *value* $t^A[e]$ of the term t in the structure \mathcal{A} with respect to the assignment e is defined by

$$x^A[e] = e(x) \quad \text{for every } x \in \text{Var},$$

$$(f(t_0, \dots, t_{n-1}))^A[e] = f^A(t_0^A[e], \dots, t_{n-1}^A[e]) \quad \text{for every } f \in \mathcal{F}.$$

- In particular, for a constant symbol c we have $c^A[e] = c^A$.
- If t is a **ground** term, its value in \mathcal{A} is independent on the assignment e .
- The value of t in \mathcal{A} depends only on the assignment of variables in t .

For example, the value of the term $x + 1$ in the structure $\mathcal{N} = \langle \mathbb{N}, +, 1 \rangle$ with respect to the assignment e with $e(x) = 2$ is $(x + 1)^{\mathcal{N}}[e] = 3$.

Values of atomic formulas

Let φ be an **atomic** formula of $L = \langle \mathcal{R}, \mathcal{F} \rangle$ in the form $R(t_0, \dots, t_{n-1})$,
 $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be an L -structure, and e be a variable assignment over A .

- The **value** $H_{at}^A(\varphi)[e]$ of the formula φ in the structure \mathcal{A} with respect to e is

$$H_{at}^A(R(t_0, \dots, t_{n-1}))[e] = \begin{cases} 1 & \text{if } (t_0^A[e], \dots, t_{n-1}^A[e]) \in R^A, \\ 0 & \text{otherwise.} \end{cases}$$

where $=^A$ is Id_A ; that is, $H_{at}^A(t_0 = t_1)[e] = 1$ if $t_0^A[e] = t_1^A[e]$, and
 $H_{at}^A(t_0 = t_1)[e] = 0$ otherwise.

- If φ is a sentence; that is, all its terms are **ground**, then its value in \mathcal{A} is independent on the assignment e .
- The value of φ in \mathcal{A} depends only on the assignment of variables in φ .

For example, the value of φ in form $x + 1 \leq 1$ in $\mathcal{N} = \langle \mathbb{N}, +, 1, \leq \rangle$ with respect to the assignment e is $H_{at}^{\mathcal{N}}(\varphi)[e] = 1$ if and only if $e(x) = 0$.

Values of formulas

The *value* $H^A(\varphi)[e]$ of the formula φ in the structure \mathcal{A} with respect to e is

$$H^A(\varphi)[e] = H_{at}^A(\varphi)[e] \text{ if } \varphi \text{ is atomic,}$$

$$H^A(\neg\varphi)[e] = \neg_1(H^A(\varphi)[e])$$

$$H^A(\varphi \wedge \psi)[e] = \wedge_1(H^A(\varphi)[e], H^A(\psi)[e])$$

$$H^A(\varphi \vee \psi)[e] = \vee_1(H^A(\varphi)[e], H^A(\psi)[e])$$

$$H^A(\varphi \rightarrow \psi)[e] = \rightarrow_1(H^A(\varphi)[e], H^A(\psi)[e])$$

$$H^A(\varphi \leftrightarrow \psi)[e] = \leftrightarrow_1(H^A(\varphi)[e], H^A(\psi)[e])$$

$$H^A((\forall x)\varphi)[e] = \min_{a \in A}(H^A(\varphi)[e(x/a)])$$

$$H^A((\exists x)\varphi)[e] = \max_{a \in A}(H^A(\varphi)[e(x/a)])$$

where $\neg_1, \wedge_1, \vee_1, \rightarrow_1, \leftrightarrow_1$ are the Boolean functions given by the tables and $e(x/a)$ for $a \in A$ denotes the assignment obtained from e by setting $e(x) = a$.

Observation $H^A(\varphi)[e]$ depends only on the assignment of *free* variables in φ .

Satisfiability with respect to assignments

The structure \mathcal{A} **satisfies** the formula φ **with assignment** e if $H^A(\varphi)[e] = 1$.

Then we write $\mathcal{A} \models \varphi[e]$, and $\mathcal{A} \not\models \varphi[e]$ otherwise. It holds that

$\mathcal{A} \models \neg\varphi[e]$	\Leftrightarrow	$\mathcal{A} \not\models \varphi[e]$
$\mathcal{A} \models (\varphi \wedge \psi)[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e]$ and $\mathcal{A} \models \psi[e]$
$\mathcal{A} \models (\varphi \vee \psi)[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e]$ or $\mathcal{A} \models \psi[e]$
$\mathcal{A} \models (\varphi \rightarrow \psi)[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e]$ implies $\mathcal{A} \models \psi[e]$
$\mathcal{A} \models (\varphi \leftrightarrow \psi)[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e]$ if and only if $\mathcal{A} \models \psi[e]$
$\mathcal{A} \models (\forall x)\varphi[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e(x/a)]$ for every $a \in A$
$\mathcal{A} \models (\exists x)\varphi[e]$	\Leftrightarrow	$\mathcal{A} \models \varphi[e(x/a)]$ for some $a \in A$

Observation Let term t be **substitutable** for x in φ and ψ be a **variant** of φ .

Then for every structure \mathcal{A} and assignment e

- 1) $\mathcal{A} \models \varphi(x/t)[e]$ if and only if $\mathcal{A} \models \varphi[e(x/a)]$ where $a = t^A[e]$,
- 2) $\mathcal{A} \models \varphi[e]$ if and only if $\mathcal{A} \models \psi[e]$.

Validity in a structure

Let φ be a formula of a language L and \mathcal{A} be an L -structure.

- φ is **valid (true) in the structure \mathcal{A}** , denoted by $\mathcal{A} \models \varphi$, if $\mathcal{A} \models \varphi[e]$ for every $e: \text{Var} \rightarrow A$. We say that \mathcal{A} **satisfies** φ . Otherwise, we write $\mathcal{A} \not\models \varphi$.
- φ is **contradictory in \mathcal{A}** if $\mathcal{A} \models \neg\varphi$; that is, $\mathcal{A} \not\models \varphi[e]$ for every $e: \text{Var} \rightarrow A$.
- For every formulas φ, ψ , variable x , and structure \mathcal{A}

$$(1) \quad \mathcal{A} \models \varphi \quad \Rightarrow \quad \mathcal{A} \not\models \neg\varphi$$

$$(2) \quad \mathcal{A} \models \varphi \wedge \psi \quad \Leftrightarrow \quad \mathcal{A} \models \varphi \text{ and } \mathcal{A} \models \psi$$

$$(3) \quad \mathcal{A} \models \varphi \vee \psi \quad \Leftrightarrow \quad \mathcal{A} \models \varphi \text{ or } \mathcal{A} \models \psi$$

$$(4) \quad \mathcal{A} \models \varphi \quad \Leftrightarrow \quad \mathcal{A} \models (\forall x)\varphi$$

- If φ is a **sentence**, it is valid or contradictory in \mathcal{A} , and thus (1) holds also in \Leftarrow . If moreover ψ is a sentence, also (3) holds in \Rightarrow .
- By (4), $\mathcal{A} \models \varphi$ if and only if $\mathcal{A} \models \psi$ where ψ is a **universal closure** of φ , i.e. a formula $(\forall x_1) \cdots (\forall x_n)\varphi$ where x_1, \dots, x_n are all **free** variables in φ .