First Order Logic (3B)

Young W. Lim
12/14/15

Please send corrections (or suggestions) to youngwlim@hotmail.com.

First Order Logic

First-order logic is a formal system used in mathematics, philosophy, linguistics, and computer science. It is also known as **first-order predicate calculus**, the **lower predicate calculus**, **quantification theory**, and predicate logic. First-order logic uses quantified variables over (non-logical) objects. This distinguishes it from propositional logic which does not use quantifiers.

A theory about some topic is usually first-order logic together with a specified domain of discourse over which the quantified variables range, finitely many functions which map from that domain into it, finitely many predicates defined on that domain, and a recursive set of axioms which are believed to hold for those things. Sometimes "theory" is understood in a more formal sense, which i: just a set of sentences in first-order logic.

The adjective "first-order" distinguishes first-order logic from higher-order logic in which there are predicates having predicates or functions as arguments, or in which one or both of predicate quantifiers or function quantifiers are permitted.[1] In first-order theories, predicates are often associated with sets. In interpreted higher-order theories, predicates may be interpreted as sets of sets.

http://en.wikipedia.org/wiki/

First Order Logic

There are many deductive systems for first-order logic that are sound (all provable statements are true in all models) and complete (all statements which are true in all models are provable). Although the logical consequence relation is only semidecidable, much progress has been made in automated theorem proving in first-order logic. First-order logic also satisfies several metalogical theorems that make it amenable to analysis in proof theory, such as the Löwenheim–Skolem theorem and the compactness theorem.

First-order logic is the standard for the formalization of mathematics into axioms and is studied in the foundations of mathematics. Mathematical theories, such as number theory and set theory, have been formalized into first-order axiom schemas such as Peano arithmetic and Zermelo–Fraenkel set theory (ZF) respectively.

No first-order theory, however, has the strength to describe fully and categorically structures with an infinite domain, such as the natural numbers or the real line. Categorical axiom systems for these structures can be obtained in stronger logics such as second-order logic.

For a history of first-order logic and how it came to dominate formal lo Second-order logic rreirós (2001).

http://en.wikipedia.org/wiki/

(**grammar**) The part of the sentence (or clause) which states something about the subject or the object of the sentence.

In "The dog barked very loudly", the subject is "the dog" and the predicate is "barked very loudly".

(**logic**) A term of a statement, where the statement may be true or false depending on whether the thing referred to by the values of the statement's **variables** has the property signified by that (predicative) term.

From Middle French predicate (French prédicat), from post-classical Late Latin praedicatum (“thing said of a subject”), a noun use of the neuter past participle of praedicare (“proclaim”).

Predicate and Proposition

- A nullary predicate is a ***proposition***.
  Also, an instance of a predicate whose terms are all constant — e.g., P(2,3) — acts as a proposition.

- A ***predicate*** can be thought of
  as either a relation (between elements of the domain of discourse)
  or as a truth-valued function (of said elements).

- A ***predicate*** is either
  valid, (**all** interpretations make the predicate true)
  satisfiable, or (**an** interpretation makes the predicate true)
  unsatisfiable. (**no** interpretations make the predicate true)

- There are two ways of binding a predicate's variables:
  one is to assign **constant values** to those variables,
  the other is to **quantify** over those variables
  (using universal or existential quantifiers).

  If all of a predicate's variables are bound,
  the resulting formula is a ***proposition***.

Quantifier

In logic, **quantification** is a construct that specifies the quantity of specimens in the domain of discourse that satisfy an open formula. For example, in arithmetic, it allows the expression of the statement that every natural number has a successor. A language element which generates a quantification (such as "every") is called a **quantifier**. The resulting expression is a quantified expression, it is said to be **quantified** over the predicate (such as "the natural number *x* has a successor") whose free variable is bound by the quantifier. In formal languages, quantification is a formula constructor that produces new formulas from old ones. The semantics of the language specifies how the constructor is interpreted. Two fundamental kinds of quantification in predicate logic are universal quantification and existential quantification. The traditional symbol for the universal quantifier "all" is "∀", a rotated letter "A", and for the existential quantifier "exists" is "∃", a rotated letter "E". These quantifiers have been generalized beginning with the work of Mostowski and Lindström.

Quantification is used as well in natural languages; examples of quantifiers in English are *for all*, *for some*, *many*, *few*, *a lot*, and *no*; see Quantifier (linguistics) for details.

http://en.wikipedia.org/wiki/

Quantifier

If *D* is a domain of *x* and *P*(*x*) is a predicate dependent on *x*, then the universal proposition can be expressed as

$$\forall x \in D \; P(x)$$

This notation is known as restricted or relativized or bounded quantification. Equivalently,

$$\forall x \; (x \in D \to P(x))$$

The existential proposition can be expressed with bounded quantification as

$$\exists x \in D \; P(x)$$

or equivalently

$$\exists x \; (x \in D \land P(x))$$

Together with negation, only one of either the universal or existential quantifier is needed to perform both tasks:

$$\neg(\forall x \in D \; P(x)) \equiv \exists x \in D \; \neg P(x),$$

which shows that to disprove a "for all *x*" proposition, one needs no more than to find an *x* for which the predicate is false. Similarly,

$$\neg(\exists x \in D \; P(x)) \equiv \forall x \in D \; \neg P(x),$$

to disprove a "there exists an *x*" proposition, one needs to show that the predicate is false for all *x*.

http://en.wikipedia.org/wiki/

References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, “Lecture Notes : Introduction to Prolog Programming”
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] http://www.cs.odu.edu/~toida/nerzic/content/logic/

References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, "Lecture Notes : Introduction to Prolog Programming"
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html
[8] http://ilppp.cs.lth.se/, P. Nugues,` An Intro to Lang Processing with Perl and Prolog