College of Computer Science
DMMMSU-SLUC
Agoo, La Union

CHAPTER 1
Logic Fundamentals

PROF. KATHRINA T. DACANAY
Instructor, Discrete Structures

COURSE OUTLINE

- Propositional Logic
- Applications of Propositional Logic
- Propositional Equivalences
- Predicates and Quantifiers
- Rules of Inference

INTRODUCTION

The rules of logic specify the meaning of mathematical statements. Logic is the basis of all mathematical reasoning, and of all automated reasoning. It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, and to other areas of computer science, as well as to many other fields of study.

PROPOSITIONAL LOGIC

INTRODUCTION

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.

INTRODUCTION

The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments. 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.

PROPOSITIONS

Propositions are the basic building blocks of logic. A **proposition** is a declarative sentence that is either true or false, but not both.

PROPOSITIONS

Examples of propositions:

1. Manila is the capital of the Philippines.
2. 1 + 1 = 2.
3. 2 + 2 = 3.
4. 7 != 7.
5. The earth is spherical.

PROPOSITIONS

Examples of sentences that are NOT propositions:

1. What is your name?
2. Read this carefully.
3. Kindly close the door.
4. $x + 1 = 2$.
5. $x + y = z$.

YOUR TURN

Which of these sentences are propositions? What are the truth values of those that are propositions?

1. Go when the traffic light is green.
2. 121 is a perfect square.
3. Ninoy Aquino is the current president of the Philippines.
4. Answer this question.
5. $2 + 3 = 5$.
6. 5 is less than 7.
7. $x + 2 = 11$.

PROPOSITIONAL VARIABLES

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, and so on.

TRUTH VALUE

The **truth value** of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition.

COMPOUND PROPOSITIONS

New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

LOGICAL OPERATORS

1. Negation
2. Conjunction
3. Disjunction
4. Exclusive OR
5. Conditional Statements
6. Biconditional Statements

LOGICAL OPERATORS
Negation

- Let p be a proposition. The negation of p, denoted by ¬p (also denoted by $\overline{p}$), is the statement “It is not the case that p.”
- The proposition ¬p is read as “not p.” The truth value of the negation of p, ¬p, is the opposite of the truth value of p.

p	¬p
T	F
F	T

LOGICAL OPERATORS

Conjunction

- 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.

p	q	p ∧ q	q ∧ p
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

LOGICAL OPERATORS
Disjunction

- Let p and q be propositions. The disjunction of p and q, denoted by p v q, is the proposition "p or q". The disjunction p v q is false when both p and q are false and is true otherwise.

p	q	p v q	q v p
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

LOGICAL OPERATORS
Exclusive OR

- Let p and q be propositions. The exclusive OR of p and q, denoted by p ⊕ q, is the proposition that is true when exactly one of p and q is true and is false otherwise.

p	q	p ⊕ q	q ⊕ p
T	T	F	F
T	F	T	T
F	T	T	T
F	F	F	F

LOGICAL OPERATORS
Conditional Statements

- Let p and q be propositions. The conditional statement $p \rightarrow q$ 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 (or antecedent or premise) and q is called the conclusion (or consequence).

LOGICAL OPERATORS
Conditional Statements

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

LOGICAL OPERATORS
Biconditional Statements

- 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.

LOGICAL OPERATORS
Biconditional Statements

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

YOUR TURN

Let p and q be the propositions "The election is decided" and "The votes have been counted," respectively. Express each of these compound propositions as an English sentence.

a) $\neg p$

b) $p \vee q$

c) $\neg p \wedge q$

d) $q \rightarrow p$

e) $\neg q \rightarrow \neg p$

f) $\neg p \rightarrow \neg q$

g) $p \leftrightarrow q$

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

YOUR TURN

Let *p*, *q*, and *r* be the propositions

p :You have the flu.

q :You miss the final examination.

r :You pass the course.

Express each of these propositions as an English sentence.

a) $p \rightarrow q$

b) $\neg q \leftrightarrow r$

c) $q \rightarrow \neg r$

d) $p \vee q \vee r$

e) $(p \rightarrow \neg r) \vee (q \rightarrow \neg r)$

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

Precedence of Logical Operators

Operator	Precedence
¬	1
∧ ∨	2 3
→ ↔	4 5

Truth Table of Compound Proposition (p ∨¬q) → (p ∧ q)

p	q	¬q	p ∨¬q	p ∧ q	(p ∨¬q) → (p ∧ 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

YOUR TURN

Show the truth table of the following compound propositions:

1. $(p \vee q) \wedge p$
2. $\overline{p} \oplus (p \wedge q)$
3. $\overline{p} \rightarrow \overline{q}$
4. $(p \rightarrow \overline{r}) \vee (q \rightarrow \overline{r})$

CONVERSE, CONTRAPOSITIVE, AND INVERSE

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.

CONVERSE, CONTRAPOSITIVE, AND INVERSE

- The **converse** of $p \rightarrow q$ is the proposition $q \rightarrow p$.
- The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.
- The **inverse** of $p \rightarrow q$ is the proposition $\neg p \rightarrow \neg q$.

CONVERSE, CONTRAPOSITIVE, AND INVERSE

We will see that of these three conditional statements formed from

$p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

LOGIC AND BIT OPERATIONS

- Computers represent information using bits. A **bit** is a symbol with two possible values, namely, 0 (zero) and 1 (one).
- A bit can be used to represent a truth value, because there are two truth values, namely, true and false.
- A variable is called a **Boolean variable** if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

LOGIC AND BIT OPERATIONS

- Computer bit operations correspond to the logical connectives. By replacing true by a one and false by a zero in the truth tables for the operators $\wedge$, $\vee$, and $\oplus$, for the corresponding bit operations are obtained.
- The notation OR, AND, and XOR will be used for the operators $\vee$,$\wedge$, and $\oplus$, as is done in various programming languages.

Truth Table for the Bit Operators OR, AND, and XOR

x	y	x ∨ y	x ∧ y	x ⊕ y
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

BIT STRING

- A bit string is a sequence of zero or more bits. The length of this string is the number of bits in the string.
- 101010011 is a bit string of length nine.
- We can extend bit operations to bit strings. We define the bitwise OR, bitwise AND, and bitwise XOR of two strings of the same length to be the strings that have as their bits the OR, AND, and XOR of the corresponding bits in the two strings, respectively.

BIT STRING
Example

01 1011 0110

11 0001 1101

11 1011 1111 bitwise OR

01 0001 0100 bitwise AND

10 1010 1011 bitwise XOR

YOUR TURN

Find the bitwise OR, bitwise AND, and bitwise XOR of each of these pairs of bit strings.

1. 101 1110, 010 0001
2. 1111 0000, 1010 1010
3. 00 0111 0001, 10 0100 1000
4. 11 1111 1111, 00 0000 0000

YOUR TURN

Evaluate each of these expressions:

1. 1 1000 ∧ (0 1011 ∨ 1 1011)
2. (0 1111 ∧ 1 0101) ∨ 0 1000
3. (0 1010 ⊕ 1 1011) ⊕ 0 1000
4. (1 1011 ∨ 0 1010) ∧ (1 0001 ∨ 1 1011)

APPLICATIONS OF PROPOSITIONAL LOGIC

INTRODUCTION

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.

APPLICATIONS

1. Translating English Sentences
2. System Specifications
3. Boolean Searches
4. Logic Puzzles
5. Logic Circuits

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. Translating sentences into compound statements removes the ambiguity. Note that this may involve making a set of reasonable assumptions based on the intended meaning of the sentence.

TRANSLATING ENGLISH SENTENCES

Moreover, once we have translated sentences from English into logical expressions we can analyze these logical expressions to determine their truth values, we can manipulate them, and we can use rules of inference to reason about them.

EXAMPLE

English Sentence

"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

EXAMPLE

Solution

There are many ways to translate this sentence into a logical expression. Although it is possible to represent the sentence by a single propositional variable, such as p, this would not be useful when analyzing its meaning or reasoning with it. Instead, we will use propositional variables to represent each sentence part and determine the appropriate logical connectives between them.

EXAMPLE

Solution

In particular, we let a, c, and f represent "You can access the Internet from campus," "You are a computer science major," and "You are a freshman," respectively. Noting that "only if" is one way a conditional statement can be expressed, this sentence can be represented as $a \rightarrow (c \lor \neg f)$.

SYSTEM SPECIFICATIONS

Translating sentences in natural language 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

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

System Specifications

1. "The diagnostic message is stored in the buffer or it is retransmitted."
2. "The diagnostic message is not stored in the buffer."
3. "If the diagnostic message is stored in the buffer, then it is retransmitted."

Determine whether these system specifications are consistent.

EXAMPLE

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.

EXAMPLE

Solution

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.

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 is used to exclude a particular search term.

LOGIC PUZZLES

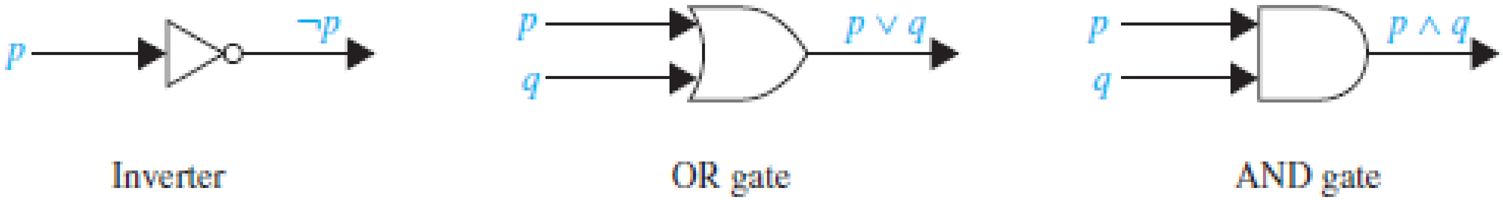
Puzzles that can be solved using logical reasoning are known as logic puzzles. Solving logic puzzles is an excellent way to practice working with the rules of logic. Also, computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities.

EXAMPLE

Smullyan posed many puzzles about an island that 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?"

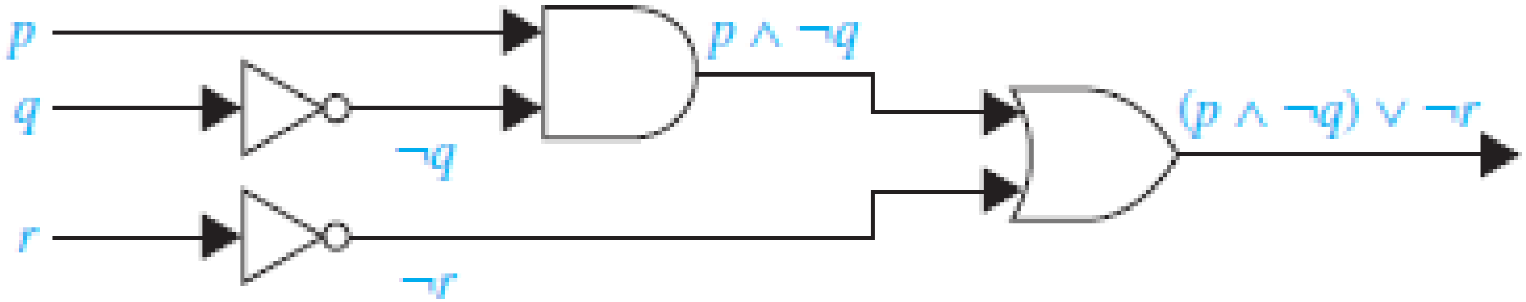
LOGIC CIRCUITS

A logic circuit (or digital circuit) receives input signals p1, p2, . . . , p_n, each a bit [either 0 (off) or 1 (on)], and produces output signals s1, s2, . . . , s_n, each a bit. In general, digital circuits may have multiple outputs.



EXAMPLE

(p ∧ ¬q) ∨ ¬r



PROPOSITIONAL EQUIVALENCES

INTRODUCTION

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

CLASSIFICATION OF COMPOUND PROPOSITIONS

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

Tautologies and contradictions are often important in mathematical reasoning.

p	¬p	p ∨ ¬p	p ∧ ¬p
T	F	T	F
F	T	T	F

LOGICAL EQUIVALENCES

Compound propositions that have the same truth values in all possible cases are called **logically equivalent**. The compound propositions p and q are called logically equivalent if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

LOGICAL EQUIVALENCES

The symbol ≡ is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology.

EXAMPLE

¬(p v q) and ¬p ∧¬q

p	q	p v q	¬(p v q)	¬p	¬q	¬p ∧¬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

EXAMPLE 2

¬p v q and p → q

p	q	¬p	¬p v q	*p → q*
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

LOGICAL EQUIVALENCES

EQUIVALENCES	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
$(\overline{\overline{p}}) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws

LOGICAL EQUIVALENCES

EQUIVALENCES	NAME
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	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
$\overline{(p \wedge q)} \equiv \overline{p} \vee \overline{q}$ $\overline{(p \vee q)} \equiv \overline{p} \wedge \overline{q}$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \overline{p} \equiv \mathbf{T}$ $p \wedge \overline{p} \equiv \mathbf{F}$	Negation laws

YOUR TURN

Show that the following compound propositions are logically equivalent:

1. $p \rightarrow q$ and $\overline{q} \rightarrow \overline{p}$
2. $p \leftrightarrow q$ and $(p \wedge q) \vee (\overline{p} \wedge \overline{q})$
3. $\overline{p \leftrightarrow q}$ and $p \leftrightarrow \overline{q}$
4. $(p \rightarrow q) \wedge (p \rightarrow r)$ and $p \rightarrow (q \wedge r)$
5. $(p \rightarrow r) \wedge (q \rightarrow r)$ and $(p \vee q) \rightarrow r$

PREDICATES AND QUANTIFIERS

INTRODUCTION

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*." No rules of propositional logic allow us to conclude the truth of the statement "*PC03 is functioning properly*."

PREDICATES

Statements involving variables, such as "$x > 3$", "$x = y + 3$", "$x + y = z$", "*computer* x *has a virus*", and "*computer y 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.

PREDICATES

The statement "x *is greater than 3*" has two parts: 1) x, is the ***subject*** and 2) "is greater than 3" is the ***predicate***. The predicate refers to a property that the subject of the statement can have.

PREDICATES

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.

EXAMPLES

Let P(x) denote the statement "x > 3." What are the truth values of:

1. P(4)
2. P(2)

Let *A(x)* denote the statement "x is a CCS instructor."

1. A(Macon Mendoza)
2. A(Franklin Quinzon)

QUANTIFIERS

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.

TYPES OF QUANTIFICATION

1. **Universal Quantification**. It tells us that a predicate is true for every element under consideration.
2. **Existential Quantification**. It tells us that there is one or more element under consideration for which the predicate is true.

UNIVERSAL QUANTIFICATION

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 xP(x)."* An element for which P(x) is false is called a **counterexample** of $\forall x P(x)$.

EXISTENTIAL QUANTIFICATION

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. The existential quantification ∃xP(x) is read as *“For some x P(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.

YOUR TURN

Let P(x) denote the statement "$x \leq 4$." What are these truth values?

1. $P(0)$
2. $P(-7)$
3. $P(4)$
4. $P(6)$

Let P(x) be the statement "the word x contains the letter a." What are these truth values?

1. *P*(orange)
2. *P*(lemon)
3. *P*(true)
4. *P*(false)

YOUR TURN

Let Q(x, y) denote the statement “x is the capital of y.” What are these truth values?

1. *P*(La Union, Agoo)
2. *P*(Manila, Philippines)
3. *P*(Japan, Tokyo)

Let N(x, y, z) denote the statement “x = y + z.” What are these truth values?

1. P(3,1,2)
2. P(5, 5, 0)
3. P(10, 8, 3)

RULES OF INFERENCE

INTRODUCTION

By an **argument**, we mean a sequence of statements that end with a conclusion. By **valid**, we mean that the conclusion must follow from the truth of the preceding statements, or **premises**, of the argument. That is, an argument is valid if and only if it is impossible for all the premises to be true and the conclusion to be false.

EXAMPLE OF AN ARGUMENT

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

and

"You have a current password."

Therefore,

"You can log onto the network."

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

DEFINITION OF TERMS

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.

DEFINITION OF TERMS

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.

RULES OF INFERENCE

RULE OF INFERENCE	NAME
p $\underline{p \to q}$ $\therefore q$	Modus ponens
$\neg q$ $\underline{p \to q}$ $\therefore \neg p$	Modus tollens
$p \to q$ $\underline{q \to r}$ $\therefore p \to r$	Hypothetical syllogism
$p \vee q$ $\underline{\neg p}$ $\therefore q$	Disjunctive syllogism

RULES OF INFERENCE

RULE OF INFERENCE	NAME
$\underline{p}$ $\therefore p \lor q$	Addition
$\underline{p \land q}$ $\therefore p$	Simplification
p $\underline{q}$ $\therefore p \land q$	Conjunction
$p \lor q$ $\underline{\neg p \lor r}$ $\therefore q \lor r$	Resolution

YOUR TURN

What rule of inference is used in each of these arguments?

1. Alice is a mathematics major. Therefore, Alice is either a mathematics major or a computer science major.
2. Jerry is a mathematics major and a computer science major. Therefore, Jerry is a mathematics major.
3. If it is rainy, then the pool will be closed. It is rainy. Therefore, the pool is closed.
4. If it snows today, the university will close. The university is not closed today. Therefore, it did not snow today.

YOUR TURN

5. If I go swimming, then I will stay in the sun too long. If I stay in the sun too long, then I will be sunburned. Therefore, if I go swimming, then I will sunburned.
6. Steve will work at a computer company this summer. Therefore, this summer Steve will work at a computer company or he will be a beach bum.
7. If I work all night on this homework, then I can answer all the exercises. If I answer all the exercises, I will understand the material. Therefore, if I work all night on this homework, then I will understand the material.

END OF CHAPTER 1