IS5311-Discrete Mathematics

Chapter 1-Introduction to Discrete Mathematics

Lecturer- MS. M.W.S. Randunu

Department of Interdisiplinary Studies,Faculty of Engineering, University of Ruhuna.

August 21, 2025

Introduction

Key areas of discrete mathematics include:

1. Sets and Relations
2. Logic
3. Combinatorics
4. Graph Theory
5. Number Theory
6. Discrete Probability

Importance of Discrete Mathematics

1. Foundation for Computer Science
2. Cryptography and Information Security
3. Networks and Communication Systems
4. Operations Research and Optimization
5. Artificial Intelligence and Machine Learning
6. Combinatorial Optimization
7. Logic and Reasoning

Propositional Logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Example

1. Wi-Fi is available in the building. **(T)**
2. Instagram is older than Facebook. **(F)**
3. $2 + 2 = 4$. **(T)**
4. Every smartphone has a headphone jack. **(F)**

Propositional Logic

Some examples of sentences that are not propositions.

1. Did the program run? (Question)
2. Charge your phone! (Command)
3. x is an even number (Open sentence)
4. $x + y = z$.

Key Concepts

- Propositional Variables
- Truth Values
- Propositional Calculus/Logic

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

Logical Operators

Negation($\neg$)

Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement "It is not the case that p." The proposition $\neg p$ is read "**not** p."

Negation

Example

1. Proposition: "The program is running."

   Negation: "It is **not** the case that the program is running."

   Simpler English: "The program is **not** running."

Negation

p	$\neg p$
T	F
F	T

Table: Truth table for the negation of a proposition p.

The negation operator constructs a new proposition from a single existing proposition. 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

Conjunction($\wedge$)

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p **and** q."

Conjunction

Example

- A: The sensor is working.
- B: The sensor is connected.

Then $A \wedge B$: The sensor is working and the sensor is connected.

Conjunction

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

Table: The Truth Table for the Conjunction of Two Propositions.

Disjunction

Disjunction ($\vee$)

Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q."

Disjunction

Example

- p: "The user pressed Enter."
- q: "The user clicked the button."

$p \vee q$: "The user pressed Enter **or** clicked the button."

Disjunction

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

Table: The Truth Table for the Disjunction of Two Propositions.

Exclusive

Exclusive($\oplus$)

Let p and q be propositions. The exclusive or of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

Exclusive

Example

- p: "The system uses a wired network."
- q: "The system uses a wireless network."

$p \oplus q$: "The system uses either a wired or wireless network, but **not both**.

Exclusive

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

Table: The Truth Table for the Exclusive Or of Two Propositions

Conditional Statements

Conditional Statements($\rightarrow$)

Let p and q be propositions. The conditional statement $p \rightarrow q$ is the proposition "**if** p**, then** q**.**"

Conditional Statements

A conditional statement, often referred to as an **"if-then"** statement, expresses a relationship between two propositions where the truth of one proposition depends on the truth of the other.

Example

1. "**If** it is raining outside, **then** I will bring an umbrella."
2. “**If** I am elected, **then** I will lower taxes.”
3. "**If** you get 100 on the final, **then** you will get an A."

Conditional Statements

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

Table: The Truth Table for the Conditional Statement $p \to q$.

Conditional Statements

Exercise:

1. Let p be the statement "The battery is low.", and q be the statement "The system sends a warning."Express $p \to q$ in English ?

Conditional Statements

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

- The proposition $q \to p$ is called the **converse** of $p \to q$.

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

Conditional Statements

- The **contrapositive** of $p \to q$ is the proposition $\neg q \to \neg p$.

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

- The proposition $\neg p \to \neg q$ is called the **inverse** of $p \to q$.

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

Conditional Statements

When two compound propositions always have the same truth value we call them **equivalent**.

- The conditional statement and its contrapositive are equivalent.
- The converse and the inverse of a conditional statement are also equivalent.

Conditional Statements

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

"If the firewall is active, then the network is protected."

Conditional Statements

Solution:

- The contrapositive of this conditional statement is

  "If the network is not protected, then the firewall is not active."

- The converse is

  "If the network is protected, then the firewall is active."

- The inverse is

  "If the firewall is not active, then the network is not protected."

Biconditional Statement

Biconditional Statement($\leftrightarrow$)

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition "p **if and only if** q."

Note that the statement $p \leftrightarrow q$ is true when both the conditional statements $p \to q$ and $q \to p$ are true and is false otherwise.

Biconditional Statement

Example:

Let p be the statement "You attend the meeting," and let q be the statement "You receive the meeting minutes."

Biconditional Statement $(p \leftrightarrow q)$:
"You can attend the meeting if and only if you receive the meeting minutes."

Truth Conditions:

- The statement is true if:
  - You attend the meeting and you receive the meeting minutes.
  - You do not attend the meeting and you do not receive the meeting minutes.
- The statement is false if:
  - You attend the meeting but do not receive the meeting minutes.
  - You do not attend the meeting but receive the meeting minutes.

Biconditional Statement

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

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

Biconditional Statement

Example: Determine whether each of the following bi-conditional statement is true or false.

No.	Statement	True/False
1	A number is divisible by 10 if and only if it ends in 0.	
2	A triangle is equilateral if and only if it has three equal angles.	
3	An animal is a mammal if and only if it lays eggs.	
4	$2 + 2 = 5$ if and only if $3 + 3 = 6$.	
5	A person is a bachelor if and only if they are an unmarried man.	
6	A square is a rhombus if and only if all its angles are right angles.	

Biconditional Statement

Solution:

No.	Statement	True/False
1	A number is divisible by 10 if and only if it ends in 0.	True
2	A triangle is equilateral if and only if it has three equal angles.	True
3	An animal is a mammal if and only if it lays eggs.	False
4	$2+2=5$ if and only if $3+3=6$.	False
5	A person is a bachelor if and only if they are an unmarried man.	True
6	A square is a rhombus if and only if all its angles are right angles.	False

Truth Tables of Compound Propositions

Example: Construct the truth table for the compound proposition $(p \wedge q) \vee r$.

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

Table: The Truth Table of $((p \wedge q) \vee r)$.

Truth Tables of Compound Propositions

Example: Construct the truth table for the compound proposition $(p \land \neg r) \lor (q \to r)$.

Solution:

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

Table: The Truth Table of $(p \land \neg r) \lor (q \to r)$.

Exercise:Construct a truth table for each of these compound propositions.

1. $p \vee \neg p$
2. $p \wedge q$
3. $p \vee q \wedge p$
4. $(p \vee q) \rightarrow (q \wedge r)$
5. $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$

Precedence of Logical Operators

Operator	Precedence
$\neg$	1
$\land$	2
$\lor$	3
$\rightarrow$	4
$\leftrightarrow$	5

Table: Precedence of Logical Operators.

Logic and Bit Operations

Computers represent information using bits. A bit is a symbol with two possible values: 0 (zero) and 1 (one). A bit can represent a truth value because there are two truth values: true and false. We use 1 to represent true (T) and 0 to represent false (F).

Truth Value	Bit
T	1
F	0

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

p	q	AND$(p \wedge q)$
1	1	1
1	0	0
0	1	0
0	0	0

p	q	OR$(p \vee q)$
1	1	1
1	0	1
0	1	1
0	0	0

p	q	XOR$(p \oplus q)$
1	1	0
1	0	1
0	1	1
0	0	0

Table: Tables for the Bit Operators OR, AND, and XOR.

Logic and Bit Operations

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.

Example: 11011000 is a bit string of length eight.

Example: Given two bit strings $01\,1011\,0110$ and $11\,0001\,1101$,find their bitwise OR, bitwise AND, and bitwise XOR.

Logic and Bit Operations

Solution: Bitwise OR:

$$\begin{array}{rr} & 01110110 \\ \vee & 11001101 \\ \hline & 11111111 \end{array}$$

Bitwise AND:

$$\begin{array}{rr} & 01110110 \\ \wedge & 11001101 \\ \hline & 01000100 \end{array}$$

Bitwise XOR:

$$\begin{array}{rr} & 01110110 \\ \oplus & 11001101 \\ \hline & 10111011 \end{array}$$

Logic and Bit Operations

Exercise:Evaluate each of these expressions.

1. $(101011 \oplus 110011) \wedge (011010 \vee 100110)$.
2. $(10101 \wedge 11011) \vee (10010 \oplus 01010)$.
3. $(111001 \wedge 100011) \oplus (110110 \vee 101010)$.
4. $(1111 \vee 0101) \wedge (1010 \oplus 0110)$.

Applications of Propositional Logic:Translating Statements

Translating English sentences into expressions with propositional variables and logical connectives serves several purposes. Once translated, we can analyze these expressions to determine their truth values.

Example:: Translate the following English sentence into a logical expression:

"If you upload your document and submit the form, then your assignment will be accepted."

Translating Statements

Solution: Let's represent:

- a: "You upload the document."
- b: "You submit the form."
- c: "Your assignment will be accepted"

So, we can translate the sentence into the logical expression

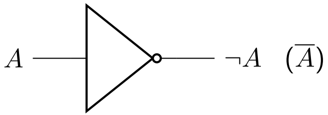
$$(a \wedge b) \to c.$$

Logic Circuits

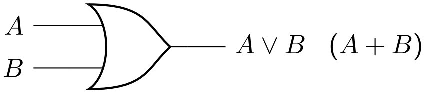
In a logic circuit, also known as a digital circuit, input signals $p_1, p_2, \ldots, p_n$ are received, each representing a bit (0 for "off" and 1 for "on"). The circuit processes these inputs and produces output signals $s_1, s_2, \ldots, s_n$, also represented as bits.

Complicated digital circuits can be constructed from **three basic circuits**, called **gates**.These gates are fundamental building blocks for constructing complex digital circuits. Here are the three basic types of gates typically used:

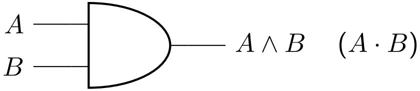
Logic Circuits



1. This gate, also known as an inverter, produces an output that is the complement of its input.



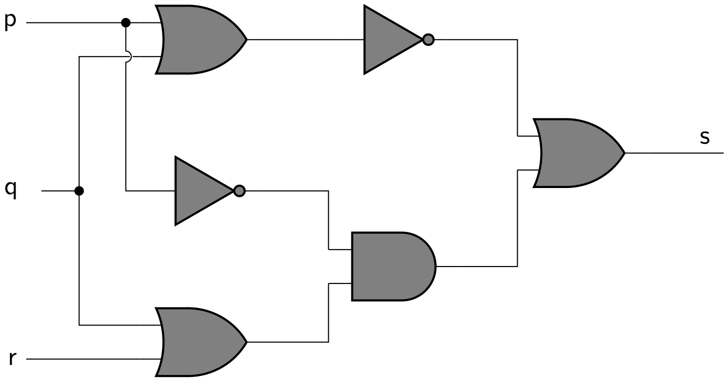
2. This gate produces a high output (1) if any of its inputs are high (1). It produces a low output (0) only if all its inputs are low (0).



3. This gate produces a high output (1) only if all of its inputs are high (1). Otherwise, it produces a low output (0).

Logic Circuits

Example: Determine the output for the combinatorial circuit in the following figures?



Logic Circuits

Let us consider the logical flow of the circuit to determine the output s based on the inputs p, q, and r.

- First OR Gate (ORa):
  - Inputs: p and q
  - Output: $p \vee q$
- **N**OT Gate (Nob):
  - Input: Output of ORa, which is $p \vee q$
  - Output: $\neg(p \vee q)$
- Second OR Gate (ORb):
  - Inputs: q and r
  - Output: $q \vee r$
- NOT Gate (Noa):
  - Input: q
  - Output: $\neg q$

Logic Circuits

- AND Gate (ANDa):
  - Inputs: Output of Noa ($\neg q$) and Output of ORb ($q \vee r$)
  - Output: $\neg q \wedge (q \vee r)$
- Third OR Gate (ORc):
  - Inputs: Output of Nob ($\neg(p \vee q)$) and Output of ANDa ($\neg q \wedge (q \vee r)$)
  - Output: $\neg(p \vee q) \vee (\neg q \wedge (q \vee r))$

Thus, the output of the circuit s is:

$$s = \neg(p \vee q) \vee (\neg q \wedge (q \vee r))$$

Logic Circuits

Example: Consider the following logical expression: $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$. Design a digital circuit that produces this output given input bits p, q, and r.

Propositional Equivalences

Tautology

A **tautology** is a compound proposition that is always true.

Example:

1. The proposition $p \vee \neg p$ is a tautology because it is always true.
2. You are online or not online.

Propositional Equivalences

Contradiction

In contrast, a **contradiction** is a compound proposition that is always false.

Example:

1. The proposition $p \wedge \neg p$ is a contradiction because it can never be true; it asserts that p is both true and false simultaneously.
2. You are online and not online at the same time.

Propositional Equivalences

Contingency

A **contingency** is a compound proposition that is neither a tautology nor a contradiction. Its truth value depends on the specific truth values assigned to its constituent propositional variables.

Example:

1. The proposition $p \vee q$ is a contingency because its truth value depends on the truth values of p and q. If both p and q are true, the proposition is true; if both are false, the proposition is false.
2. The sun is shining.

Logical Equivalences

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.

Table: De Morgan's Laws

Expression	**Equivalent Expression**
$\neg(p \wedge q)$	$\neg p \vee \neg q$
$\neg(p \vee q)$	$\neg p \wedge \neg q$

Logical Equivalences

Example: To show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent, we can use a truth table.

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

Table: Truth Table for $\neg(p \vee q)$ and $\neg p \wedge \neg q$

Exercise: Show that $p \to q$ and $\neg p \vee q$ are logically equivalent.

Logical Equivalences

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
$\neg(\neg p) \equiv p$	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

Table: Logical Equivalences

Logical Equivalences

Equivalence	**Name**
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	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
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation laws

Table: Logical Equivalences

Logical Equivalences

1. Using De Morgan' s Laws show that $\neg(p \rightarrow q)$ is logically equivalent to $p \wedge \neg q$.
2. Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent by developing a series of logical equivalences.
3. Show that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

Logical Equivalences

1.

$$\begin{aligned}\neg(p \to q) &\equiv \neg(\neg p \vee q) && \text{by definition of implication}\\ &\equiv \neg(\neg p) \wedge \neg q && \text{by the second De Morgan's law}\\ &\equiv p \wedge \neg q && \text{by the double negation law}\end{aligned}$$

2.

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan's law}\\ &\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan's law}\\ &\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law}\\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law}\\ &\equiv F \vee (\neg p \wedge \neg q) && \text{because } \neg p \wedge p \equiv F\\ &\equiv (\neg p \wedge \neg q) \vee F && \text{by the commutative law for disjunctio}\\ &\equiv \neg p \wedge \neg q && \text{by the identity law for } F\end{aligned}$$

Predicates and Quantifiers

Consider the statement

"x is greater than 3"

It consists of two parts:

1. The variable x, representing the subject,
2. And the predicate "is greater than 3" .

We denote this statement as $P(x)$, where

1. P represents the predicate "is greater than 3,"
2. x is the variable.

$P(x)$ is also known as the value of the propositional function P at x. Only when a specific value is assigned to x, $P(x)$ becomes a proposition with a definite truth value.

Predicates

Example: Let $Q(x)$ denote the statement "x is even." What are the truth values of $Q(4)$ and $Q(3)$?

Solution: For $Q(x)$ denoting the statement "x is even," the truth values are:

$Q(4)$: True, because 4 is an even number.

$Q(3)$: False, because 3 is not an even number.

Predicates

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 the truth values of $A(\text{CS1})$, $A(\text{CS2})$, and $A(\text{MATH1})$?

Quantifiers

Universal Quantifier

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 $x P(x)$."

Quantifiers

Example: Let $C(x)$ denote "x is a cat," and let $A(x)$ denote "x is an animal." The statement "All cats are animals" can be written as:

$$\forall x(C(x) \rightarrow A(x))$$

Explanation:

- $\forall x$: This is the universal quantifier, which means "for all x."
- $C(x) \rightarrow A(x)$: This is a conditional statement that reads as "if x is a cat, then x is an animal."

Quantifiers

Example: Let $Q(x)$ be the statement " $x < 2$." The truth value of the quantification $\forall x Q(x)$, where the domain consists of all real numbers, is as follows:
Since $Q(x)$ is not true for every real number x

$$Q(3) \text{ is false,}$$

$x = 3$ serves as a counterexample for the statement $\forall x Q(x)$.
Hence, $\forall x Q(x)$ is false.

Quantifiers

Existential Quantifier

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**.

Quantifiers

Example: Let $R(x)$ be the statement "Device x has more than 8GB of RAM." What is the truth value of the quantification $\exists x\, R(x)$, where the domain consists of all the computing devices in a laboratory?

Quantifiers

Solution: Let $R(x)$ denote the statement "Device x has more than 8GB of RAM." The truth value of the quantification $\exists x\, R(x)$, where the domain consists of all computing devices in a lab, is as follows: If there exists at least one device in the lab that has more than 8GB of RAM (e.g., a 16GB workstation), then the statement $\exists x\, R(x)$ is **true**.

Observe that $\exists x\, R(x)$ is **false** only if *no* device in the lab has more than 8GB of RAM. That is, it is false if every device in the lab has RAM less than or equal to 8GB.

Exercise: Let $S(x)$ denote the statement "Device x never runs out of battery." What is the truth value of $\forall x\, S(x)$, where the domain consists of all mobile devices in the building?

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.

Table: Quantifiers

Negating Quantified Expressions

Consider the statement:

"Every IoT device in the building is connected to the secure network."

This is a universal quantification, represented as:

$$\forall x\, C(x), \text{ where } C(x)$$

is the statement:

"x is connected to the secure network,"

and the domain consists of all IoT devices in the building.

The negation of this statement is:

"It is not the case that every IoT device in the building is connected to the secure network,"

Negating Quantified Expressions

which is equivalent to:

"There is at least one IoT device in the building that is not connected to the secure network."

This is the existential quantification of the negation:

$$\exists x \, \neg C(x)$$

This example illustrates the equivalence:

$$\neg \forall x \, C(x) \equiv \exists x \, \neg C(x)$$

Negating Quantified Expressions

Quantifier	Equ. Statement	When is Negation True?	When is Negation False?
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There exists an x for which $P(x)$ is false	Every x satisfies $P(x)$
$\neg\exists x P(x)$	$\forall x \neg P(x)$	Every x for which $P(x)$ is false	There exists an x for which $P(x)$ is true

Table: De Morgan's Laws for Quantifiers

Example: What are the negations of the statements "Every software update fixes at least one known bug." ?

Negating Quantified Expressions

Solution:
Let U(x) denote "software update x fixes at least one known bug."
Original statement: $\forall x\, U(x)$

Negation: $\neg(\forall x\, U(x)) \equiv \exists x\, \neg U(x)$
English interpretation of the negation:

There exists at least one software update that does not fix any known bugs.

Negating Quantified Expressions

Exercise: What is the negation of the statement:

$$\forall x\, \exists y\, (T(x,y) \to E(x,y))$$

where $T(x,y)$ denotes "device x transmits packet y" and $E(x,y)$ denotes "packet y is encrypted by x".

Hint: Use logical equivalences to express the negation in simplified form.

Solution: Negating Quantified Expressions

Original: $\forall x\, \exists y\, (T(x, y) \to E(x, y))$

Negation:

$$\begin{aligned}\neg\forall x\, \exists y\, (T(x, y) \to E(x, y)) &\equiv \exists x\, \forall y\, \neg(T(x, y) \to E(x, y))\\ &\equiv \exists x\, \forall y\, (T(x, y) \wedge \neg E(x, y))\end{aligned}$$

Interpretation: There is a device that transmits every packet without encrypting any of them.