About Discrete mathematics

Discrete mathematics may be taken as real-world mathematics. It encourages students to examine problems on topic like reasoning, counting, combinatory, etc. to explore the real world which are interesting and challenging. It teaches mathematical reasoning and proof techniques. A discrete mathematics course has more than one purpose. Students should learn a particular set of mathematical facts and how to apply them. In particular such a course teaches students how to think logically and mathematically. To achieve these goals, this text stresses mathematical reasoning and the different ways problems are solved. Five important themes in this course are: mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modeling. A successful discrete mathematics course should carefully blend and balance all five themes. In this course we study first three themes.

Mathematical Reasoning: Students must understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments. This text starts with a discussion of mathematical logic, which serves as the foundation for the subsequent discussions of methods of proof. Both the science and the art of constructing proofs are addressed. The technique of mathematical induction is stressed through many different types of examples of such proofs and a careful explanation of why mathematical induction is a valid proof technique.

Combinatorial Analysis: An important problem-solving skill is the ability to count or enumerate objects. Here we study the method of recurrence relation.

Discrete Structures: A course in discrete mathematics should teach students how to work with discrete structures, which are the abstract mathematical structures used to represent discrete objects and relationships between these objects. These discrete structures include sets, permutations, relations, graphs, semigroups, groups, ring and field.

Unit 1 -- Logic and Induction

Introduction

Logic is a set or a system of principles or rules. The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments and specify the meaning of mathematical statements. Logic is the basis of all mathematical reasoning, and of all automated reasoning. Besides the importance of logic in understanding mathematical reasoning, logic has numerous applications to computer science.

These rules are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs, and in many other ways. To understand mathematics, we must understand what makes up a correct mathematical argument, that is, a proof. Everyone knows that proofs are important throughout mathematics, but many people find it surprising how important proofs are in computer science. In fact, proofs are used to verify that computer programs produce the correct output for all possible input values, to show that algorithms always produce the correct result, to establish the security of a system, and to create artificial intelligence. Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs. First, we will explain what makes up a correct mathematical argument and then introduce tools to construct these arguments.

Proposition

Our discussion begins with an introduction to the basic building blocks of logic—propositions.
A **proposition** is a declarative sentence that is either true or false, but not both.

Example: All the following declarative sentences are propositions.

1. Washington, D.C., is the capital of the United States of America.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Example: Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of sentences 3 and 4 can be turned into a proposition if we assign some values to the variables x, y and z.

We use letters to denote **propositional variables** (or **statement variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables. The conventional letters used for propositional variables are $p, q, r, s, \ldots$. The **truth value** of a proposition is true, denoted by $\boldsymbol{T}$, if it is a true proposition, and the truth value of a proposition is false, denoted by $\boldsymbol{F}$, if it is a false proposition. The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**. It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.

Atomic propositions

Propositions that cannot be expressed in terms of simpler propositions are called **atomic propositions**.

Compound propositions

Many mathematical statements are constructed by combining one or more atomic propositions by using logical operators. These are called compound propositions. Thus, **compound propositions** are formed from existing propositions using logical operators.

Note: To study the truth values of a compound statement we represent all the possible cases in a table called **truth table**.

Negation

Let p be a proposition. The **negation of $\boldsymbol{p}$**, denoted by $\boldsymbol{\neg p}$, is the statement "It is not the case that p." The proposition $\neg p$ is also read as "not p." The truth value of $\neg p$, is the opposite of the truth value of p. The negation of a proposition can also be considered as the result of the operation of the **negation operator** on a proposition. The negation operator constructs a new proposition from a single existing proposition without loss of information, which alter the truth value of the original proposition. The Truth Table for the Negation of a Proposition is given below

p	$\neg p$
T	F
F	T

Example: Find the negation of the proposition "Michael's PC runs Linux" and express this in simple English.

Solution: The negation is "It is not the case that Michael's PC runs Linux."

This negation can be more simply expressed as "Michael's PC does not run Linux."

Example: Find the negation of the proposition "Vandana's smartphone has at least 32GB of memory" and express this in simple English.

Solution: The negation is "It is not the case that Vandana's smartphone has at least 32GB of memory."

This negation can also be expressed as "Vandana's smartphone does not have at least 32GB of memory"

or even more simply as "Vandana's smartphone has less than 32GB of memory."

Note: The notation for the negation operator is not standardized, although $-\mathrm{p}, \neg\mathrm{p}, \sim\mathrm{p}$ and $\bar{\mathrm{p}}$ are the most common notations used in mathematics to express the negation of p.

Connectives

We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

Conjunction

Let p and q be propositions. The **conjunction** of p and q, denoted by $\boldsymbol{p} \wedge \boldsymbol{q}$, is the proposition "$p$ and q." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise. The Truth Table for the conjunction $p \wedge q$ is given below:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Example: Find the conjunction of the propositions p and q where p is the proposition "Rebecca's PC has more than 16 GB free hard disk space" and q is the proposition "The processor in Rebecca's PC runs faster than 1 GHz."

Solution: The conjunction of these propositions is the proposition "Rebecca's PC has more than 16 GB free hard disk space, and the processor in Rebecca's PC runs faster than 1 GHz." This conjunction can be expressed more simply as "Rebecca's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz." For this conjunction to be true, both conditions given must be true. It is false, when one or both of these conditions are false.

Note: In logic the word "but" sometimes is used instead of "and" in a conjunction. For example, the statement "The sun is shining, but it is raining" is another way of saying "The sun is shining and it is raining."

Disjunction

Let p and q be propositions. The **disjunction** of p and q, denoted by $\boldsymbol{p \vee q}$, is the proposition "p or q." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise. The Truth Table for the disjunction $p \vee q$ is given below

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Example: Translate the statement "Students who have taken calculus or introductory computer science can take this class" in a statement in propositional logic using the propositions p: "A student who has taken calculus can take this class" and q: "A student who has taken introductory computer science can take this class."

Solution: We assume that this statement means that students who have taken both calculus and Introductory computer science can take the class, as well as the students who have taken only one of the two subjects. Hence, this statement can be expressed as $p \vee q$, the disjunction of p and q.

Note: There are two types of 'or' are used in English, **inclusive or** and **exclusive or.** The use of the connective 'or' in a disjunction corresponds to '**inclusive or'**. For instance, the inclusive or is being used in the previous example. On the other hand, we are using the **exclusive or** when we say "Students who have taken calculus or computer science, but not both, can enroll in this class." Here, we mean that students who have taken both calculus and a computer science course cannot take the class. Only those who have taken exactly one of the two courses can take the class. In everyday conversation when we say "p or q" we mean p is true or q is true, but not both

p and q are true. For example, “the door is open or the door is closed.” Similarly, when a menu at a restaurant states, “Soup or salad comes with an entrée,” the restaurant almost always means that customers can have either soup or salad, but not both. Hence, this is an exclusive, rather than an inclusive or. In this course we refer to ‘inclusive or’ whenever we use ‘or’.

Exclusive or

The **exclusive or** of two propositions p and q, denoted by $\boldsymbol{p \oplus q}$, is the proposition “Either p or q”. The proposition $p \oplus q$ is true when exactly one of p and q is true and is false otherwise.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Example: Express the statement “I will use all my savings to travel to Europe or to buy an electric car” in propositional logic using the statement p: “I will use all my savings to travel to Europe” and the statement q: “I will use all my savings to buy an electric car.”

Solution: To translate this statement, we first note that the or in this statement must be an exclusive or because this person can either use all his or her savings to travel to Europe or use all these savings to buy an electric car, but cannot both go to Europe and buy an electric car. (This is clear because either option requires all his savings.) Hence, this statement can be expressed as $p \oplus q$.

Conditional statement

Let p and q be propositions. The **conditional statement** $\boldsymbol{p \rightarrow q}$ is the proposition “if p then q.” The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the hypothesis and q is called the conclusion.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Example: Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job." Express the statement $p \rightarrow q$ as a statement in English.

Solution: From the definition of conditional statements, we see that when p is the statement "Maria learns discrete mathematics" and q is the statement "Maria will find a good job," $p \rightarrow q$ represents the statement

"If Maria learns discrete mathematics then she will find a good job."

There are many other ways to express this conditional statement in English. Among the most natural of these are:

"Maria will find a good job when she learns discrete mathematics."

"For Maria to get a good job, it is sufficient for her to learn discrete mathematics."

"Maria will find a good job unless she does not learn discrete mathematics."

Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \rightarrow q$. You will encounter most if not all of the following ways to express this conditional statement:

if p, then q

if p, q

p is sufficient for q

q if p

q when p

a necessary condition for p is q

q unless $\neg p$

p implies q

p only if q

a sufficient condition for q is p

q whenever p

q is necessary for p

q follows from p

q provided that p

Note: A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract. For example, the pledge many politicians make when election comes,

"If I am elected, then I will lower petrol price."

If the politician is elected, voters would expect this politician to lower petrol price. Furthermore, if the politician is not elected, then voters will not have any expectation that this person will lower petrol price, although the person may have sufficient influence to cause those in power to lower petrol price. It is only when the politician is elected but does not lower petrol price that voters can say that the politician has broken the campaign pledge. This last scenario corresponds to the case when p is true but q is false in $p \to q$.

Note: Some people have difficulty using the truth table for $p \to q$ because of this ambiguity in English. Suppose that I hold an ordinary playing card (with its back to you) and say "If this card is a heart, then it is a queen." In which of the following four circumstances would you say I lied: 1. the card is a heart and a queen

2. the card is a heart and a king

3. the card is a diamond and a queen

4. the card is a diamond and a king

You would certainly say I lied in the case the card is the king of hearts, and you would certainly say I didn't lie if the card is the queen of hearts. What about the other two? Hopefully in this example, the inconsistency of English language seems out of place to you and you would not say I am a liar in either of the other cases. Now we apply the principle called the principle of the excluded middle, "A statement is true exactly when it is not false." This principle tells us that that my statement is true in these two cases where you wouldn't say I lied.

Note: The way we have defined conditional statements is more general than the meaning attached to such statements in the English language. For instance, the conditional statement

"If it is sunny, then we will go to the beach"

is true unless it is indeed sunny, but we do not go to the beach. On the other hand, the statement

"If Juan has a smartphone, then $2 + 3 = 5$"

is true from the definition of a conditional statement, because its conclusion is true. The conditional statement

"If Juan has a smartphone, then $2 + 3 = 6$"

is true if Juan does not have a smartphone, even though $2 + 3 = 6$ is false.

We would not use these last two conditional statements in natural language, because there is no relationship between the hypothesis and the conclusion in either statement. In mathematical reasoning, we consider conditional statements of a more general sort than we use in English. The mathematical concept of a conditional statement is independent of a cause and effect relationship between hypothesis and conclusion. Our definition of a conditional statement specifies its truth values; it is not based on English usage. Propositional language is an artificial language; we only parallel English usage to make it easy to use and remember.

Biconditional

Let p and q be propositions. The **biconditional** statement $\boldsymbol{p} \leftrightarrow \boldsymbol{q}$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Example: Let p be the statement "You can take the flight," and let q be the statement "You buy a ticket." Then biconditional statement $p \leftrightarrow q$ is the statement

"You can take the flight if and only if you buy a ticket."

There are some other common ways to express $p \leftrightarrow q$:

"p is necessary and sufficient for q."

"if p then q, and conversely."

"p if and only if q."

"p exactly when q."

Note: We should be aware that biconditionals are not always explicit in natural language. In particular, the "if and only if" construction used in biconditionals is rarely used in common language. Instead, biconditionals are often expressed using an "if, then" or an "only if" construction. The other part of the "if and only if" is implicit. That is, the converse is implied, but not stated. For example, consider the statement in English "If you finish your meal, then you can have dessert." What is really meant is "You can have dessert if and only if you finish your meal." This last statement is logically equivalent to the two statements "If you finish your meal, then you can have dessert" and "You can have dessert only if you finish your meal." Because of this imprecision in natural language, we need to make an assumption whether a conditional statement in natural language implicitly includes its converse. Because precision is essential in mathematics and in logic, we will always distinguish between the conditional statement $p \rightarrow q$ and the biconditional statement $p \leftrightarrow q$.

Tautology, contradiction and contingency

A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a **tautology**. A compound proposition that is always false is called a **contradiction.** A compound proposition that is neither a tautology nor a contradiction is called a **contingency.**

Example: We can construct examples of tautologies and contradictions using just one propositional variable. Consider the truth tables of $p \vee \neg p$ and $p \wedge \neg p$.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p.$
T	F	T	F
F	T	T	F

Because $p \vee \neg p$ is always true, it is a tautology. Because $p \wedge \neg p$ is always false, it is a contradiction.

Converse, Inverse, and Contrapositive

We can form some new conditional statements starting with a conditional statement $p \rightarrow q$. In particular, there are three related conditional statements that occur so often that they have special names.

(i) The proposition $q \rightarrow p$ is called the **converse** of $p \rightarrow q$.

(ii) The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$.

(iii) The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.

Example: What are the contrapositive, the converse, and the inverse of the conditional statement

"The home team wins whenever it is raining?"

Solution: Because "q whenever p" is one of the ways to express the conditional statement $p \rightarrow q$, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently, the contrapositive of this conditional statement is:

"If the home team does not win, then it is not raining."

The converse is: "If the home team wins, then it is raining."

The inverse is: "If it is not raining, then the home team does not win."

Note: We will see that of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

p	q	$p \rightarrow q$	$\neg q$	$\neg p$	$q \rightarrow p$	$\neg p \rightarrow \neg q$	$\neg q \rightarrow \neg p$
T	T	T	F	F	T	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

Equivalent Statements

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them **equivalent**. Hence, a conditional statement and its contrapositive are equivalent. Also, the converse and the inverse of a conditional statement are also equivalent, but neither is equivalent to the original conditional statement. The notation $r \equiv s$ denotes that r and s are logically equivalent.

Example: $p \rightarrow q \equiv \neg p \vee q$.

p	q	$p \rightarrow q$	$\neg p$	$\neg p \vee q$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Example: The proposition $p \leftrightarrow q$ and $(p \rightarrow q) \wedge (q \rightarrow p)$ are equivalent.

p	q	$p \leftrightarrow q$	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \wedge (q \rightarrow p)$
T	T	T	T	T	T
T	F	F	F	T	F
F	T	F	T	F	F
F	F	T	T	T	T

Example: A conditional statement and its contrapositive are equivalent. Also, the converse and the inverse of a conditional statement are also equivalent.

Example: **De Morgan's Laws.** $\neg(\mathbf{p} \wedge \mathbf{q}) \equiv \neg\mathbf{p} \vee \neg\mathbf{q}$; $\neg(\mathbf{p} \vee \mathbf{q}) \equiv \neg\mathbf{p} \wedge \neg\mathbf{q}$.

p	q	$\neg p$	$\neg q$	$p \wedge q$	$p \vee q$	$\neg(p \wedge q)$	$\neg p \vee \neg q$	$\neg(p \vee q)$	$\neg p \wedge \neg q$
T	T	F	F	T	T	F	F	F	F
T	F	F	T	F	T	T	T	F	F
F	T	T	F	F	T	T	T	F	F
F	F	T	T	F	F	T	T	T	T

Example: Use De Morgan's laws to express the negations of "Miguel has a cell phone and he has a laptop computer" and "Heather will go to the concert or Steve will go to the concert."

Solution: Let p be "Miguel has a cell phone" and q be "Miguel has a laptop computer." Then "Miguel has a cell phone and he has a laptop computer" can be represented by $p \wedge q$. By the first of De Morgan's laws, $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$. Consequently, we can express the negation of our original statement as "Miguel does not have a cell phone or he does not have a laptop computer." Let r be "Heather will go to the concert" and s be "Steve will go to the concert." Then "Heather will go to the concert or Steve will go to the concert" can be represented by $r \vee s$. By the second of De Morgan's laws, $\neg(r \vee s)$ is equivalent to $\neg r \wedge \neg s$. Consequently, we can express the negation of our original statement as "Heather will not go to the concert and Steve will not go to the concert." This can be simply written as "neither Heather nor Steve will go to the concert."

Note: The compound propositions r and s are logically equivalent if $r \leftrightarrow s$ is a tautology. The symbol ≡ is not a logical connective, and $r \equiv s$ is not a compound proposition but rather is the statement that $r \leftrightarrow s$ is a tautology. The symbol ⇔ is sometimes used instead of ≡ to denote logical equivalence.

Some Important Equivalence

In the following equivalences, T denotes the compound proposition that is always true and F denotes the compound proposition that is always false.

Equivalence	***Name***
$p \wedge T \equiv p$; $p \vee F \equiv p$	Identity laws
$p \vee T \equiv T$; $p \wedge F \equiv F$	Domination laws
$p \vee p \equiv p$; $p \wedge p \equiv p$	Idempotent laws

$p \vee \neg p \equiv T;\ p \wedge \neg p \equiv F$	Negation laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p;\ p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r);\ (p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (p \wedge q) \equiv p;\ p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r);\ p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q;\ \neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws

Some more Logical Equivalences

$p \to q \equiv \neg p \vee q$
$p \to q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Well-formed formula: A proposition formula is said to be well formed formula if it has the following properties:

1. Every atomic proposition is well-formed formula.
2. If p is wff, then $\neg p$ is also well-formed formula.
3. If p and q are well-formed formula, then $p \vee q, p \wedge q$ and $p \to q$ are well-formed formula.
4. Nothing else is well-formed formula.

Example: The proposition $(p \wedge q) \vee r$ is a well-formed-formula whereas $p \wedge q \vee r$ is not well-formed formula. To evaluate the formula $p \wedge q \vee r$ we can apply $\wedge$ first and then $\vee$ or apply $\vee$ first and then $\wedge$. That is we have two formulae $(p \wedge q) \vee r$ and $p \wedge (q \vee r)$.

p	q	r	$p \wedge q$	$(p \wedge q) \vee r$	$q \vee r$	$p \wedge (q \vee r)$
T	T	T	T	T	T	T

T	T	F	T	T	T	T
T	F	T	F	T	T	T
T	F	F	F	F	F	F
F	T	T	F	T	T	F
F	T	F	F	F	T	F
F	F	T	F	T	T	F
F	F	F	F	F	F	F

From the above truth table, it is clear that $(p \land q) \lor r$ and $p \land (q \lor r)$ have different truth values. So, they are not equivalent. So $p \land q \lor r$ is not well-formed formula.

Rules of Precedence: If a given formula is not well-formed formula, then we can convert it into a well-formed formula by using the order of Precedence of Logical Operators which is as follows:

Operators	**Precedence**
$\neg$	1
$\land$	2
$\lor$ $\oplus$	3
$\rightarrow$	4
$\leftrightarrow$	5

Functionally complete set of connectives: A set of connectives is called functionally complete if every compound proposition can be expressed as a logically equivalent proposition involving only these connectives.

Example: The sets $\{\neg, \land\}$, $\{\neg, \lor\}$ and $\{\neg, \land, \lor\}$ are functionally complete.

Satisfiable: A compound proposition is **satisfiable** if there is an assignment of truth values to its variables that makes it true (that is, when it is a tautology or a contingency). When no such assignment exists, that is, when the compound proposition is false for all assignments of truth values to its variables, the compound proposition is **unsatisfiable**. Note that a compound proposition is unsatisfiable if and only if its negation is true for all assignments of truth values to the variables, that is, if and only if its negation is a tautology.

Example: Determine whether each of the compound propositions is satisfiable.

(i) $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p)$.

(ii) $(p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$

(iii) $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p) \land (p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$

Solution:

(i) Note that $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p)$ is true when the three variables *p*, *q*, and *r* have the same truth value. Hence, it is satisfiable.

p	q	r	$\neg p$	$\neg q$	$\neg r$	$p \lor \neg q$	$q \lor \neg r$	$r \lor \neg p$	$(p \lor \neg q)$ $\land (q \lor \neg r)$ $\land (r \lor \neg p)$
T	T	T	F	F	F	T	T	T	T
T	T	F	F	F	T	T	T	F	F
T	F	T	F	T	F	T	F	T	F
T	F	F	F	T	T	T	T	F	F
F	T	T	T	F	F	F	T	T	F
F	T	F	T	F	T	F	T	T	F
F	F	T	T	T	F	T	F	T	F
F	F	F	T	T	T	T	T	T	T

Instead of using a truth table to solve this problem, we will reason about truth values.

(ii) Similarly, note that $(p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$ is true when at least one of *p*, *q*, and *r* is true and at least one is false. Hence, $(p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$ is satisfiable.

(iii) Finally, note that for $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p) \land (p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$ to be true, $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p)$ and $(p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$ must both be true. For the first to be true, the three variables must have the same truth values, and for the second to be true, at least one of the three variables must be true and at least one must be false. However, these conditions are contradictory. From these observations we conclude that no assignment of truth values to p, q, and r makes $(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p) \land (p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)$ true. Hence, it is unsatisfiable.

Some Applications of Propositional Logic

Logic has many important applications to mathematics, computer science, and numerous other disciplines. Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic. For example, logic is used in the specification of software and hardware, because these specifications need to be precise before development begins. Furthermore, propositional logic and its rules can be used to design computer circuits, to construct computer programs, to verify the correctness of programs, and to build expert systems. Logic can be used to analyze and solve many familiar puzzles. Software systems based on the rules of logic have been developed for constructing some, but not all, types of proofs automatically.

Translating English Sentences: There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives. In particular, English is often ambiguous and translating sentences into compound statements removes the ambiguity. Moreover, once we have translated sentences from English into logical expressions, we can analyze these logical expressions to determine their truth values and we can manipulate them.

System Specifications**:** Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems. System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development. System specifications should be **consistent**, that is, they should not contain conflicting requirements that could be used to derive a contradiction. When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

Example: Determine whether these system specifications are consistent:

“The diagnostic message is stored in the buffer or it is retransmitted.”

“The diagnostic message is not stored in the buffer.”

“If the diagnostic message is stored in the buffer, then it is retransmitted.”

Solution: To determine whether these specifications are consistent, we first express them using logical expressions. Let p denote “The diagnostic message is stored in the buffer” and let q denote “The diagnostic message is retransmitted.” The specifications can then be written as $p \vee$

q, $\neg p$, and $p \rightarrow q$. An assignment of truth values that makes all three specifications true must have p false to make $\neg p$ true. Because we want $p \vee q$ to be true but p must be false, q must be true. Because $p \rightarrow q$ is true when p is false and q is true, we conclude that these specifications are consistent, because they are all true when p is false and q is true. We could come to the same conclusion by use of a truth table to examine the four possible assignments of truth values to p and q.

p	q	$\neg p$	$p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	T	F
F	T	T	T	T
F	F	T	F	T

Logic Puzzles: Puzzles that can be solved using logical reasoning are known as **logic puzzles**. Solving logic puzzles is an excellent way to practice working with the rules of logic. Also, computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities. Many people enjoy solving logic puzzles, published in periodicals, books, and on the Web, as a recreational activity.

Example: A father tells his two children, a boy and a girl, to play in their backyard without getting dirty. However, while playing, both children get mud on their foreheads. When the children stop playing, the father says “At least one of you has a muddy forehead,” and then asks the children to answer “Yes” or “No” to the question: “Do you know whether you have a muddy forehead?” The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.

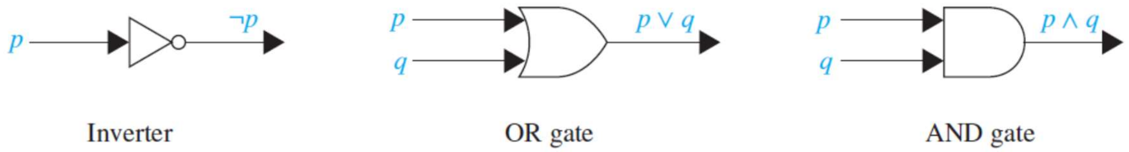
Solution: Let s be the statement that the son has a muddy forehead and let d be the statement that the daughter has a muddy forehead. When the father says that at least one of the two children has a muddy forehead, he is stating that the disjunction $s \vee d$ is true. Both children will answer “No” the first time the question is asked because each sees mud on the other child’s forehead. That is, the son knows that d is true, but does not know whether s is true, and the daughter knows that s is true, but does not know whether d is true. After the son has answered “No” to the first question, the daughter can determine that d must be true. This follows because when the first question is asked, the son knows that $s \vee d$ is true, but cannot determine whether s is true. Using this information, the daughter can conclude that d must be true, for if d were false, the son could have reasoned that because $s \vee d$ is true, then s must be true, and he would have answered "Yes" to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer "Yes" the second time the question is asked.

Example: An island has two kinds of inhabitants, knights, who always tell the truth, and their opposites, knaves, who always lie. You encounter two people *A* and *B*. What are *A* and *B* if *A* says "*B* is a knight" and *B* says "The two of us are opposite types?"

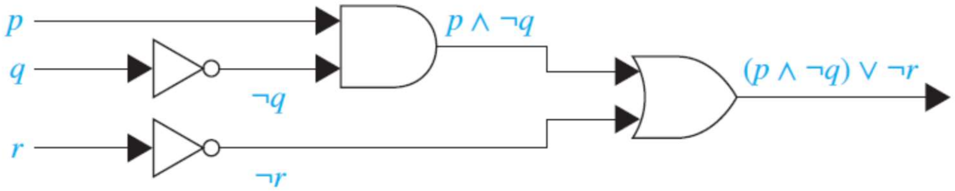
Solution*:* Let *p* and *q* be the statements that *A* is a knight and *B* is a knight, respectively, so that $\neg p$ and $\neg q$ are the statements that *A* is a knave and *B* is a knave, respectively. We first consider the possibility that *A* is a knight; this is the statement that p is true. If *A* is a knight, then he is telling the truth when he says that *B* is a knight, so that q is true, and *A* and *B* are the same type. However, if *B* is a knight, then *B*'s statement that *A* and *B* are of opposite types, the statement $(p \wedge \neg q) \vee (p \wedge q)$, would have to be true, which it is not, because *A* and *B* are both knights. Consequently, we can conclude that *A* is not a knight, that is, that *p* is false. If *A* is a knave, then because everything a knave says is false, *A*'s statement that *B* is a knight, that is, that q is true, is a lie. This means that q is false and *B* is also a knave. Furthermore, if *B* is a knave, then *B*'s statement that *A* and *B* are opposite types is a lie, which is consistent with both *A* and *B* being knaves. We can conclude that both *A* and *B* are knaves.

Boolean Searches: Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. Because these searches employ techniques from propositional logic, they are called **Boolean searches**. In Boolean searches, the connective *AND* is used to match records that contain both of two search terms, the connective *OR* is used to match one or both of two search terms, and the connective *NOT* (sometimes written as *AND NOT)* is used to exclude a particular search term. Careful planning of how logical connectives are used is often required when Boolean searches are used to locate information of potential interest.

Logic Circuits: Propositional logic can be applied to the design of computer hardware. A **logic circuit** (or **digital circuit**) receives input signals $p_1, p_2, \dots, p_n$, each a bit [either 0 (off) or 1 (on)], and produces output signals $s_1, s_2, \ldots, s_n$, each a bit. In this section we will restrict our attention to logic circuits with a single output signal; in general, digital circuits may have multiple outputs. Complicated digital circuits can be constructed from three basic circuits, called **gates**, as shown in the bellow figure.



The **inverter**, or **NOT gate**, takes an input bit p, and produces as output $\neg p$. The **OR gate** takes two input signals p and q, each a bit, and produces as output the signal $p \ \vee \ q$. Finally, the **AND gate** takes two input signals p and q, each a bit, and produces as output the signal $p \ \wedge \ q$. We use combinations of these three basic gates to build more complicated circuits, such as that shown in the bellow figure.



Rules of Inference for Propositional Logic

Proofs in mathematics are valid arguments that establish the truth of mathematical statements. By an **argument**, we mean a sequence of statements that end with a conclusion. By **valid argument**, we mean that the conclusion, or final statement of the argument, must follow from the truth of the preceding statements of the argument. To deduce new statements from statements we already have, we use some rules which are templates for constructing valid arguments. These rules are our basic tools for establishing the truth of statements and we call them rules of inference for propositional logic

Argument: An ***argument*** in propositional logic is a sequence of propositions. All but the final proposition in the argument are called ***premises*** and the final proposition is called the ***conclusion***. An argument is ***valid*** if all its premises are true implies that the conclusion is true. An ***argument form*** in propositional logic is a sequence of compound propositions involving propositional variables. An argument form is *valid* no matter which particular propositions are substituted for the propositional variables in its premises; the conclusion is true if the premises are all true.

To check the validity of an argument: Consider the argument with Premises $p,\ q$ and conclusion r. For a valid argument all premises are true implies that the conclusion is true. i.e. $p \wedge q$ is true implies r is true. i.e. $p \wedge q$ is true and r is true. Moreover, the argument is invalid only when $p \wedge q$ is true and r is false. So, argument is valid even if $p \wedge q$ is false and r is true or false. Thus, argument is valid if $(p \wedge q) \longrightarrow r$ is true irrespective of the truth values of p, q, r, i.e. $(p \wedge q) \longrightarrow r$ is a tautology.

Example: Consider the following argument involving propositions:

"If you have a current password, then you can log onto the network."

"You have a current password."

Therefore, "You can log onto the network."

We would like to determine whether this is a valid argument. That is, we would like to determine whether the conclusion "You can log onto the network" must be true when the premises "If you have a current password, then you can log onto the network" and "You have a current password" are both true. Before we discuss the validity of this particular argument, we will look at its form. Use p to represent "You have a current password" and q to represent "You can log onto the network." Then, the argument has the form

$$\begin{array}{l} p \rightarrow q \\ \underline{p \qquad\quad} \\ \therefore\ q; \end{array}$$

where ∴ is the symbol that denotes "therefore."

This can be written as $((p \rightarrow q) \wedge p) \rightarrow q$ and we have to show it is a tautology. This can be done by truth table as below:

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$((p \rightarrow q) \wedge p)) \rightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

In particular, when both $p \rightarrow q$ and p are true, we know that q must also be true. We say this form of argument is **valid** because whenever all its premises are true, the conclusion must also be true. Now suppose that both “If you have a current password, then you can log onto the network” and “You have a current password” are true statements. When we replace p by “You have a current password” and q by “You can log onto the network,” it necessarily follows that the conclusion “You can log onto the network” is true. This argument is valid because its form is valid. Note that whenever we replace p and q by propositions where $p \rightarrow q$ and p are both true, then q must also be true.

Note: What happens when we replace p and q in this argument form by propositions where not both p and $p \rightarrow q$ are true? For example, suppose that p represents “You have access to the network” and q represents “You can change your grade” and that p is true, but $p \rightarrow q$ is false. The argument we obtain by substituting these values of p and q into the argument form is

“If you have access to the network, then you can change your grade.”
“You have access to the network.”
____
∴ “You can change your grade.”

The argument we obtained is a valid argument, because one of the premises, namely the first premise, is false, we cannot conclude that the conclusion is true.

Note: We can always use a truth table to show that an argument form is valid. We do this by showing that whenever the premises are true, the conclusion must also be true. However, this can be a tedious approach. For example, when an argument form involves 10 different propositional variables, to use a truth table to show this argument form is valid, requires $2^{10} = 1024$ different rows. Fortunately, we do not have to resort to truth tables. Instead, we can first establish the validity of some relatively simple argument forms, called **rules of inference**. These rules of inference can be used as building blocks to construct more complicated valid argument forms.

Example: The tautology $(p \wedge (p \rightarrow q)) \rightarrow q$ is the basis of the rule of inference called **modus ponens**, or the **law of detachment**. This tautology leads to the following valid argument form,

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

Example: Suppose that the conditional statement “If it snows today, then we will go skiing” and its hypothesis, “It is snowing today,” are true. Then, by modus ponens, it follows that the conclusion of the conditional statement, “We will go skiing,” is true.

There are many useful rules of inference for propositional logic.

No.	Rule of Inference	Tautology	Name
1	p $\underline{p \rightarrow q}$ $\therefore\ q$	$(p \wedge (p \rightarrow q)) \rightarrow q$	Modus ponens
2	$\neg q$ $\underline{p \rightarrow q}$ $\therefore\ \neg p$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$	Modus tollens
3	$p \rightarrow q$ $\underline{q \rightarrow r}$ $\therefore\ p \rightarrow r$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	Hypothetical syllogism
4	$p \vee q$ $\underline{\neg p}$ $\therefore\ q$	$((p \vee q) \wedge \neg p) \rightarrow q$	Disjunctive syllogism
5	$p \vee q$ $\underline{\neg p \vee r}$ $\therefore\ q \vee r$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$	Resolution
6	$\underline{p}$ $\therefore\ p \vee q$	$p \rightarrow (p \vee q)$	Addition
7	$\underline{p \wedge q}$ $\therefore\ p$	$(p \wedge q) \rightarrow p$	Simplification
8	p $\underline{q}$ $\therefore\ p \wedge q$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction

Example: State which rule of inference is the basis of the following argument?

“It is below freezing now. Therefore, it is either below freezing or raining now.”

Solution: Addition.

Example: State which rule of inference is the basis of the following argument?

“It is below freezing and raining now. Therefore, it is below freezing now.”

Solution: Simplification.

Example: State which rule of inference is used in the argument:

"If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow."

Solution: Hypothetical syllogism.

Example: Show that the premises

"It is not sunny this afternoon and it is colder than yesterday,"

"We will go swimming only if it is sunny,"

"If we do not go swimming, then we will take a canoe trip," and

"If we take a canoe trip, then we will be home by sunset"

lead to the conclusion

"We will be home by sunset."

Solution: Let p be the proposition "It is sunny this afternoon," q the proposition "It is colder than yesterday," r the proposition "We will go swimming," s the proposition "We will take a canoe trip," and t the proposition "We will be home by sunset." Then the premises become $\neg p \land q, r \to p, \neg r \to s$, and $s \to t$. The conclusion is simply t. We need to show, it is a valid argument with premises (i) $\neg p \land q$, (ii) $r \to p$, (iii) $\neg r \to s$, and (iv) $s \to t$ and conclusion t. We construct an argument to show that our premises lead to the desired conclusion as follows.

Step 1

$\neg p \land q$	Premise (i)
$\therefore \neg p$	simplification

Step 2

$r \to p$	Premise (ii)
$\neg p$	Step 1
$\therefore \neg r$	Modus tollens

Step 3

$\neg r \to s$	Premise (iii)
$\neg r$	step 2
$\therefore s$	Modus ponens

Step 4

$s \to t$	Premise (iv)
s	step 3
$\therefore t$	Modus ponens

Therefore, the argument is valid.

Note: we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, $p, q, r, s,$ and t , such a truth table would have 32 rows.

Example: Show that the premises "If you send me an e-mail message, then I will finish writing the program," "If you do not send me an e-mail message, then I will go to sleep early," and "If I go to sleep early, then I will wake up feeling refreshed" lead to the conclusion "If I do not finish writing the program, then I will wake up feeling refreshed."

Solution: Let p be the proposition "You send me an e-mail message," q the proposition "I will finish writing the program," r the proposition "I will go to sleep early," and s the proposition "I will wake up feeling refreshed." Then the premises are (i) $p \rightarrow q$, (ii)$\neg p \rightarrow r$, and (iii) $r \rightarrow s$. The desired conclusion is $\neg q \rightarrow s$. We need to give a valid argument with premises $p \rightarrow q$, $\neg p \rightarrow r$,and $r \rightarrow s$ and conclusion $\neg q \rightarrow s$. This argument form shows that the premises lead to the desired conclusion.

Step 1

$$\begin{array}{ll} p \rightarrow q & \text{Premise (i)} \\ \hline \therefore \neg q \rightarrow \neg p & \text{Contrapositive} \end{array}$$

Step 2

$$\begin{array}{ll} \neg q \rightarrow \neg p & \text{step 1} \\ \neg p \rightarrow r & \text{premise (ii)} \\ \hline \therefore \neg q \rightarrow r & \text{Hypothetical syllogism} \end{array}$$

Step 3

$$\begin{array}{ll} \neg q \rightarrow r & \text{step 2} \\ r \rightarrow s & \text{premise (iii)} \\ \hline \therefore \neg q \rightarrow s & \text{Hypothetical syllogism} \end{array}$$

This argument form shows that the premises lead to the desired conclusion.

Note: Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as **resolution**. This rule of inference is based on the tautology $((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$. The final disjunction in the resolution rule, $q \vee r$, is called the **resolvent**. When we let $r = q$ in this tautology, we obtain $(p \vee q) \wedge (\neg p \vee q) \rightarrow q$. Furthermore, when we let $r = \boldsymbol{F}$, we obtain $(p \vee q) \wedge (\neg p) \rightarrow q$ (because $q \vee \boldsymbol{F} \equiv q$), which is the tautology on which the rule of disjunctive syllogism is based.

Example: Use resolution to show that the hypotheses "Jasmine is skiing or it is not snowing" and "It is snowing or Bart is playing hockey" imply that "Jasmine is skiing or Bart is playing hockey."

Fallacies: Several common fallacies (mistaken believes) arise in incorrect arguments. These fallacies resemble rules of inference, but are based on contingencies rather than tautologies.

(i) The proposition $((p \to q) \land q) \to p$ is not a tautology, because it is false when p is false and q is true. However, there are many incorrect arguments that treat this as a tautology. In other words, they treat the argument with premises $p \to q$ and q and conclusion p as a valid argument form, which it is not. This type of incorrect reasoning is called the **fallacy of affirming the conclusion**.

(ii) The proposition $\big((p \to q) \land \neg p\big) \to \neg q$ is not a tautology, because it is false when p is false and q is true. Many incorrect arguments use this incorrectly as a rule of inference. This type of incorrect reasoning is called the **fallacy of denying the hypothesis.**

Example: Is the following argument valid?

"If you do every problem in this book, then you will learn discrete mathematics."

"You learned discrete mathematics."

"Therefore, you did every problem in this book."

Solution: Let p be the proposition "You did every problem in this book." Let q be the proposition "You learned discrete mathematics." Then this argument is of the form: if $p \to q$ and q, then p. This is not a valid argument and is an example of an incorrect argument using the fallacy of affirming the conclusion. Indeed, it is possible for you to learn discrete mathematics in some way other than by doing every problem in this book. (You may learn discrete mathematics by reading, listening to lectures, doing some, but not all, the problems in this book, and so on.) This is the case of **fallacy of affirming the conclusion.**

Example: Is the following argument valid?

"If you do every problem in this book, then you will learn discrete mathematics."

"You did not do every problem in this book."

"Therefore, you did not learn discrete mathematics."

Solution: This is not a valid argument. It is possible that you learned discrete mathematics even if you did not do every problem in this book. This is the case of **fallacy of denying the hypothesis.**

Predicates and Quantifiers

Propositional logic 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.”

And “MATH3 is one of the computers connected to the university network.

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

“MATH3 is functioning properly,”

Likewise, we cannot use the rules of propositional logic to conclude from the statement

“CS2 is under attack by an intruder,”

and “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.”

A more powerful type of logic called **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 is the part of a sentence or clause containing a verb and stating something about the subject. 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.

Example: 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*(CS1), *A*(CS2), and *A*(MATH1)?

Solution: We obtain the statement *A*(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*(CS1) is false. Similarly, because CS2 and MATH1 are on the list of computers under attack, we say that *A*(CS2) and *A*(MATH1) are true.

Example: 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.

Quantifier is a determiner or pronoun indicative of quantity. 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**:** 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.

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

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

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

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

Solution: Because $P(x)$ is true for all real numbers x, the quantification $\forall x\ P(x)$ is true.

Example: Let $Q(x)$ be the statement “$x < 2$.” What is the truth value of the quantification $\forall x\ Q(x)$, 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 $\forall x\ Q(x)$. Thus $\forall x\ Q(x)$ is false.

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

Solution*:* The statement $\forall x\ P(x)$ 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 $\forall x\ P(x)$ is false.

Example: 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.

“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.”

Again, 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.

Note: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. Note that if the domain is empty, then $\forall x\ P(x)$ 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." 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: "There is not any reason to avoid studying." A statement $\forall x\ P(x)$ 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 $\forall x\ P(x)$. Note that a single counterexample is all we need to establish that $\forall x\ P(x)$ is false. Looking for counterexamples to universally quantified statements is an important activity in the study of mathematics.

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.

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

Note: A domain must always be specified when a statement $\exists x\ P(x)$ is used. Furthermore, the meaning of $\exists x\ P(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists x\ P(x)$ 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 $\exists x\ P(x)$ is read as "There is an x such that $P(x)$," "There is at least one x such that $P(x)$," "For some $x\ P(x)$." Observe that the statement $\exists x\ P(x)$ is false if and only if there is no element x in the domain for which $P(x)$ is true. That is, $\exists x\ P(x)$ is false if and only if $P(x)$ is false for every element of the domain. Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. If the domain is empty, then $\exists x\ P(x)$ is false whenever $P(x)$ is a propositional function because when the domain is empty, there can be no element x in the domain for which $P(x)$ is true.

Example: Let $P(x)$ denote the statement "$x > 3$." What is the truth value of the quantification $\exists x\ P(x)$, 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 $\exists x\ P(x)$, is true.

Example: Let $Q(x)$ denote the statement "$x = x + 1$." What is the truth value of the quantification $\exists x\ Q(x)$, 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 $\exists x\ Q(x)$, is false.

Example: Express the statements "Some student in this class has visited 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 $\exists x\ M(x)$.

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

Quantifiers Over Finite Domains

When the domain of a quantifier is finite, that is, when all its elements can be listed, quantified statements can be expressed using propositional logic. In particular, when the elements of the domain are $x_1, x_2, \dots, x_n$, where n is a positive integer, the universal quantification $\forall x\, P(x)$ 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 $P(x_1),\ P(x_2), \cdots, P(x_n)$ are all true. Similarly, the existential quantification $\exists x\, P(x)$ is the same as the disjunction

$$P(x_1) \vee\ P(x_2) \vee \cdots \vee\ P(x_n).$$

Quantifiers with Restricted Domains: Consider the statement "For every natural number x, $x^2 < 100$." In this case the domain of discourse is the set of all-natural numbers $\{1,2,3,\dots\}$. The given statement is $\forall x\, P(x)$, where $P(x)$ is the statement "$x^2 < 100$" and it is false as $P(11)$ is false. But if we consider our domain of discourse to be the set $A = \{1,2,3,4,5,6,7,8,9,10\}$, then $x^2 < 100$ is true. So, if we change the domain of a quantifier the truth value of proposition may be changed. Let $Q(x)$ be the proposition " x is in A." Then the proposition "For every x in A, $x^2 < 100$" is written as $\forall x\, (Q(x) \to P(x))$, where the domain of discourse may be the set of all-natural numbers or real number. Note that the proposition "For every x in A, $x^2 < 100$" is equivalent to "For all x, if it is in A, then $x^2 < 100$." An abbreviated notation is often used to restrict the domain of a quantifier. In this notation, a condition satisfied by the variable must be included after the quantifier. So, the other way of writing the statement $\forall x\, (Q(x) \to P(x))$ is $\forall x \in A\ \ P(x)$.

The restricted domain statement $\exists x \in A\, P(x)$ reads, "There exists some x in A, where the predicate $P(x)$ holds". This is equivalent to saying, "There exists some x, that is in A and the predicate $P(x)$ holds." Which is $\exists x\, (x \in A\ \wedge\ P(x))$.

Example: 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 $\forall x\, (x\ <\ 0\ \to\ x^2\ >\ 0)$. 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 $\forall y\, (y\ \neq\ 0\ \to\ y^3\ \neq\ 0)$. 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: The restriction of a universal quantification is the same as the universal quantification of a conditional statement. The quantifiers ∀ and ∃ have higher precedence than all logical operators from propositional calculus. For example, $\forall x\, P(x) \vee Q(x)$ is the disjunction of $\forall x\, P(x)$ and $Q(x)$. In other words, it means $(\forall x\, P(x)) \vee Q(x)$ rather than $\forall x\ (P(x) \vee Q(x))$.

Precedence of Quantifiers

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

Logical Equivalences Involving Quantifiers: 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.

Example: Show that $\forall x\ (P(x) \wedge Q(x))$ and $\forall x\ P(x) \wedge \forall x Q(x)$ are logically equivalent.

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 $\forall x\ P(x) \wedge \forall x\ Q(x)$ are logically equivalent by doing two things. First, we show that if $\forall x\ (P(x) \wedge Q(x))$ is true, then $\forall x\ P(x) \wedge \forall x\ Q(x)$ is true. Second, we show that if $\forall x\ P(x) \wedge \forall x\ Q(x)$ 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 $\forall x P(x)$ and $\forall x\ Q(x)$ are both true. This means that $\forall x\ P(x) \wedge \forall x\ Q(x)$ is true. Next, suppose that $\forall x\ P(x) \wedge \forall x\ Q(x)$ is true. It follows that $\forall x\ P(x)$ is true and $\forall x\ Q(x)$ 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 $\forall x\,\big(P(x) \wedge Q(x)\big) \equiv \forall x\,P(x) \wedge \forall x\,Q(x)$.

Note: 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.

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,

$$\forall x\,P(x),$$

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: $\neg\big(\forall x\,P(x)\big) \equiv\ \exists x\,(\neg P(x))$.

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 $\exists x\,\big(\neg P(x)\big)$ is true. Putting these steps together, we can conclude that $\neg(\forall x\,P(x))$ is true if and only if $\exists x\,\big(\neg P(x)\big)$ is true. It follows that $\neg\big(\forall x\,P(x)\big) \equiv\ \exists x\,(\neg P(x))$.

Similarly, we have 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. Thus, we conclude that $\neg\exists x\ Q(x)$ and $\forall x\ \neg Q(x)$are logically equivalent.

The rules for negations for quantifiers are called **De Morgan's laws for quantifiers.**

Negations	Equivalent statement	When is Negation true?	When is Negation false?
$\exists x\ Q(x)$	$\forall x\ \neg Q(x)$	For every x, $Q(x)$ is false.	There is an x for which $Q(x)$ is true
$\neg\forall x\ Q(x)$	$\exists x\ \neg Q(x)$	There is an x for which $Q(x)$ is false	For every x, $Q(x)$ is true.

Example: 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."

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 $\neg\forall x\ C(x)$, 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."

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.

Using Quantifiers in System Specifications: Previously, we used propositions to represent system specifications. However, many system specifications involve predicates and quantifications.

Example: Use predicates and quantifiers to express the system specifications

"All lions are fierce."

"Some lions do not drink coffee."

"Some fierce creatures do not drink coffee."

Solution: Let $P(x)$, $Q(x)$, and $R(x)$ be the statements "x is a lion," "x is fierce," and "x drinks coffee," respectively. We assume that the domain consists of all creatures. 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 $\exists x\ (Q(x) \rightarrow \neg R(x))$.

Rules of Inference for Quantified Statements: We have discussed rules of inference for propositions. We will now describe some important rules of inference for statements involving quantifiers. These rules of inference are used extensively in mathematical arguments, often without being explicitly mentioned.

Universal instantiation is the rule of inference used to conclude that $P(c)$ is true, where c is a particular member of the domain, given the premise $\forall x\ P(x)$. Universal instantiation is used when we conclude from the statement "All women are wise" that "Lisa is wise," where Lisa is a member of the domain of all women.

Universal generalization is the rule of inference that states that $\forall x\ P(x)$ is true, given the premise that $P(c)$ is true for all elements *c* in the domain. Universal generalization is used when we show that $\forall x\ P(x)$ is true by taking an arbitrary element *c* from the domain and showing that $P(c)$ is true. The element *c* that we select must be an arbitrary, and not a specific, element of the domain. That is, when we assert from $\forall x\ P(x)$ the existence of an element *c* in the domain, we have no control over *c* and cannot make any other assumptions about c other than it comes from the domain. Universal generalization is used implicitly in many proofs in mathematics and is seldom mentioned explicitly. However, the error of adding unwarranted assumptions about the arbitrary element c when universal generalization is used is all too common in incorrect reasoning.

Existential instantiation is the rule that allows us to conclude that there is an element c in the domain for which $P(c)$ is true if we know that $\exists x\ P(x)$ is true. We cannot select an arbitrary value of *c* here, but rather it must be a *c* for which $P(c)$ is true. Usually we have no knowledge of what c is, only that it exists. Because it exists, we may give it a name 'c'and continue our argument.

Existential generalization is the rule of inference that is used to conclude that $\exists x\ P(x)$ is true when a particular element c with $P(c)$ true is known. That is, if we know one element c in the domain for which *P(c)* is true, then we know that $\exists x\ P(x)$ is true.

Rules of Inference for Quantified Statements.	
Rule of Inference	***Name***
$\forall x\ P(x)$ $\therefore\ P(c)$	Universal instantiation
$P(c)$ for an arbitrary *c* $\therefore\ \forall x\ P(x)$	Universal generalization
$\exists x\ P(x)$ $\therefore\ P(c)$ for some element *c*	Existential instantiation
$P(c)$ for some element *c* $\therefore\ \exists x\ P(x)$	Existential generalization

Example: Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Maria is a student in this class” imply the conclusion “Maria has taken a course in computer science.”

Solution: Let $D(x)$ denote "x is in this discrete mathematics class," and let $C(x)$ denote "x has taken a course in computer science." Then the premises are (i) $\forall x\ (D(x)\ \rightarrow\ C(x))$ and (ii) *D*(Maria). The conclusion is *C*(Maria). The following steps can be used to establish the conclusion from the premises.

Step 1	**Reason**
$\forall x(D(x)\ \rightarrow\ C(x))$	Premise (i)
∴ *D*(Maria)→*C*(Maria)	Universal instantiation

Step 2	**Reason**
D(Maria)→*C*(Maria)	step 1
D(Maria)	Premise (ii)
∴ *C*(Maria)	Modus ponens

Example: Show that the premises "A student in this class has not read the book," and "Everyone in this class passed the first exam" imply the conclusion "Someone who passed the first exam has not read the book."

Solution*:* Let $C(x)$ be "x is in this class," $B(x)$ be "x has read the book," and $P(x)$ be "x passed the first exam." The premises are (i) $\exists x\ (C(x) \wedge \neg B(x))$ and (ii) $\forall x\ (C(x)\ \rightarrow\ P(x))$. The conclusion is $\exists x\ (P(x) \wedge \neg B(x))$. These steps can be used to establish the conclusion from the premises.

Step 1	**Reason**
$\exists x(C(x) \wedge \neg B(x))$	Premise (i)
$C(a) \wedge \neg B(a)$	Existential instantiation

Step 2	**Reason**
$C(a) \wedge \neg B(a)$	step 1
C(a)	Simplification

Step 3	**Reason**
$C(a) \wedge \neg B(a)$	step 1
$\neg B(a)$	Simplification

Step 4	**Reason**
$\forall x(C(x) \rightarrow P(x))$	Premise (ii)
$C(a) \rightarrow P(a)$	Universal instantiation

Step 5	**Reason**
$C(a) \rightarrow P(a)$	step 4
C(a)	*step 2*
P(a)	Modus ponens from (3) and (5)

Step 6	**Reason**
P(a)	*step 5*
$\neg B(a)$	step 3
$P(a) \wedge \neg B(a)$	Conjunction

Step 7	**Reason**
$P(a) \wedge \neg B(a)$	step 6
$\exists x(P(x) \wedge \neg B(x))$	Existential generalization

Combining Rules of Inference for Propositions and Quantified Statements: We have developed rules of inference both for propositions and for quantified statements. Sometimes we use both a rule of inference for quantified statements, and a rule of inference for propositional logic. Here we give some of them

(i) **Universal modus ponens** rule tells us that if $\forall x\ (P(x) \rightarrow Q(x))$ is true, and if $P(a)$ is true for a particular element *a* in the domain of the universal quantifier, then $Q(a)$ must also be true. i.e.

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

$P(a)$, *where* a is a particular element in the domain

____

$\therefore Q(a)$

(ii) **Universal modus tollens** combines universal instantiation and modus tollens and can be expressed in the following way:

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

$\neg Q(a)$, where a is a particular element in the domain

$\therefore \neg P(a)$

Mathematical Induction

Now we introduce a notion of proof and describe methods for constructing proofs. A proof is a valid argument that establishes the truth of a mathematical statement. A proof can use the hypotheses of the theorem, if any, axioms assumed to be true and previously proven theorems. Using these ingredients and rules of inference, the final step of the proof establishes the truth of the statement being proved. There are two types of proofs, formal proofs and informal proofs. The proof that an argument is true is **formal proof**, where all steps were supplied, and the rules for each step in the argument were given. However, formal proofs of useful theorems can be extremely long and hard to follow. In practice, the proofs of theorems designed for human consumption are almost always **informal proofs**, where more than one rule of inference may be used in each step, where steps may be skipped, where the axioms being assumed and the rules of inference used are not explicitly stated. Informal proofs can often explain to humans why theorems are true, while computers are perfectly happy producing formal proofs using automated reasoning systems. When you read proofs, you will often find the words "obviously" or "clearly." These words indicate that steps have been omitted, however we will try to avoid using these words and try not to omit too many steps. There are many proof techniques that can be used to prove a wide variety of theorems. For example: direct proof, proof by contraposition, vacuous and trivial proofs, proofs by contradiction, proofs of equivalence, counterexamples, mathematical induction. A major goal of this section is to provide a thorough understanding of mathematical induction. In this section, we will describe how mathematical induction can be used and why it is a valid proof technique. It is extremely important to note that mathematical induction can be used only to prove results obtained in some other way. It is not a tool for discovering formulae or theorems.

Many mathematical statements assert that a property is true for all positive integers n with $n \geq b$. Mathematical induction can be used to prove this type of statements $P(n),\ n \geq b$. Mathematical induction can be used to prove a tremendous variety of results. Understanding how to read and construct proofs by mathematical induction is a key goal of learning discrete mathematics.

Mathematical induction is based on the rule of inference that tells us that if $P(b)$ and $\forall k > b\ (P(k) \rightarrow P(k+1))$ are true for the domain of positive integers, then $\forall n \geq b\ P(n)$ is true. i.e.

$$\begin{array}{l} \forall k > b\ \big(P(k) \rightarrow P(k+1)\big) \\ \underline{P(b) \qquad\qquad\quad} \\ \therefore \forall n \geq b\ P\ (n) \end{array}$$

Proofs using mathematical induction have two parts. First, **basis step,** we show that the statement holds for the positive integer b. Second, **inductive step,** we show that if the statement holds for a positive integer $> b$ then it must also hold for the next larger integer. To complete the inductive step of a proof using the principle of mathematical induction, we assume that $P(k)$ is true for an arbitrary positive integer k and show that under this assumption, $P(k+1)$ must also be true. The assumption that $P(k)$ is true is called the **inductive hypothesis**. Once we complete both steps in a proof by mathematical induction, we conclude that $P(n)$ is true for all positive integers $> b$, that is, we have shown that $\forall n \geq b\ P\ (n)$ is true.

Example: Show that if n is a positive integer, then

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Solution: Let $P(n)$ be the proposition that the sum of the first n positive integers, is $\frac{n(n+1)}{2}$. We must do two things to prove that $P(n)$ is true for $n = 1, 2, 3, \ldots .$ Namely, we must show that $P(1)$ is true and that the conditional statement $P(k)$ implies $P(k+1)$ is true for $k > 1$.

Basis step: $P(1)$ is true, because $1 = \frac{1(1+1)}{2}$.

(The left-hand side of this equation is 1 because 1 is the sum of the first positive integer. The right-hand side is found by substituting 1 for n in $\frac{n(n+1)}{2}$.)

Inductive step: For the inductive hypothesis we assume that $P(k)$ holds for an arbitrary positive integer k. That is, we assume that

$$1 + 2 + \cdots + k = \frac{k(k + 1)}{2}.$$

Under this assumption, it must be shown that $P(k + 1)$ is true, namely, that

$$1 + 2 + \cdots + k + (k + 1) = \frac{(k + 1)[(k + 1) + 1]}{2} = \frac{(k + 1)(k + 2)}{2}$$

is also true. When we add $k + 1$ to both sides of the equation in $P(k)$, we obtain

$$\begin{aligned} 1 + 2 + \cdots + k + (k + 1) &= \frac{k(k + 1)}{2} + (k + 1) \\ &= \frac{k(k + 1) + 2(k + 1)}{2} \\ &= \frac{(k + 1)(k + 2)}{2}. \end{aligned}$$

This last equation shows that $P(k + 1)$ is true under the assumption that $P(k)$ is true. This completes the inductive step.

We have completed the basis step and the inductive step, so by mathematical induction we know that $P(n)$ is true for all positive integers n. That is, we have proven that

$$1 + 2 + \cdots + n = \frac{n(n + 1)}{2}$$

for all positive integers n.

Example: Conjecture a formula for the sum of the first *n* positive odd integers. Then prove your conjecture using mathematical induction.

Example: Use mathematical induction to show that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$

Example: Use mathematical induction to prove that $n^3 - n$ is divisible by 3 whenever n is a positive integer.

Example: Use mathematical induction to prove that $7^{n+2} + 8^{2n+1}$ is divisible by 57 for every nonnegative integer n.

Example: If $a_n = 2^n + 3^n$ show that $a_n = 5a_{n-1} - 6a_{n-2}$, for $n \geq 2$ $(a_0 = 1$ and $a_1 = 5)$.

Example: The **harmonic numbers** $H_j, j = 1, 2, 3, \ldots$, are defined by

$$H_j = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{j}.$$

Use mathematical induction to show that

$$H_{2^n} \geq 1 + \frac{n}{2}$$

whenever n is a nonnegative integer.

Strong Induction: There is another form of mathematical induction, called strong induction, which can often be used when we cannot easily prove a result using mathematical induction. The basis step of a proof by strong induction is the same as a proof of the same result using mathematical induction. That is, in a strong induction proof that $P(n)$ is true for all positive integers $n \geq b$, the basis step shows that $P(b)$ is true. However, the inductive steps in these two proof methods are different. In a proof by mathematical induction, the inductive step shows that if the inductive hypothesis $P(k)$ is true, then $P(k + 1)$ is also true. In a proof by strong induction, the inductive step shows that if $P(j)$ is true for all positive integers j not exceeding k, then $P(k + 1)$ is true. That is, for the inductive hypothesis we assume that $P(j)$ is true for $j = b, b+1, , \ldots, k$. Thus to prove that $P(n)$ is true for all positive integers $n \geq b$, where $P(n)$ is a propositional function, we complete two steps:

Basis step: We verify that the proposition $P(b)$ is true.

Inductive step: We show that the conditional statement $[P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)] \rightarrow P(k + 1)$ is true for all positive integers $k > b$.

Example: Given that $d_1 = 1, d_2 = 2, d_3 = 3$, $d_{n+3} = d_{n+2} + d_{n+1} + d_n$for all positive integer n. Show by method of Strong Induction that $d_n < 2^n$.

Solution: Given $d_1 = 1, d_2 = 2, d_3 = 3$, and for all positive integer n,

$$d_{n+3} = d_{n+2} + d_{n+1} + d_n.$$

To prove that, $d_n < 2^n$, for all $n \in \mathbb{N}$ by method of strong induction.

Step 1. To prove $d_n < 2^n$ for all $n \in \mathbb{N}$, let $P(n)$: $d_n < 2^n$.

Step 2. (Basis Step) Given that $d_1 = 1, d_2 = 2, d_3 = 3$.

Also $d_1 = 1 < 2^1, d_2 = 2 < 2^2, d_3 = 3 < 2^3$.

Therefore $P(1)$, $P(2)$ and $P(3)$ are true.

Step 3. (Inductive Step) Assume that $P(4), P(5), \cdots, P(k)$ are true for some $k > 3$. We have to show that the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k + 1)$ is true. i.e $P(k+1)$ is true.

Now, $P(k+1)$: $d_{k+1} < 2^{k+1}$. Thus, we have to show $d_{k+1} < 2^{k+1}$.

As $P(k)$, $P(k-1)$ and $P(k-2)$ are true,

$$d_k < 2^k, d_{k-1} < 2^{k-1}, d_{k-2} < 2^{k-2}.$$

Given that

$$d_{n+3} = d_{n+2} + d_{n+1} + d_n.$$

So

$$d_{k+1} = d_k + d_{k-1} + d_{k-2} < 2^k + 2^{k-1} + 2^{k-2} = 2^k\left(1 + \frac{1}{2} + \frac{1}{4}\right) < 2^k \times 2 = 2^{k+1}.$$

i.e. $d_{k+1} < 2^{k+1}$. i.e. $P(k+1)$ is true.

Thus, the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ is true.

Step 4. (Conclusion) As we have already shown $P(1)$, $P(2)$ and $P(3)$ are true, and the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ is true, by method of strong induction $P(n)$ is true for all $n \in \mathbb{N}$. That is $d_n < 2^n$ for all $n \in \mathbb{N}$.

Example: Given a_0=2,$a_1 = 7$, and for all $n \geq 2$,$a_n = 5a_{n-1} - 6a_{n-2}$. Prove that, $a_n = 3^{n+1} - 2^n$ for all $n \in \mathbb{N}$ by method of strong induction.

Solution: Given a_0=2,$a_1 = 7$, and for all $n \geq 2$,

$$a_n = 5a_{n-1} - 6a_{n-2}.$$

To prove that, $a_n = 3^{n+1} - 2^n$, for all $n \in \mathbb{N}$ by method of strong induction.

Step 1. *To prove* $a_n = 3^{n+1} - 2^n$ for all $n \in \mathbb{N}$, let $P(n)$: $a_n = 3^{n+1} - 2^n$.

Step 2.(Basis Step) Given that a_0=2,$a_1 = 7$. Also $a_0 = 3^{0+1} - 2^0 = 2$ and $a_1 = 3^{1+1} - 2^1 = 7$.

Now for all $n \geq 2$, $a_n = 5a_{n-1} - 6a_{n-2}$.

So, $a_2 = 5a_1 - 6a_0 = 5 \times 7 - 6 \times 2 = 23$. Also $a_2 = 3^{2+1} - 2^2 = 23$.

Therefore $P(0)$, $P(1)$ and $P(2)$ are true.

Step 3. (Inductive Step) Assume that $P(1), P(2), \cdots, P(k)$ are true for some $k \in \mathbb{N}$. We have to show that the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ is true. i.e $P(k+1)$ is true.

Now, $P(k+1)$: $a_{k+1} = 3^{k+1+1} - 2^{k+1} = 3^{k+2} - 2^{k+1}$.

Thus, we have to show $a_{k+1} = 3^{k+2} - 2^{k+1}$.

Given that

$$a_n = 5a_{n-1} - 6a_{n-2}.$$

Therefore, $a_{k+1} = 5a_k - 6a_{k-1}$.

As $P(k)$ and $P(k-1)$ are true, $a_k = 3^{k+1} - 2^k$ and $a_{k-1} = 3^k - 2^{k-1}$.

Thus,

$$\begin{aligned} a_{k+1} &= 5a_k - 6a_{k-1} \\ &= 5(3^{k+1} - 2^k) - 6(3^k - 2^{k-1}) \\ &= 15 \times 3^k - 10 \times 2^{k-1} - 6 \times 3^k + 6 \times 2^{k-1} \\ &= (15-6)3^k - (10-6)2^{k-1} \\ &= 9 \times 3^k - 4 \times 2^{k-1} \\ &= 3^{k+2} - 2^{k+1} \end{aligned}$$

i.e.

$$a_{k+1} = 3^{k+2} - 2^{k+1}.$$

i.e. $P(k+1)$ is true.

Thus, the conditional statement $[P(1) \land P(2) \land \cdots \land P(k)] \rightarrow P(k+1)$ is true.

Step 4. (Conclusion) As we have already shown $P(0), P(1)$ and $P(2)$ are true, and the conditional statement $[P(1) \land P(2) \land \cdots \land P(k)] \rightarrow P(k+1)$ is true, by method of strong induction $P(n)$ is true for all $n \in \mathbb{N}$. That is $a_n = 3^{n+1} - 2^n$ for all $n \in \mathbb{N}$.

Unit II---- Set, Relation and Function

Set

A *set* is an unordered collection of objects, called *elements* or *members* of the set. A set is said to *contain* its elements. We write $a \in A$ to denote that a is an element of the set A. The notation $a \notin A$ denotes that a is not an element of the set A.

It is common for sets to be denoted using uppercase letters. Lowercase letters are usually used to denote elements of sets. There are several ways to describe a set. One way is to list all the members of a set, when this is possible. We use a notation where all members of the set are listed between braces. For example, the notation $\{a, b, c, d\}$ represents the set with the four elements a, b, c, and d. This way of describing a set is known as the **roster method**.

Example: The set V of all vowels in the English alphabet can be written as $V = \{\mathrm{a}, \mathrm{e}, \mathrm{i}, \mathrm{o}, \mathrm{u}\}$.

Example: The set O of odd positive integers less than 10 can be expressed by $O = \{1, 3, 5, 7, 9\}$.

Although sets are usually used to group together elements with common properties, there is nothing that prevents a set from having seemingly unrelated elements. For instance, {a*,* 2*,* Fred, New Jersey} is the set containing the four elements a, 2, Fred, and New Jersey.

Sometimes the roster method is used to describe a set without listing all its members. Some members of the set are listed, and then *ellipses* (. . .) are used when the general pattern of the elements is obvious.

Example: The set of positive integers less than 100 can be denoted by $\{1, 2, 3, \ldots, 99\}$.

Another way to describe a set is to use **set builder** notation. We characterize all those elements in the set by stating the property or properties they must have to be members. For instance, the set O of all odd positive integers less than 10 can be written as

O = {x | x is an odd positive integer less than 10}*,*

or, specifying the universe $\mathbb{N}$ as the set of positive integers, as

$$O = \{x \in \mathbb{N} : x \text{ is odd and } x < 10\}.$$

Equal sets: Two sets are *equal* if and only if they have the same elements. Therefore, if *A* and *B* are sets, then *A* and *B* are equal if and only if $\forall x(x \in A \leftrightarrow x \in B)$. We write $A = B$ if *A* and *B* are equal sets.

Example: The sets $\{1, 3, 5\}$ and $\{3, 5, 1\}$ are equal, because they have the same elements.

Note that the order in which the elements of a set are listed does not matter. Note also that it does not matter if an element of a set is listed more than once, so $\{1, 3, 3, 3, 5, 5, 5, 5\}$ is the same as the set $\{1, 3, 5\}$ because they have the same elements.

There is a special set that has no elements. This set is called the **empty set**, or **null set**, and is denoted by ∅. The empty set can also be denoted by { } (that is, we represent the empty set with a pair of braces that encloses all the elements in this set). Often, a set of elements with certain properties turns out to be the null set. For instance, the set of all positive integers that are greater than their squares is the null set. A set with one element is called a **singleton set**.

A common error is to confuse the empty set $\emptyset$ with the set $\{\emptyset\}$, which is a singleton set. The single element of the set $\{\emptyset\}$ is the empty set itself!

It is common to encounter situations where the elements of one set are also the elements of a second set. We now introduce some terminology and notation to express such relationships between sets.

Subset: The set *A* is a *subset* of *B* if and only if every element of *A* is also an element of *B*. We use the notation $A \subseteq B$ to indicate that *A* is a subset of the set *B*.

We see that $A \subseteq B$ if and only if the quantification $\forall x(x \in A \rightarrow x \in B)$ is true. Note that to show that *A* is not a subset of *B* we need only find one element $x \in A$ with $x \notin B$. Such an *x* is a counterexample to the claim that $x \in A$ implies $x \in B$.

Example: For every set S, (i) $\emptyset \subseteq S$ and (ii) $S \subseteq S$.

Solution: To show that $\emptyset \subseteq S$, we must show that $\forall x(x \in \emptyset \rightarrow x \in S)$ is true. Because the empty set contains no elements, it follows that $x \in \emptyset$ is always false. It follows that the conditional statement $x \in \emptyset \rightarrow x \in S$ is always true, because its hypothesis is always false and a conditional statement with a false hypothesis is true. Therefore, $\forall x(x \in \emptyset \rightarrow x \in S)$ is true. Thus, $\emptyset \subseteq S$. Since $\forall x(x \in S \rightarrow x \in S)$ is true, $S \subseteq S$.

Note that this is an example of a vacuous proof. To show that two sets A and B are equal, show that $A \subseteq B$ and $B \subseteq A$.

Sets may have other sets as members. For instance, we have the sets $A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ and $B = \{x \mid x$ is a subset of the set $\{a, b\}\}$. Note that these two sets are equal, that is, $A = B$. Also note that $\{a\} \in A$, but $a \notin A$.

Sets are used extensively in counting problems, and for such applications we need to discuss the sizes of sets.

The Size of a Set: Let S be a set. If there are exactly n distinct elements in S where n is a nonnegative integer, we say that S is a *finite set* and that n is the *cardinality* of S. The cardinality of S is denoted by $|S|$.

Example: Let *A* be the set of odd positive integers less than 10. Then $|A| = 5$.

Example: Let S be the set of letters in the English alphabet. Then $|S| = 26$.

Example: Because the null set has no elements, it follows that $|\emptyset| = 0$.

Infinite set: A set is said to be *infinite* if it is not finite.

Example: The set of positive integers is infinite.

Many problems involve testing all combinations of elements of a set to see if they satisfy some property. To consider all such combinations of elements of a set *S*, we build a new set that has as its members all the subsets of *S*.

Power Sets: Given a set S, the *power set* of S is the set of all subsets of the set S. The power set of S is denoted by $P(S)$.

Example: What is the power set of the set $\{0, 1, 2\}$?

Solution: The power set $P(\{0, 1, 2\})$ is the set of all subsets of $\{0, 1, 2\}$. Hence,

$$P(\{0,1,2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}.$$

Note that the empty set and the set itself are members of this set of subsets.

Example: What is the power set of the empty set? What is the power set of the set $\{\emptyset\}$?

Solution: The empty set has exactly one subset, namely, itself. Consequently, $P(\emptyset) = \{\emptyset\}$.

The set $\{\emptyset\}$ has exactly two subsets, namely, $\emptyset$ and the set $\{\emptyset\}$ itself. Therefore,

$$P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}.$$

Note: If a set has n elements, then its power set has 2^n elements.

Set Operations

Two, or more, sets can be combined in many different ways. For instance, starting with the set of mathematics majors at your school and the set of computer science majors at your school, we can form the set of students who are mathematics majors or computer science majors, the set of students who are joint majors in mathematics and computer science, the set of all students not majoring in mathematics, and so on.

Union of the sets: Let *A* and *B* be sets. The *union* of the sets *A* and *B*, denoted by $A \cup B$, is the set that contains those elements that are either in *A* or in *B*, or in both. An element x belongs to the union of the sets *A* and *B* if and only if x belongs to *A* or x belongs to *B*. This tells us that

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

Example: The union of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 2, 3, 5\}$; that is, $\{1, 3, 5\} \cup \{1, 2, 3\} = \{1, 2, 3, 5\}$.

Example: The union of the set of all computer science majors at your school and the set of all mathematics majors at your school is the set of students at your school who are majoring either in mathematics or in computer science (or in both).

Intersection of the sets: Let *A* and *B* be sets. The *intersection* of the sets *A* and *B*, denoted by $A \cap B$, is the set containing those elements in both *A* and *B*. An element x belongs to the intersection of the sets *A* and *B* if and only if x belongs to *A* and x belongs to *B*. This tells us that

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$

Example: The intersection of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 3\}$; that is, $\{1, 3, 5\} \cap \{1, 2, 3\} = \{1, 3\}$.

Disjoint sets: Two sets are called *disjoint* if their intersection is the empty set.

Example: Let $A = \{1, 3, 5, 7, 9\}$ and $B = \{2, 4, 6, 8, 10\}$. Because $A \cap B = \emptyset$, *A* and *B* are disjoint.

Difference of sets: Let *A* and *B* be sets. The *difference* of *A* and *B,* denoted by $A - B,$ is the set containing those elements that are in A but not in B. The difference of *A* and *B* is also called the *complement of B with respect to A*. An element x belongs to the difference of A and B if and only if $x \in A$ and $x \notin B$. This tells us that

$$A - B = \{x \mid x \in A \wedge x \notin B\}.$$

Remark: The difference of sets *A* and *B* is sometimes denoted by $A \setminus B$.

Example: The difference of $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{5\}$; that is, $\{1, 3, 5\} - \{1, 2, 3\} = \{5\}$. This is different from the difference of $\{1, 2, 3\}$ and $\{1, 3, 5\}$, which is the set $\{2\}$.

Complement of a set: Let *U* be the universal set. The *complement* of the set *A,* denoted by $\bar{A},$ is the complement of *A* with respect to *U*. Therefore, the complement of the set *A* is $U - A$. An element belongs to $\bar{A}$ if and only if $x \notin A$. This tells us that

$$\bar{A} = \{x \in U \mid x \notin A\}.$$

Example: Let $A = \{$a, e, i, o, u$\}$ (where the universal set is the set of letters of the English alphabet). Then

$\bar{A} = \{$b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z$\}$.

Example: Let *A* be the set of positive integers greater than 10 (with universal set the set of all positive integers). Then $\bar{A} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

Example: Prove that $\overline{A \cap B} = \bar{A} \cup \bar{B}$.

Solution: We will prove that the two sets $\overline{A \cap B}$ and $\overline{A} \cup \bar{B}$ are equal by showing that each set is a subset of the other. First, we will show that $\overline{A \cap B} \subseteq \overline{A} \cup \bar{B}$. We do this by showing that if *x* is in $\overline{A \cap B}$, then it must also be in $\overline{A} \cup \bar{B}$. Now suppose that $x \in \overline{A \cap B}$. By the definition of complement, $x \notin A \cap B$. Using the definition of intersection, we see that the proposition$\neg((x \in A) \wedge (x \in B))$ is true. By applying De Morgan's law for propositions, we see that $\neg(x \in A)$ or $\neg(x \in B)$. Using the definition of negation of propositions, we have $x \notin A$ or $x \notin B$. Using the definition of the complement of a set, we see that this implies that $x \in \bar{A}$ or $x \in \bar{B}$. Consequently, by the definition of union, we see that $x \in \bar{A} \cup \bar{B}$. We have now shown that $\overline{A \cap B} \subseteq \overline{A} \cup \bar{B}$. Next, we will show that $\bar{A} \cup \bar{B} \subseteq \overline{A \cap B}$. We do this by showing that if *x* is in $\bar{A} \cup \bar{B}$, then it must also be in $\overline{A \cap B}$. Now suppose that $x \in \bar{A} \cup \bar{B}$. By the definition of union, we know that $x \in \overline{A}$ or $x \in \bar{B}$. Using the definition of complement, we see that $x \notin A$ or $x \notin B$. Consequently, the proposition $\neg(x \in A) \vee \neg(x \in B)$ is true. By De Morgan's law for propositions, we conclude that $\neg((x \in A) \wedge (x \in B))$ is true. By the definition of intersection, it follows that $\neg(x \in A \cap B)$. We now use the definition of complement to conclude that $x \in \overline{A \cap B}$. This shows that $\bar{A} \cup \bar{B} \subseteq \overline{A \cap B}$. Because we have shown that each set is a subset of the other, the two sets are equal, and the identity is proved.

Principle of inclusion–exclusion.

The subtraction rule: Suppose that a task can be done in one of two ways, but some of the ways to do it are common to both ways. In this situation, we cannot use the sum rule to count the number of ways to do the task. If we add the number of ways to do the tasks in these two ways, we get an overcount of the total number of ways to do it, because the ways to do the task that are common to the two ways are counted twice. To correctly count the number of ways to do the two tasks, we must subtract the number of ways that are counted twice. This leads us to an important counting rule.

"If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways."

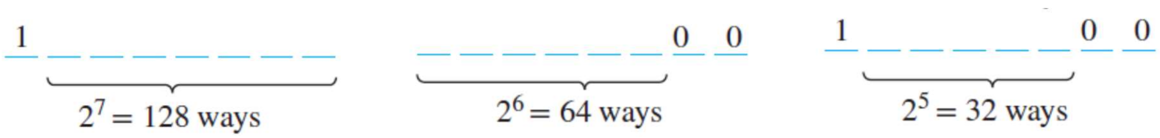
The subtraction rule is also known as the **principle of inclusion–exclusion**, especially when it is used to count the number of elements in the union of two sets. Suppose that A_1 and A_2 are sets.

Then, there are $|A_1|$ ways to select an element from A_1 and $|A_2|$ ways to select an element from A_2. The number of ways to select an element from A_1 or from A_2, that is, the number of ways to select an element from their union, is the sum of the number of ways to select an element from A_1 and the number of ways to select an element from A_2, minus the number of ways to select an element that is in both A_1 and A_2. Because there are $|A_1 \cup A_2|$ ways to select an element in eitherA_1or in A_2, and $|A_1 \cap A_2|$ ways to select an element common to both sets, we have

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Example: How many bit strings of length eight start with a 1 bit or end with the two bits 00?

Solution: we need three counting problems to solve before we can apply the principle of inclusion–exclusion. We can construct a bit string of length eight that starts with a 1 bit or ends with the two bits 00, by constructing a bit string of length eight beginning with a 1 bit or by constructing a bit string of length eight that ends with the two bits 00.



We can construct a bit string of length eight that begins with a 1 in $2^7 = 128$ ways. This follows by the product rule, because the first bit can be chosen in only one way and each of the other seven bits can be chosen in two ways. Similarly, we can construct a bit string of length eight ending with the two bits 00, in $2^6 = 64$ ways. This follows by the product rule, because each of the first six bits can be chosen in two ways and the last two bits can be chosen in only one way. Some of the ways to construct a bit string of length eight starting with a 1 are the same as the ways to construct a bit string of length eight that ends with the two bits 00. There are $2^5 = 32$ ways to construct such a string. This follows by the product rule, because the first bit can be chosen in only one way, each of the second through the sixth bits can be chosen in two ways, and the last two bits can be chosen in one way. Consequently, the number of bit strings of length eight that begin with a 1 or end with a 00, which equals the number of ways to construct a bit string of length eight that begins with a 1 or that ends with 00, equals 128 + 64 − 32 = 160.

Example: How many positive integers not exceeding 100 are divisible either by 4 or by 6?

Solution: Let A be the set of positive integers not exceeding 100 that are divisible by 4 and B be the set of positive integers not exceeding 100 that are divisible by 6. Then $A \cap B$ is the set of positive integers not exceeding 100 that are divisible by 4 and 6. That is, $A \cap B$ is the set of positive integers not exceeding 100 that are divisible by 12. Also $A \cup B$ is the set of positive integers not exceeding 100 are divisible either by 4 or by 6. Then $|A| = [100/4] = 25$, $|B| = [100/6] = 16$, $|A \cap B| = [100/12] = 8$. Thus,

$$|A \cup B| = |A| + |B| - |A \cap B| = 25 + 16 - 8 = 33.$$

Therefore, there are 33 positive integers not exceeding 100 are divisible either by 4 or by 6.

More generally if $A_1, A_2, \ldots A_m$ are finite sets, then

$$|A_1 \cup A_2 \cup \cdot \cdot \cdot \cup A_m| = \sum_{i=1}^{m} |A_i| - \sum_{1 \le i < j \le m} |A_i \cap A_j| + \sum_{1 \le i < j < k \le m} |A_i \cap A_j \cap A_k| - \ldots$$

$$+(-1)^{m-1} |A_1 \cap A_2 \cap \cdot \cdot \cdot \cap A_m|.$$

Example: How many positive integers not exceeding 100 are divisible by 4,or by 5, or by 6?

Solution: Let A be the set of positive integers not exceeding 100 that are divisible by 4,

B be the set of positive integers not exceeding 100 that are divisible by 5 and

C be the set of positive integers not exceeding 100 that are divisible by 6. Then

$A \cap B$ is the set of positive integers not exceeding 100 that are divisible by 4 and 5,

$A \cap C$ is the set of positive integers not exceeding 100 that are divisible by 4 and 6 ,

$B \cap C$ is the set of positive integers not exceeding 100 that are divisible by 5 and 6 and

$A \cap B \cap C$ is the set of positive integers not exceeding 100 that are divisible by 4 and 5 and 6.

Then

$$|A| = \left[\frac{100}{4}\right] = 25, |B| = \left[\frac{100}{5}\right] = 20, |C| = \left[\frac{100}{6}\right] = 16,$$

$$|A \cap B| = \left[\frac{100}{20}\right] = 5. |A \cap C| = \left[\frac{100}{12}\right] = 8, |B \cap C| = \left[\frac{100}{30}\right] = 3, |A \cap B \cap C| = \left[\frac{100}{60}\right] = 1.$$

Thus,

$$|A \cup B \cup c| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap BC|$$
$$= 25 + 20 + 16 - 5 - 8 - 3 + 1 = 46.$$

Therefore, there are 46 positive integers not exceeding 100 are divisible by 4,or by 5, or by 6.

Cartesian Products

The order of elements in a collection is often important. Because sets are unordered, a different structure is needed to represent ordered collections. This is provided by ordered *n*-tuples. The *ordered n-tuple* $(a_1, a_2, \ldots, a_n)$ is the ordered collection that has a_1 as its first element, a_2 as its second element, . . . , and a_n as its nth element. We say that two ordered *n*-tuples are equal if and only if each corresponding pair of their elements are equal. In other words, $(a_1, a_2, \ldots, a_n) = (b_1, b_2, \ldots, b_n)$ if and only if $a_i = b_i$, for $i = 1, 2, \ldots, n$. In particular, ordered 2-tuples are called **ordered pairs**. The ordered pairs (a, b) and (c, d) are equal if and only if $a = c$ and $b = d$. Note that (a, b) and (b, a) are not equal unless $a = b$.

Let *A* and *B* be nonempty sets. The *Cartesian product* of *A* and *B*, denoted by $A \times B$, is the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$. Hence,

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

Example: What is the Cartesian product of $A = \{1, 2\}$ and $B = \{a, b, c\}$?

Solution: The Cartesian product $A \times B$ is

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}.$$

Note that the Cartesian products $A \times B$ and $B \times A$ are not equal, unless $A = B$.

Definition: The *Cartesian product* of the sets $A_1, A_2, \ldots, A_n$, denoted by $A_1 \times A_2 \times \cdot \cdot \cdot \times A_n$, is the set of ordered *n*-tuples $(a_1, a_2, \ldots, a_n)$, where a_i belongs to A_i for $i = 1, 2, \ldots, n$. In other words,

$$A_1 \times A_2 \times \cdot \cdot \cdot \times A_n = \{(a_1, a_2, \ldots, a_n) | a_i \in A_i \text{ for } i = 1, 2, \ldots, n\}.$$

We use the notation A^2 to denote $A \times A$, the Cartesian product of the set *A* with itself. Similarly, $A^3 = A \times A \times$ A, $A^4 = A \times A \times A \times A$, and so on. More generally,

$$A^n = \{(a_1, a_2, \ldots, a_n) | a_i \in A \text{ for } i = 1, 2, \ldots, n\}.$$

Example: Suppose that $A = \{1, 2\}$. It follows that $A^2 = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ and $A^3 = \{(1, 1, 1), (1, 1, 2), (1, 2, 1), (1, 2, 2), (2, 1, 1), (2, 1, 2), (2, 2, 1), (2, 2, 2)\}$.

Relations

The most direct way to express a relationship between elements of two sets is to use ordered pairs made up of two related elements. For this reason, sets of ordered pairs are called binary relations.

Binary relation: Let *A* and *B* be sets. A binary relation *from A to B* is a subset of $A \times B$.

In other words, a binary relation from *A* to *B* is a set *R* of ordered pairs where the first element of each ordered pair comes from *A* and the second element comes from *B*. We use the notation $a\ R\ b$ to denote that $(a, b) \in R$. Moreover, when (a, b) belongs to *R*, a is said to be related to b by *R*. Binary relations represent relationships between the elements of two sets.

Example: Let $A = \{0, 1, 2\}$ and $B = \{a, b\}$. Then $\{(0, a), (0, b), (1, a), (2, b)\}$ is a relation from *A* to *B*.

Note: A *relation on a set* A is a relation from A to A. In other words, a relation on a set A is a subset of $A \times A$.

Example: Let *A* be the set $\{1, 2, 3, 4\}$. Which ordered pairs are in the relation $R = \{(a, b) \mid a \text{ divides } b\}$?

Solution: Because (a, b) is in R if and only if *a* and *b* are positive integers not exceeding 4 such that a divides b, we see that

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)\}.$$

Example: How many relations are there on a set with *n* elements?

Solution: A relation on a set *A* is a subset of $A \times A$. Because $A \times A$ has n^2 elements when A has n elements, and a set with *m* elements has 2^m subsets, there are 2^{n^2} subsets of $A \times A$. Thus, there are 2^{n^2} relations on a set with n elements. For example, there are $2^{3^2} = 2^9 = 512$ relations on the set {*a, b, c*}.

Properties of Relations: Let R be a relation on a set A. The relation R is called *reflexive* if $(a, a) \in R$ for every element $a \in A$. The relation R is called *symmetric* if $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$. In the relation R, if $(a, b) \in R$ and $(b, a) \in R$, imply $a = b$ then R is called *antisymmetric*. The relation R *is* called *transitive* if $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$.

Example: Consider the following relations on $\{1, 2, 3, 4\}$:

$R_1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$,

$R_2 = \{(1, 1), (1, 2), (2, 1)\}$,

$R_3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$,

$R_4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$,

$R_5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$,

$R_6 = \{(3, 4)\}$.

Which of these relations are reflexive, symmetric, antisymmetric and transitive?

Solution: The relations R_3 and R_5 are reflexive because they both contain all pairs of the form (a, a), namely, $(1, 1)$, $(2, 2)$, $(3, 3)$, and $(4, 4)$. The other relations are not reflexive because they do not contain all of these ordered pairs. In particular, R_1, R_2, R_4, and R_6 are not reflexive because $(3, 3)$ is not in any of these relations.

The relations R_2 and R_3 are symmetric, because in each case (b, a) belongs to the relation whenever (a, b) does. For R_2, the only thing to check is that both $(2, 1)$ and $(1, 2)$ are in the relation. For R_3, it is necessary to check that both $(1, 2)$ and $(2, 1)$ belong to the relation, and $(1, 4)$ and $(4, 1)$ belong to the relation. None of the other relations is symmetric. This is done by finding a pair (a, b) such that it is in the relation but (b, a) is not.

The relations R_4, R_5, and R_6 are all antisymmetric. For each of these relations there is no pair of elements a and b with $a \neq b$ such that both (a, b) and (b, a) belong to the relation. None of the other relations is antisymmetric. This is done by finding a pair (a, b) with $a \neq b$ such that (a, b) and (b, a) are both in the relation.

The relations R_4, R_5, and R_6 are transitive. For each of these relations, we can show that it is transitive by verifying that if (a, b) and (b, c) belong to this relation, then (a, c)also does. For instance, R_4 is transitive, because $(3, 2)$ and $(2, 1)$, $(4, 2)$ and $(2, 1)$, $(4, 3)$ and $(3, 1)$, and $(4, 3)$ and $(3, 2)$ are the only such sets of pairs, and $(3, 1)$, $(4, 1)$, and $(4, 2)$ belong to R_4. R_1 is not transitive because $(3, 4)$ and $(4, 1)$ belong to R_1, but $(3, 1)$ does not. R_2 is not transitive because $(2, 1)$ and $(1, 2)$ belong to R_2, but $(2, 2)$ does not. R_3 is not transitive because $(4, 1)$ and $(1, 2)$ belong to R_3, but $(4, 2)$ does not.

Example: Is the "divides" relation on the set of positive integers reflexive, symmetric, antisymmetric and transitive?

Solution*:* Because $a \mid a$ whenever a is a positive integer, the "divides" relation is reflexive. (Note that if we replace the set of positive integers with the set of all integers the relation is not reflexive because by definition 0 does not divide 0.) This relation is not symmetric because $1|2$, but $2 \nmid 1$. It is antisymmetric, for if a and b are positive integers with $a\,|b$ and $b\,|a$, then $a = b$. Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. It follows that this relation is transitive.

Example: How many reflexive relations are there on a set with n elements?

Solution: A relation R on a set A is a subset of $A \times A$. Consequently, a relation is determined by specifying whether each of the n^2 ordered pairs in $A \times A$ is in R. However, if R is reflexive, each of the n ordered pairs (a, a) must be in *R* for $a \in A$. Each of the other $n^2 - n$ ordered pairs of the form (a, b), where $a \neq b$, may or may not be in R. Hence, by the product rule for counting, there are $2^{n(n-1)}$ reflexive relations [this is the number of ways to choose whether each element (a, b), with $a \neq b$, belongs to R].

Example: How many symmetric relations are there on a set with n elements?

Example: How many reflexive and symmetric relations are there on a set with n elements?

Example: The relation $\Delta = \{(a, a) \mid a \in A\}$ is called the **diagonal relation** on *A*.

Combining Relations

Example: Let $A = \{1, 2, 3\}$ and $B = \{1, 2, 3, 4\}$. The relations $R_1 = \{(1, 1), (2, 2), (3, 3)\}$ and $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4)\}$ can be combined to obtain $R_1 \cup R_2 = \{(1,1), (1,2), ((1,3), (1, 4), (2, 2), (3, 3)\}$, $R_1 \cap R_2 = \{(1, 1)\}$, $R_1 - R_2 = \{(2, 2), (3, 3)\}$ and $R_2 - R_1 = \{(1, 2), (1, 3), (1, 4)\}$.

Example: Let *A* and *B* be the set of all students and the set of all courses at a school, respectively. Suppose that R_1 consists of all ordered pairs (a, b), where a is a student who has taken course b, and R_2 consists of all ordered pairs (a, b), where a is a student who requires course b to graduate. What are the relations $R_1 \cup R_2, R_1 \cap R_2, R_1 \oplus R_2, R_1 - R_2$, and $R_2 - R_1$?

Solution: The relation $R_1 \cup R_2$ consists of all ordered pairs (a, b), where a is a student who either has taken course b or needs course b to graduate, and $R_1 \cap R_2$ is the set of all ordered pairs (a, b), where a is a student who has taken course b and needs this course to graduate. Also, $R_1 \oplus R_2$ consists of all ordered pairs (a, b), where student a has taken course b but does not need it to graduate or needs course b to graduate but has not taken it. $R_1 - R_2$is the set of ordered pairs (a, b), where a has taken course b but does not need it to graduate; that is, b is an elective course that a has taken. $R_2 - R_1$is the set of all ordered pairs (a, b), where b is a course that a needs to graduate but has not taken by a.

Example: Let R_1 be the "less than" relation on the set of real numbers and let R_2 be the "greater than" relation on the set of real numbers, that is, $R_1 = \{(x, y) \mid x < y\}$ and $R_2 = \{(x, y) \mid x > y\}$. What are $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 - R_2$, $R_2 - R_1$, and $R_1 \oplus R_2$?

Solution: We note that $(x, y) \in R_1 \cup R_2$ if and only if $(x, y) \in R_1$ *or* $(x, y) \in R_2$. Hence, $(x, y) \in R_1 \cup R_2$ if and only if $x < y$ or $x > y$. Because the condition $x < y$ or $x > y$ is the same as the condition $x \neq y$, it follows that $R_1 \cup R_2 = \{(x, y) \mid x \neq y\}$. In other words, the union of the "less than" relation and the "greater than" relation is the "not equals" relation. It is impossible for a pair (x, y) to belong to both R_1 and R_2 because it is impossible that $x < y$ and $x > y$. It follows that $R_1 \cap R_2 = \emptyset$. We also see that $R_1 - R_2 = R_1$, $R_2 - R_1 = R_2$, and $R_1 \oplus R_2 = (R_1 \cup R_2) - (R_1 \cap R_2) = \{(x, y) \mid x \neq y\}$.

Composition of relations: Let R be a relation from a set A to a set B and S a relation from B to a set C. The *composite* of R and S is the relation consisting of ordered pairs (a, c), where $a \in$

$A, c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$. We denote the composite of R and S by $S \circ R$.

Example: What is the composite of the relations R and S, where R is the relation from $\{1, 2, 3\}$ to $\{1, 2, 3, 4\}$ with $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$ and S is the relation from $\{1, 2, 3, 4\}$ to $\{0, 1, 2\}$ with $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$?

Solution: $S \circ R$ is constructed using all ordered pairs in R and ordered pairs in S, where the second element of the ordered pair in R agrees with the first element of the ordered pair in S. For example, the ordered pairs $(2, 3) \in R$ and $(3, 1) \in S$ produce the ordered pair $(2, 1) \in S \circ R$. Computing all the ordered pairs in the composite, we find $S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}$.

Example: Composing the Parent Relation with Itself. Let R be the relation on the set of all people such that $(a, b) \in R$ if person a is a parent of person b. Then $(a, c) \in R \circ R$ if and only if there is a person b such that $(a, b) \in R$ and $(b, c) \in R$, that is, if and only if there is a person b such that a is a parent of b and b is a parent of c. In other words, $(a, c) \in R \circ R$ if and only if a is a grandparent of c.

The powers of a relation R can be recursively defined from the definition of a composite of two relations. Let R be a relation on the set A. The powers $R^n, n = 1, 2, 3, \ldots$, are defined recursively by $R^1 = R$ and $R^{n+1} = R^n \circ R$.

The definition shows that $R^2 = R \circ R$, $R^3 = R^2 \circ R = (R \circ R) \circ R$, and so on.

Example: Let $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$. Find the powers $Rn, n = 2, 3, 4, \ldots$.

Solution: Because $R^2 = R \circ R$, we find that $R^2 = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$. Furthermore, because $R^3 = R^2 \circ R = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. Additional computation shows that R^4 is the same as R^3, so $R^4 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. It also follows that $R^n = R^3$ for $n = 5, 6, 7, \ldots$.

Theorem: The relation R on a set A is transitive if and only if $R^n \subseteq R$ for $n = 1, 2, 3, \ldots$.

Proof: We first prove the "if" part of the theorem. We suppose that $R^n \subseteq R$ for $n = 1, 2, 3, \ldots$. In particular, $R^2 \subseteq R$. To see that this implies *R* is transitive, note that if $(a, b) \in R$ and $(b, c) \in R$, then by the definition of composition, $(a, c) \in R^2$. Because $R^2 \subseteq R$, this means that $(a, c) \in R$. Hence, R is transitive. We will use mathematical induction to prove the only if part of the theorem. Note that this part of the theorem is trivially true for $n = 1$. Assume that $R^n \subseteq R$, where n is a positive integer. This is the inductive hypothesis. To complete the inductive step, we must show that this implies that R^{n+1} is also a subset of R. To show this, assume that $(a, b) \in R^{n+1}$. Then, because $R^{n+1}= R^n \circ R$, there is an element x with $x \in A$ such that $(a, x) \in R$ and $(x, b) \in R^n$. The inductive hypothesis, namely, that $R^n \subseteq R$, implies that $(x, b) \in R$. Furthermore, because *R* is transitive, and $(a, x) \in R$ and $(x, b) \in R$, it follows that $(a, b) \in R$. This shows that $R^{n+1} \subseteq R$, completing the proof.

Inverse relation: Let R be a relation from a set A to a set B. The inverse relation of R is a relation from B to A, denoted by R^{-1}, given by the set $\{(b, a) \mid (a, b) \in R\}$. The complementary relation $\bar{R}$ is the set of ordered pairs $\{(a, b) \mid (a, b) \notin R\}$.

Example: Let $R = \{(a, b) \mid a < b\}$ be the relation on the set of integers. Then the inverse relation of

$$R^{-1} = \{(b, a)|(a, b) \in R\} = \{(b, a)|a < b\} = \{(b, a)|b > a\}.$$

And the complementary relation of R

$$\bar{R} = \{(a, b)|(a, b) \notin R\} = \{(a, b)|a \nless b\} = \{(a, b)|a \geq b\}.$$

Example: Let $R = \{(a, b) \mid a \text{ divides } b\}$ be the relation on the set of positive integers. Then

$$R^{-1} = \{(b, a)|(a, b) \in R\} = \{(b, a)|a \text{ divides } b\} = \{(b, a)|\, b \text{ is divisible } by\ a\}.$$

And

$$\bar{R} = \{(a, b)|(a, b) \notin R\} = \{(a, b)|\, a \text{ does not divides } b\}.$$

Example: Let R *and* S be the relations with $R \subseteq S$, then $R^{-1} \subseteq S^{-1}$.

Example: Let R *and* S be two relations, then $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$, $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$ and $(S \circ R)^{-1} = R^{-1} \circ S^{-1}$.

Example: On a set of n elements the number of reflexive relations is $2^{n(n-1)}$, number of symmetric relation is $2^{\frac{n(n+1)}{2}}$ and the number of reflexive and symmetric relations is $2^{\frac{n(n-1)}{2}}$.

Representing Relations Using Matrices

A relation between finite sets can be represented using a zero–one matrix. Suppose that R is a relation from $A = \{a_1, a_2, \ldots, a_m\}$ to $B = \{b_1, b_2, \ldots, b_n\}$. The relation R can be represented by the matrix $\boldsymbol{M}_R = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{if } (a_i, b_j) \in R \\ 0 & \text{if } (a_i, b_j) \notin R \end{cases}$$

In other words, the zero–one matrix representing R has a 1 as its (i, j) entry when a_i is related to b_j, and a 0 in this position if a_i is not related to b_j. (Such a representation depends on the orderings used for A and B.)

Example: Suppose that $A = \{1, 2, 3\}$ and $B = \{1, 2\}$. Let R be the relation from A to B containing (a, b) if $a > b$. What is the matrix representing R ?

Solution: Because $R = \{(2, 1), (3, 1), (3, 2)\}$, the matrix for R is

$$M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Example: Let $A = \{a_1, a_2, a_3\}\ and\ B = \{b_1, b_2, b_3, b_4, b_5\}$. Which ordered pairs are in the relation R represented by the matrix

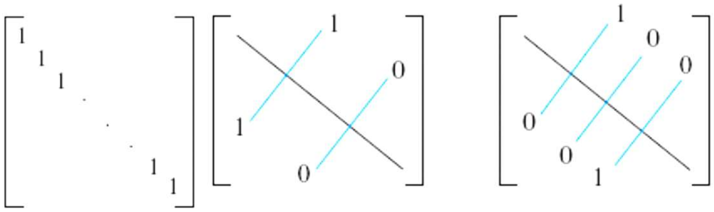
$$M_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}?$$

Solution: Because R consists of those ordered pairs (a_i, b_j) with $m_{ij} = 1$, it follows that $R = \{(a_1, b_2), (a_2, b_1), (a2, b_3), (a_2, b_4), (a_3, b_1), (a_3, b_3), (a_3, b_5)\}$.

The matrix of a relation on a set, which is a square matrix, can be used to determine whether the relation has certain properties. Recall that a relation R on A is reflexive if $(a, a) \in R$ whenever $a \in A$. Thus, R is reflexive if and only if $(a_i, a_i) \in R$ for $i = 1, 2, \ldots, n$. Hence, R is reflexive if and only if $m_{ii} = 1$, for $i = 1, 2, \ldots, n$. In other words, R is reflexive if all the elements on the main diagonal of M_R are equal to 1. Note that the elements off the main diagonal can be either 0 or 1.

The relation R is symmetric if $(a,b) \in R$ implies that $(b,a) \in R$. Consequently, the relation R on the set $A = \{a_1, a_2, \ldots, a_n\}$ is symmetric if and only if $(a_j, a_i) \in R$ whenever $(a_i, a_j) \in R$. In terms of the entries of M_R, R is symmetric if and only if $m_{ji} = 1$ whenever $m_{ij} = 1$. This also means $m_{ji} = 0$ whenever $m_{ij} = 0$. Consequently, R is symmetric if and only if $m_{ij} = m_{ji}$, for all pairs of integers i and j with $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$. Recalling the definition of the transpose of a matrix, we see that R is symmetric if and only if $M_R = M_R^T$ that is, if M_R is a symmetric matrix.

The relation R is antisymmetric if and only if $(a,b) \in R$ and $(b,a) \in R$ imply that $a = b$. Consequently, the matrix of an antisymmetric relation has the property that if $m_{ij} = 1$ with $i \neq j$, then $m_{ji} = 0$. Also, it may be both $m_{ij} = 0$ and $m_{ji} = 0$. In other words, $m_{ij} = 0$ or $m_{ji} = 0$ when $i \neq j$.



Example: Suppose that the relation R on a set is represented by the matrix $M_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. Is R reflexive, symmetric, and/or antisymmetric?

Solution: Because all the diagonal elements of this matrix are equal to 1, R is reflexive. Moreover, because M_R is symmetric, it follows that R is symmetric. It is also easy to see that R is not antisymmetric as $m_{12} = 1 = m_{21}$.

A matrix all of whose entries are either 0 or 1 is called a **zero–one matrix**. Zero–one matrices are often used to represent discrete structures. Algorithms using these structures are based on Boolean arithmetic with zero–one matrices. This arithmetic is based on the Boolean operations ∧ and ∨, which operate on pairs of bits, defined by

$$b_1 \wedge b_2 = \begin{cases} 1 & \text{if } b_1 = b_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$b_1 \vee b_2 = \begin{cases} 1 & \text{if } b_1 = 1 \; or \; b_2 = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Definition: Let $A = [a_{ij}]$ and $B = [b_{ij}]$ are two $m \times n$ zero–one matrices. Then the *join* of A and B is the zero–one matrix with (i,j)th entry $a_{ij} \vee b_{ij}$. The join of A and B is denoted by $A \vee B$. The *meet* of A and B is the zero–one matrix with (i,j)th entry $a_{ij} \wedge b_{ij}$. The meet of A and B is denoted by $A \wedge B$.

Example: Find the join and meet of the zero–one matrices

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solution: We find that the join of A and B is

$$A \vee B = \begin{bmatrix} 1 \vee 1 & 0 \vee 0 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

and the meet of A and B is

$$A \wedge B = \begin{bmatrix} 1 \wedge 1 & 0 \wedge 0 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Definition: Let $A = [a_{ij}]$ be an $m \times k$ zero–one matrix and $B = [b_{ij}]$ be a $k \times n$ zero–one matrix. Then the *Boolean product* of A and B, denoted by $A \odot B$, is the $m \times n$ matrix with (i,j)th entry c_{ij} where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdot \cdot \cdot \vee (a_{ik} \wedge b_{kj}).$$

Example: Find the Boolean product of A and B, where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solution: The Boolean product $A \odot B$ is given by

$$A \odot B = \begin{bmatrix} (1 \wedge 1) \vee (1 \wedge 1) & (1 \wedge 0) \vee (1 \wedge 1) & (1 \wedge 1) \vee (1 \wedge 0) \\ (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) & (0 \wedge 1) \vee (1 \wedge 0) \\ (0 \wedge 1) \vee (0 \wedge 1) & (0 \wedge 0) \vee (0 \wedge 1) & (0 \wedge 1) \vee (0 \wedge 0) \end{bmatrix}$$

$$= \begin{bmatrix} 1 \vee 1 & 0 \vee 1 & 1 \vee\vee 0 \\ 0 \vee 1 & 0 \vee 1 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 0 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Definition: Let A be a square zero–one matrix and let r be a positive integer. The rth *Boolean power* of A is the Boolean product of r factors of A. The rth Boolean product of A is denoted by A$^{[r]}$.

Hence

$$A^{[r]} = A \odot A \odot A \odot \cdot \cdot \cdot \odot A, \ r \text{ times}$$

(This is well defined because the Boolean product of matrices is associative.) We also define $A^{[0]}$ to be the identity matrix I_n.

The Boolean operations join and meet can be used to find the matrices representing the union and the intersection of two relations. Suppose that R_1 and R_2 are relations on a set A represented by the matrices M_{R_1} and M_{R_2} , respectively. The matrix representing the inverse, union and intersection of these relations are

$$M_{R_1^{-1}} = \left(M_{R_1}\right)^T, \ M_{R_1 \cup R_2} = M_{R_1} \vee M_{R_2} \text{ and } M_{R_1 \cap R_2} = M_{R_1} \wedge M_{R_2}.$$

We now turn our attention to determining the matrix for the composite of relations. This matrix can be found using the Boolean product of the matrices for these relations. In particular, suppose that R is a relation from A to B and S is a relation from B to C. Suppose that A, B, and C have m, n, and p elements, respectively. Let the zero–one matrices for $S \circ R$, R, and S be $M_{S \circ R} = [t_{ij}]$, $M_R = [r_{ij}]$, and $M_S = [s_{ij}]$, respectively (these matrices have sizes $m \times p$, $m \times n$, and $n \times p$, respectively). The ordered pair (a_i, c_j) belongs to $S \circ R$ if and only if there is an element b_k such that (a_i, b_k) belongs to R and (b_k, c_j) belongs to S. It follows that $t_{ij} = 1$ if and only if $r_{ik} = s_{kj} = 1$ for some k. From the definition of the Boolean product, this means that

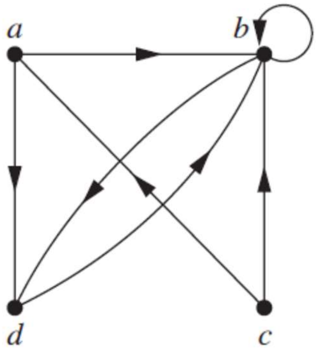
$$M_{S \circ R} = M_R \odot M_S.$$

Note: We have $M_{R^n} = M_R \odot M_R \odot \cdots \odot M_R = M_R^{[n]}$.

Representing Relations Using Digraphs

Definition: A *directed graph*, or *digraph*, consists of a set V of *vertices* (or *nodes*) together with a set E of ordered pairs of elements of V called *edges* (or *arcs*). The vertex a is called the *initial* *vertex* of the edge (a, b), and the vertex b is called the *terminal vertex* of this edge. An edge of the form (a, a) is represented using an arc from the vertex a back to itself. Such an edge is called a **loop**.

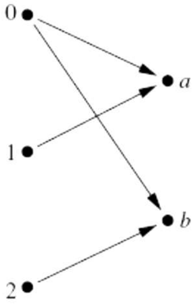
Example: The directed graph with vertices $\{a, b, c, d\}$, and edges $\{(a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (d, b)\}$ is displayed below:



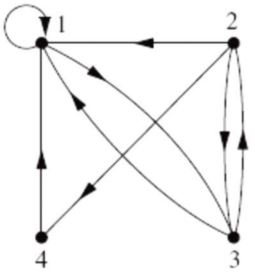
The relation R on a set A is represented by the directed graph that has the elements of A as its vertices and the ordered pairs (a, b), where $(a, b) \in R$, as edges. This assignment sets up a one-to-one correspondence between the relations on a set A and the directed graphs with A as their set of vertices. Thus, every statement about relations corresponds to a statement about directed graphs, and vice versa. Directed graphs give a visual display of information about relations. As such, they are often used to study relations and their properties.

Note that relations from a set A to a set B can be represented by a directed graph where there is a vertex for each element of A and a vertex for each element of B. However, when A = B, such representation provides much less insight than the digraph representations described here.

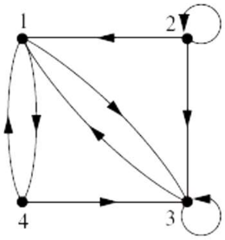
Example: Let $A = \{0, 1, 2\}$ and $B = \{a, b\}$. $R = \{(0, a), (0, b), (1, a), (2, b)\}$ is a relation from A to B. This Relation can be represented graphically,



Example: The directed graph of the relation $R = \{(1,1),(1,3),(2,1),(2,3),\ (2,4),(3,1),(3,2),(4,1)\}$ on the set $\{1,2,3,4\}$ is shown bellow:

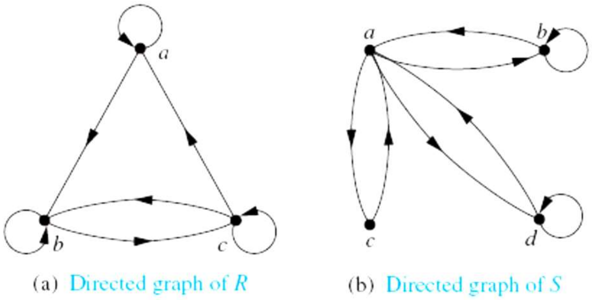


Example: What are the ordered pairs in the relation R represented by the directed graph shown in the bellow figure?



Solution: The ordered pairs (x,y) in the relation are $R = \{(1,3),(1,4),(2,1),(2,2),(2,3),(3,1),(3,3),(4,1),(4,3)\}$. Each of these pairs corresponds to an edge of the directed graph, with $(2,2)$ and $(3,3)$ corresponding to loops.

Example: Determine whether the relations for the directed graphs shown in bellow Figure are reflexive, symmetric, antisymmetric, and/or transitive.



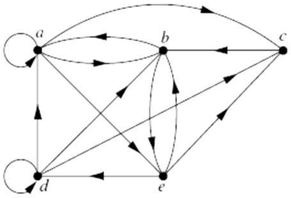
(a) Directed graph of R (b) Directed graph of S

Solution: Because there are loops at every vertex of the directed graph of R, it is reflexive. R is neither symmetric nor antisymmetric because there is an edge from a to b but not one from b to a, but there are edges in both directions connecting b and c. Finally, R is not transitive because there is an edge from a to b and an edge from b to c, but no edge from a to c.

Because loops are not present at all the vertices of the directed graph of S, this relation is not reflexive. It is symmetric but not antisymmetric, because every edge between distinct vertices is accompanied by an edge in the opposite direction. It is also not hard to see from the directed graph that S is not transitive, because (c, a) and (a, b) belong to S, but (c, b) does not belong to S.

Paths in Directed Graphs: A *path* from a to b in the directed graph G is a sequence of edges (x_0, x_1), (x_1, x_2), (x_2, x_3), …, (x_{n-1}, x_n) in G, where n is a nonnegative integer, and $x_0 = a$ and $x_n = b$, that is, a sequence of edges where the terminal vertex of an edge is the same as the initial vertex in the next edge in the path. This path is denoted by $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$ and has *length* n. We view the empty set of edges as a path of length zero from a to a. A path of length $n \geq 1$ that begins and ends at the same vertex is called a *circuit* or *cycle*.

Example: Which of the following are paths in the directed graph shown in the bellow figure: (i) a, b, e, d; (ii) a, e, c, d, b; (iii) b, a, c, b, a, a, b; (iv) d, c; (v) c, b, a; (vi) e, b, a, b, a, b, e? What are the lengths of those that are paths? Which of the paths in this list are circuits?



Solution: (i) Because each of (a, b), (b, e), and (e, d) is an edge, a, b, e, d is a path of length three. (ii) Because (c, d) is not an edge, a, e, c, d, b is not a path. (iii) Also, b, a, c, b, a, a, b is a path of length six because (b, a), (a, c), (c, b), (b, a), (a, a), and (a, b) are all edges. (iv) We see that d, c is a path of length one, because (d, c) is an edge. (v) Also c, b, a is a path of length two, because (c, b) and (b, a) are edges. (vi) All of (e, b), (b, a), (a, b), (b, a), (a, b), and (b, e) are edges, so e, b, a, b, a, b, e is a path of length six. The two paths b, a, c, b, a, a, b and e, b, a, b, a, b, e are circuits because they begin and end at the same vertex. The paths a, b, e, d; c, b, a; and d, c are not circuits.

Theorem: Let R be a relation on a set A. Then $(a, b) \in R^n$ if and only if there is a path of length n from a to b, where n is a positive integer.

Proof: We will use mathematical induction. By definition, there is a path from a to b of length one if and only if $(a, b) \in R$, so the theorem is true when $n = 1$. Assume that the theorem is true for the positive integer n. This is the inductive hypothesis. There is a path of length $n + 1$ from a to b if and only if there is an element $c \in A$ such that there is a path of length one from a to c, so $(a, c) \in R$, and a path of length n from c to b, that is, $(c, b) \in R^n$. Consequently, by the inductive hypothesis, there is a path of length $n + 1$ from a to b if and only if there is an element c with $(a, c) \in R$ and $(c, b) \in R^n$. But there is such an element if and only if $(a, b) \in R^{n+1}$. Therefore, there is a path of length $n + 1$ from a to b if and only if $(a, b) \in R^{n+1}$. This completes the proof.

Closures of Relations

A computer network has data centers in Boston, Chicago, Denver, Detroit, New York, and San Diego. There are direct, one-way telephone lines from Boston to Chicago, from Boston to Detroit, from Chicago to Detroit, from Detroit to Denver, and from New York to San Diego. Let R be the relation containing (a, b) if there is a telephone line from the data center in a *to* that in b. How can we determine if there is some (possibly indirect) link composed of one or more telephone lines from one center to another? Because not all links are direct, such as the link from Boston to Denver that goes through Detroit, R cannot be used directly to answer this. In the language of relations, R is not transitive, so it does not contain all the pairs that can be linked. As we will show, we can find all pairs of data centers that have a link by constructing a transitive relation S containing R such that S is a subset of every transitive relation containing R. Here, S is the smallest transitive relation that contains R. This relation is called the transitive closure of R. In general, let R be a relation on a set A and R may or may not have some property $\boldsymbol{P}$, such as reflexivity, symmetry, or transitivity. If there is a relation S with property $\boldsymbol{P}$ containing R such that S is a subset of every relation with property $\boldsymbol{P}$ containing R, then S is called the **closure** of R with respect to $\boldsymbol{P}$. (Note that the closure of a relation with respect to a property may not exist.) We will show how reflexive, symmetric, and transitive closures of relations can be found.

The relation $R = \{(1,1),(1,2),(2,1),(3,2)\}$ on the set $A = \{1,2,3\}$ is not reflexive. How can we produce a reflexive relation containing R that is as small as possible? This can be done by adding $(2,2)$ and $(3,3)$ to R, because these are the only pairs of the form (a,a) that are not in R. Clearly, this new relation contains R. Furthermore, *any* reflexive relation that contains R must also contain $(2,2)$ and $(3,3)$. Because this relation contains R, is reflexive, and is contained within every reflexive relation that contains R, it is called the **reflexive closure** of R.

Reflexive closure: Let R be a relation on a set A and S is reflexive closure of R. Then S is a reflexive relation containing R and if T is a reflexive relation containing R then, $S \subseteq T$. The reflexive closure of R can be formed by adding all pairs of the form (a,a) with $a \in A$ to R. The addition of these pairs produces a new relation that is reflexive, contains R, and is contained within any reflexive relation containing R. We see that the reflexive closure of R equals $R \cup \Delta$, where $\Delta = \{(a,a) \mid a \in A\}$ the diagonal relation on A.

Example: What is the reflexive closure of the relation $R = \{(a,b) \mid a < b\}$ on the set of integers?

Solution: The reflexive closure of *R* is

$$R \cup \Delta = \{(a,b) \mid a < b\} \cup \{(a,a) \mid a \in \boldsymbol{Z}\} = \{(a,b) \mid a \leq b\}.$$

The relation $\{(1,1),(1,2),(2,2),(2,3),(3,1),(3,2)\}$ on $\{1,2,3\}$ is not symmetric. How can we produce a symmetric relation that is as small as possible and contains R ? To do this, we need only add $(2,1)$ and $(1,3)$, because these are the only pairs of the form (b,a) with $(a,b) \in R$ that are not in R. This new relation is symmetric and contains R. Furthermore, *any* symmetric relation that contains R must contain this new relation, because a symmetric relation that contains R must contain $(2,1)$ and $(1,3)$. Consequently, this new relation is called the **symmetric closure** of R.

Symmetric closure: Let R be a relation on a set A and S is symmetric closure of R. Then S is a symmetric relation containing R and if T is a symmetric relation containing R then, $S \subseteq T$. The symmetric closure of a relation R can be constructed by adding all ordered pairs of the form (b,a), where (a,b) is in the relation, that are not already present in R. Adding these pairs produces a relation that is symmetric, that contains R, and that is contained in any symmetric relation that contains R. The symmetric closure of a relation can be constructed by taking the union of a relation with its inverse that is, $R \cup R^{-1}$ is the symmetric closure of R, where $R^{-1} = \{(b,a) \mid (a,b) \in R\}$.

Example: What is the symmetric closure of the relation $R = \{(a,b) \mid a > b\}$ on the set of positive integers?

Solution: The symmetric closure of *R* is the relation

$$R \cup R^{-1} = \{(a,b) \mid a > b\} \cup \{(b,a) \mid a > b\} = \{(a,b) \mid a \neq b\}.$$

This last equality follows because *R* contains all ordered pairs of positive integers where the first element is greater than the second element and *R*−1 contains all ordered pairs of positive integers where the first element is less than the second.

Transitive closures: Suppose that a relation R is not transitive. How can we produce a transitive relation that contains *R* such that this new relation is contained within any transitive relation that contains *R*? Can the transitive closure of a relation R be produced by adding all the pairs of the form (a,c), where (a,b) and (b,c) are already in the relation?

Consider the relation $R = \{(1,3),(1,4),(2,1),(3,2)\}$ on the set $\{1,2,3,4\}$. This relation is not transitive because it does not contain $(3{,}1)$ where $(3{,}2)$ and $(2{,}1)$ are in R. The pairs of this form not in R are $(1,2)$, $(2,3)$, $(2,4)$, and $(3,1)$. Adding these pairs does *not* produce a transitive relation, because the resulting relation contains $(3,1)$ and $(1,4)$ but does not contain $(3,4)$. This shows that constructing the transitive closure of a relation is more complicated than constructing either the reflexive or symmetric closure.

Definition: Let R be a relation on a set A. The *connectivity relation* R^* consists of the pairs (a,b) such that there is a path of length at least one from a to b in R.

Because R^n consists of the pairs (a,b) such that there is a path of length n from a to b, it follows that R^* is the union of all the sets R^n. In other words,

$$R^* = \bigcup_{n=1}^{\infty} R^n.$$

Theorem: The transitive closure of a relation R equals the connectivity relation R^*.

Now that we know that the transitive closure equals the connectivity relation, we turn our attention to the problem of computing this relation. We do not need to examine arbitrarily long paths to determine whether there is a path between two vertices in a finite directed graph. As the following theorem shows, it is sufficient to examine paths containing no more than n edges, where n is the number of elements in the set.

Theorem: Let R be a relation on a set A with n elements. If there is a path of length at least one in R from a to b, then there is such a path with length not exceeding n. Moreover, when $a \neq b$, if there is a path of length at least one in R from a to b, then there is such a path with length not exceeding $n - 1$.

From above theorem, we see that the transitive closure of R is the union of $R, R^2, R^3, \ldots$, and R^n. This follows because there is a path in R^* between two vertices if and only if there is a path between these vertices in R^i, for some positive integer i with $i \leq n$. Because

$$R^* = R \cup R^2 \cup R^3 \cup \cdots \cup R^n$$

and the zero–one matrix representing a union of relations is the join of the zero–one matrices of these relations, the zero–one matrix for the transitive closure is the join of the zero–one matrices of the first n powers of the zero–one matrix of R.

Theorem: Let M_R be the zero–one matrix of the relation R on a set with n elements. Then the zero–one matrix of the transitive closure R^* is

$$M_{R^*} = M_R \vee M_{R^2} \vee M_{R^3} \vee \cdots \vee M_{R^n} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \cdots \vee M_R^{[n]}.$$

Example: Find the zero–one matrix of the transitive closure of the relation R where

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solution: The zero–one matrix of the transitive closure R^* is

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]}.$$

Because $M_R^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ and $M_R^{[3]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$, it follows that

$$M_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Warshall's Algorithm: Warshall's algorithm, named after Stephen Warshall, who described it in 1960, is an efficient method for computing the transitive closure of a relation.

Lemma: Let $W_k = [w_{ij}^{[k]}]$ be the zero–one matrix that has a 1 in its (i, j)th position if and only if there is a path from v_i to v_j with interior vertices from the set $\{v_1, v_2, \ldots, v_k\}$. Then

$$w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]}),$$

whenever i, j , and k are positive integers not exceeding *n*.

Warshall's algorithm computes M_{R^*} by efficiently computing $W_0 = M_R, W_1, W_2, \ldots, W_n = M_{R^*}$.

Example: Let $A = \{a_1, a_2\ a_3, a_4, a_5\}$ and R be a relation on A given by $R = \{(a_1, a_1), (a_1, a_2), (a_1, a_4), (a_2, a_3), (a_3, a_3), (a_3, a_5), (a_4, a_4), (a_5, a_2)\}$. Find the transitive closure of R using Warshall's algorithm.

Solution: The matrix of the relation R $M_R = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$.

Step 1. We set $W_0 = M_R$, i.e.,

$$W_0 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Step 2. Construct W_1. First transfer all 1's of $W_0$to W_1

$$W_1 = \begin{bmatrix} 1 & 1 & & 1 & \\ & & 1 & & \\ & & 1 & & 1 \\ & & & 1 & \\ & 1 & & & \end{bmatrix}.$$

In column 1 of W_0: Nonzero entry at position 1. In row 1 of W_0: Nonzero entry at positions 1,2 and 4. Thus, at the position $(1,1)$, $(1,2)$, and $(1,4)$ of W_1make the entries 1. Therefore

$$W_1 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Step 3. Construct W_2. First transfer all 1's of $W_1$to W_2

$$W_2 = \begin{bmatrix} 1 & 1 & & 1 & \\ & & 1 & & \\ & & 1 & & 1 \\ & & & 1 & \\ & 1 & & & \end{bmatrix}.$$

In column 2 of W_1: Nonzero entry at positions 1 and 5. In row 2 of W_1: Nonzero entry at position 3. Thus at the position (1, 3), and (5, 3) of W_2 make the entries 1. Therefore

$$W_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Step 4. Construct W_3.

$$W_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Step 5. Construct W_4.

$$W_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Step 6. Construct W_5.

$$W_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

From W_5, we can conclude that the transitive closure of R is:

$$R^* = \{(a_1, a_1), (a_1, a_2), (a_1, a_3), (a_1, a_4), (a_1, a_5), (a_2, a_2), (a_2, a_3), (a_2, a_5),$$
$$(a_3, a_2)(a_3, a_3), (a_3, a_5), (a_4, a_4), (a_5, a_2), (a_5, a_3), (a_5, a_5)\}.$$

Example: Let $R = \{(a, c), (b, d), (c, a), (d, b), (e, d)\}$ be a relation on the set $A = \{a, b, c, d, e, f\}$. Check whether R is reflexive, symmetric, antisymmetric or transitive. Find reflexive, symmetric and Transitive closure of R. Use Warshall's algorithm to find the transitive closuer.

Solution: Given that $A = \{a, b, c, d, e, f\}$ and $R = \{(a, c), (b, d), (c, a), (d, b), (e, d)\}$ is a relation on A.

(i) R is not reflexive as $(a, a) \notin R$.

(ii) R is not symmetric as $(e, d) \in R$, but $(d, e) \notin R$.

(iii) R is not antisymmetric as $(a, c) \in R$ and $(c, a) \in R$ but $a \neq c$.

(iv) R is not transitive as $(a, c) \in R$ and $(c, a) \in R$ but $(a, a) \notin R$.

(v) Reflexive closure of R is $R \cup \{(a, a), (b, b), (c, c), (d, d), (e, e), (f, f)\}$

= $\{(a, c), (b, d), (c, a), (d, b), (e, d), (a, a), (b, b), (c, c), (d, d), (e, e), (f, f)\}$.

(vi) Symmetric closure of R is $R \cup R^{-1} = \{(a, c), (b, d), (c, a), (d, b), (e, d), (d, e)\}$

For Transitive closure we use Warshall's algorithm.

The matrix of the relation R $\qquad M_R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$

Step 1. We set $W_0 = M_R$, i.e. ,

$$W_0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Step 2. Construct W_1. First transfer all 1's of $W_0$to W_1

$$W_1 = \begin{bmatrix} & & 1 & & \\ & & & 1 & \\ 1 & & & & \\ & 1 & & & \\ & & & 1 & \end{bmatrix}.$$

In column 1 of W_0: Nonzero entry at position 3.

In row 1 of W_0: Nonzero entry at position 3.

So, at the position (3, 3) of W_1make the entries 1. Rest are zero. Therefore

$$W_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Step 3. Construct W_2. First transfer all 1's of $W_1$to W_2

In column 2 of W_1: Nonzero entry at position 4.

In row 2 of W_1: Nonzero entry at position 4

So, at the position $(4,4)$ of W_2 make the entries 1. Rest are zero. Therefore

$$W_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly

Step 4. Construct W_3.

$$W_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Step 5. Construct W_4.

$$W_4 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Step 6. Construct W_5.

$$W_5 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

From W_5, we can conclude that the transitive closure of R is:

$$\{(a,a),(a,c),(b,b),(b,d),(c,a),(c,c),(d,b),(d,d),(e,b),(e,d)\}.$$

Equivalence Relations

Definition: A relation on a set A is called an *equivalence relation* if it is reflexive, symmetric, and transitive. Two elements a and b that are related by an equivalence relation are called *equivalent*. The notation $a \sim b$ is often used to denote that a and b are equivalent elements with respect to a particular equivalence relation.

Example: Let R be the relation on the set of integers such that $(a,b) \in R$ if and only if $a = b$ or $a = -b$. Then R is reflexive, symmetric, and transitive. It follows that R is an equivalence relation.

Example: Let R be the relation on the set of real numbers such that $(a,b) \in R$ if and only if $a - b$ is an integer. Is R an equivalence relation?

Solution: Because $a - a = 0$ is an integer for all real numbers a, $(a,a) \in R$ for all real numbers a. Hence, R is reflexive. Now suppose that $(a,b) \in R$. Then $a - b$ is an integer, so $b - a$ is also an integer. Hence, $(b,a) \in R$. It follows that R is symmetric. If $(a,b) \in R$ and $(b,a) \in R$, then $a - b$ and $b - c$ are integers. Therefore, $a - c = (a - b) + (b - c)$ is also an integer. Hence, $(a,c) \in R$. Thus, R is transitive. Consequently, R is an equivalence relation.

Example: Congruence Modulo $\boldsymbol{m}$. Let m be an integer with $m > 1$. Show that the relation $R = \{(a,b) \mid a \equiv b \pmod{m}\}$ is an equivalence relation on the set of integers.