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

} SIGNATURE

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

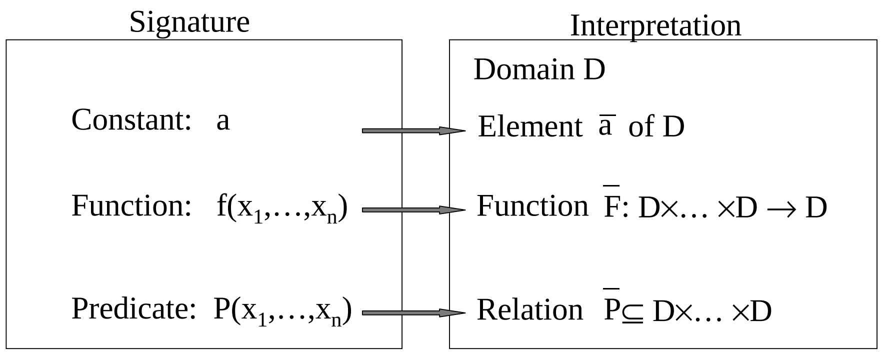
Predicates used to describe relations on (tuples of) objects

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;



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 a even integer
   $\exists x. (A = 2*x)$

2) The set of even integers
   $\forall y. (Even(y) \equiv \exists x. (y = 2*x)$

3) A set of even integers
   $\forall y. (Even(y) \rightarrow \exists x. (y = 2*x)$

4) All the employees of Doc are headed by Jeff
   $\forall x. (employee\_Doc(x) \rightarrow headed(x, Jeff))$

5) Every first-year student has a tutor
   $\forall x. ((student(x) \wedge first\_year(x)) \rightarrow \exists y.tutor(y,x))$

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. Xs = [X1, …, Xn],

P is of this arity: $P \subseteq X1 \times \ldots \times Xn$

A set of function symbols, each with a given arity

Arity is a pair (Xs,Y) where Xs is a list of sorts and Y is a sort,

F is a function of this arity: $F: X1 \times \ldots \times Xn \rightarrow Y$

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,

↪ a corresponding set [X], or carrier (or domain) of X

For each constant a of sort X,

↪ an element $\overline{a}$ of [X],

For each predicate symbol $P \subseteq X1 \times \ldots \times Xn$,

↪ a corresponding subset $\overline{P}$ of the Cartesian product $[X1] \times \ldots \times [Xn]$

For each function symbol f: $X1 \times \ldots \times Xn \rightarrow Y$,

↪ a corresponding function $\overline{F}$ from $[X1] \times \ldots \times [Xn]$ to [Y].

Examples

1) All IC students graduate with a first

   $\forall$s:StudentIC. (grade(s)=first)

2) All employees whose salary is under 20000 pounds

   $\forall$e:Employees. (salary(e) < 20000).

3) Italians read only Topolino

   $\forall$b:books, $\forall$p:people. [(italian(p)$\wedge$read(p,b)) $\rightarrow$ b=topolino]

4) All borrowed books are not in the library

   $\forall$b:Lib_books. (borrowed(b) $\rightarrow$ status(b) = out_library).

5) Students can borrow only one book

   $\forall$t:Time, $\forall$s:Students, $\forall$b,b1:Lib_books.
   (borrowing(s,b,t) $\wedge$ borrowing(s,b1,t) $\rightarrow$b=b1).