Tutorial on Predicate logic

Two types of predicate logic:

First-order logic

Extends propositional logic by including predicates and function symbols with "arity" greater or equal to 0, and quantifiers to range over objects.

Many-sorted logic

Extends first-order logic by allowing domains composed of more then one sort (i.e. type of objects).

First-order logic

A predicate logic language is like a programming language in that:

- it provides us with connectives and quantifiers, as a programming language would provide us with basic operations, e.g. addition on numbers, concatenation of lists,....
- it allows us to define (extra-logical) symbols that we can use to write formulae that describe specific problems, in the same was as we define, in a programming language, specific variables, types, etc. that we use to write a specific program.

The set of extra-logical symbols include:

- Constants
- Function symbols
- Predicate symbols

A first-order signature

Constants used to refer to objects in a given domain:

- e.g. a to refer to Peter, if the domain is a set of people
- 2, if the domain is the set Z,

Functions used to map (tuples of) objects to objects in a domain

- e.g. f( ) to refer to the mother of a person
- f( , ) to refer to minus operation over Z

Predicates used to describe relations on (tuples of) objects

- e.g. P( , ) to refer to the brotherhood relation
- Q( ) to refer to a set of even numbers

Structures

Interpretations define possible meanings of a given signature

Domain D: range of possible values over which variables can vary;

Signature | Interpretation
--- | ---
Constant: a | Domain D
Function: f(x₁,...,xₙ) | Function \( \bar{f}: \mathbb{D} \times \ldots \times \mathbb{D} \rightarrow \mathbb{D} \)
Predicate: P(x₁,...,xₙ) | Relation \( \bar{P} \subseteq \mathbb{D} \times \ldots \times \mathbb{D} \)
Models

Given a set of formulae, or theory, written in a given signature, models are those interpretations that make each formula in the theory true.

A theory is effectively a description of a particular set of models.

Examples

1) A is an even integer
   \[ \exists x. (A = 2 \times x) \]

2) The set of even integers
   \[ \forall y. (\text{Even}(y) \iff \exists x. (y = 2 \times x)) \]

3) A set of even integers
   \[ \forall y. (\text{Even}(y) \implies \exists x. (y = 2 \times x)) \]

4) All the employees of Doc are headed by Jeff
   \[ \forall x. (\text{employee}_{Doc}(x) \implies \text{headed}(x, \text{Jeff})) \]

5) Every first-year student has a tutor
   \[ \forall x. ((\text{student}(x) \land \text{first}\_\text{year}(x)) \implies \exists y. \text{tutor}(y, x)) \]

Many-sorted Logic

Signature

A set of sorts
Sorts are like types in a programming language

A set of constant symbols, each with its own sort

A set of predicate symbols, each with a given arity
*Arity* is a finite list of sorts, e.g. \( X_s = [X_1, \ldots, X_n] \)
\( P \) is of this arity: \( P \subseteq X_1 \times \cdots \times X_n \)

A set of function symbols, each with a given arity
*Arity* is a pair \((X_s, Y)\) where \( X_s \) is a list of sorts and \( Y \) is a sort,
\( F \) is a function of this arity: \( F : X_1 \times \cdots \times X_n \to Y \)

Formulae can be meaningless simply because they are not “well-typed”.

Structures for many-sorted logic

Assume a given signature, then a *structure* for it comprises:

For each sort \( X \),
\[ \mathcal{S} \] a corresponding set \([X]\), or carrier (or domain) of \( X \)

For each constant \( a \) of sort \( X \),
\[ \mathcal{S} \] an element \( a \) of \([X]\),

For each predicate symbol \( P \subseteq X_1 \times \cdots \times X_n \),
\[ \mathcal{S} \] a corresponding subset \( \tilde{P} \) of the Cartesian product \([X_1] \times \cdots \times [X_n]\)

For each function symbol \( f : X_1 \times \cdots \times X_n \to Y \),
\[ \mathcal{S} \] a corresponding function \( \tilde{F} \) from \([X_1] \times \cdots \times [X_n]\) to \([Y]\).
Examples

1) All IC students graduate with a first
\( \forall s: \text{StudentIC}. \ (\text{grade}(s) = \text{first}) \)

2) All employees whose salary is under 20000 pounds
\( \forall e: \text{Employees}. \ (\text{salary}(e) < 20000) \)

3) Italians read only Topolino
\( \forall b: \text{books}, \ \forall p: \text{people}. \ ((\text{italian}(p) \land \text{read}(p,b)) \rightarrow b = \text{topolino}) \)

4) All borrowed books are not in the library
\( \forall b: \text{Lib_books}. \ (\text{borrowed}(b) \rightarrow \text{status}(b) = \text{out_library}) \)

5) Students can borrow only one book
\( \forall t: \text{Time}, \ \forall s: \text{Students}, \ \forall b, b_1: \text{Lib_books}. \ (\text{borrowing}(s,b,t) \land \text{borrowing}(s,b_1,t) \rightarrow b = b_1) \)