CISC 204   Class 14

Predicate Logic as a Formal Language

Text Correspondence: pp. 95–101

Main Concepts:

- *Variable: symbol for referring to an object in the universe of discourse*
- *Predicate: set of variable values*
- *Universal quantifier: the "for all" quantifier*
- *Existential quantifier: the "there exists" quantifier*

14.1 Predicate Logic as a Formal Language

The axiomatic definition of predicate logic begins with new concepts that must be accepted and understood intuitively. These are concepts we already know from basic mathematics:

Variable: a thing that varies. It is a member of some set, which is the universe of discourse (what the logic discusses).

Function: a map from variables to a variable value. We will call a function of zero variables a nullary function; of one variable, a unary function; of two variables, a binary function; and so on, with a function of n variables called an n-ary function. The set of all functions is denoted as $\mathcal{F}$. From this, we specify a special kind of function:

> **Constant:** A function of zero variables. This corresponds to a proposition.

Predicate: a thing that maps variables to truth values. When we get to semantics, we will see that a predicate is a set; for now, we will think of a predicate as a mapping from one or more variables to either **T** or **F**. The space of all predicates is denoted as $\mathcal{P}$. In general, $\mathcal{P} \neq \mathcal{F}$.

From these concepts and usages, we can define a *term* and a *formula*.

Definition: Term

- A variable is a term
- A constant is a term; a constant c is a nullary function in the space of all functions $\mathcal{F}$
- If f is an n-ary function in $\mathcal{F}$ and $t_1, ..., t_n$ are terms, then $f(t_1, ..., t_n)$ is a term
- nothing else is a term

A variable is often written as a lower-case symbol from the "end" of the Latin alphabet, such as x, y, etc. A constant is often written as a lower-case symbol from the "start" of the Latin alphabet, such as a, b, etc. A function is often written as a lower-case symbol from the "late start" of the Latin alphabet, such as f, g, etc.

Examples of terms:

a	constant
x	variable
$P(z)$	predicate
$f(a, x)$	function
$g(x, f(a, b), y)$	function

The last term is an example of how we can *nest* functions. This comes from the definition of a term, and how a function maps from variables to a variable-like value.

Axiomatically, we understand that if $P \in \mathcal{P}$ is an n-ary predicate symbol, and $t_1, t_2, \ldots, t_n$ are terms, then $P(t_1, t_2, \ldots, t_n)$ expresses the relation P among the terms. We usually use the word "predicate" to indicate a unary function, and "relation" to indicate a binary function, but these usages are not definitions and are not strictly observed.

We have already discussed the quantifiers. For conciseness, these are

$\forall x$ is the symbolic representation of "for all x"

$\exists x$ is the symbolic representation of "there exists an x"

The syntactic specification of predicate logic is finalized by defining a *formula* to include everything in propositional logic, plus simple predicate and quantified predicates.

Definition: Formula

- If $P \in \mathcal{P}$ is an n-ary predicate symbol, and $t_1, t_2, \ldots, t_n$ are terms, then $P(t_1, t_2, \ldots, t_n)$ is a formula
- If ϕ is a formula, the $(\neg\, \phi)$ is a formula
- If ϕ and ψ are formulas, then $(\phi \wedge \psi)$ and $(\phi \vee \psi)$ and $(\phi \to \psi)$ are formulas
- If ϕ is a formula and x is a variable, then $(\forall x\, \phi)$ is a formula and $(\exists x\, \phi)$ is a formula
- Nothing else is a formula

The first three items in this definition capture the syntax of propositional logic. If we think of propositions, such as p or q, as constant functions that are either **T** or **F**, then we can see that they meet the syntactic and semantic requirements of predicate logic. The new concepts are in the fourth item, where quantification is defined.

The fourth item is often mis-understood and may need to be the subject of further study by some students. For example, the symbolic string

$$(\forall x\, P(x, y))$$

is a formula but the variable y is not "mentioned" in the quantifier. This is still a formula! We will discuss this later in the course when we get to the *scope* of a quantifier. For now, as computer scientists, we understand that the variable y is "global" to the formula.

Extra Notes

Practice Problems for Predicates

Many students need practice in translating natural language into symbolic logic. These are some problems of the type that students are expected to be able to translate.

Translations

Using these predicates, and the nullary or constant function m,

$A(x, y)$:	x admires y
$B(x, y)$:	x attended y
$P(x)$:	x is a professor
$S(x)$:	x is a student
$L(x)$:	x is a lecture
m:	is Mary

Translate the following English sentences into predicate logic:

- Mary admires every professor.
- Some professor admires Mary.
- Mary admires herself.
- No student attended every lecture.
- No lecture was attended by every student.
- No lecture was attended by any student.

End of Extra Notes