COMP340-08B

Reasoning About Programs

5. Syntax of Predicate Logic

Robi Malik

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato
DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

Moving on: Predicate Logic

- Propositional logic is a subset of predicate logic.
- Predicate logic adds **quantifiers**:
  - $\exists$ "there exists"
  - $\forall$ "for all"
- These additional components allow us to express relationships propositional logic cannot express.

Applications of Predicate Logic

- Reasoning in Mathematics, Philosophy, …
- Reasoning about Computer Programs
- Software Verification
- Artificial Intelligence

An Example of Predicate Logic

Every man is mortal. Socrates is a man.	*Premises*
Therefore, Socrates is mortal.	*Conclusion*

Formalising the argument:

$\text{man}(x)$ = "x is a man" $\text{mortal}(x)$ = "x is mortal"

$\forall x\,(\text{man}(x) \rightarrow \text{mortal}(x))$ $\text{man}(\text{socrates})$	*Premises*
$\text{mortal}(\text{socrates})$	*Conclusion*

Predicates

- Predicates are used to describe properties of objects or relations between them.
- A predicate with one argument describes the property of an object (e.g. $\text{mortal}(x)$)
- A predicate with several arguments describes a relation between objects (e.g. $\text{smaller}(x, y)$)

Atoms of Predicate Logic

Propositional logic has **propositions**

$$\text{p},\ \text{p}_1,\ \text{p}_2,\ldots,\ \text{q},\ \text{r},\ldots$$

Predicate logic replaces this by **predicates**

$$\text{p}(T_1,T_2,\ldots,T_n)$$

where $T_1,T_2,\ldots,T_n$ are **terms**.

Terms can be

- **variables**: $x,\ x_1,\ x_2,\ y,\ z,\ \ldots$
- names of objects: $\text{a},\ \text{a}_1,\ \text{a}_2,\ \text{b},\ \text{c},\ \ldots$ "**constants**"
- constructed using **functions**: $\text{f}(x),\ \text{f}(\text{g}(x,\text{a})),\ \ldots$

Representation of Objects

Predicate logic talks about **objects**.

Objects can be referenced using:

- Variable symbols: $x, y, z, x_1, y_1, z_1, \ldots$
- Constant symbols: socrates, 0, 1, a, b, …
- Function symbols: $\mathrm{f}(x)$, plus(0,1), …

All the above will be called **terms**.

First-Order Signature

Definition:

A **signature** is a triple $\Sigma = (\mathcal{P}, \mathcal{F}, \mathcal{V})$, where

- $\mathcal{P}$ is a finite set of **predicate symbols**;
- $\mathcal{F}$ is a finite set of **function symbols**;
- $\mathcal{V}$ is a finite set of **variable symbols**.
- Each predicate and function symbol has an attached **arity**, i.e. a number identifying the number of arguments it can take.

Example

$x \in \mathcal{V}$
a **variable** symbol

mortal $\in \mathcal{P}$
a **1-ary predicate** symbol

$\forall x\,(\mathrm{man}(x) \rightarrow \mathrm{mortal}(x))$
man(socrates)

---

mortal(socrates)

socrates $\in \mathcal{F}$
a **0-ary function** symbol,
i.e. a **constant** symbol

Constructing Terms

Definition:

Let $\Sigma = (\mathcal{P}, \mathcal{F}, \mathcal{V})$ be a signature.
The set $\mathbf{TERM}_\Sigma$ of terms for Σ is defined recursively:

- For each $x \in \mathcal{V}$, we have $x \in \mathbf{TERM}_\Sigma$.
- If $\mathrm{a} \in \mathcal{F}$ is a constant symbol, then $\mathrm{a} \in \mathbf{TERM}_\Sigma$.
- If $\mathrm{f} \in \mathcal{F}$ is an n-ary function symbol, $n \geq 1$, and $T_1,\ldots,T_n \in \mathbf{TERM}_\Sigma$, then $\mathrm{f}(T_1,\ldots,T_n) \in \mathbf{TERM}_\Sigma$.

Predicate Logic Formulas

Definition:

Let $\Sigma = (\mathcal{P}, \mathcal{F}, \mathcal{V})$ be a signature.
The set $\mathbf{WFF}_\Sigma$ of well-formed formulas of predicate logic for Σ is defined recursively:

- If $\mathrm{p} \in \mathcal{P}$ is an n-ary predicate symbol, and $T_1,\ldots,T_n \in \mathbf{TERM}_\Sigma$, then $\mathrm{p}(T_1,\ldots,T_n) \in \mathbf{WFF}_\Sigma$.
- If $A, B \in \mathbf{WFF}_\Sigma$, then $\mathit{true}, \mathit{false} \in \mathbf{WFF}_\Sigma$, $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$, $(A \oplus B) \in \mathbf{WFF}_\Sigma$.
- If $x \in \mathcal{V}$ and $A \in \mathbf{WFF}_\Sigma$, then $\forall x\, A$, $\exists x\, A \in \mathbf{WFF}_\Sigma$.

Examples

There exists a man.
$\exists x\ \mathrm{man}(x)$

Socrates has a parent.
$\exists x\ \mathrm{parent}(x, \mathrm{socrates})$

Every man has a parent.
$\forall x\,(\mathrm{man}(x) \rightarrow \exists y\ \mathrm{parent}(y, x))$

Every man who has a child is happy.
$\forall x\,(\mathrm{man}(x) \wedge \exists z\ \mathrm{parent}(x, z) \rightarrow \mathrm{happy}(x))$

Scope of a Quantifier

Scope of $\forall x$

$$\forall x\ (\text{man}(x) \land \exists y\ \text{parent}(x, y) \rightarrow \text{brother}(x, z))$$

Scope of $\exists y$

Occurrence of x **bound** to $\forall x$.

Free occurrence of z.

Free Variables

- A variable x that does not occur in the scope of any quantifier ranging over x is called **free**.
- The meaning of free variables is unclear.
- Therefore, we prefer to consider formulas without any free variables.
- Such formulas are called **closed** formulas.

More Examples

Some mathematics …

For every natural number x, there exists a natural number y greater than x.

$$\forall x\ (\text{nat}(x) \rightarrow \exists y\ (\text{nat}(y) \land \text{gt}(y,x)))$$

For all natural numbers x and y, x is greater than or less than y, or they are equal.

$$\forall x \forall y\ (\text{nat}(x) \land \text{nat}(y) \rightarrow (\text{gt}(x,y) \lor \text{gt}(y,x) \lor \text{eq}(x,y)))$$

Primitive Encoding of Numbers

Non-negative integers can be characterised using two function symbols:

- Constant symbol 0 ("zero")
- Unary function symbol s ("successor")

This enables us to encode numbers:

0	1	2	3	…
0	s(0)	s(s(0))	s(s(s(0)))	…

More Examples

Using the primitive number encoding …

$$\forall x\ (\text{nat}(x) \rightarrow \text{gt}(\text{s}(x),0))$$

$$\forall x \forall y\ (\text{nat}(x) \land \text{nat}(y) \rightarrow (\text{gt}(x,y) \rightarrow \text{gt}(\text{s}(x),\text{s}(y))))$$

$$\forall x\ (\text{nat}(x) \rightarrow (\text{eq}(x,0) \lor \exists y\ (\text{nat}(y) \land \text{eq}(x,\text{s}(y)))))$$

Reading

Huth & Ryan:
Sections 2.1–2.2
pp. 93–107.