Logical Form and Logical Equivalence

The central concept of deductive logic is the concept of argument form. An **argument** is a sequence of statements aimed at demonstrating the truth of an assertion. The assertion at the end of the sequence is called the **conclusion**, and the preceding statements are called **premises**. To have confidence in the conclusion that you draw from an argument, you must be sure that the premises are acceptable on their own merits or follow from other statements that are known to be true.

Consider the following two arguments, for example. Although their content is very different, their logical form is the same.

> **Argument 1** If the program execution results in division by zero, then the computer will generate an error message.
>
> **Argument 2** If x is a real number such that $x < -2$ or $x > 2$, then $x^2 > 4$.

1. Identify the premise(s) and conclusion of each of the arguments above.

2. Fill in the blanks below so that argument (b) has the same form as argument (a). Then represent the common form of the arguments using letters to stand for component sentences.

a. If Jane is a math major or Jane is a computer science major, then Jane will take Math 150.
Jane is a computer science major. Therefore, Jane will take Math 150.

b. If logic is easy or __________ , then ______________ .
I will study hard.
Therefore, I will get an A in this course.

Representing Common Form

To illustrate the logical form of these arguments, you will use letters of the alphabet (such as *p*, *q*, and *r*) to represent the component sentences. Both arguments in question 2 are in the form:

If p or q, then r .
q.
Therefore, r .

3. Rewrite 2a so that it follows the form:

If p or q, then r .
p.
Therefore, r .

and the expression “not *p*” to refer to the sentence “It is not the case that *p*.” Then the *common logical form* of both the previous arguments is as follows:

If p or q, then r .
Therefore, if not r , then not p and not q.

Statements

Most of the definitions of formal logic have been developed so that they agree with the natural or intuitive logic used by people who have been educated to think clearly and use language carefully. The differences that exist between formal and intuitive logic are necessary to avoid ambiguity and obtain consistency.

In any mathematical theory, new terms are defined by using those that have been previously defined. However, this process has to start somewhere. A few initial terms necessarily remain undefined. In logic, the words sentence, true, and false are the initial undefined terms.

For example, “Two plus two equals four” and “Two plus two equals five” are both statements, the first because it is true and the second because it is false. On the other

hand, the truth or falsity of “He is a college student” depends on the reference for the pronoun he. For some values of he the sentence is true; for others it is false. If the sen- tence were preceded by other sentences that made the pronoun’s reference clear, then the sentence would be a statement. Considered on its own, however, the sentence is neither true nor false, and so it is not a statement. We will discuss ways of transforming sentences of this form into statements in Section 3.1.

Similarly, “$x + y > 0$” is not a statement because for some values of x and y the sentence is true, whereas for others it is false. For instance, if $x = 1$ and $y = 2$, the sentence is true; if $x = -1$ and $y = 0$, the sentence is false.

Compound Statements

We now introduce three symbols that are used to build more complicated logical expressions out of simpler ones. The symbol □ denotes *not,* ∧ denotes *and,* and □ denotes *or.* Given a statement *p*, the sentence "□*p*" is read "not *p*" or "It is not the case that *p*" and is called the **negation of *p***. In some computer languages the symbol ' is used in place of □. Given another statement *q*, the sentence "*p* □ *q*" is read "*p* and *q*" and is called the **conjunction of *p* and *q***. The sentence "*p* □ *q*" is read "*p* or *q*" and is called the **disjunction of *p* and *q***.

In expressions that include the symbol □ as well as □ or □, the **order of operations** specifies that ~ is performed first. For instance, *~p* □ *q* = (□*p*) □ *q*. In logical expres- sions, as in ordinary algebraic expressions, the order of operations can be overridden through the use of parentheses. Thus □(*p* □ *q*) represents the negation of the conjunction of *p* and *q*. In this, as in most treatments of logic, the symbols □ and □ are considered coequal in order of operation, and an expression such as *p* □ *q* □ *r* is considered ambigu- ous. This expression must be written as either (*p* □ *q*) □ *r* or *p* □ (*q* □ *r*) to have meaning. A variety of English words translate into logic as □, □, or □. For instance, the word *but* translates the same as *and* when it links two independent clauses, as in "Jim is tall but he is not heavy." Generally, the word *but* is used in place of *and* when the part of the sentence that follows is, in some way, unexpected. Another example involves the words *neither-nor*. When Shakespeare wrote, "Neither a borrower nor a lender be," he meant, "Do not be a borrower and do not be a lender." So if *p* and *q* are statements, then

Table

Example 2.1.2 Translating from English to Symbols: *But* and *Neither-Nor*

Write each of the following sentences symbolically, letting *h* = "It is hot" and *s* = "It is sunny."

a. It is not hot but it is sunny.

b. It is neither hot nor sunny.

Solution

a. The given sentence is equivalent to "It is not hot and it is sunny," which can be written symbolically as □*h* □ *s*.

To say it is neither hot nor sunny means that it is not hot and it is not sunny. Therefore, the given sentence can be written symbolically as □*h* □ □*s*.

The notation for inequalities involves *and* and *or* statements. For instance, if *x*, *a*, and *b* are particular real numbers, then

Note that the inequality $2 \leqslant x \leq 1$ is not satisfied by any real numbers because $2 \leq x \leq 1$ means $2 \leq x$ and $x \leq 1$,

and this is false no matter what number x happens to be. By the way, the point of specify- ing x, a, and b to be *particular* real numbers is to ensure that sentences such as "$x < a$" and "$x \geq b$" are either true or false and hence that they are statements.

Example 2.1.3 *And, Or,* and Inequalities

Suppose x is a particular real number. Let p, q, and r symbolize "$0 < x$," "$x < 3$," and "$x = 3$," respectively. Write the following inequalities symbolically:

a. $x \leq 3$ b. $0 < x < 3$ c. $0 < x \leq 3$

Solution

a. q □ r b. p □ q c. p □ $(q$ □ $r)$ ■

Truth Values

In Examples 2.1.2 and 2.1.3 we built compound sentences out of component statements and the terms *not, and,* and *or.* If such sentences are to be statements, however, they must have well-defined **truth values**—they must be either true or false. We now define such compound sentences as statements by specifying their truth values in terms of the statements that compose them.

The negation of a statement is a statement that exactly expresses what it would mean for the statement to be false.

If p is a statement variable, the **negation** of p is "not p" or "It is not the case that p" and is denoted □p. It has opposite truth value from p: if p is true, □p is false; if p is false, □p is true.

The truth values for negation are summarized in a *truth table.*

Truth Table for □ *p*

In ordinary language the sentence "It is hot and it is sunny" is understood to be true when both conditions—being hot and being sunny—are satisfied. If it is hot but not sunny, or sunny but not hot, or neither hot nor sunny, the sentence is understood to be false. The formal definition of truth values for an *and* statement agrees with this general understanding.

Def

If p and q are statement variables, the **conjunction** of p and q is " p and q," denoted $p \wedge q$. It is true when, and only when, both p and q are true. If either p or q is false, or if both are false, p □ q is false.

The truth values for conjunction can also be summarized in a truth table. The table is obtained by considering the four possible combinations of truth values for p and q. Each combination is displayed in one row of the table; the corresponding truth value for the whole statement is placed in the right-most column of that row. Note that the only row containing a T is the first one since the only way for an *and* statement to be true is for both component statements to be true.

Truth Table for $p \wedge q$

By the way, the order of truth values for p and q in the table above is TT, TF, FT, FF. It is not absolutely necessary to write the truth values in this order, although it is customary to do so. We will use this order for all truth tables involving two statement variables. In Example 2.1.5 we will show the standard order for truth tables that involve three statement variables.

In the case of disjunction—statements of the form " p or q"—intuitive logic offers two alternative interpretations. In ordinary language *or* is sometimes used in an exclusive sense (p or q but not both) and sometimes in an inclusive sense (p or q or both). A waiter who says you may have "coffee, tea, or milk" uses the word *or* in an exclusive sense: Extra payment is generally required if you want more than one beverage. On the other hand, a waiter who offers "cream or sugar" uses the word *or* in an inclusive sense: You are entitled to both cream and sugar if you wish to have them.

Mathematicians and logicians avoid possible ambiguity about the meaning of the word *or* by understanding it to mean the inclusive "and/or." The symbol $\vee$ comes from the Latin word *vel*, which means *or* in its inclusive sense. To express the exclusive *or*, the phrase *p or q but not both* is used.

Def

If p and q are statement variables, the **disjunction** of p and q is " p or q," denoted $p \vee q$. It is true when either p is true, or q is true, or both p and q are true; it is false only when both p and q are false.

Here is the truth table for disjunction:

Truth Table for $p \vee q$

Evaluating the Truth of More General Compound Statements

Now that truth values have been assigned to □*p*, *p* □ *q*, and *p* □ *q*, consider the question of assigning truth values to more complicated expressions such as □*p* □ *q*, *(p* □ *q)* □ □*(p* □ *q)*, and *(p* □ *q)* □ *r*. Such expressions are called *statement forms* (or *propositional forms*). The close relationship between statement forms and *Boolean expressions* is discussed in Section 2.4.

Def

A **statement form** (or **propositional form**) is an expression made up of statement variables (such as *p*, *q*, and *r*) and logical connectives (such as □, □, and □) that becomes a statement when actual statements are substituted for the component state- ment variables. The **truth table** for a given statement form displays the truth values that correspond to all possible combinations of truth values for its component state- ment variables.

To compute the truth values for a statement form, follow rules similar to those used to evaluate algebraic expressions. For each combination of truth values for the statement variables, first evaluate the expressions within the innermost parentheses, then evaluate the expressions within the next innermost set of parentheses, and so forth until you have the truth values for the complete expression.

Example 2.1.4 Truth Table for *Exclusive Or*

Construct the truth table for the statement form *(p* □ *q)* □ □*(p* □ *q)*. Note that when *or* is used in its exclusive sense, the statement " *p* or *q*" means " *p* or *q* but not both" or " *p* or *q* and not both *p* and *q*," which translates into symbols as *(p* □ *q)* □ □*(p* □ *q)*. This is sometimes abbreviated *p* □ *q* or *p* XOR *q*.

Solution Set up columns labeled *p*, *q*, *p* □ *q*, *p* □ *q*, □*(p* □ *q)*, and *(p* □ *q)* □ □*(p* □ *q)*. Fill in the *p* and *q* columns with all the logically possible combinations of T's and F's. Then use the truth tables for □ and □ to fill in the *p* □ *q* and *p* □ *q* columns with the appropriate truth values. Next fill in the ∼*(p* □ *q)* column by taking the opposites of the truth values for *p* □ *q*. For example, the entry for □*(p* □ *q)* in the first row is F because in the first row the truth value of *p* □ *q* is T.

Finally, fill in the $(p \vee q)$ □ □$(p$ □ $q)$ col- umn by considering the truth table for an *and* statement together with the computed truth values for p □ q and □$(p$ □ $q)$. For example, the entry in the first row is F because the entry for p □ q is T, the entry for □$(p$ □ $q)$ is F, and an *and* statement is false unless both components are true. The entry in the second row is T because both components are true in this row.

Truth Table for *Exclusive Or:* $(p$ □ $q)$ □ □$(p$ □ $q)$

Example 2.1.5 Truth Table for $(p \wedge q)$ □ □r

Construct a truth table for the statement form $(p \wedge q)$ □ □r.

Solution Make columns headed p, q, r, $p \wedge q$, □r, and $(p \wedge q)$ □ □r. Enter the eight logically possible combinations of truth values for p, q, and r in the three left-most columns. Then fill in the truth values for $p \wedge q$ and for □r. Complete the table by con- sidering the truth values for $(p \wedge q)$ and for □r and the definition of an *or* statement. Since an *or* statement is false only when both components are false, the only rows in which the entry is F are the third, fifth, and seventh rows because those are the only rows in which the expressions $p \wedge q$ and □r are both false. The entry for all the other rows is T.

The essential point about assigning truth values to compound statements is that it allows you—using logic alone—to judge the truth of a compound statement on the basis of your knowledge of the truth of its component parts. Logic does not help you determine the truth or falsity of the component statements. Rather, logic helps link these separate pieces of information together into a coherent whole.

Logical Equivalence

The statements

6 is greater than 2 and 2 is less than 6

are two different ways of saying the same thing. Why? Because of the definition of the phrases *greater than* and *less than.* By contrast, although the statements

(1) Dogs bark and cats meow and (2) Cats meow and dogs bark

are also two different ways of saying the same thing, the reason has nothing to do with the definition of the words. It has to do with the logical form of the statements. Any two statements whose logical forms are related in the same way as (1) and (2) would either both be true or both be false. You can see this by examining the following truth table, where the statement variables p and q are substituted for the component statements “Dogs bark” and “Cats meow,” respectively. The table shows that for each combination of truth values for p and q, $p \wedge q$ is true when, and only when, $q \wedge p$ is true. In such a case, the statement forms are called *logically equivalent*, and we say that (1) and (2) are *logically equivalent statements.*

Def

Two *statement forms* are called **logically equivalent** if, and only if, they have identical truth values for each possible substitution of statements for their statement variables. The logical equivalence of statement forms P and Q is denoted by writing $P \equiv Q$. Two *statements* are called **logically equivalent** if, and only if, they have logically equivalent forms when identical component statement variables are used to replace identical component statements.

Testing Whether Two Statement Forms *P* and *Q* Are Logically Equivalent

1. Construct a truth table with one column for the truth values of P and another column for the truth values of Q.
2. Check each combination of truth values of the statement variables to see whether the truth value of P is the same as the truth value of Q.
   a. If in each row the truth value of P is the same as the truth value of Q, then P and Q are logically equivalent.
   b. If in some row P has a different truth value from Q, then P and Q are not logically equivalent.

Example 2.1.6 Double Negative Property: $\sim(\sim p) \equiv p$

Construct a truth table to show that the negation of the negation of a statement is logically equivalent to the statement, annotating the table with a sentence of explanation.

Table

\

There are two ways to show that statement forms P and Q are *not* logically equiva- lent. As indicated previously, one is to use a truth table to find rows for which their truth values differ. The other way is to find concrete statements for each of the two forms, one of which is true and the other of which is false. The next example illustrates both of these ways.

Example 2.1.7 Showing Nonequivalence

Show that the statement forms $\sim(p \wedge q)$ and $\sim p \wedge \sim q$ are not logically equivalent.

Solution

a. This method uses a truth table annotated with a sentence of explanation.

Table

a. This method uses an example to show that $\sim(p \wedge q)$ and $\sim p \wedge \sim q$ are not logically equivalent. Let p be the statement "$0 < 1$" and let q be the statement "$1 < 0$." Then

$\sim(p \wedge q)$ is "It is not the case that both $0 < 1$ and $1 < 0$," which is true. On the other hand,

$\sim p \wedge \sim q$ is "$0 \not< 1$ and $1 \not< 0$,"

which is false. This example shows that there are concrete statements you can substi- tute for p and q to make one of the statement forms true and the other false. Therefore, the statement forms are not logically equivalent.

Example 2.1.8 Negations of *And* and *Or:* De Morgan's Laws

For the statement "John is tall and Jim is redheaded" to be true, both components must be true. So for the statement to be false, one or both components must be false. Thus the negation can be written as "John is not tall or Jim is not redheaded." In general, the negation of the conjunction of two statements is logically equivalent to the disjunction of their negations. That is, statements of the forms $\sim(p \wedge q)$ and $\sim p \vee \sim q$ are logically equivalent. Check this using truth tables.

Table

Symbolically, $\sim(p \wedge q) \equiv \sim p \vee \sim q$.

In the exercises at the end of this section you are asked to show the analogous law that the negation of the disjunction of two statements is logically equivalent to the conjunction of their negations:

$\sim(p \vee q) \equiv \sim p \wedge \sim q$.

The two logical equivalences of Example 2.1.8 are known as **De Morgan's laws** of logic in honor of Augustus De Morgan, who was the first to state them in formal mathematical terms.

De Morgan's Laws

The negation of an *and* statement is logically equivalent to the *or* statement in which each component is negated.

The negation of an *or* statement is logically equivalent to the *and* statement in which each component is negated.

Example 2.1.9 Applying De Morgan's Laws

Write negations for each of the following statements:

a. John is 6 feet tall and he weighs at least 200 pounds.

b. The bus was late or Tom's watch was slow.

Solution

a. John is not 6 feet tall or he weighs less than 200 pounds.

b. The bus was not late and Tom's watch was not slow.

Since the statement "neither p nor q" means the same as "$\sim p$ and $\sim q$," an alternative answer for (b) is "Neither was the bus late nor was Tom's watch slow." ■

If x is a particular real number, saying that x is not less than 2 $(x \not< 2)$ means that x does not lie to the left of 2 on the number line. This is equivalent to saying that either $x = 2$ or x lies to the right of 2 on the number line $(x = 2$ or $x > 2)$. Hence, $x \not< 2$ is equivalent to $x \geq 2$. Pictorially,

Similarly,

Example 2.1.10 Inequalities and De Morgan's Laws

Use De Morgan's laws to write the negation of $-1 < x \leqslant 4$.

Solution The given statement is equivalent to

$$-1 < x \text{ and } x \leq 4.$$

By De Morgan's laws, the negation is

Caution! The negation of $-1 < x \leq 4$ is *not* -1 □ x $ 4. It is also not $-1 \geq x > 4$.

which is equivalent to

-1 □ x or x i 4,

$$-1 \geq x \text{ or } x > 4.$$

Pictorially, if $-1 \geq x$ or $x > 4$, then x lies in the shaded region of the number line, as shown below.

De Morgan's laws are frequently used in writing computer programs. For instance, suppose you want your program to delete all files modified outside a certain range of dates, say from date 1 through date 2 inclusive. You would use the fact that

$$\sim(date1 \leq file\_modification\_date \leq date2)$$

is equivalent to

$$(file\_modification\_date < date1) \quad \text{or} \quad (date2 < file\_modification\_date).$$

Example 2.1.11 A Cautionary Example

According to De Morgan's laws, the negation of

p: Jim is tall and Jim is thin

i s

$\sim p$: Jim is not tall or Jim is not thin

because the negation of an *and* statement is the *or* statement in which the two components are negated.

Unfortunately, a potentially confusing aspect of the English language can arise when you are taking negations of this kind. Note that statement p can be written more com- pactly as

Caution! Although the laws of logic are extremely useful, they should be used as an *aid* to thinking, not as a mechanical substitute for

p': Jim is tall and thin.

When it is so written, another way to negate it is

$\sim(p')$: Jim is not tall and thin.

But in this form the negation looks like an *and* statement. Doesn't that violate De Morgan's laws?

Actually no violation occurs. The reason is that in formal logic the words *and* and *or* are allowed only between complete statements, not between sentence fragments.

One lesson to be learned from this example is that when you apply De Morgan’s laws, you must have complete statements on either side of each *and* and on either side of each *or*.

it.

■

Tautologies and Contradictions

It has been said that all of mathematics reduces to tautologies. Although this is formally true, most working mathematicians think of their subject as having substance as well as form. Nonetheless, an intuitive grasp of basic logical tautologies is part of the equipment of anyone who reasons with mathematics.

Definition

A **tautology** is a statement form that is always true regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a tautology is a **tautological statement.**

A **contradication** is a statement form that is always false regardless of the truth val- ues of the individual statements substituted for its statement variables. A statement whose form is a contradication is a **contradictory statement.**

According to this definition, the truth of a tautological statement and the falsity of a contradictory statement are due to the logical structure of the statements themselves and are independent of the meanings of the statements.

Example 2.1.12 Tautologies and Contradictions

Show that the statement form *p* □ □*p* is a tautology and that the statement form *p* □ □*p*

is a contradiction

Example 2.1.13 Logical Equivalence Involving Tautologies and Contradictions

If **t** is a tautology and **c** is a contradiction, show that $p \wedge \mathbf{t} \equiv p$ and $p \wedge \mathbf{c} \equiv \mathbf{c}$.

Summary of Logical Equivalences

Knowledge of logically equivalent statements is very useful for constructing arguments. It often happens that it is difficult to see how a conclusion follows from one form of a statement, whereas it is easy to see how it follows from a logically equivalent form of the statement. A number of logical equivalences are summarized in Theorem 2.1.1 for future reference.

Theorem 2.1.1 Logical Equivalences

Table

The proofs of laws 4 and 6, the first parts of laws 1 and 5, and the second part of law 9 have already been given as examples in the text. Proofs of the other parts of the theorem are left as exercises. In fact, it can be shown that the first five laws of Theorem 2.1.1 form a core from which the other laws can be derived. The first five laws are the axioms for a mathematical structure known as a Boolean algebra, which is discussed in Section 6.4.

The equivalences of Theorem 2.1.1 are general laws of thought that occur in all areas of human endeavor. They can also be used in a formal way to rewrite complicated state- ment forms more simply.

Example 2.1.14 Simplifying Statement Forms

Use Theorem 2.1.1 to verify the logical equivalence

$$\square(\square p \;\square\; q) \;\square\; (p \;\square\; q) \equiv p.$$

Solution Use the laws of Theorem 2.1.1 to replace sections of the statement form on the left by logically equivalent expressions. Each time you do this, you obtain a logically equivalent statement form. Continue making replacements until you obtain the statement form on the right.

$\square(\square p \;\square\; q) \;\square\; (p \;\square\; q) \equiv (\square(\square p) \;\square\; \square q) \;\square\; (p \;\square\; q)$ by De Morgan's laws

$\equiv (p \;\square\; \square q) \;\square\; (p \;\square\; q)$ by the double negative law

≡ *p* □ (□*q* □ *q*) by the distributive law

≡ *p* □ (*q* □ □*q*) by the commutative law for ∧

≡ *p* □ **c** by the negation law

≡ *p* by the identity law. ■

Skill in simplifying statement forms is useful in constructing logically efficient computer programs and in designing digital logic circuits.

Although the properties in Theorem 2.1.1 can be used to prove the logical equivalence of two statement forms, they cannot be used to prove that statement forms are not logically equivalent. On the other hand, truth tables can always be used to determine both equivalence and nonequivalence, and truth tables are easy to program on a computer. When truth tables are used, however, checking for equivalence always requires 2^n steps, where *n* is the number of variables. Sometimes you can quickly see that two statement forms are equivalent by Theorem 2.1.1, whereas it would take quite a bit of calculating to show their equivalence using truth tables. For instance, it follows immediately from the associative law for □ that *p* □ (□*q* □ □*r*) ≡ (*p* □ □*q*) □ □*r*, whereas a truth table verification requires constructing a table with eight rows.

Concepts and Procedures

1. An *and* statement is true if, and only if, both components are _____.

2. An *or* statement is false if, and only if, both components are _____.

3. Two statement forms are logically equivalent if, and only if, they always have_______________.

4. De Morgan's laws say (1) that the negation of an *and* state- ment is logically equivalent to the___________statement in which each component is , and (2) that the negation of an *or* statement is logically equivalent to the statement in which each component is .

1. A tautology is a statement that is always_____.

2. A contradiction is a statement that is always__.

In each of 1–4 represent the common form of each argument using letters to stand for component sentences, and fill in the blanks so that the argument in part (b) has the same logical form as the argument in part (a).

1. a. If all integers are rational, then the number 1 is rational. All integers are rational.
Therefore, the number 1 is rational.
b. If all algebraic expressions can be written in prefix notation, then_________.
____.
Therefore, $(a + 2b)(a^2 - b)$ can be written in prefix notation.

2. a. If all computer programs contain errors, then this program contains an error.

This program does not contain an error.
Therefore, it is not the case that all computer programs contain errors.

b. If_____, then_____. 2 is not odd.
Therefore, it is not the case that all prime numbers are odd.

3. a. This number is even or this number is odd. This number is not even.
Therefore, this number is odd.
b. or logic is confusing. My mind is not shot. Therefore,_________.

4. a. If n is divisible by 6, then n is divisible by 3.
If n is divisible by 3, then the sum of the digits of n is divisible by 3.

Therefore, if n is divisible by 6, then the sum of the dig- its of n is divisible by 3.
(Assume that n is a particular, fixed integer.)
b. If this function is__________then this function is differen- tiable.

If this function is then this function is continuous. Therefore, if this function is a polynomial, then this function

4. Indicate which of the following sentences are statements.
a. 1,024 is the smallest four-digit number that is a perfect square.

b. She is a mathematics major.

c. $128 = 2^6$ d. $x = 2^6$

Write the statements in 6–9 in symbolic form using the symbols
□, □, and □ and the indicated letters to represent component statements.

6. Let s = "stocks are increasing" and i = "interest rates are steady."

a. Stocks are increasing but interest rates are steady.
b. Neither are stocks increasing nor are interest rates steady.

1. Juan is a math major but not a computer science major. (m = "Juan is a math major," c = "Juan is a computer science major")

2. Let h = "John is healthy," w = "John is wealthy," and s = "John is wise."
a. John is healthy and wealthy but not wise.
b. John is not wealthy but he is healthy and wise.
c. John is neither healthy, wealthy, nor wise.
d. John is neither wealthy nor wise, but he is healthy.
e. John is wealthy, but he is not both healthy and wise.

3. Either this polynomial has degree 2 or it has degree 3 but not both. (n = "This polynomial has degree 2," k = "This polynomial has degree 3")

4. Let p be the statement "DATAENDFLAG is off," q the statement "ERROR equals 0," and r the statement "SUM is less than 1,000." Express the following sentences in sym- bolic notation.

a. DATAENDFLAG is off, ERROR equals 0, and SUM is less than 1,000.
b. DATAENDFLAG is off but ERROR is not equal to 0.
c. DATAENDFLAG is off; however, ERROR is not 0 or SUM is greater than or equal to 1,000.
d. DATAENDFLAG is on and ERROR equals 0 but SUM is greater than or equal to 1,000.

e. Either DATAENDFLAG is on or it is the case that both ERROR equals 0 and SUM is less than 1,000.

11. In the following sentence, is the word *or* used in its inclu- sive or exclusive sense? A team wins the playoffs if it wins two games in a row or a total of *three* games

Write truth tables for the statement forms in 12–15.

12. □*p* □ *q* 13. □(*p* □ *q*) □ (*p* □ *q*)

14. *p* □ (*q* □ *r*) 15. *p* □ (□*q* □ *r*)

Determine whether the statement forms in 16–24 are logically equivalent. In each case, construct a truth table and include a sentence justifying your answer. Your sentence should show that you understand the meaning of logical equivalence

16. *p* □ (*p* □ *q*) and *p* 17. □(*p* □ *q*) and □*p* □ □*q*

18. $p \vee \mathbf{t}$ *and* $\mathbf{t}$

19. $p \wedge \mathbf{t}$ *and* p

20. $p \wedge \mathbf{c}$ *and* $p \vee \mathbf{c}$

21. $(p \wedge q) \wedge r$ *and* $p \wedge (q \wedge r)$

22. $p \wedge (q \vee r)$ *and* $(p \wedge q) \vee (p \wedge r)$

23. $(p \wedge q) \vee r$ *and* $p \wedge (q \vee r)$

24. $(p \vee q) \vee (p \wedge r)$ *and* $(p \vee q) \wedge r$

Use De Morgan's laws to write negations for the statements in 25–31.

25. *Hal is a math major and Hal's sister is a computer science major.*

26. Sam is an orange belt and Kate is a red belt.

27. *The connector is loose or the machine is unplugged.*

28. The units digit of 4^{67} *is 4 or it is 6.*

29. This computer program has a logical error in the first ten lines or it is being run with an incomplete data set.

30. The dollar is at an all-time high and the stock market is at a record low.

31. The train is late or my watch is fast.

Assume x is a particular real number and use De Morgan's laws to write negations for the statements in 32–37.

32. $-2 < x < 7$ 33. $-10 < x < 2$

34. $x < 2$ or $x > 5$ 35. $x \leqslant 1$ or $x > 1$

36. $1 > x \geq -3$ 37. $0 > x \geq -7$

In 38 and 39, imagine that *num_orders* and *num_instock* are par- ticular values, such as might occur during execution of a com- puter program. Write negations for the following statements.

38. (*num_orders* > 100 and *num_instock* ⩽ 500) or
num_instock < 200

39. (*num_orders* < 50 and *num_instock* > 300) or (50 ≤ *num_orders* < 75 and *num_instock* > 500)

Use truth tables to establish which of the statement forms in 40–43 are tautologies and which are contradictions

40. (*p* □ *q*) □ (□*p* □ (*p* □ □*q*))

41. (*p* □ □*q*) □ (□*p* □ *q*)

42. ((□*p* □ *q*) □ (*q* □ *r*)) □ □*q*

43. (□*p* □ *q*) □ (*p* □ □*q*)

In 44 and 45, determine whether the statements in (a) and (b) are logically equivalent.

Problem Solving and Reasoning

44. Assume *x* is a particular real number.

a. $x < 2$ or it is not the case that $1 < x < 3$.

$x \leq 1$ or either $x < 2$ or $x \geq 3$

45. a. Bob is a double math and computer science major and Ann is a math major, but Ann is not a double math and computer science major.

b. It is not the case that both Bob and Ann are dou- ble math and computer science majors, but it is the case that Ann is a math major and Bob is a double math and computer science major.

46. In Example 2.1.4, the symbol ⊕ was introduced to denote *exclusive or*, so *p* ⊕ *q* ≡ (*p* □ *q*)□ □(*p* □ *q*). Hence the truth table for *exclusive or* is as follows:

table

◆ . In logic and in standard English, a double negative is equiv- alent to a positive. There is one fairly common English usage in which a “double positive” is equivalent to a nega- tive. What is it? Can you think of others?

In 48 and 49 below, a logical equivalence is derived from Theo- rem 2.1.1. Supply a reason for each step.

p □ □*q*) □ (*p* □ *q*) ≡ *p* □ (□*q* □ *q*) by (a)

48. (

≡ *p* □ (*q* □ □*q*) by (b)

≡ *p* □ **t** by (c)

≡ *p* by (d)

Therefore, (*p* □ □*q*) □ (*p* □ *q*) ≡ *p*.

49. (*p* □ □*q*) □ (□*p* □ □*q*)

≡ (□*q* □ *p*) □ (□*q* □ □*p*) by

(b)

(c)

≡ □*q* □ (*p* □ □*p*) by _______

≡ □*q* □ **c** by _______

≡ □*q* by (d)

Therefore, (*p* □ □*q*) □ (□*p* □ □*q*) ≡ □*q*.

Use Theorem 2.1.1 to verify the logical equivalences in 50–54. Supply a reason for each step.

50. (*p* □ □*q*) □ *p* ≡ *p* 51. *p* □ (□*q* □ *p*) ≡ *p*

52. □(*p* □ □*q*) □ (□*p* □ □*q*) ≡ □*p*

53. □((□*p* □ *q*) □ (□*p* □ □*q*)) □ (*p* □ *q*) ≡ *p*

54. (*p* □ (□(□*p* □ *q*))) □ (*p* □ *q*) ≡ *p*