

1.4 1.4 Predicates and Quantifiers

1.4 Predicates and Quantifiers

Introduction

Propositional logic, studied in Sections 1.1–1.3, cannot adequately express the meaning of all statements in mathematics and in natural language. For example, suppose that we know that

“Every computer connected to the university network is functioning properly.”

Page 37

No rules of propositional logic allow us to conclude the truth of the statement

“MATH3 is functioning properly,”

where MATH3 is one of the computers connected to the university network. Likewise, we cannot use the rules of propositional logic to conclude from the statement

“CS2 is under attack by an intruder,”

where CS2 is a computer on the university network, to conclude the truth of

“There is a computer on the university network that is under attack by an intruder.”

In this section we will introduce a more powerful type of logic called **predicate logic**. We will see how predicate logic can be used to express the meaning of a wide range of statements in mathematics and computer science in ways that permit us to reason and explore relationships between objects. To understand predicate logic, we first need to introduce the concept of a predicate. Afterward, we will introduce the notion of quantifiers, which enable us to reason with statements that assert that a certain property holds for all objects of a certain type and with statements that assert the existence of an object with a particular property.

Predicates

Statements involving variables, such as

$x > 3,$ $x = y + 3,$ $x + y = z,$

and

“computer x is under attack by an intruder,”

and

“computer x is functioning properly,”

are often found in mathematical assertions, in computer programs, and in system specifications. These statements are neither true nor false when the values of the variables are not specified. In this section, we will discuss the ways that propositions can be produced from such statements.

The statement “ x is greater than 3” has two parts. The first part, the variable x , is the subject of the

statement. The second part—the **predicate**, “is greater than 3”—refers to a property that the subject of the statement can have. We can denote the statement “ x is greater than 3” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the **propositional function** P at x . Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value. Consider Examples 1 and 2.

EXAMPLE 1

Let $P(x)$ denote the statement “ $x > 3$.” What are the truth values of $P(4)$ and $P(2)$?

Solution:

We obtain the statement $P(4)$ by setting $x = 4$ in the statement “ $x > 3$.” Hence, $P(4)$, which is the statement “ $4 > 3$,” is true. However, $P(2)$, which is the statement “ $2 > 3$,” is false.

Page 38

EXAMPLE 2

Let $A(x)$ denote the statement “Computer x is under attack by an intruder.” Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are truth values of $A(\text{CS1})$, $A(\text{CS2})$, and $A(\text{MATH1})$?

Solution:

We obtain the statement $A(\text{CS1})$ by setting $x = \text{CS1}$ in the statement “Computer x is under attack by an intruder.” Because CS1 is not on the list of computers currently under attack, we conclude that $A(\text{CS1})$ is false. Similarly, because CS2 and MATH1 are on the list of computers under attack, we know that $A(\text{CS2})$ and $A(\text{MATH1})$ are true.

We can also have statements that involve more than one variable. For instance, consider the statement “ $x = y + 3$.” We can denote this statement by $Q(x, y)$, where x and y are variables and Q is the predicate. When values are assigned to the variables x and y , the statement $Q(x, y)$ has a truth value.

EXAMPLE 3

Let $Q(x, y)$ denote the statement “ $x = y + 3$.” What are the truth values of the propositions $Q(1, 2)$ and $Q(3, 0)$?



Solution:

To obtain $Q(1, 2)$, set $x = 1$ and $y = 2$ in the statement $Q(x, y)$. Hence, $Q(1, 2)$ is the statement “ $1 = 2 + 3$,” which is false. The statement $Q(3, 0)$ is the proposition “ $3 = 0 + 3$,” which is true.



CHARLES SANDERS PEIRCE (1839–1914)

Many consider Charles Peirce, born in Cambridge, Massachusetts, to be the most original and versatile American intellect. He made important contributions to an amazing number of disciplines, including mathematics, astronomy, chemistry, geodesy, metrology, engineering, psychology, philology, the history of science, and economics. Peirce was also an inventor, a lifelong student of medicine, a book reviewer, a dramatist and an actor, a short story writer, a phenomenologist, a logician, and a metaphysician. He is noted as the preeminent system-building philosopher competent and productive in logic, mathematics, and a wide range of sciences. He was encouraged by his father, Benjamin Peirce, a professor of mathematics and natural philosophy at Harvard, to pursue a career in science. Instead, he decided to study logic and scientific methodology. Peirce attended Harvard (1855–1859) and received a Harvard master of arts degree (1862) and an advanced degree in chemistry from the Lawrence Scientific School (1863).

In 1861, Peirce became an aide in the U.S. Coast Survey, with the goal of better understanding scientific methodology. His service for the Survey exempted him from military service during the Civil War. While working for the Survey, Peirce did astronomical and geodesic work. He made fundamental contributions to the design of pendulums and to map projections, applying new mathematical developments in the theory of elliptic functions. He was the first person to use the wavelength of light as a unit of measurement. Peirce rose to the position of Assistant for the Survey, a position he held until forced to resign in 1891 when he disagreed with the direction taken by the Survey's new administration.

While making his living from work in the physical sciences, Peirce developed a hierarchy of sciences, with mathematics at the top rung, in which the methods of one science could be adapted for use by those sciences under it in the hierarchy. During this time, he also founded the American philosophical theory of pragmatism.

The only academic position Peirce ever held was lecturer in logic at Johns Hopkins University in Baltimore (1879–1884). His mathematical work during this time included contributions to logic, set theory, abstract algebra, and the philosophy of mathematics. His work is still relevant today, with recent applications of this work on logic to artificial intelligence. Peirce believed that the study of mathematics could develop the mind's powers of imagination, abstraction, and generalization. His diverse activities after retiring from the Survey included writing for periodicals, contributing to scholarly dictionaries, translating scientific papers, guest lecturing, and textbook writing. Unfortunately, his income from these pursuits was insufficient to protect him and his second wife from abject poverty. He was supported in his later years by a fund created by his many admirers and administered by the philosopher William James, his lifelong friend. Although Peirce wrote and published voluminously in a vast range of subjects, he left more than 100,000 pages of unpublished manuscripts. Because of the difficulty of studying his unpublished writings, scholars have only recently started to understand some of his varied contributions. A group of people is devoted to making his work available over the Internet to bring a better appreciation of Peirce's accomplishments to the world.

Page 39

EXAMPLE 4

Let $A(c, n)$ denote the statement “Computer c is connected to network n ,” where c is a variable representing a computer and n is a variable representing a network. Suppose that the computer MATH1 is connected to network CAMPUS2, but not to network CAMPUS1. What are the values of $A(\text{MATH1}, \text{CAMPUS1})$ and $A(\text{MATH1}, \text{CAMPUS2})$?

Solution:

Because MATH1 is not connected to the CAMPUS1 network, we see that $A(\text{MATH1}, \text{CAMPUS1})$ is

false. However, because MATH1 is connected to the CAMPUS2 network, we see that $A(\text{MATH1}, \text{CAMPUS2})$ is true.

Similarly, we can let $R(x, y, z)$ denote the statement “ $x + y = z$.” When values are assigned to the variables x , y , and z , this statement has a truth value.

EXAMPLE 5

What are the truth values of the propositions $R(1, 2, 3)$ and $R(0, 0, 1)$?

Solution:

The proposition $R(1, 2, 3)$ is obtained by setting $x = 1$, $y = 2$, and $z = 3$ in the statement $R(x, y, z)$. We see that $R(1, 2, 3)$ is the statement “ $1 + 2 = 3$,” which is true. Also note that $R(0, 0, 1)$, which is the statement “ $0 + 0 = 1$,” is false.

In general, a statement involving the n variables x_1, x_2, \dots, x_n can be denoted by

$$P(x_1, x_2, \dots, x_n).$$

A statement of the form $P(x_1, x_2, \dots, x_n)$ is the value of the propositional function P at the n -tuple (x_1, x_2, \dots, x_n) , and P is also called an **n -place predicate** or a **n -ary predicate**.

Propositional functions occur in computer programs, as Example 6 demonstrates.

EXAMPLE 6

Consider the statement

if $x > 0$ then $x := x + 1$.

When this statement is encountered in a program, the value of the variable x at that point in the execution of the program is inserted into $P(x)$, which is “ $x > 0$.” If $P(x)$ is true for this value of x , the assignment statement $x := x + 1$ is executed, so the value of x is increased by 1. If $P(x)$ is false for this value of x , the assignment statement is not executed, so the value of x is not changed.

PRECONDITIONS AND POSTCONDITIONS

PRECONDITIONS AND POSTCONDITIONS Predicates are also used to establish the correctness of computer programs, that is, to show that computer programs always produce the desired output when given valid input. (Note that unless the correctness of a computer program is established, no amount of testing can show that it produces the desired output for all input values, unless every input value is tested.) The statements that describe valid input are known as **preconditions** and the conditions that the output should satisfy when the program has run are known as **postconditions**. As Example 7 illustrates, we use predicates to describe both preconditions and postconditions. We will study this process in greater detail in Section 5.5.

EXAMPLE 7

Consider the following program, designed to interchange the values of two variables x and y .

```
temp := x
x := y
y := temp
```

```
temp := x x := y y := temp
```

Find predicates that we can use as the precondition and the postcondition to verify the correctness of this program. Then explain how to use them to verify that for all valid input the program does what is intended.

Page 40

Solution:

For the precondition, we need to express that x and y have particular values before we run the program. So, for this precondition we can use the predicate $P(x, y)$, where $P(x, y)$ is the statement “ $x = a$ and $y = b$,” where a and b are the values of x and y before we run the program. Because we want to verify that the program swaps the values of x and y for all input values, for the postcondition we can use $Q(x, y)$, where $Q(x, y)$ is the statement “ $x = b$ and $y = a$.”

To verify that the program always does what it is supposed to do, suppose that the precondition $P(x, y)$ holds. That is, we suppose that the statement “ $x = a$ and $y = b$ ” is true. This means that $x = a$ and $y = b$. The first step of the program, $temp := x$, assigns the value of x to the variable $temp$, so after this step we know that $x = a$, $temp = a$, and $y = b$. After the second step of the program, $x := y$, we know that $x = b$, $temp = a$, and $y = b$. Finally, after the third step, we know that $x = b$, $temp = a$, and $y = a$. Consequently, after this program is run, the postcondition $Q(x, y)$ holds, that is, the statement “ $x = b$ and $y = a$ ” is true.



Quantifiers

When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there is another important way, called **quantification**, to create a proposition from a propositional function. Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words *all*, *some*, *many*, *none*, and *few* are used in quantifications. We will focus on two types of quantification here: universal quantification, which tells us that a predicate is true for every element under consideration, and existential quantification, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and quantifiers is called the **predicate calculus**.



THE UNIVERSAL QUANTIFIER

THE UNIVERSAL QUANTIFIER Many mathematical statements assert that a property is true for all values of a variable in a particular domain, called the domain of discourse (or the **universe of discourse**), often just referred to as the **domain**. Such a statement is expressed using universal quantification. The universal quantification of $P(x)$ for a particular domain is the proposition that asserts that $P(x)$ is true for all values of x in this domain. Note that the domain specifies the possible values of the variable x . The meaning of the universal quantification of $P(x)$ changes when we change the domain. The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined.

DEFINITION 1

The *universal quantification* of $P(x)$ is the statement

“ $P(x)$ for all values of x in the domain.”

The notation $\forall xP(x)$ denotes the universal quantification of $P(x)$. Here \forall is called the universal quantifier. We read $\forall xP(x)$ as “for all $xP(x)$ ” or “for every $xP(x)$.” An element for which $P(x)$ is false is called a **counterexample** of $\forall xP(x)$.

The meaning of the universal quantifier is summarized in the first row of Table 1. We illustrate the use of the universal quantifier in Examples 8–13.

TABLE 1 Quantifiers.		
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 .

Table 1
Quantifiers.

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 .

Page 41

EXAMPLE 8

Let $P(x)$ be the statement “ $x + 1 > x$.” What is the truth value of the quantification $\forall xP(x)$, where the domain consists of all real numbers?



Solution:

Because $P(x)$ is true for all real numbers x , the quantification

$\forall xP(x)$

is true.

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. Note that if the domain is empty, then $\forall xP(x)$ is true for any propositional function $P(x)$

because there are no elements x in the domain for which $P(x)$ is false.

Besides “for all” and “for every,” universal quantification can be expressed in many other ways, including “all of,” “for each,” “given any,” “for arbitrary,” “for each,” and “for any.”

marginal text

Remember that the truth value of $\forall xP(x)$ depends on the domain!

Remark: It is best to avoid using “for any x ” because it is often ambiguous as to whether “any” means “every” or “some.” In some cases, “any” is unambiguous, such as when it is used in negatives, for example, “there is not any reason to avoid studying.”

A statement $\forall xP(x)$ is false, where $P(x)$ is a propositional function, if and only if $P(x)$ is not always true when x is in the domain. One way to show that $P(x)$ is not always true when x is in the domain is to find a counterexample to the statement $\forall xP(x)$. Note that a single counterexample is all we need to establish that $\forall xP(x)$ is false. Example 9 illustrates how counterexamples are used.

EXAMPLE 9

Let $Q(x)$ be the statement “ $x < 2$.” What is the truth value of the quantification $\forall xQ(x)$, where the domain consists of all real numbers?

Solution:

$Q(x)$ is not true for every real number x , because, for instance, $Q(3)$ is false. That is, $x = 3$ is a counterexample for the statement $\forall xQ(x)$. Thus

$\forall xQ(x)$

is false.

EXAMPLE 10

Suppose that $P(x)$ is “ $x^2 > 0$.” To show that the statement $\forall xP(x)$ is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$.

Looking for counterexamples to universally quantified statements is an important activity in the study of mathematics, as we will see in subsequent sections of this book.

When all the elements in the domain can be listed—say, x_1, x_2, \dots, x_n —it follows that the universal quantification $\forall xP(x)$ is the same as the conjunction

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \dots, P(x_n)$ are all true.

Page 42

EXAMPLE 11

What is the truth value of $\forall xP(x)$, where $P(x)$ is the statement “ $x^2 < 10$ ” and the domain consists of the positive integers not exceeding 4?

Solution:

The statement $\forall xP(x)$ is the same as the conjunction

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4),$$

because the domain consists of the integers 1, 2, 3, and 4. Because $P(4)$, which is the statement “ $4^2 < 10$,” is false, it follows that $\forall xP(x)$ is false.

EXAMPLE 12

What does the statement $\forall xN(x)$ mean if $N(x)$ is “Computer x is connected to the network” and the domain consists of all computers on campus?

Solution:

The statement $\forall xN(x)$ means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as “Every computer on campus is connected to the network.”

As we have pointed out, specifying the domain is mandatory when quantifiers are used. The truth value of a quantified statement often depends on which elements are in this domain, as Example 13 shows.

EXAMPLE 13

What is the truth value of $\forall x(x^2 \geq x)$ if the domain consists of all real numbers? What is the truth value of this statement if the domain consists of all integers?

Solution:

The universal quantification $\forall x(x^2 \geq x)$, where the domain consists of all real numbers, is false. For example, $(\frac{1}{2})^2 \not\geq \frac{1}{2}$. Note that $x^2 \geq x$ if and only if $x^2 - x = x(x - 1) \geq 0$. Consequently, $x^2 \geq x$ if and only if $x \leq 0$ or $x \geq 1$. It follows that $\forall x(x^2 \geq x)$ is false if the domain consists of all real numbers (because the inequality is false for all real numbers x with $0 < x < 1$). However, if the domain consists of the integers, $\forall x(x^2 \geq x)$ is true, because there are no integers x with $0 < x < 1$.

THE EXISTENTIAL QUANTIFIER

THE EXISTENTIAL QUANTIFIER Many mathematical statements assert that there is an element with a certain property. Such statements are expressed using existential quantification. With existential quantification, we form a proposition that is true if and only if $P(x)$ is true for at least one value of x in the domain.

DEFINITION 2

The *existential quantification* of $P(x)$ is the proposition

“There exists an element x in the domain such that $P(x)$.”

We use the notation $\exists xP(x)$ for the existential quantification of $P(x)$. Here \exists is called the *existential quantifier*.

A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning.

Besides the phrase “there exists,” we can also express existential quantification in many other ways, such as by using the words “for some,” “for at least one,” or “there is.” The existential quantification $\exists xP(x)$ is read as

“There is an x such that $P(x)$,”

“There is at least one x such that $P(x)$,”

or

“For some $xP(x)$.”

Page 43

The meaning of the existential quantifier is summarized in the second row of Table 1. We illustrate the use of the existential quantifier in Examples 14–16.

EXAMPLE 14

Let $P(x)$ denote the statement “ $x > 3$.” What is the truth value of the quantification $\exists xP(x)$, where the domain consists of all real numbers?



Solution:

Because “ $x > 3$ ” is sometimes true—for instance, when $x = 4$ —the existential quantification of $P(x)$, which is $\exists xP(x)$, is true.

Observe that the statement $\exists xP(x)$ is false if and only if there is no element x in the domain for which $P(x)$ is true. That is, $\exists xP(x)$ is false if and only if $P(x)$ is false for every element of the domain. We illustrate this observation in Example 15.

EXAMPLE 15

Let $Q(x)$ denote the statement “ $x = x + 1$.” What is the truth value of the quantification $\exists xQ(x)$, where the domain consists of all real numbers?

Solution:

Because $Q(x)$ is false for every real number x , the existential quantification of $Q(x)$, which is $\exists xQ(x)$, is false.

marginal text

Remember that the truth value of $\exists xP(x)$ depends on the domain!

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. If the domain is empty, then $\exists xQ(x)$ is false whenever $Q(x)$ is a propositional function because when the domain is empty, there can be no element x in the domain for which $Q(x)$ is true.

When all elements in the domain can be listed—say, x_1, x_2, \dots, x_n —the existential quantification $\exists xP(x)$ is the same as the disjunction

$$P(x_1) \vee P(x_2) \vee \dots \vee P(x_n),$$

because this disjunction is true if and only if at least one of $P(x_1), P(x_2), \dots, P(x_n)$ is true.

EXAMPLE 16

What is the truth value of $\exists xP(x)$, where $P(x)$ is the statement “ $x^2 > 10$ ” and the universe of discourse consists of the positive integers not exceeding 4?

Solution:

Because the domain is $\{1, 2, 3, 4\}$, the proposition $\exists xP(x)$ is the same as the disjunction

$$P(1) \vee P(2) \vee P(3) \vee P(4).$$

Because $P(4)$, which is the statement “ $4^2 > 10$,” is true, it follows that $\exists xP(x)$ is true.

It is sometimes helpful to think in terms of looping and searching when 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 to see whether $P(x)$ is always true. If we encounter a value x for which $P(x)$ is false, then we have shown that $\forall xP(x)$ is false. Otherwise, $\forall xP(x)$ is true. To see whether $\exists xP(x)$ is true, we loop through the n values of x searching for a value for which $P(x)$ is true. If we find one, then $\exists xP(x)$ is true. If we never find such is true, we loop through the n values of x searching for a value for which $P(x)$ is true. If we find one, then $\exists xP(x)$ is true. If we never find such an x , then we have determined that $\exists xP(x)$ is false. (Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)

THE UNIQUENESS QUANTIFIER We have now introduced universal and existential quantifiers. These are the most important quantifiers in mathematics and computer science. However, there is no limitation on the number of different quantifiers we can define, such as “there are exactly two,” “there are no more than three,” “there are at least 100,” and so on. Of these other quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$ or $\exists 1$. The notation $\exists!xP(x)$ [or $\exists 1xP(x)$] states “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.”) For instance, $\exists!x(x - 1 = 0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x - 1 = 0$. This is a true statement, as $x = 1$ is the unique real number such that $x - 1 = 0$. Observe that we can use quantifiers and propositional logic to express uniqueness (see Exercise 52 in Section 1.5), so the uniqueness quantifier can be avoided. Generally, it is best to stick with existential and universal quantifiers so that rules of inference for these quantifiers can be used.

Quantifiers with Restricted Domains

An abbreviated notation is often used to restrict the domain of a quantifier. In this notation, a condition a variable must satisfy is included after the quantifier. This is illustrated in Example 17. We will also describe other forms of this notation involving set membership in Section 2.1.

EXAMPLE 17

What do the statements $\forall x < 0 (x^2 > 0)$, $\forall y \neq 0 (y^3 \neq 0)$, and $\exists z > 0 (z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

Solution:

The statement $\forall x < 0 (x^2 > 0)$ states that for every real number x with $x < 0$, $x^2 > 0$. That is, it states “The square of a negative real number is positive.” This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y \neq 0 (y^3 \neq 0)$ states that for every real number y with $y \neq 0$, we have $y^3 \neq 0$. That is, it states “The cube of every nonzero real number is nonzero.” Note that this statement is equivalent to $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.

Finally, the statement $\exists z > 0 (z^2 = 2)$ states that there exists a real number z with $z > 0$ such that $z^2 = 2$. That is, it states “There is a positive square root of 2.” This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$.

Note that the restriction of a universal quantification is the same as the universal quantification of a conditional statement. For instance, $\forall x < 0 (x^2 > 0)$ is another way of expressing $\forall x(x < 0 \rightarrow x^2 > 0)$. On the other hand, the restriction of an existential quantification is the same as the existential quantification of a conjunction. For instance, $\exists z > 0 (z^2 = 2)$ is another way of expressing $\exists z(z > 0 \wedge z^2 = 2)$.

Precedence of Quantifiers

The quantifiers \forall and \exists have higher precedence than all logical operators from propositional calculus. For example, $\forall xP(x) \vee Q(x)$ is the disjunction of $\forall xP(x)$ and $Q(x)$. In other words, it means $(\forall xP(x)) \vee Q(x)$.

$Q(x)$ rather than $\forall x(P(x) \vee Q(x))$.

Binding Variables

When a quantifier is used on the variable x , we say that this occurrence of the variable is bound. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be free. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition. This can be done using a combination of universal quantifiers, existential quantifiers, and value assignments.

Page 45

The part of a logical expression to which a quantifier is applied is called the scope of this quantifier. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable.

EXAMPLE 18

In the statement $\exists x(x + y = 1)$, the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it is not bound by a quantifier and no value is assigned to this variable. This illustrates that in the statement $\exists x(x + y = 1)$, x is bound, but y is free.

In the statement $\exists x(P(x) \wedge Q(x)) \vee \forall xR(x)$, all variables are bound. The scope of the first quantifier, $\exists x$, is the expression $P(x) \wedge Q(x)$ because $\exists x$ is applied only to $P(x) \wedge Q(x)$, and not to the rest of the statement. Similarly, the scope of the second quantifier, $\forall x$, is the expression $R(x)$. That is, the existential quantifier binds the variable x in $P(x) \wedge Q(x)$ and the universal quantifier $\forall x$ binds the variable x in $R(x)$. Observe that we could have written our statement using two different variables x and y , as $\exists x(P(x) \wedge Q(x)) \vee \forall yR(y)$, because the scopes of the two quantifiers do not overlap. The reader should be aware that in common usage, the same letter is often used to represent variables bound by different quantifiers with scopes that do not overlap.

Logical Equivalences Involving Quantifiers

In Section 1.3 we introduced the notion of logical equivalences of compound propositions. We can extend this notion to expressions involving predicates and quantifiers.

DEFINITION 3

Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain of discourse is used for the variables in these propositional functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.

Example 19 illustrates how to show that two statements involving predicates and quantifiers are logically equivalent.

EXAMPLE 19

Show that $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent (where the same domain is

used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a disjunction, nor can we distribute an existential quantifier over a conjunction. (See Exercises 50 and 51.)

Solution:

To show that these statements are logically equivalent, we must show that they always take the same truth value, no matter what the predicates P and Q are, and no matter which domain of discourse is used. Suppose we have particular predicates P and Q , with a common domain. We can show that $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent by doing two things. First, we show that if $\forall x(P(x) \wedge Q(x))$ is true, then $\forall xP(x) \wedge \forall xQ(x)$ is true. Second, we show that if $\forall xP(x) \wedge \forall xQ(x)$ is true, then $\forall x(P(x) \wedge Q(x))$ is true.

So, suppose that $\forall x(P(x) \wedge Q(x))$ is true. This means that if a is in the domain, then $P(a) \wedge Q(a)$ is true. Hence, $P(a)$ is true and $Q(a)$ is true. Because $P(a)$ is true and $Q(a)$ is true for every element in the domain, we can conclude that $\forall xP(x)$ and $\forall xQ(x)$ are both true. This means that $\forall xP(x) \wedge \forall xQ(x)$ is true.

Next, suppose that $\forall xP(x) \wedge \forall xQ(x)$ is true. It follows that $\forall xP(x)$ is true and $\forall xQ(x)$ is true. Hence, if a is in the domain, then $P(a)$ is true and $Q(a)$ is true [because $P(x)$ and $Q(x)$ are both true for all elements in the domain, there is no conflict using the same value of a here]. It follows that for all a , $P(a) \wedge Q(a)$ is true. It follows that $\forall x(P(x) \wedge Q(x))$ is true. We can now conclude that

$$\forall x(P(x) \wedge Q(x)) \equiv \forall xP(x) \wedge \forall xQ(x).$$

Page 46

Negating Quantified Expressions

We will often want to consider the negation of a quantified expression. For instance, consider the negation of the statement

“Every student in your class has taken a course in calculus.”

This statement is a universal quantification, namely,

$$\forall xP(x),$$



where $P(x)$ is the statement “ x has taken a course in calculus” and the domain consists of the students in your class. The negation of this statement is “It is not the case that every student in your class has taken a course in calculus.” This is equivalent to “There is a student in your class who has not taken a course in calculus.” And this is simply the existential quantification of the negation of the original propositional function, namely,

$$\exists x \neg P(x).$$

This example illustrates the following logical equivalence:

$$\neg \forall x P(x) \equiv \exists x \neg P(x).$$

To show that $\neg \forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent no matter what the propositional function $P(x)$ is and what the domain is, first note that $\neg \forall x P(x)$ is true if and only if $\forall x P(x)$ is false. Next, note that $\forall x P(x)$ is false if and only if there is an element x in the domain for which $P(x)$ is false. This holds if and only if there is an element x in the domain for which $\neg P(x)$ is true. Finally, note that there is an element x in the domain for which $\neg P(x)$ is true if and only if $\exists x \neg P(x)$ is true. Putting these steps together, we can conclude that $\neg \forall x P(x)$ is true if and only if $\exists x \neg P(x)$ is true. It follows that $\neg \forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent.

Suppose we wish to negate an existential quantification. For instance, consider the proposition “There is a student in this class who has taken a course in calculus.” This is the existential quantification

$$\exists x Q(x),$$

where $Q(x)$ is the statement “ x has taken a course in calculus.” The negation of this statement is the proposition “It is not the case that there is a student in this class who has taken a course in calculus.” This is equivalent to “Every student in this class has not taken calculus,” which is just the universal quantification of the negation of the original propositional function, or, phrased in the language of quantifiers,

$$\forall x \neg Q(x).$$

This example illustrates the equivalence

$$\neg \exists x Q(x) \equiv \forall x \neg Q(x).$$

To show that $\neg \exists x Q(x)$ and $\forall x \neg Q(x)$ are logically equivalent no matter what $Q(x)$ is and what the domain is, first note that $\neg \exists x Q(x)$ is true if and only if $\exists x Q(x)$ is false. This is true if and only if no x exists in the domain for which $Q(x)$ is true. Next, note that no x exists in the domain for which $Q(x)$ is true if and only if $Q(x)$ is false for every x in the domain. Finally, note that $Q(x)$ is false for every x in the domain if and only if $\neg Q(x)$ is true for all x in the domain, which holds if and only if $\forall x \neg Q(x)$ is true. Putting these steps together, we see that $\neg \exists x Q(x)$ is true if and only if $\forall x \neg Q(x)$ is true. We conclude that $\neg \exists x Q(x)$ and $\forall x \neg Q(x)$ are logically equivalent.

Page 47

The rules for negations for quantifiers are called **De Morgan's laws for quantifiers**. These rules are summarized in Table 2.

TABLE 2 De Morgan's Laws for Quantifiers.			
<i>Negation</i>	<i>Equivalent Statement</i>	<i>When Is Negation True?</i>	<i>When False?</i>
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x , $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x .

Table 2
De Morgan's Laws for Quantifiers.

Negation *Equivalent Statement* *When Is Negation True?* *When False?*

$\neg\exists x P(x)$ $\forall x \neg P(x)$ For every x , $P(x)$ is false. There is an x for which $P(x)$ is true.

$\neg\forall x P(x)$ $\exists x \neg P(x)$ There is an x for which $P(x)$ is false. $P(x)$ is true for every x .

Remark: When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements x_1, x_2, \dots, x_n , it follows that $\neg\forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$. Similarly, $\neg\exists x P(x)$ is the same as $\neg(P(x_1) \vee P(x_2) \vee \dots \vee P(x_n))$, which by De Morgan's laws is equivalent to $\neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n)$, and this is the same as $\forall x \neg P(x)$.

We illustrate the negation of quantified statements in Examples 20 and 21.

EXAMPLE 20

What are the negations of the statements “There is an honest politician” and “All Americans eat cheeseburgers”?

Solution:

Let $H(x)$ denote “ x is honest.” Then the statement “There is an honest politician” is represented by $\exists x H(x)$, where the domain consists of all politicians. The negation of this statement is $\neg\exists x H(x)$, which is equivalent to $\forall x \neg H(x)$. This negation can be expressed as “Every politician is dishonest.” (*Note:* In English, the statement “All politicians are not honest” is ambiguous. In common usage, this statement often means “Not all politicians are honest.” Consequently, we do not use this statement to express this negation.)



Let $C(x)$ denote “ x eats cheeseburgers.” Then the statement “All Americans eat cheeseburgers” is represented by $\forall x C(x)$, where the domain consists of all Americans. The negation of this statement is

$\neg\forall xC(x)$, which is equivalent to $\exists x\neg C(x)$. This negation can be expressed in several different ways, including “Some American does not eat cheeseburgers” and “There is an American who does not eat cheeseburgers.”

EXAMPLE 21

What are the negations of the statements $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$?

Solution:

The negation of $\forall x(x^2 > x)$ is the statement $\neg\forall x(x^2 > x)$, which is equivalent to $\exists x\neg(x^2 > x)$. This can be rewritten as $\exists x(x^2 \leq x)$. The negation of $\exists x(x^2 = 2)$ is the statement $\neg\exists x(x^2 = 2)$, which is equivalent to $\forall x\neg(x^2 = 2)$. This can be rewritten as $\forall x(x^2 \neq 2)$. The truth values of these statements depend on the domain.

We use De Morgan's laws for quantifiers in Example 22.

Page 48

EXAMPLE 22

Show that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution:

By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(\neg(P(x) \rightarrow Q(x)))$ are logically equivalent. By the fifth logical equivalence in Table 7 in Section 1.3, we know that $\neg(P(x) \rightarrow Q(x))$ and $P(x) \wedge \neg Q(x)$ are logically equivalent for every x . Because we can substitute one logically equivalent expression for another in a logical equivalence, it follows that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Translating from English into Logical Expressions

Translating sentences in English (or other natural languages) into logical expressions is a crucial task in mathematics, logic programming, artificial intelligence, software engineering, and many other disciplines. We began studying this topic in Section 1.1, where we used propositions to express sentences in logical expressions. In that discussion, we purposely avoided sentences whose translations required predicates and quantifiers. Translating from English to logical expressions becomes even more complex when quantifiers are needed. Furthermore, there can be many ways to translate a particular sentence. (As a consequence, there is no “cookbook” approach that can be followed step by step.) We will use some examples to illustrate how to translate sentences from English into logical expressions. The goal in this translation is to produce simple and useful logical expressions. In this section, we restrict ourselves to sentences that can be translated into logical expressions using a single quantifier; in the next section, we will look at more complicated sentences that require multiple quantifiers.

EXAMPLE 23

Express the statement “Every student in this class has studied calculus” using predicates and quantifiers.

Solution:

First, we rewrite the statement so that we can clearly identify the appropriate quantifiers to use. Doing so, we obtain:

“For every student in this class, that student has studied calculus.”



Next, we introduce a variable x so that our statement becomes

“For every student x in this class, x has studied calculus.”

Continuing, we introduce $C(x)$, which is the statement “ x has studied calculus.” Consequently, if the domain for x consists of the students in the class, we can translate our statement as $\forall x C(x)$.

However, there are other correct approaches; different domains of discourse and other predicates can be used. The approach we select depends on the subsequent reasoning we want to carry out. For example, we may be interested in a wider group of people than only those in this class. If we change the domain to consist of all people, we will need to express our statement as

“For every person x , if person x is a student in this class then x has studied calculus.”



If $S(x)$ represents the statement that person x is in this class, we see that our statement can be expressed as $\forall x(S(x) \rightarrow C(x))$. [Caution! Our statement *cannot* be expressed as $\forall x(S(x) \wedge C(x))$ because this statement says that all people are students in this class and have studied calculus!]

Finally, when we are interested in the background of people in subjects besides calculus, we may prefer to use the two-variable quantifier $Q(x, y)$ for the statement “student x has studied subject y .” Then we would replace $C(x)$ by $Q(x, \text{calculus})$ in both approaches to obtain $\forall x Q(x, \text{calculus})$ or $\forall x(S(x) \rightarrow Q(x, \text{calculus}))$.

Page 49

In Example 23 we displayed different approaches for expressing the same statement using predicates and quantifiers. However, we should always adopt the simplest approach that is adequate for use in subsequent reasoning.

EXAMPLE 24

Express the statements “Some student in this class has visited Mexico” and “Every student in this class has visited either Canada or Mexico” using predicates and quantifiers.

Solution:

The statement “Some student in this class has visited Mexico” means that

“There is a student in this class with the property that the student has visited Mexico.”

We can introduce a variable x , so that our statement becomes

“There is a student x in this class having the property that x has visited Mexico.”

We introduce $M(x)$, which is the statement “ x has visited Mexico.” If the domain for x consists of the students in this class, we can translate this first statement as $\exists x M(x)$.

However, if we are interested in people other than those in this class, we look at the statement a little differently. Our statement can be expressed as

“There is a person x having the properties that x is a student in this class and x has visited Mexico.”



In this case, the domain for the variable x consists of all people. We introduce $S(x)$ to represent “ x is a student in this class.” Our solution becomes $\exists x(S(x) \wedge M(x))$ because the statement is that there is a person x who is a student in this class and who has visited Mexico. [*Caution!* Our statement cannot be expressed as $\exists x(S(x) \rightarrow M(x))$, which is true when there is someone not in the class because, in that case, for such a person x , $S(x) \rightarrow M(x)$ becomes either $\mathbf{F} \rightarrow \mathbf{T}$ or $\mathbf{F} \rightarrow \mathbf{F}$, both of which are true.]

Similarly, the second statement can be expressed as

“For every x in this class, x has the property that x has visited Mexico or x has visited Canada.”

(Note that we are assuming the inclusive, rather than the exclusive, or here.) We let $C(x)$ be “ x has visited Canada.” Following our earlier reasoning, we see that if the domain for x consists of the students in this class, this second statement can be expressed as $\forall x(C(x) \vee M(x))$. However, if the domain for x consists of all people, our statement can be expressed as

“For every person x , if x is a student in this class, then x has visited Mexico or x has visited Canada.”

In this case, the statement can be expressed as $\forall x(S(x) \rightarrow (C(x) \vee M(x)))$.

Instead of using $M(x)$ and $C(x)$ to represent that x has visited Mexico and x has visited Canada, respectively, we could use a two-place predicate $V(x, y)$ to represent “ x has visited country y .” In this case, $V(x, \text{Mexico})$ and $V(x, \text{Canada})$ would have the same meaning as $M(x)$ and $C(x)$ and could replace them in our answers. If we are working with many statements that involve people visiting different countries, we might prefer to use this two-variable approach. Otherwise, for simplicity, we would stick with the one-variable predicates $M(x)$ and $C(x)$.

Page 50

Using Quantifiers in System Specifications

In Section 1.2 we used propositions to represent system specifications. However, many system specifications involve predicates and quantifications. This is illustrated in Example 25.

EXAMPLE 25

Use predicates and quantifiers to express the system specifications “Every mail message larger than one

megabyte will be compressed” and “If a user is active, at least one network link will be available.”



Solution:

Let $S(m, y)$ be “Mail message m is larger than y megabytes,” where the variable x has the domain of all mail messages and the variable y is a positive real number, and let $C(m)$ denote “Mail message m will be compressed.” Then the specification “Every mail message larger than one megabyte will be compressed” can be represented as $\forall m(S(m, 1) \rightarrow C(m))$.

Let $A(u)$ represent “User u is active,” where the variable u has the domain of all users, let $S(n, x)$ denote “Network link n is in state x ,” where n has the domain of all network links and x has the domain of all possible states for a network link. Then the specification “If a user is active, at least one network link will be available” can be represented by $\exists u A(u) \rightarrow \exists n S(n, \text{available})$.

marginal text

Remember the rules of precedence for quantifiers and logical connectives!

Examples from Lewis Carroll

Lewis Carroll (really C. L. Dodgson writing under a pseudonym), the author of *Alice in Wonderland*, is also the author of several works on symbolic logic. His books contain many examples of reasoning using quantifiers. Examples 26 and 27 come from his book *Symbolic Logic*; other examples from that book are given in the exercises at the end of this section. These examples illustrate how quantifiers are used to express various types of statements.

EXAMPLE 26

Consider these statements. The first two are called *premises* and the third is called the *conclusion*. The entire set is called an *argument*.

“All lions are fierce.”

“Some lions do not drink coffee.”

“Some fierce creatures do not drink coffee.”

(In Section 1.6 we will discuss the issue of determining whether the conclusion is a valid consequence of the premises. In this example, it is.) Let $P(x)$, $Q(x)$, and $R(x)$ be the statements “ x is a lion,” “ x is fierce,” and “ x drinks coffee,” respectively. Assuming that the domain consists of all creatures, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, and $R(x)$.

Solution:

We can express these statements as:

$$\forall x(P(x) \rightarrow Q(x)).$$

$$\exists x(P(x) \wedge \neg R(x)).$$

$$\exists x(Q(x) \wedge \neg R(x)).$$

Notice that the second statement cannot be written as $\exists x(P(x) \rightarrow \neg R(x))$. The reason is that $P(x) \rightarrow \neg R(x)$ is true whenever x is not a lion, so that $\exists x(P(x) \rightarrow \neg R(x))$ is true as long as there is at least one creature that is not a lion, even if every lion drinks coffee. Similarly, the third statement cannot be written as

$\exists x(Q(x) \rightarrow \neg R(x))$.



CHARLES LUTWIDGE DODGSON (1832–1898)

We know Charles Dodgson as Lewis Carroll—the pseudonym he used in his literary works. Dodgson, the son of a clergyman, was the third of 11 children, all of whom stuttered. He was uncomfortable in the company of adults and is said to have spoken without stuttering only to young girls, many of whom he entertained, corresponded with, and photographed (sometimes in poses that today would be considered inappropriate). Although attracted to young girls, he was extremely puritanical and religious. His friendship with the three young daughters of Dean Liddell led to his writing *Alice in Wonderland*, which brought him money and fame.

Dodgson graduated from Oxford in 1854 and obtained his master of arts degree in 1857. He was appointed lecturer in mathematics at Christ Church College, Oxford, in 1855. He was ordained in the Church of England in 1861 but never practiced his ministry. His writings published under this real name include articles and books on geometry, determinants, and the mathematics of tournaments and elections. (He also used the pseudonym Lewis Carroll for his many works on recreational logic.)

Page 51

EXAMPLE 27

Consider these statements, of which the first three are premises and the fourth is a valid conclusion.

- “All hummingbirds are richly colored.”
- “No large birds live on honey.”
- “Birds that do not live on honey are dull in color.”
- “Hummingbirds are small.”

Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a hummingbird,” “ x is large,” “ x lives on honey,” and “ x is richly colored,” respectively. Assuming that the domain consists of all birds, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

Solution:

We can express the statements in the argument as

$\forall x(P(x) \rightarrow S(x)).$

$\neg \exists x(Q(x) \wedge R(x)).$

$\forall x(\neg R(x) \rightarrow \neg S(x)).$

$\forall x(P(x) \rightarrow \neg Q(x)).$

(Note we have assumed that “small” is the same as “not large” and that “dull in color” is the same as “not richly colored.” To show that the fourth statement is a valid conclusion of the first three, we need to use rules of inference that will be discussed in Section 1.6.)

Logic Programming
comp:marginal text



An important type of programming language is designed to reason using the rules of predicate logic. Prolog (from *Programming in Logic*), developed in the 1970s by computer scientists working in the area of artificial intelligence, is an example of such a language. Prolog programs include a set of declarations consisting of two types of statements, **Prolog facts** and **Prolog rules**. Prolog facts define predicates by specifying the elements that satisfy these predicates. Prolog rules are used to define new predicates using those already defined by Prolog facts. Example 28 illustrates these notions.

EXAMPLE 28

Consider a Prolog program given facts telling it the instructor of each class and in which classes students are enrolled. The program uses these facts to answer queries concerning the professors who teach particular students. Such a program could use the predicates *instructor(p, c)* and *enrolled(s, c)* to represent that professor *p* is the instructor of course *c* and that student *s* is enrolled in course *c*, respectively. For example, the Prolog facts in such a program might include:

Page 52

```
instructor(chan,math273) instructor(patel,ee222) instructor(grossman,cs301) enrolled(kevin,math273)
enrolled(juana,ee222) enrolled(juana,cs301) enrolled(kiko,math273) enrolled(kiko,cs301)
```

(Lowercase letters have been used for entries because Prolog considers names beginning with an uppercase letter to be variables.)

A new predicate *teaches(p, s)*, representing that professor *p* teaches student *s*, can be defined using the Prolog rule

```
teaches(P,S) :- instructor(P,C), enrolled(S,C)
```

which means that *teaches(p, s)* is true if there exists a class *c* such that professor *p* is the instructor of class *c* and student *s* is enrolled in class *c*. (Note that a comma is used to represent a conjunction of predicates in Prolog. Similarly, a semicolon is used to represent a disjunction of predicates.)

Prolog answers queries using the facts and rules it is given. For example, using the facts and rules listed,

the query

?enrolled(kevin,math273)

produces the response

yes

because the fact *enrolled*(kevin, math273) was provided as input. The query

?enrolled(X,math273)

produces the response

kevin

kiko

To produce this response, Prolog determines all possible values of X for which *enrolled*(X , math273) has been included as a Prolog fact. Similarly, to find all the professors who are instructors in classes being taken by Juana, we use the query

?teaches(X,juana)

This query returns

patel

grossman