Discrete Mathematics

Chapter	*Core*	*Optional CS*	*Optional Math*
1	1.1–1.8 (as needed)		
2	2.1–2.4, 2.6 (as needed)		2.5
3		3.1–3.3 (as needed)	
4	4.1–4.4 (as needed)	4.5, 4.6	
5	5.1–5.3	5.4, 5.5	
6	6.1–6.3	6.6	6.4, 6.5
7	7.1	7.4	7.2, 7.3
8	8.1, 8.5	8.3	8.2, 8.4, 8.6
9	9.1, 9.3, 9.5	9.2	9.4, 9.6
10	10.1–10.5		10.6–10.8
11	11.1	11.2, 11.3	11.4, 11.5
12		12.1–12.4	
13		13.1–13.5	

> *Discrete mathematics provides the mathematical foundations for many computer science courses, including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems. Students find these courses much more difficult when they have not had the appropriate mathematical foundations from discrete mathematics. One student sent me an e-mail message saying that she used the contents of this book in every computer science course she took!*

> *Although there are plenty of exercises in this text similar to those addressed in the examples, a large percentage of the exercises require **original thought***

> *The material discussed in the text provides the tools needed to solve these exercises, but your job is to successfully apply these tools using your own creativity. One of the primary goals of this course is to learn how to attack problems that may be somewhat different from any you may have previously seen.*

You will learn the most by actively working exercises. I suggest that you solve as many as you possibly can

The more work you do yourself rather than passively reading or copying solutions, the more you will learn.

> *Most of you will return to this book as a useful tool throughout your future studies, especially for those of you who continue in computer science, mathematics, and engineering.*

1.1 Propositional Logic

> *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.*

1.1.2 Proposition

1. A proposition is a **declarative sentence** (that is, a sentence that declares a fact) that is **either**
   true or false, but not both.
2. The conventional letters used for propositional variables are **p, q, r, s,....**
3. Propositions that **cannot** be expressed in terms of simpler propositions are called **atomic propositions.**
4. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

> ### *Definition 1*
>
> *Let p be a proposition. The negation of p, denoted by ¬p (also denoted by p), is the statement "It*
> *is not the case that p."*
> *The proposition ¬p is read "not p." The truth value of the negation of p, ¬p, is the opposite of*
> *the truth value of p.*

Remark: The notation for the negation operator is not standardized. Although ¬p and p are the most common notations used in mathematics to express the negation of p, other notations you might see are ~p, -p, p′, Np, and !p.

> ### *Definition 2*
>
> *Let p and q be propositions. The conjunction of p and q, denoted by p ∧ q, is the proposition "p and q." The conjunction p ∧ q is true when both p and q are true and is false otherwise.*

Note that 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."

Definition 3

Let p and q be propositions. The disjunction of p and q, denoted by 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

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

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

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

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

Definition 4

Let p and q be propositions. The `exclusive or` *of p and q, denoted by p ⊕ q (or p XOR q), is the proposition that is true when **exactly** one of p and q is true and is false otherwise*

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

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

1.1.3 Conditional Statements

several other important ways in which propositions can be combined

> ## *Definition 5*
>
> *Let p and q be propositions. The conditional statement $p \to q$ is the proposition "if p, then q." The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \to q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).*

TABLE 5 **The Truth Table for the Conditional Statement $p \to q$.**

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Example 1

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 running for office is "If I am elected, then I will lower taxes." If the politician is elected, voters would expect this politician to lower taxes. Furthermore, if the politician is not elected, then voters will not have any expectation that this person will lower taxes, although the person may have sufficient influence to cause those in power to lower taxes.

It is only when the politician is elected but does not lower taxes 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 $p \to q$.

It is sufficient relationship

Example 2

"If Juan has a smartphone, then 2 + 3 = 5" is true from the definition of a conditional statement, because its conclusion is true. (The truth value of the hypothesis does not matter then.)

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.

> *The proposition $q \to p$ is called the converse of $p \to q$. (逆命题)*
>
> *The contrapositive of $p \to q$ is the proposition $\neg q \to \neg p$.(逆否命题)*
>
> *The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$. (否命题)*
>
> *We will see that of these three conditional statements formed from $p \to q$, only the contrapositive always has the same truth value as $p \to q$*

> ## *Definition 6*
>
> *Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ 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.*

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

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

(same with bitwise ^)

"p is necessary and sufficient for q"

1.1.4 Truth Table of Compound Propositions

We use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

EXAMPLE 14 Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

TABLE 7 **The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.**

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

1.1.5 Precedence of Logical Operators

TABLE 8 **Precedence of Logical Operators.**

Operator	*Precedence*
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

1.1.6 Logic and Bit Operations

TABLE 9 **Table for the Bit Operators *OR*, *AND*, and *XOR*.**

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

> ### *Definition 7*
>
> *A bit string is a sequence of zero or more bits. The length of this string is the number of bits in the string.*

Exercise

> *Which of these sentences are propositions? What are the truth values of those that are propositions?*
>
> *a) Boston is the capital of Massachusetts.*
>
> *b) Miami is the capital of Florida.*
>
> *c) 2 + 3 = 5.*
>
> *d) 5 + 7 = 10.*
>
> *e) x + 2 = 11.*
>
> *f) Answer this question.*

a) is proposition, and it is T

b) is proposition, and it is F   (Tallahassee is it's capital)

c) is proposition, and it is T

d)is proposition, and it is F

e) is not proposition, the value of x is unknown

f) is not proposition, it is not a declarative sentence

> *What is the negation of each of these propositions?*
>
> *a) Linda is younger than Sanjay.*
>
> *b) Mei makes more money than Isabella.*
>
> *c) Moshe is taller than Monica.*
>
> *d) Abby is richer than Ricardo.*

a) Linda is not younger than Sanjay

b) Mei does not make more money than Isabella

c) Moshe isn't taller than Monica

d) Abby is not richer than Ricardo

> *What is the negation of each of these propositions?*

> *a) Mei has an MP3 player.*
>
> *b) There is no pollution in New Jersey.*
>
> *c) 2 + 1 = 3.*
>
> *d) The summer in Maine is hot and sunny.*

a) Mei does not have an MP3 player

b) There is pollution in New Jersey

c) $2 + 1 \neq 3$

~~d) The summer in Maine is not hot and sunny~~

d) The summer in Maine is not hot or it is not sunny

> *What is the negation of each of these propositions?*
>
> *a) Steve has more than 100 GB free disk space on his laptop.*
>
> *b) Zach blocks e-mails and texts from Jennifer.*
>
> *c) 7 · 11 · 13 = 999.*
>
> *d) Diane rode her bicycle 100 miles on Sunday.*

a) Steve has no more than 100 GB free disk space on his laptop

b) Zach doesn't block e-mails from Jennifer or he doesn't block texts from Jennifer

c) $7 * 11 * 13 \neq 999$

d) Dian didn't ride her bicycle 100 miles on Sunday

> *Suppose that during the most recent fiscal year, the annual revenue of Acme Computer was 138 billion dollars and its net profit was 8 billion dollars, the annual revenue of Nadir Software was 87 billion dollars and its net profit was 5 billion dollars, and the annual revenue of Quixote Media was 111 billion dollars and its net profit was 13 billion dollars. Determine the truth value of each of these propositions for the most recent fiscal year.*
>
> *a) Quixote Media had the largest annual revenue.*
>
> *b) Nadir Software had the lowest net profit and Acme Computer had the largest annual revenue.*
>
> *c) Acme Computer had the largest net profit or Quixote Media had the largest net profit.*
>
> *d) If Quixote Media had the smallest net profit, then Acme Computer had the largest annual revenue.*
>
> *e) Nadir Software had the smallest net profit if and only if Acme Computer had the largest annual revenue.*

a) F

b) $T \wedge T$, T

c) $F \vee T$, T

d) $F \implies T$, T

e) $T \longleftrightarrow T$, T

> *Let p and q be the propositions "Swimming at the New Jersey shore is allowed" and "Sharks have been spotted near the shore," respectively. Express each of these compound propositions as an English sentence.*
>
> *a)* $\neg q$
>
> *b)* $p \wedge q$
>
> *c)* $\neg p \vee q$
>
> *d)* $p \to \neg q$
>
> *e)* $\neg q \to p$
>
> *f)* $\neg p \to \neg q$
>
> *g)* $p \leftrightarrow \neg q$
>
> *h)* $\neg p \wedge (p \vee \neg q)$

a) Sharks haven't been spotted near the shore

b) Swimming at the New Jersey shore is allowed, and sharks have been spotted near the shore

c) Swimming at the New Jersey shore is not allowed, or sharks have been spotted near the shore

d) If Swimming at the New Jersey shore is allowed, then sharks haven't been spotted near the shore

e)If Sharks haven't been spotted near the shore, then Swimming at the New Jersey shore is allowed

f) if Swimming at the New Jersey shore is not allowed, then Sharks haven't been spotted near the shore

g) Swimming at the New Jersey shore is allowed if and only if Sharks haven't been spotted near the shore

~~h)Swimming at the New Jersey shore is not allowed or Swimming at the New Jersey shore is not allowed and Sharks haven't been spotted near the shore~~

h) Swimming at the New Jersey shore is not allowed, and either swimming at the New Jersey shore is allowed or sharks have not been spotted near the shore. (Note that we were able to incorporate the parentheses by using the word "either" in the second half of the sentence.)

> *Let p and q be the propositions p: It is below freezing. q: It is snowing. Write these propositions using p and q and logical connectives (including negations).*
>
> *a) It is below freezing and snowing.*
>
> *b) It is below freezing but not snowing.*
>
> *c) It is not below freezing and it is not snowing.*
>
> *d) It is either snowing or below freezing (or both).*
>
> *e) If it is below freezing, it is also snowing.*
>
> *f) Either it is below freezing or it is snowing, but it is not snowing if it is below freezing.*
>
> *g) That it is below freezing is necessary and sufficient for it to be snowing.*

a) $p \wedge q$

b) $p \wedge \neg q$

c) $\neg p \wedge \neg q$

d) $p \vee q$

e) $p \implies q$

f) $(p \vee q) \wedge (p \implies \neg q)$

g) $p \longleftrightarrow q$

> *Let p and q be the propositions*
>
> *p: You drive over 65 miles per hour.*
>
> *q: You get a speeding ticket.*
>
> *Write these propositions using p and q and logical connectives (including negations).*
>
> *a) You do not drive over 65 miles per hour.*
>
> *b) You drive over 65 miles per hour, but you do not get a speeding ticket.*
>
> *c) You will get a speeding ticket if you drive over 65 miles per hour.*
>
> *d) If you do not drive over 65 miles per hour, then you will not get a speeding ticket.*
>
> *e) Driving over 65 miles per hour is sufficient for getting a speeding ticket.*
>
> *f) You get a speeding ticket, but you do not drive over 65 miles per hour.*
>
> *g) Whenever you get a speeding ticket, you are driving over 65 miles per hour.*

a) $\neg p$

b) $p \wedge \neg q$

c) $p \implies q$

d) $\neg p \implies \neg q$

e) $p \implies q$

f) $q \land \neg p$

~~g) p → q~~

g) q → p

> *Let p, q, and r be the propositions*
>
> *p: Grizzly bears have been seen in the area.*
>
> *q: Hiking is safe on the trail.*
>
> *r: Berries are ripe along the trail.*
>
> *Write these propositions using p, q, and r and logical connectives (including negations).*
>
> *a) Berries are ripe along the trail, but grizzly bears have not been seen in the area.*
>
> *b) Grizzly bears have not been seen in the area and hiking on the trail is safe, but berries are ripe along the trail.*
>
> *c) If berries are ripe along the trail, hiking is safe if and only if grizzly bears have not been seen in the area.*
>
> *d) It is not safe to hike on the trail, but grizzly bears have not been seen in the area and the berries along the trail are ripe.*
>
> *e) For hiking on the trail to be safe, it is necessary but not sufficient that berries not be ripe along the trail and for grizzly bears not to have been seen in the area.*
>
> *f) Hiking is not safe on the trail whenever grizzly bears have been seen in the area and berries are ripe along the trail.*

a) $r \land \neg p$

b) $\neg p \land q \land r$

c) $r \implies (q \longleftrightarrow \neg p)$

d) $\neg q \land \neg p \land r$

~~e) ¬p ∧ ¬r → q~~

e) (q →(¬r ∧ ¬p)) ∧ ¬((¬r ∧ ¬p) → q)

f) $p \land r \to \neg q$

> *Determine whether each of these conditional statements is true or false.*
>
> *a) If 1 + 1 = 2, then 2 + 2 = 5.*
>
> *b) If 1 + 1 = 3, then 2 + 2 = 4.*

> *c) If 1 + 1 = 3, then 2 + 2 = 5.*
>
> *d) If monkeys can fly, then 1 + 1 = 3.*

a) (T $\rightarrow$ F) F

b) (F $\rightarrow$ T) T

c) (F $\rightarrow$ F) T

d) (F $\rightarrow$ F) T

> *21*
>
> *For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.*
>
> *a) Coffee or tea comes with dinner.*
>
> *b) A password must have at least three digits or be at least eight characters long.*
>
> *c) The prerequisite for the course is a course in number theory or a course in cryptography.*
>
> *d) You can pay using U.S. dollars or euros.*

a) Exclusive or: You get only one beverage.

b) Inclusive or: Long passwords can have any combination of symbols.

c) Inclusive or: A student with both courses is even more qualified.

d) Either interpretation possible; a traveler might wish to pay with a mixture of the two currencies, or the store may not allow that.

1.2 Applications of Propositional Logic

1.1.2 Translate English Sentence

In particular, English (and every other human language) is often ambiguous. Translating sentences into compound statements (and other types of logical expressions, which we will introduce later in this chapter) removes the ambiguity

1.2.3 System Specifications

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 4 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 $p \to q$. 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 $p \to q$ 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. ◀

1.2.4 Boolean Search

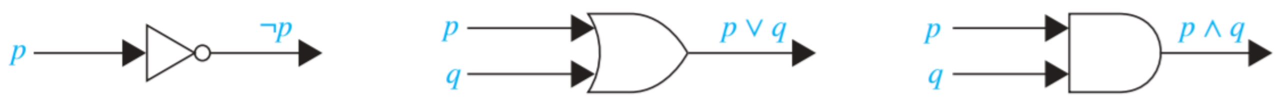
Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. Because these searches employ techniques from propositional logic, Links they are called Boolean searches.

1.2.5 Logical Puzzle

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 Links the Web, as a recreational activity

1.2.6 Logical Circuits

Logical Gates



Inverter OR gate AND gate

Exercises(some logical puzzle games)

1.3 Propositional Equivalence

1.3.1 Introduction

we will use the term “compound proposition” to refer to an expression formed from propositional variables using logical operators, such as p ∧ q.

> ### *Definition 1*
>
> *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**.*

TABLE 1 Examples of a Tautology and a Contradiction.

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

> ### *Definition 2*
>
> *The compound propositions p and q are called logically equivalent if **p ↔ q is a tautology**. The notation p ≡ q denotes that p and q are logically equivalent.*
>
> *(Compound propositions that **have the same truth values in all possible cases** are called logically Demo equivalent.)*

One way to determine whether two compound propositions are equivalent is to use a truth table. In particular, the compound propositions p and q are equivalent if and only if the columns Extra giving their truth values agree.

if p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent

TABLE 2 De Morgan's Laws.

$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

TABLE 3 Truth Tables for $\neg(p \vee q)$ and $\neg p \wedge \neg q$.

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

TABLE 4 Truth Tables for $\neg p \vee q$ and $p \rightarrow q$.

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

TABLE 5 **A Demonstration That $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ Are Logically Equivalent.**

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

Conclusions

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent 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
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	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
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

TABLE 7 **Logical Equivalences Involving Conditional Statements.**
$p \to q \equiv \neg p \vee q$
$p \to q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

TABLE 8 **Logical Equivalences Involving Biconditional Statements.**
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

1. Furthermore, note that De Morgan's laws extend to
   - $\neg(p_1 \vee p_2 \vee p_3 \vee \ldots . \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \neg p_3 \ldots . \wedge \neg p_n$
   - $\neg(p_1 \wedge p_2 \wedge p_3 \wedge \ldots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \neg p_3 \ldots . \vee \neg p_n$
2. Be careful not to apply logical identities, such as associative laws, distributive laws, or De Morgan's laws, to expressions that have a mix of conjunctions and disjunctions when the identities only apply when all these operators are the same.
3.

We will sometimes use the notation $\bigvee_{j=1}^{n} p_j$ for $p_1 \vee p_2 \vee \cdots \vee p_n$ and $\bigwedge_{j=1}^{n} p_j$ for $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. Using this notation, the extended version of De Morgan's laws can be written concisely as $\neg\left(\bigvee_{j=1}^{n} p_j\right) \equiv \bigwedge_{j=1}^{n} \neg p_j$ and $\neg\left(\bigwedge_{j=1}^{n} p_j\right) \equiv \bigvee_{j=1}^{n} \neg p_j$. (Methods for proving these identities will be given in Section 5.1.)

1.3.4 Construction New Logical Equivalences

1. use truth table
2. use logical identities that we already know to establish new logical identities

1.3.5 Satisfiability

> *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 assignments exists, that is, when the compound proposition is false for all assignments of truth values to its variables, the compound proposition is unsatisfiable.*
>
> *When we find a particular assignment of truth values that makes a compound proposition true, we have shown that it is satisfiable; such an assignment is called a solution of this particular satisfiability problem. However, to show that a compound proposition is unsatisfiable, we need to show that every assignment of truth values to its variables makes it false.*

1.3.6 Application of Satisfiability

please finish the two following proof

1. The n-Queens Problem
2. Sudoku(If you are so boring , you can write a program to auto-solve it)

Exercises

1.4 Predicates and quantifiers

1.4.2 Predicates

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.*

> *The statements that describe valid input are known as preconditions and the conditions that the output should satisfy when the program has run are known as postconditions.*

EXAMPLE 7 Consider the following program, designed to interchange the values of two variables x and y.

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

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

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

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

1.4.3 Quantifiers

The area of logic that deals with predicates and quantifiers is called the predicate calculus.

Universal quantifiers

> ## *Definition*
>
> *The universal quantification of $P(x)$ is the statement*
>
> *"$P(x)$ for all values of x in the domain."*
>
> *The notation $\forall xP(x)$ denotes the universal quantification of $P(x)$. Here $\forall$ is called the universal quantifier. We read $\forall xP(x)$ as "for all $xP(x)$" or "for every $xP(x)$." An element for which $P(x)$ is false is called a counterexample to $\forall xP(x)$.*

The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined.

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. Note that if the domain is empty, then $\forall xP(x)$ is true for any propositional function $P(x)$ because there are no elements x in the domain for which $P(x)$ is false.

TABLE 1 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.

Existential quantifier

> ## *Definition*
>
> *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 ∃xP(x) for the existential quantification of P(x). Here ∃ is called the existential quantifier*

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

1.44 Quantifiers over finite domains

EXAMPLE 15 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. ◀

Similarly, when the elements of the domain are $x_1, x_2, \ldots, x_n$, where n is a positive integer, the existential quantification $\exists x P(x)$ is the same as the disjunction

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

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

Connection between quantification and looping

It is sometimes helpful to think in terms of looping and searching when determining the truth value of a quantification. Suppose that there are n objects in the domain for the variable x. To determine whether ∀xP(x) is true, we can loop through all n values of x to see whether P(x) is always true. If we encounter a value x for which P(x) is false, then we have shown that ∀xP(x) is false. Otherwise, ∀xP(x) is true. To see whether ∃xP(x) is true, we loop through the n values of x searching for a value for which P(x) is true. If we find one, then ∃xP(x) is true. If we never find such an x, then we have determined that ∃xP(x) is false.

(Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)

1.4.5 Quantifiers with restricted domains

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

1.4.6 Precedence of Quantifiers

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

1.4.7 Logical equivalences involving quantifiers

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

EXAMPLE 19 Show that $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent (where the same domain is used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a disjunction, nor can we distribute an existential quantifier over a conjunction. (See Exercises 52 and 53.)

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

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

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

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

◀

Negating Quantified Expressions

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

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

TABLE 2 De Morgan's Laws for Quantifiers.

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

1.4.10 Translating from English into logical Expression

EXAMPLE 24 Express the statements "Some student in this class has visited Mexico" and "Every student in this class has visited either Canada or Mexico" using predicates and quantifiers.

Solution: The statement "Some student in this class has visited Mexico" means that

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

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

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

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

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

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

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

Similarly, the second statement can be expressed as

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

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

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

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

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

1.4.11 Using Quantifiers in System Specifications

1.4.12 Examples from Lewis Carroll

1.4.13 Logic Programming

Prolog and lisp

Exercise

1.5 Nested Quantifiers

1.5.1 Introduction

nested quantifiers, where one quantifier is within the scope of another, such as ∀x∃y(x + y = 0).

For example, ∀x∃y(x + y = 0) is the same thing as ∀x Q(x), where Q(x) is ∃y P(x, y), where P(x, y) is x + y = 0.

THINKING OF QUANTIFICATION AS LOOPS

In working with quantifications of more than one variable, it is sometimes helpful to think in terms of nested loops. (If there are infinitely many elements in the domain of some variable, we cannot actually loop through all values. Nevertheless, this way of thinking is helpful in understanding nested quantifiers.) For example, to see whether ∀x∀y P(x, y) is true, we loop through the values for x, and for each x we loop through the values for y. If we find that for all values of x that P(x, y) is true for all values of y,we have determined that ∀x∀y P(x, y) is true. If we ever hit a value x for which we hit a value y for which P(x, y) is false, we have shown that ∀x∀y P(x, y) is false

1.5.3 The order of quantifiers

It is important to note that the order of the quantifiers is important, unless all the quantifiers are universal quantifiers or all are existential quantifiers.

EXAMPLE 4 Let $Q(x, y)$ denote "$x + y = 0$." What are the truth values of the quantifications $\exists y \forall x Q(x, y)$ and $\forall x \exists y Q(x, y)$, where the domain for all variables consists of all real numbers?

Solution: The quantification

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

denotes the proposition

"There is a real number y such that for every real number x, $Q(x, y)$."

No matter what value of y is chosen, there is only one value of x for which $x + y = 0$. Because there is no real number y such that $x + y = 0$ for all real numbers x, the statement $\exists y \forall x Q(x, y)$ is false.

The quantification

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

denotes the proposition

"For every real number x there is a real number y such that $Q(x, y)$."

Given a real number x, there is a real number y such that $x + y = 0$; namely, $y = -x$. Hence, the statement $\forall x \exists y Q(x, y)$ is true. ◀

TABLE 1 Quantifications of Two Variables.

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

1.5.4 Translating Mathematical Statements into Statements Involving Nested Quantifiers

1.5.5 Translating from Nested Quantifiers into English

Expressions with nested quantifiers expressing statements in English can be quite complicated. The first step in translating such an expression is to write out what the quantifiers and predicates in the expression mean. The next step is to express this meaning in a simpler sentence.

1.5.6 Translating English Sentences into Logical Expressions

1.5.7 Negating Nested Quantifiers

Statements involving nested quantifiers can be negated by successively applying the rules for negating statements involving a single quantifier.

Exercises

1.6 Rules of inference

1.6.1 Introduction

1.6.2 Valid Arguments in Propositional Logic

> ## *Definition 1*
>
> *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 the truth of all its premises 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 if no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true.*

Remark: From the definition of a valid argument form we see that the argument form with premises p1, p2, ... , p n and conclusion q is valid exactly when $(p_1 \wedge p_2 \wedge \cdots \wedge p_n)$ → q is a tautology.

The key to showing that an argument in propositional logic is valid is to show that its argument form is valid.

1.6.3 Rules of Inference for Propositional Logic

The tautology (p ∧ (p → q)) → q is the basis of the rule of inference called modus ponens, or the law of detachment.

$$\begin{array}{rl} & p \\ & p \to q \\ \hline \therefore & q \end{array}$$

EXAMPLE 1 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. ◀

EXAMPLE 2 Determine whether the argument given here is valid and determine whether its conclusion must be true because of the validity of the argument.

"If $\sqrt{2} > \frac{3}{2}$, then $\left(\sqrt{2}\right)^2 > \left(\frac{3}{2}\right)^2$. We know that $\sqrt{2} > \frac{3}{2}$. Consequently, $\left(\sqrt{2}\right)^2 = 2 > \left(\frac{3}{2}\right)^2 = \frac{9}{4}$."

Solution: Let p be the proposition "$\sqrt{2} > \frac{3}{2}$" and q the proposition "$2 > (\frac{3}{2})^2$." The premises of the argument are $p \to q$ and p, and q is its conclusion. This argument is valid because it is constructed by using modus ponens, a valid argument form. However, one of its premises, $\sqrt{2} > \frac{3}{2}$, is false. Consequently, we cannot conclude that the conclusion is true. Furthermore, note that the conclusion of this argument is false, because $2 < \frac{9}{4}$. ◀

TABLE 1 **Rules of Inference.**

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

1.6.4 Using Rules of Inference to Build Arguments

EXAMPLE 6 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."

Extra Examples

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 $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$. The conclusion is simply t. We need to give a valid argument with premises $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$ and conclusion t.

We construct an argument to show that our premises lead to the desired conclusion as follows.

Step	Reason
1. $\neg p \wedge q$	Premise
2. $\neg p$	Simplification using (1)
3. $r \to p$	Premise
4. $\neg r$	Modus tollens using (2) and (3)
5. $\neg r \to s$	Premise
6. s	Modus ponens using (4) and (5)
7. $s \to t$	Premise
8. t	Modus ponens using (6) and (7)

1.6.5 Resolution

Many of these programs make use of a rule of inference known as resolution. This rule of inference is based on the tautology $((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$.

1.6.6 Fallacies

EXAMPLE 10 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 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.) ◀

The proposition $((p \to q) \wedge \neg p) \to \neg q$ 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**.

1.6.7 Rules of Inference for Quantified Statements

TABLE 2 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 12 Show that the premises "Everyone in this discrete mathematics class has taken a course in computer science" and "Marla is a student in this class" imply the conclusion "Marla has taken a course in computer science."

Extra Examples

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 $\forall x(D(x) \to C(x))$ and $D(\text{Marla})$. The conclusion is $C(\text{Marla})$.

The following steps can be used to establish the conclusion from the premises.

Step	**Reason**
1. $\forall x(D(x) \to C(x))$	Premise
2. $D(\text{Marla}) \to C(\text{Marla})$	Universal instantiation from (1)
3. $D(\text{Marla})$	Premise
4. $C(\text{Marla})$	Modus ponens from (2) and (3)

◀

1.6.8 Combining Rules of Inference for Propositions and Quantified Statements

EXAMPLE 14 Assume that "For all positive integers n, if n is greater than 4, then n^2 is less than 2^n" is true. Use universal modus ponens to show that $100^2 < 2^{100}$.

Solution: Let $P(n)$ denote "$n > 4$" and $Q(n)$ denote "$n^2 < 2^n$." The statement "For all positive integers n, if n is greater than 4, then n^2 is less than 2^n" can be represented by $\forall n(P(n) \to Q(n))$, where the domain consists of all positive integers. We are assuming that $\forall n(P(n) \to Q(n))$ is true. Note that $P(100)$ is true because $100 > 4$. It follows by universal modus ponens that $Q(100)$ is true, namely, that $100^2 < 2^{100}$. ◀

Exercise

1. Find the argument form for the following argument and determine whether it is valid. Can we conclude that the conclusion is true if the premises are true?

If Socrates is human, then Socrates is mortal.
Socrates is human.

∴ Socrates is mortal.

$$P \longrightarrow Q$$

$$P$$

$$------$$

$$Q$$

valid

3. What rule of inference is used in each of these arguments?
   a) Alice is a mathematics major. Therefore, Alice is either a mathematics major or a computer science major.
   b) Jerry is a mathematics major and a computer science major. Therefore, Jerry is a mathematics major.
   c) If it is rainy, then the pool will be closed. It is rainy. Therefore, the pool is closed.
   d) If it snows today, the university will close. The university is not closed today. Therefore, it did not snow today.
   e) If I go swimming, then I will stay in the sun too long. If I stay in the sun too long, then I will sunburn. Therefore, if I go swimming, then I will sunburn.

a) addition

b) simplification

c) modus ponens

d) modus tollens

e) hypothesis syllogism

> 5. *Use rules of inference to show that the hypotheses "Randy works hard," "If Randy works hard, then he is a dull boy," and "If Randy is a dull boy, then he will not get the job" imply the conclusion "Randy will not get the job."*

Randy works hard P

Randy is a dull boy Q

Randy will not get the job R

$$P \tag{P}$$

$$P \longrightarrow Q \tag{P}$$
