Predicate Logic

Definition: a **predicate** is a function that yields a Boolean value.

A predicate is effectively a “parameterized Boolean value” — it may be true for some arguments, and false for others. For instance $x > 0$ is a predicate with a single argument, we could name it gt0(x). Then gt0(5) is true and gt0(0) is false.

The well-formed formulas (wffs) of predicate logic are more involved than those of propositional logic, and their elaboration is comprised of several definitions.

Definition: the elements of predicate logic wffs consist of the following components:
- variable identifiers – a set (normally infinite) of **variable names**, often $x, x_1, x_2, \dots, y, y_1, y_2, \dots$
- constant identifiers – a set (finite, infinite, or empty) of **constant names**, often $a, a_1, a_2, \dots, b, b_1, b_2, \dots$
- predicate identifiers – a (non-empty) set of **predicate names**, often $p, p_1, p_2, \dots, q, q_1, q_2, \dots$
- function identifiers – a set of **function names**, often $f, f_1, f_2, \dots, g, g_1, g_2, \dots$

Each function and predicate identifier has a fixed **arity** — the number of arguments it accepts.

Definition: the **terms** of the predicate logic are defined inductively as:
(i) variable names and constant names are terms, and
(ii) if $t_1, \dots, t_k$ are terms and f is a function name of arity k, then $f(t_1, t_2, \dots, t_k)$ is a term.
A term that contains no variables is called a **ground term**.

Definition: if $t_1, \dots, t_k$ are terms and predicate name p has arity k, then $p(t_1, \dots, t_k)$ is an **atomic formula** of the predicate logic.

The additional logical operations in predicate logic are **universal quantification**, ∀x — read as “for all x” (e.g., $\forall x.x^2 \geq 0$), and **existential quantification**, ∃x — read as “there exists x” (e.g., $\exists x.x^2 = 3$). In the precedence scheme for avoiding parentheses in formulas, ∀ and ∃ have the lowest precedence of all the connectives.

Definition: the **well-formed formulas** (wffs) of predicate logic are inductively defined as:
(i) each atomic formula is a wff, and
(ii) if □ and □ are wffs and x is a variable name, then each of the following is also a wff
- (□□)
- (□□□) • (□x.□)
- (□ □) • (□x.□)
- (□□ □) • (□⇔□).

These two quantification operations provide an indispensable means to express assertions about the truth outcomes of predicates. The interpretation of each of these logical operations depends on an understanding about the universe from which values of variables may be drawn. If this universe is finite, say $\{c_1, c_2, \dots, c_k\}$, then these new operations can be expressed using the propositional logic connectives. A formula (∀x.□) is equivalent to a conjunction of wffs obtained by replacing x by each of the items of the universe (e.g., $\forall x.p(x,y) \equiv p(c_1,y)$ □ $p(c_2,y)$ □ … □ $p(c_k,y)$). Similarly a formula (∃x.□) is equivalent to a disjunction of wffs obtained by replacing x by each of the items of the universe (e.g., $\exists x.p(x,y) \equiv p(c_1,y)$ $p(c_2,y)$ … $p(c_k,y)$). However, typically the universe is not finite so in general these new operations provide increased expressiveness.

The quantification operations require us to differentiate the use of variables. For instance, the formula p(x) has a parameter x, and may be true for some values and false for others. However, the formula ∀x.p(x) effectively has no parameters and represents a single unique value — the variable x is said to be *bound* in the latter case, and *free* in the former case. This illustrates two different roles for variables in predicate logic wffs that must be carefully distinguished.

Definition: the **bound** occurrences of variables in (□x.□) are the bound occurrences of variables in □, plus all occurrences of x in □; □ is called the **scope** of the quantification. All variable occurrences that are not bound are **free**. Similar definitions apply to (□x.□). A wff is called **closed** if it has no free variable occurrences.

Definition: an **interpretation** i consists of:
(i) a non-empty set D — the domain (or universe of values),
(ii) an assignment □ of
- each n-ary predicate name to an n-place relation on D,
- each n-ary function name to an n-place function on D,
- each constant identifier to an element of D.

We write i = (D, □).

An interpretation is a **term interpretation** if D is all terms, and the assignments for each function name is the corresponding term constructor, $□(f)(t_1, \dots, t_k) = f(t_1, \dots, t_k)$. A term interpretation using only ground terms is called a **Herbrand interpretation**.

Definition: given an interpretation i = (D, □), a **variable assignment** (or **state**) □ is a function on the set of variables V, □: V □ D. The assignment is inductively extended to yield a value for all terms and formulas,

(i) for terms
- for variable x, $val_□(x) = □(x)$, and for constant c, $val_□(c) = □(c)$,
- for a compound term $val_□(f(t_1, \dots, t_k)) = □(f)(val_□(t_1), \dots, val_□(t_k))$

(ii) for formulas
- for an atomic formula $val_□(p(t_1, \dots, t_k)) = □(p)(val_□(t_1), \dots, val_□(t_k))$
- for compound formulas

$val_□(□□) = □val_□(□)$,

$val_□(□□□) = val_□(□)\ □\ val_□(□)$,

$val_□(□\ □) = val_□(□)\ \ val_□(□)$,

$val_□(□□\ □) = val_□(□)\ □\ val_□(□)$,

$val_□(□ \Leftrightarrow □) = val_□(□) \Leftrightarrow val_□(□)$,

$$val_□(□x.□) = \begin{cases} \text{true if } val_{□'}(□) = \text{true for each } □' \text{ where } □'(y) = □(y) \text{ for } y \neq x \\ \text{false otherwise} \end{cases}$$

$$val_□(□x.□) = \begin{cases} \text{true if } val_{□'}(□) = \text{true for some } □' \text{ where } □'(y) = □(y) \text{ for } y \neq x \\ \text{false otherwise} \end{cases}$$

The last two definitions indicate that, given an interpretation and a state, a unique value is determined for each term and each formula by "evaluating" each logical operation. This provides the truth values we use to categorize formulas.

Definition: Let □ be a wff, i be an interpretation, and □ be a state. Then □ **satisfies** □ under i, $i \models_□ □$, if $val_□(□)$ = true. The wff □ is **true in i**, $i \models □$, if every state **satisfies** □ under i, and i is called a **model** of □; the wff □ is **false in i** if no state **satisfies** □ under i. An interpretation is called a **model of a set of wffs** if it is a model of each wff in the set, and if it is a term interpretation, it is called a **term model**.

Definition: A wff is **logically valid** (a **tautology**) if it is true in every interpretation, **satisfiable** if there exists an interpretation and state that satisfies it, and a **contradiction** if it is unsatisfiable.

Definition: A wff □ is a **logical consequence** of a set of wffs □, □ |= □, if every interpretation and state which satisfies each □□□ also satisfies □; □ and □ are **logically equivalent**, □ ≡ □, if for every interpretation and every state □, val□(□) = val□(□).

It turns out (i.e., can be proven) that logical validity and logical consequence can be determined by examining only term models. Term interpretations are completely determined by describing the ground atomic formulas that are true — this set of formulas is sometimes called a *Herbrand interpretation*. Herbrand models are often compared by comparing these sets — the *least model* assigns true values to the smallest set of formulas required to obtain a model.

There are a variety of helpful logical equivalencies in predicate logic and we note a few.

- □x.(□y.□) ≡ □y.(□x.□)
- □x.(□y.□) ≡ □y.(□x.□)
- □(□x.□) ≡ □x.(□□)
- □(□x.□) ≡ □x.(□□)

A **proof system** in predicate logic also comprises a collection of axioms, plus rules of inference. Of course, modus ponens and resolution are still sound rules of inference. A variety of rules of inference pertaining to quantification may be utilized. A few are cited here.

- universal generalization □ / □x.□ (UG)
- universal instantiation □x.□ / □[x∅ t] (UI)

  for each term t that "avoids name clashes", where □[x□ t] is wff □ with each occurrence of x in the scope replaced by t
- existential generalization □[x□ t] / □x.□ (EG)

  for each term t that "avoids name clashes", where □[x□ t] is wff □ with each occurrence of x in the scope replaced by t.

Sample Proof

This is a complete system for the propositional logic, where we only consider the logical operations □ and □ . This is not a limitation since every formula can be rewritten using just these two operations — for instance □ □ □ ≡ □(□ □ □□), and □ □ ≡ (□□) □ □. We can consideration these other operations as "abbreviations" with the reduced operation set.

<u>Axioms</u> (for all legal wffs □, □, □):

A1. □ □ (□ □ □)
A2. (□ □ (□ □ □)) □ ((□□ □) □ ((□ □ □))
A3. (□ □□ □□) □ ((□□ □ □) □ □)

<u>Proof</u>: for any wff □, |— □□ □

1. (□ □ ((□ □ □) □ □) □ ((□□ (□□ □))□ (□□ □)) by A2
2. □□ ((□□ □)□ □) by A1
3. (□□ (□□ □)) □ (□□ □) by modus ponens on 1 and 2
4. □□ (□□ □) by A1
5. □□ □ by modus ponens on 3 and 4