Brief Introduction to Predicate Logic

This document provides a brief introduction to predicate logic. It is created for the students in C-S 743 - Software Verification and Validation. Interested readers who would like to read more about predicate logic should refer to books on Discrete Mathematics or Logic. This document uses some notations from the Z formal specification language.

Symbol	**Meaning**
$\mathbb{Z}$	set of integers
$\mathbb{N}$	set of natural numbers (integers greater than zero)
mod	binary operator that gives the remainder of integer division
div	binary operator that gives the quotient of integer division
$\in$	binary operator that indicates set membership

Definitions

A **predicate** is an assertion and is written as a sentence that contains a finite set of variables. When these variables are substituted by values, the sentence becomes a *proposition*. Each variable has a set of values that can be substituted in the predicate. This set of values is called the *domain* of that variable. A subset of the domain values will make the predicate become *true*; this subset is called the *truth set* of the variable with respect to the predicate. Consider the examples

$$successor(n) = 16$$
$$odd(n) \Rightarrow n \bmod 2 = 1$$

These predicates have only one variable, n. The domain of the variable in the first predicate is the set of all integer values and its truth set is $\{15\}$. The domain of the variable in the second predicate is the same set of all integers but the truth set of this predicate is the set of odd integers.
See the two-variables example given below:

$$ProfessorTeachesCourse(p, c)$$

The domain of the variable p is the set of all professors and the domain of the variable c is the set of all courses. The values 'Kasi Periyasamy' for p and 'CS 743' for c make this predicate true and hence these values belong to the truth set. The actual truth set in this case includes the set of all professors in the world and the set of courses they teach.

Quantified Statements

The examples mentioned in the previous paragraphs are said to be simple predicates which do not have any information on where to get the domain values or what the truth set is. Often, this additional information is given through quantifiers. There are two types of quantifiers - *universal quantifier* represented as '$\forall$', and *existential quantifier* represented as '$\exists$'.

A universal quantifier, when used in a logical statement, asserts that *for every value in the domain of each variable in the statement, the statement is true.*

An existential quantifier, when used in a logical statement, asserts that *there exists at least one value in the domain of each variable in the statement for which the statement is true.*

In both cases, the domain can be identified from the declaration of the variables given as part of the quantified statement. The following sections illustrate the use of these quantifiers in detail with several examples.

A general form of a quantified statement is given below:

$$< \textit{quantifier} > < \textit{declaration}(s) > \bullet < \textit{predicate} >$$

The symbol $\bullet$ is a separator between the declaration(s) and predicate; it is read as "such that". Consider the example below:

$$\forall\, n : \mathbb{Z} \bullet n + 1 = 1 + n$$

This statement asserts that "for every integer n, it is true that $n+1$ is the same as $1+n$". In other words, the operator '+' is commutative. See another example below:

$$\exists\, n : \mathbb{Z} \bullet n < \mathit{Max}$$

This statement asserts that there exists at least one integer n that is less than Max. It is expected that the variable Max is declared as an integer (or real) and its value is known. The variable n is said to be *bound* in the above statement because outside the statement, the value of n is undefined. On the other hand, the variable Max is said to be *free* in the statement because its value must be known before this statement and it continues to exist even after this statement. The notions of *free variable* and *bound variable* define the scope of variables used in a predicate.

As an exercise, find out the truth values of the following statements:

$$\forall\, n : \mathbb{Z} \bullet n \;\mathsf{mod}\; 2 = 0$$
$$\forall\, n : \mathbb{Z} \bullet (n - 1) < n$$
$$\exists\, n : \mathbb{Z} \bullet n - 1 = n + 1$$
$$\exists\, n : \mathbb{Z} \bullet n = n \;\mathsf{div}\; 2$$
$$\forall\, c : \mathit{Cat} \bullet c \in \mathit{Animal}$$
$$\exists\, c : \mathit{Tiger} \bullet c \in \mathit{Cat}$$

Negation of Quantified Statements

The negation of a universally quantified statement becomes an existentially quantified statement and vice versa. Formally,

$$\neg\, (\forall\, x : D \bullet P(x)) \text{ is equivalent to } \exists\, x : D \bullet \neg\, P(x)$$
$$\neg\, (\exists\, x : D \bullet P(x)) \text{ is equivalent to } \forall\, x : D \bullet \neg\, P(x)$$

Notice that the declaration part remains unchanged during negation.

Converse and Inverse Statements

The goal of studying predicates in this course is to translate informal statements in a natural language (such as English) to quantified statements so that we can derive additional statements (often called 'conclusion') from a given set of statements (often called 'premises'). This exercise is similar to translating informal statements into propositions, and consequently we will face two problems due to converse errors and inverse errors.

Converse Errors

Given the following statement

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$

the converse of the statement is

$$\forall\, x : D \bullet Q(x) \Rightarrow P(x)$$

The notion of converse is very important because sometimes the specification writer translates an informal statement into the converse of the original statement. This situation is called *converse error*.

Example:

Consider the statement

For all drinks, if a drink contains sugar then it is sweet.

The converse of this statement is

For all drinks, if a drink is sweet then it contains sugar.

Justify that the converse is not true! The above example is written in predicate notation as below:

$$\forall\, d : \mathit{Drink} \bullet \mathit{contains\_sugar}(d) \Rightarrow \mathit{sweet}(d)$$
$$\forall\, d : \mathit{Drink} \bullet \mathit{sweet}(d) \Rightarrow \mathit{contains\_sugar}(d)$$

As another example, consider the statement

For all integers n, if $n > 0$ then $n^2 > 0$

Rewrite the statement in predicate notation, write is converse and then prove that the converse is not true.

Inverse Errors

Given the following statement

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$

the inverse of the statement is

$$\forall\, x : D \bullet \neg\, P(x) \Rightarrow \neg\, Q(x)$$

As with converse error, the notion of inverse statement is also important. Once again, while translating informal statements, a specification writer may translate into the inverse instead of the original statement; this situation is called *inverse error.*

Write the inverse of the statements given in the two examples discussed earlier and argue whether or not they are true.
Notice that both converse and inverse errors will occur only with universally quantified statements with implication in the predicate part.

Inference Rules

Similar to the exercises on propositional logic, we will work out several problems using predicate logic. The problem statements will be given as informal statements. First, we will translate them into formal statements (predicates). For the purposes of this course, these statements will always be quantified statements with implications. After rewriting the statements, we will derive one or more conclusions using inference rules. We use three inference rules in the derivation process.

Universal Modus Ponens
Given a universally quantified statement

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$

and if $P(a)$ is true for a particular element $a \in D$, then the Universal Modus Ponens asserts that $Q(a)$ is true. In essence, Universal Modus Ponens implies substitution of arguments from the domain. The derivation steps in this case are written as below:

$$\begin{array}{l} \forall\, x : D \bullet P(x) \Rightarrow Q(x) \\ P(a)\ \mathit{for\ some}\ a \in D \\ \therefore Q(a) \end{array}$$

We can also use the following derivation step which is based on the previous rule:

$$\begin{array}{l} \forall\, x : D \bullet P(x) \Rightarrow Q(x) \\ \forall\, x : D \bullet Q(x) \Rightarrow R(x) \\ \therefore \forall\, x : D \bullet P(x) \Rightarrow R(x) \end{array}$$

Universal Modus Tollens
Given a universally quantified statement

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$

and if $\neg\ Q(a)$ is true for a particular element $a \in D$, then the Universal Modus Tollens asserts that $\neg\ P(a)$ is true. A derivation step using this rule will look like

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$
$$\neg\ Q(a)\ \textit{for some}\ a \in D$$
$$\therefore \neg\ P(a)$$

We will also be able to use the following:

$$\forall\, x : D \bullet P(x) \Rightarrow Q(x)$$
$$\therefore \forall\, x : D \bullet \neg\ Q(x) \Rightarrow \neg\ P(x)$$
$$\forall\, x : D \bullet Q(x) \Rightarrow R(x)$$
$$\therefore \forall\, x : D \bullet \neg\ R(x) \Rightarrow \neg\ Q(x)$$
$$\therefore \forall\, x : D \bullet \neg\ R(x) \Rightarrow \neg\ P(x)$$

Existential Elimination

This rule ie used when (and only when) an existential quantified statement appears in a problem. Informally, the rules says that if we are able to find at least one value in the domain of a variable for which the statement is true, then we could as well eliminate the quantified statement, substitute the value in place of the variable in the predicate and make it into a proposition. As an example, consider the statement

There exists an integer n which is even.

Formally, this statement becomes

$$\exists\, n : \mathbb{Z} \bullet n \bmod 2 = 0$$

Since $n = 10$ satisfies the property stated above, we could as well use 10 in place of this predicate. This rule is also called "One Point Rule".

Given an existentially quantified statement

$$\exists\, x : D \bullet P(x)$$

and if $P(a)$ is true for a particular value $x = a$, then you can eliminate the existential statement and use $P(a)$ as an assertion in its place.

Consider the following statements

$$\forall n : \mathbb{Z} \bullet n + 1 > 0 \Rightarrow n \geq 0$$
$$\forall n : \mathbb{Z} \bullet n > 0 \Rightarrow n^2 > 0$$
$$\exists n : \mathbb{Z} \bullet n > 3$$

In this set of statements, we start with the existential statement and try to find a value that satisfies this statement. Let us choose $n = 4$. Since $4 > 3$, we could eliminate the existential statement and use 4 in place of n in the other statements.
The first statement now becomes

$$4 + 1 > 0 \Rightarrow 4 \geq 0$$

Since the left side is true, the right side must be true (using Universal Modus Ponens). So we arrive at a statement $4 > 0$. Using this in the next predicate and using Universal Modus Ponens,

$$4 > 0 \Rightarrow 4^2 > 0$$

we arrive at the conclusion $4^2 > 0$.

<u>To Remember</u>

When the statements in the problem are translated into only universally quantified statements, then conclusion will also be another universally quantified statement. This is because we will use only the two rules Universal Modus Ponens and Universal Modus Tollens. If, on the other hand, the problem includes at least one existentially quantified statement, we will then start from this statement, eliminate the existential statement using One-point rule, use the value to substitute in other predicates and finally arrive at a conclusion based on this value.