Chapter 1: Part 2

Predicates and quantifiers are two fundamental concepts in mathematical logic and are used to express mathematical statements precisely.

A **predicate** is a statement that describes a property or relation that can be attributed to an object or set of objects. For example, "x is a prime number" is a predicate where x is a variable that can take on different values. We can make the statement true or false by substituting a value for x.

> Predicates are used to express both preconditions and post-conditions in a precise and formal way. By specifying preconditions and postconditions using predicates, we can reason about the correctness of a program using mathematical logic. This allows us to prove that a program will always produce the desired output for valid input, rather than relying on testing alone, which can only demonstrate the correctness of a program for a finite set of inputs.
>
> A **precondition** is a statement that specifies the conditions that must be met before a program can be executed correctly. It describes the valid input that the program can accept. For example, if we have a program that calculates the square root of a number, a precondition might be that the input must be a non-negative number.
>
> A **post-condition** is a statement that specifies the desired output or behavior of the program after it has been executed correctly. It describes the conditions that the output should satisfy when the program has run. For example, if we have a program that sorts a list of numbers, a post-condition might be that the output should be a sorted list in ascending order.

Quantifiers are used to express the scope of a predicate. There are two types of quantifiers:

The **universal quantifier**, denoted by the symbol $\forall$, is used to assert that a predicate is true for all objects in a set.

The **existential quantifier**, denoted by the symbol $\exists$, is used to assert that a predicate is true for at least one object in a set.

For example, the statement **$\forall x(x > 0) \land \exists y(y < 0)$** means **"for all x, x is greater than 0 and there exists a y that is less than 0"**

Statement	When True?	When False?
$\forall xP(x)$	P(x) is true for every x.	There is an x for which P(x) is false.
$\exists xP(x)$	There is an x for which P(x) is true.	P(x) is false for every x.

The uniqueness quantifier, denoted by $\exists!$ or $\exists 1$, is a type of quantifier that is used to express that there is exactly one element in the domain that satisfies a given property.

The statement **$\exists!xP(x)$** means "There exists a unique x such that P(x) is true." Other phrases for uniqueness quantification include "there is exactly one" and "there is one and only one."

However, uniqueness quantification can be expressed using universal and existential quantifiers and propositional logic.

For instance, the statement **$\exists!xP(x)$** is equivalent to **$(\exists xP(x) \land \forall y\forall z((P(y) \land P(z)) \rightarrow y=z))$**, which means "There exists an x such that P(x) is true, and for all y and z, if P(y) and P(z) are both true, then y and z are equal."

Connecting quantification *(can't apply to infinite numbers)* to looping and searching can be a useful way of determining the truth value of a quantification. Suppose that there are n objects in the domain for the variable x. To determine whether **$\forall xP(x)$** is **true**, we can loop through all n values of x and check whether P(x) is true for each value of x. If we encounter a value of x for which P(x) is false, then we have shown that $\forall xP(x)$ is false. Otherwise, $\forall xP(x)$ is true.

On the other hand, to determine whether **$\exists x P(x)$ is true**, we loop through the n values of x and search for a value of x for which P(x) is true. If we find such an x, then $\exists x P(x)$ is true. If we never find such an x, then we have determined that $\exists x P(x)$ is false.

Binding Variables

> When a quantifier is used on a variable, we say that this occurrence of the variable is bound. For example, in the expression $\forall x P(x)$, the variable x is bound by the universal quantifier $\forall$. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be free. For example, in the expression $P(x) \vee Q(y)$, the variable x is free and the variable y is free.

> To turn a propositional function into a proposition, all the variables that occur in the function must be bound or set equal to a particular value. This can be done using a combination of universal quantifiers, existential quantifiers, and value assignments. For example, the propositional function P(x) could be turned into the proposition $\forall x P(x)$ if we specify that the domain of x is some set that contains all the values that x can take.

> The part of a logical expression to which a quantifier is applied is called the scope of this quantifier. For example, in the expression **$\forall x(P(x) \rightarrow Q(x))$**, the scope of the universal quantifier $\forall x$ is $(P(x) \rightarrow Q(x))$. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable. For example, in the expression $\forall x(P(x) \vee Q(y))$, the variable y is free because it is outside the scope of the universal quantifier $\forall x$.

Negation	Equivalent Statement	When Is Negation True?	When Is Negation False?
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which P(x) is false.	P(x) is true for every x.
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, P(x) is false.	There is an x for which P(x) is true.

👤 **Lewis Carroll**