normally. If the system is operating normally, the kernel is functioning. The kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode."

36. Are the following specifications consistent? "If the file system is not locked, then new messages will be queued. If the file system is not locked, then the system is functioning normally, and conversely. If new messages are not queued, then they will be sent to the message buffer. If the file system is not locked, then new messages will be sent to the message buffer. New messages will not be sent to the message buffer."

37. What Boolean search would you use to look for Web pages about beaches in New Jersey? What if you wanted to find Web pages about beaches on the isle of Jersey (in the English Channel)?

38. What Boolean search would you use to look for Web pages about hiking in West Virginia? What if you wanted to find Web pages about hiking in Virginia, but not in West Virginia?

Exercises 39–42 are puzzles that can be solved by translating statements into logical expressions and reasoning from these expressions using truth tables.

39. Steve would like to determine the relative salaries of three coworkers using two facts. First, he knows that if Fred is not the highest paid of the three, then Janice is. Second, he knows that if Janice is not the lowest paid, then Maggie is paid the most. Is it possible to determine the relative salaries of Fred, Maggie, and Janice from what Steve knows? If so, who is paid the most and who the least? Explain your reasoning.

40. Five friends have access to a chat room. Is it possible to determine who is chatting if the following information is known? Either Kevin or Heather, or both, are chatting. Either Randy or Vijay, but not both, are chatting. If Abby is chatting, so is Randy. Vijay and Kevin are either both chatting or neither is. If Heather is chatting, then so are Abby and Kevin. Explain your reasoning.

41. A detective has interviewed four witnesses to a crime. From the stories of the witnesses the detective has concluded that if the butler is telling the truth then so is the cook; the cook and the gardener cannot both be telling the truth; the gardener and the handyman are not both lying; and if the handyman is telling the truth then the cook is lying. For each of the four witnesses, can the detective determine whether that person is telling the truth or lying? Explain your reasoning.

42. Four friends have been identified as suspects for an unauthorized access into a computer system. They have made statements to the investigating authorities. Alice said "Carlos did it." John said "I did not do it." Carlos said "Diana did it." Diana said "Carlos lied when he said that I did it."

a) If the authorities also know that exactly one of the four suspects is telling the truth, who did it? Explain your reasoning.
b) If the authorities also know that exactly one is lying, who did it? Explain your reasoning.

1.2

Propositional Equivalences

INTRODUCTION

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments.

We begin our discussion with a classification of compound propositions according to their possible truth values.

DEFINITION 1. A compound proposition that is always true, no matter what the truth values of the propositions that occur in it, is called a tautology. A compound proposition that is always false is called a contradiction. Finally, a proposition that is neither a tautology nor a contradiction is called a contingency.

Tautologies and contradictions are often important in mathematical reasoning. The following example illustrates these types of propositions.
1.2 Propositional Equivalences

We can construct examples of tautologies and contradictions using just one proposition. Consider the truth tables of $p \lor \neg p$ and $p \land \neg p$, shown in Table 1. Since $p \lor \neg p$ is always true, it is a tautology. Since $p \land \neg p$ is always false, it is a contradiction.

### Logical Equivalences

Compound propositions that have the same truth values in all possible cases are called logically equivalent. We can also define this notion as follows.

**Definition 2.** The propositions $p$ and $q$ are called logically equivalent if $p \rightarrow q$ is a tautology. The notation $p \iff q$ denotes that $p$ and $q$ are logically equivalent.

One way to determine whether two propositions are equivalent is to use a truth table. In particular, the propositions $p$ and $q$ are equivalent if and only if the columns giving their truth values agree. The following example illustrates this method.

Show that $\neg(p \lor q)$ and $\neg p \land \neg q$ are logically equivalent. This equivalence is one of De Morgan's laws for propositions, named after the English mathematician Augustus De Morgan, of the mid-nineteenth century.

**Solution:** The truth tables for these propositions are displayed in Table 2. Since the truth values of the propositions $\neg(p \lor q)$ and $\neg p \land \neg q$ agree for all possible combinations of the truth values of $p$ and $q$, it follows that these propositions are logically equivalent.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \lor q$</th>
<th>$\neg(p \lor q)$</th>
<th>$\neg p$</th>
<th>$\neg q$</th>
<th>$\neg p \land \neg q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
### TABLE 3  Truth Tables for $\neg p \lor q$ and $p \rightarrow q$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\neg p$</th>
<th>$\neg p \lor q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

**Example 3**

Show that the propositions $p \rightarrow q$ and $\neg p \lor q$ are logically equivalent.

**Solution:** We construct the truth table for these propositions in Table 3. Since the truth values of $\neg p \lor q$ and $p \rightarrow q$ agree, these propositions are logically equivalent.

**Example 4**

Show that the propositions $p \lor (q \land r)$ and $(p \lor q) \land (p \lor r)$ are logically equivalent. This is the distributive law of disjunction over conjunction.

**Solution:** We construct the truth table for these propositions in Table 4. Since the truth values of $p \lor (q \land r)$ and $(p \lor q) \land (p \lor r)$ agree, these propositions are logically equivalent.

**Remark:** A truth table of a compound proposition involving three different propositions requires eight rows, one for each possible combination of truth values of the three propositions. In general, $2^n$ rows are required if a compound proposition involves $n$ propositions.

---

Augustus De Morgan (1806–1871). Augustus De Morgan was born in India, where his father was a colonel in the Indian army. De Morgan's family moved to England when he was 7 months old. He attended private schools, where he developed a strong interest in mathematics in his early teens. De Morgan studied at Trinity College, Cambridge, graduating in 1827. Although he considered entering medicine or law, he decided on a career in mathematics. He won a position at University College, London, in 1828, but resigned when the college dismissed a fellow professor without giving reasons. However, he resumed this position in 1836 when his successor died, staying there until 1866.

De Morgan was a noted teacher who stressed principles over techniques. His students included many famous mathematicians, including Ada Augusta, Countess of Lovelace, who was Charles Babbage's collaborator in his work on computing machines (see page 19 for biographical notes on Ada Augusta). (De Morgan cautioned the countess against studying too much mathematics, since it might interfere with her childbearing abilities!)

De Morgan was an extremely prolific writer. He wrote more than 1000 articles for more than 15 periodicals. De Morgan also wrote textbooks on many subjects, including logic, probability, calculus, and algebra. In 1838 he presented what was perhaps the first clear explanation of an important proof technique known as *mathematical induction* (discussed in Section 3.2 of this text), a term he coined. In the 1840s De Morgan made fundamental contributions to the development of symbolic logic. He invented notations that helped him prove propositional equivalences, such as the laws that are named after him. In 1842 De Morgan presented what was perhaps the first precise definition of a limit and developed some tests for convergence of infinite series. De Morgan was also interested in the history of mathematics and wrote biographies of Newton and Halley.

In 1837 De Morgan married Sophia Freind, who wrote his biography in 1882. De Morgan’s research, writing, and teaching left little time for his family or social life. Nevertheless, he was noted for his kindness, humor, and wide range of knowledge.
TABLE 4: A Demonstration That \( p \lor (q \land r) \) and \( (p \lor q) \land (p \lor r) \) Are Logically Equivalent.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>( q \land r )</th>
<th>( p \lor (q \land r) )</th>
<th>( p \lor q )</th>
<th>( p \lor r )</th>
<th>( (p \lor q) \land (p \lor r) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 5 contains some important equivalences.* In these equivalences, T denotes any proposition that is always true and F denotes any proposition that is always false. The reader is asked to verify these equivalences in the exercises at the end of the section. The associative law for disjunction shows that the expression \( p \lor q \lor r \) is well defined, in the sense that it does not matter whether we first take the disjunction of \( p \) and

TABLE 5: Logical Equivalences.

<table>
<thead>
<tr>
<th>Equivalence</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p \land T \iff p )</td>
<td>Identity laws</td>
</tr>
<tr>
<td>( p \lor F \iff p )</td>
<td></td>
</tr>
<tr>
<td>( p \lor T \iff T )</td>
<td>Domination laws</td>
</tr>
<tr>
<td>( p \land F \iff F )</td>
<td></td>
</tr>
<tr>
<td>( p \lor p \iff p )</td>
<td>Idempotent laws</td>
</tr>
<tr>
<td>( p \land p \iff p )</td>
<td></td>
</tr>
<tr>
<td>( \neg(\neg p) \iff p )</td>
<td>Double negation law</td>
</tr>
<tr>
<td>( p \lor q \iff q \lor p )</td>
<td>Commutative laws</td>
</tr>
<tr>
<td>( p \land q \iff q \land p )</td>
<td></td>
</tr>
<tr>
<td>( (p \lor q) \lor r \iff p \lor (q \lor r) )</td>
<td>Associative laws</td>
</tr>
<tr>
<td>( (p \land q) \lor r \iff p \lor (q \land r) )</td>
<td></td>
</tr>
<tr>
<td>( p \lor (q \land r) \iff (p \lor q) \land (p \lor r) )</td>
<td>Distributive laws</td>
</tr>
<tr>
<td>( p \land (q \lor r) \iff (p \land q) \lor (p \land r) )</td>
<td></td>
</tr>
<tr>
<td>( \neg(p \lor q) \iff \neg p \land \neg q )</td>
<td>De Morgan's laws</td>
</tr>
<tr>
<td>( \neg(p \land q) \iff \neg p \lor \neg q )</td>
<td></td>
</tr>
</tbody>
</table>

*These identities are a special case of identities that hold for any Boolean algebra. Compare them with set identities in Table 1 in Section 1.5 and with Boolean identities in Table 5 in Section 9.1.
<table>
<thead>
<tr>
<th>TABLE 5</th>
<th>Some Useful Logical Equivalences.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \lor \neg p \iff T$</td>
<td></td>
</tr>
<tr>
<td>$p \land \neg p \iff F$</td>
<td></td>
</tr>
<tr>
<td>$(p \rightarrow q) \iff (\neg p \lor q)$</td>
<td></td>
</tr>
</tbody>
</table>

$q$ and then the disjunction of $p \lor q$ with $r$, or if we first take the disjunction of $q$ and $r$ and then take the disjunction of $p$ and $q \lor r$. Similarly, the expression $p \land q \lor r$ is well defined. By extending this reasoning, it follows that $p_1 \lor p_2 \lor \cdots \lor p_n$, $p_1 \land p_2 \land \cdots \land p_n$, and $p_1 \lor p_2 \lor \cdots \lor p_n$ are well defined whenever $p_1$, $p_2$, $\ldots$, $p_n$ are propositions. Furthermore, note that De Morgan's laws extend to

$$
\neg (p_1 \lor p_2 \lor \cdots \lor p_n) \iff (\neg p_1 \land \neg p_2 \land \cdots \land \neg p_n)
$$

and

$$
\neg (p_1 \land p_2 \land \cdots \land p_n) \iff (\neg p_1 \lor \neg p_2 \lor \cdots \lor \neg p_n).
$$

(Methods for proving these identities will be given in Chapter 3.)

The logical equivalences in Table 5, as well as any others that have been established (such as those shown in Table 6), can be used to construct additional logical equivalences. The reason for this is that a proposition in a compound proposition can be replaced by one that is logically equivalent to it without changing the truth value of the compound proposition. This technique is illustrated in Examples 5 and 6, where we also use the fact that if $p$ and $q$ are logically equivalent and $q$ and $r$ are logically equivalent, then $p \land q$ and $r$ are logically equivalent (see Exercise 40).

**EXAMPLE 5**

Show that $\neg (p \lor (\neg p \land q))$ and $\neg p \land \neg q$ are logically equivalent.

**Solution:** We could use a truth table to show that these compound propositions are equivalent. Instead, we will establish this equivalence by developing a series of logical equivalences, using one of the equivalences in Table 5 at a time, starting with $\neg (p \lor (\neg p \land q))$ and ending with $\neg p \land \neg q$. We have the following equivalences.

$$
\begin{align*}
\neg (p \lor (\neg p \land q)) & \iff \neg p \land (\neg p \lor q) & \text{from the second De Morgan's law} \\
& \iff \neg p \land [\neg (\neg p) \lor q] & \text{from the first De Morgan's law} \\
& \iff \neg p \land (p \lor \neg q) & \text{from the double negation law} \\
& \iff (\neg p \land p) \lor (\neg p \land \neg q) & \text{from the distributive law} \\
& \iff F \lor (\neg p \land \neg q) & \text{since } \neg p \land p \iff F \\
& \iff (\neg p \land \neg q) \lor F & \text{from the commutative law} \\
& \iff \neg p \land \neg q & \text{for disjunction} \\
\end{align*}
$$

Consequently $\neg (p \lor (\neg p \land q))$ and $\neg p \land \neg q$ are logically equivalent.

**EXAMPLE 6**

Show that $(p \land q) \rightarrow (p \lor q)$ is a tautology.

**Solution:** To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to $T$. (Note: This could also be done using a truth table.)
\[(p \land q) \rightarrow (p \lor q) \iff -(p \land q) \lor (p \lor q)\]
\[\iff (-p \lor -q) \lor (p \lor q)\]
\[\iff (-p \lor -q) \lor (-q \lor q)\]
\[\iff T \lor T\]
\[\iff T\]

A truth table can be used to determine whether a compound proposition is a tautology. This can be done by hand for a proposition with a small number of variables, but when the number of variables grows, this becomes impractical. For instance, there are \(2^{20} = 1,048,576\) rows in the truth value table for a proposition with 20 variables. Clearly, you need a computer to help you determine, in this way, whether a compound proposition in 20 variables is a tautology. But when there are 1000 variables, can even a computer determine in a reasonable amount of time whether a compound proposition is a tautology? Checking every one of the \(2^{1000}\) (a number with more than 300 decimal digits) possible combinations of truth values simply cannot be done by a computer in even trillions of years. Furthermore, no other procedures are known that a computer can follow to determine in a reasonable amount of time whether a compound proposition in such a large number of variables is a tautology. We will study questions such as this in Chapter 2, when we study the complexity of algorithms.

### Exercises

1. Use truth tables to verify the following equivalences.
   a) \(p \land T \iff p\)
   b) \(p \lor F \iff p\)
   c) \(p \land F \iff F\)
   d) \(p \lor T \iff T\)
   e) \(p \lor p \iff p\)
   f) \(p \land p \iff p\)

2. Show that \(\neg(p \land q)\) and \(p\) are logically equivalent.

3. Use truth tables to verify the commutative laws
   a) \(p \lor q \iff q \lor p\)
   b) \(p \land q \iff q \land p\)

4. Use truth tables to verify the associative laws
   a) \((p \lor q) \lor r \iff p \lor (q \lor r)\)
   b) \((p \land q) \land r \iff p \land (q \land r)\)

5. Use truth tables to verify the distributive law \(p \land (q \lor r) \iff (p \land q) \lor (p \land r)\).

6. Use a truth table to verify the equivalence \(\neg (p \land q) \iff \neg p \lor \neg q\).

7. Show that each of the following implications is a tautology by using truth tables.
   a) \((p \land q) \rightarrow p\)
   b) \(p \rightarrow (p \lor q)\)
   c) \(\neg p \rightarrow (p \rightarrow q)\)
   d) \((p \lor q) \rightarrow (p \lor q)\)
   e) \((p \rightarrow q) \rightarrow p\)
   f) \((p \rightarrow q) \rightarrow \neg q\)

8. Show that each of the following implications is a tautology by using truth tables.
   a) \((\neg p \lor (p \lor q)) \rightarrow q\)
   b) \(((p \lor q) \land (q \lor r)) \rightarrow (p \lor r)\)
   c) \((p \lor (p \rightarrow q)) \rightarrow q\)
   d) \(((p \lor q) \land (p \lor r)) \land (q \lor r)) \rightarrow r\)

---

Ada Augusta, Countess of Lovelace (1815–1852). Ada Augusta was the only child from the marriage of the famous poet Lord Byron and Annabella Milbanke, who separated when Ada was 1 month old. She was raised by her mother, who encouraged her intellectual talents. She was taught by the mathematicians William Frend and Augustus De Morgan. In 1838 she married Lord King, later elevated to Earl of Lovelace. Together they had three children.

Ada Augusta continued her mathematical studies after her marriage, assisting Charles Babbage in his work on an early computing machine, called the Analytic Engine. The most complete accounts of this machine are found in her writings. After 1845 she and Babbage worked toward the development of a system to predict horse races. Unfortunately, their system did not work well, leaving Ada heavily in debt at the time of her death. The programming language Ada is named in honor of the Countess of Lovelace.
9. Show that each implication in Exercise 7 is a tautology without using truth tables.

10. Show that each implication in Exercise 8 is a tautology without using truth tables.

11. Verify the following equivalences, which are known as the absorption laws.
   a) \( p \vee (p \land q) \iff p \)
   b) \( p \land (p \lor q) \iff p \)

12. Determine whether \( (\neg p \land (p \to q)) \to \neg q \) is a tautology.

13. Determine whether \( (\neg q \land (p \to q)) \to \neg p \) is a tautology.

14. Show that \( p \iff q \) and \( (p \lor q) \lor (\neg p \land \neg q) \) are equivalent.

15. Show that \( (p \to q) \to r \) and \( p \to (q \to r) \) are not equivalent.

16. Show that \( p \to q \) and \( \neg q \to \neg p \) are logically equivalent.

17. Show that \( \neg p \iff q \) and \( p \iff \neg q \) are logically equivalent.

18. Show that \( (p \lor q) \) and \( p \lor q \) are logically equivalent.

19. Show that \( \neg (p \iff q) \) and \( \neg p \iff q \) are logically equivalent.

The dual of a compound proposition that contains only the logical operators \( \lor, \land \), and \( \neg \) is the proposition obtained by replacing each \( \lor \) by \( \land \), each \( \land \) by \( \lor \), each \( T \) by \( F \), and each \( F \) by \( T \). The dual of proposition \( s \) is denoted by \( s' \).

20. Find the dual of each of the following propositions.
   a) \( p \land \neg q \land \neg r \)
   b) \( (p \land q) \lor r \lor s \)

21. Show that \( (s')' = s \).

22. Show that the logical equivalences in Table 5, except for the double negation law, come in pairs, where each pair contains propositions that are duals of each other.

**23. Why are the duals of two equivalent compound propositions also equivalent, where these compound propositions contain only the operators \( \land, \lor, \land \)?

24. Find a compound proposition involving the propositions \( p, q, \) and \( r \) that is true when \( p \) and \( q \) are true and \( r \) is false, but is false otherwise. \( Hint: \) Use a conjunction of each proposition or its negation.

25. Find a compound proposition involving the propositions \( p, q, \) and \( r \) that is true when exactly two of \( p, q, \) and \( r \) are true and is false otherwise. \( Hint: \) Form a conjunction of disjunctions. Include a conjunction for each combination of values for which the proposition is true. Each conjunction should include each of the three propositions or their negations.

26. Suppose that a truth table in \( n \) propositional variables is specified. Show that a compound proposition with this truth table can be formed by taking the disjunction of conjunctions of the variables or their negations, with one conjunction included for each combination of values for which the compound proposition is true. The resulting compound proposition is said to be in disjunctive normal form.

A collection of logical operators is called functionally complete if every compound proposition is logically equivalent to a compound proposition involving only those logical operators.

27. Show that \( \neg, \land, \) and \( \lor \) form a functionally complete collection of logical operators. \( Hint: \) Use the fact that every proposition is logically equivalent to one in disjunctive normal form, as shown in Exercise 26.

28. Show that \( \neg \land \) and \( \lor \) form a functionally complete collection of logical operators. \( Hint: \) First use De Morgan's law to show that \( p \lor q \) is equivalent to \( \neg (\neg p \land \neg q) \).

29. Show that \( \neg \) and \( \lor \) form a functionally complete collection of logical operators.

The following exercises involve the logical operators NAND and NOR. The proposition \( p \land q \) is true when both \( p \) or \( q \), or both, are false; and it is false when both \( p \) and \( q \) are true. The proposition \( p \lor q \) is true when both \( p \) and \( q \) are false, and it is false otherwise. The propositions \( p \land q \) and \( p \lor q \) are denoted by \( p \downarrow q \) and \( p \downarrow q \), respectively. (The operators \( | \) and \( \downarrow \) are called the Sheffer stroke and the Peirce arrow after H. M. Sheffer and C. S. Peirce, respectively.)

30. Construct a truth table for the logical operator NAND.

31. Show that \( p \downarrow q \) is logically equivalent to \( \neg (p \land q) \).

32. Construct a truth table for the logical operator NOR.

33. Show that \( p \downarrow q \) is logically equivalent to \( \neg (p \lor q) \).

34. In this exercise we will show that \( \{ \} \) is a functionally complete collection of logical operators.
   a) Show that \( p \downarrow p \) is logically equivalent to \( \neg p \).
   b) Show that \( (p \downarrow q) \downarrow (p \downarrow q) \) is logically equivalent to \( p \lor q \).
   c) Conclude from parts (a) and (b), and Exercise 29, that \( \{ \} \) is a functionally complete collection of logical operators.

35. Find a proposition equivalent to \( p \to q \) using only the logical operator \( \downarrow \).

36. Show that \( \{ \} \) is a functionally complete collection of logical operators.

37. Show that \( p \downarrow q \) and \( q \downarrow p \) are equivalent.

38. Show that \( p \downarrow (q \downarrow r) \) and \( (p \downarrow q) \downarrow r \) are not equivalent, so that the logical operator \( | \) is not associative.

39. How many different truth tables of compound propositions are there that involve the propositions \( p \) and \( q \)?

40. Show that if \( p, q, \) and \( r \) are compound propositions such that \( p \) and \( q \) are logically equivalent and \( q \) and \( r \) are logically equivalent, then \( p \) and \( r \) are logically equivalent.

41. The following sentence is taken from the specification of a telephone system: "If the directory database is opened, then the monitor is put in a closed state, if the system is not in its initial state." This specification is hard to understand since it involves two implications. Find an equivalent, easier-to-understand specification that involves disjunctions and negations but not implications.