Fall 2021 - MATH 1101 Discrete Structures
Lectures 2-3

PART 1. Propositions. Logical Operations. Algebra of Propositions.

PART 2. Predicates and Quantifiers

Exercises. Set 1 (Solved Problems)

Exercises. Set 2 (Supplementary Problems)

Introduction

Logic (mathematical logic) provides a universal language for constructing mathematical reasoning. The scope of the logic toolkit is extremely wide. In fact, the tools of logic are used in almost all areas of knowledge because of its universality.

The language and methods of logic are used widely in computer science for computer programming, creating system specifications, programming languages, artificial intelligence, creating automated reasoning systems that allow computers to create their own proofs, etc.

The rules of logic give precise meaning to mathematical statements.

In Lectures 1-2 we introduced our first algebraic structure – Algebra of Sets. In the current Lecture Notes we define the second algebraic structure – Algebra of Propositions - which is quite similar to the first one. In fact, this similarity is the consequence of the fact that both structures are special cases of a more general structure (Boolean Algebra), which we will define later.

PART 1. PROPOSITIONS. LOGICAL OPERATIONS.

ALGEBRA OF PROPOSITIONS.

Section 0. Introduction.

Many algorithms and proofs use logical expressions such as:

"IF p THEN q" or "If $p1$ AND $p2$, THEN $q1$ OR $q2$"

Therefore, it is necessary to know the cases in which these expressions are TRUE or FALSE, that is, to know the "truth value" of such expressions.

We start our discussions with an introduction to the basic building blocks of logic - propositions.

Section 1. Propositions.

The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**.

Definition 1. 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. All the following declarative sentences are propositions.

1. Baku is the capital of Azerbaijan. **2**. Ganja is the capital of Azerbaijan.

3. 2+2=4. **4**. 2+2=3.

5. Japan is in Africa

Propositions 1 and 3 are true, whereas 2, 4, and 5 are false. ■

Some sentences that are not propositions are given in Example 2.

EXAMPLE 2. Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. x+1=2.
4. x+y=z.
5. Where are you going?
6. Do your homework.

Sentences 1, 2, 5, and 6 **are not propositions** because they are not declarative sentences. Sentences 3 and 4 **are not propositions** because they are neither true nor false. Note that each of sentences 3 and 4 **can be turned into a proposition** if we assign values to the variables. Later in Part 2 we will also discuss other ways to turn sentences such as these into propositions. ■

Compound Propositions

Definition 2. A proposition is said to be a **compound** proposition if it is composed of sub-propositions and various connectives. A proposition is said to be **primitive** if it cannot be broken down into simpler propositions, that is, if it is not composite. 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**. The conventional letters used for propositional variables are *p, q, r, s, …* ■

For example, propositions in Example 1 are primitive propositions. On the other hand, the following two propositions are composite (compound propositions):

- "The mountain is high **and** the water has no color"
- "Farid is smart **or** he studies every night".

Important! *The fundamental property of a compound proposition is that its truth value is completely determined by:*

- *the truth values of its sub-propositions and*
- *the way in which they are connected to form the compound propositions.*

The next section studies some of these connectives which are called **logical operations**.

Section 2. Logical Operations.

Basic Logical Operations.

First, we define **three basic logical operations**:

conjunction, **disjunction**, and **negation**

which correspond to the English words "and," "or," and "not" respectively. First two, conjunction and disjunction, are binary operations (they are similar to the intersection and union operations in the case of sets) and last one, negation, is an unary operation (like the complement operation in the case of sets).

Any two propositions can be combined by the word "and" to form a compound proposition called the **conjunction** of the original propositions. Symbolically, $p \wedge q$ (or p&q, or pq, or p·q), read "p and q", denotes the conjunction of p and q. Since $p \wedge q$ is a proposition it has a truth value, and this truth value depends only on the truth values of p and q. Specifically:

Definition 1. If propositions p and q are true, then their conjunction $p \wedge q$ is true; otherwise $p \wedge q$ is false. Equivalently, the conjunction operation is defined by the truth table 1. ■

Any two propositions can be combined by the word "or" to form a compound proposition called the **disjunction** of the original propositions. Symbolically, $p \vee q$ (or p+q), read "p or q" denotes the disjunction of p and q. The truth value of $p \vee q$ depends only on the truth values of p and q as follows.

Definition 2. If both propositions p and q are false, then their disjunction p$\vee$q is false; otherwise p$\vee$q is true. Equivalently, the disjunction operation is defined by the truth table 2. ■

Thus, p$\vee$q always means **"p and/or q".** In other words, the use of the connective ***or*** in a disjunction corresponds to **inclusive or**. A disjunction is true when at least one of the two propositions is true.

Given proposition p, a new proposition, called the **negation** of p, can be formed by writing "It is not a case that…" or "It is not true that ...", or "It is false that ..." before p or, if possible, by inserting in p the word "not". Symbolically, the negation of p, read "not p" is denoted by ¬p (or p', or $\bar{p}$). The truth value of ¬p depends on the truth value of p as follows:

Definition 3. If p is true, then ¬p is false; and if p is false, then ¬p is true. Equivalently, the negation operation is defined by the truth table 3. ■

Table 1 "p and q"

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

Table 2 "p or q"

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

Table 3 "not p"

p	¬p
T	F
F	T

EXAMPLE 3. Consider the following four compound propositions:

(i) Oil is lighter than water **and** 2+2=4. (iii) Japan is in Africa **and** 2+2=4.

(ii) Oil is lighter than water **and** 2+2=5. (iv) Japan is in Africa **and** 2+2=5.

Only the first proposition is true. Each of the others is false since at least one of its components (primitive propositions) is false.

EXAMPLE 4. Find the conjunction of the propositions p and q where p is the proposition "Farid's PC has more than 100 GB free hard disk space" and q is the proposition "The processor in Farid's PC runs faster than 2 GHz.".

Solution: The conjunction of these propositions, p$\wedge$q, is the proposition "Farid's PC has more than 100 GB free hard disk space, and the processor in Farid's PC runs faster than 2 GHz." This conjunction can be expressed more simply as "Farid's PC has more than 100 GB free hard disk space, and its processor runs faster than 2 GHz." For this conjunction to be true, both conditions given must be true. It is false, when one or both conditions are false. ■

EXAMPLE 5. Consider the following four compound propositions:

(i) Oil is lighter than water **or** 2+2=4. (iii) Japan is in Africa **or** 2+2=4.

(ii) Oil is lighter than water **or** 2+2=5. (iv) Japan is in Africa **or** 2+2=5.

Only the last proposition (iv) is false. Each of the others is true since at least one of its components (primitive propositions) is true.

EXAMPLE 6. What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 4?

Solution: The disjunction of p and q, p$\vee$q, is the proposition "Farid's PC has at least 100 GB free hard disk space, or the processor in Farid's PC runs faster than 2 GHz." This proposition is true when Farid's PC has at least 100 GB free hard disk space, when the PC's processor runs faster than 2 GHz, and when both conditions are true. It is false when both of these conditions are false, that is, when Farid's PC has less than 100 GB free hard disk space and the processor in his PC runs at 2 GHz or slower. ■

EXAMPLE 7. Find the negation of the proposition "Farid's PC runs Linux" and express this in simple English.

Solution: The negation is "It is not the case that Farid's PC runs Linux." This negation can be more simply expressed as "Farid's PC does not run Linux."

Additional Logical Operations (three more operations):

We discuss now several other important ways in which propositions can be combined. There are three more binary logical operations which are frequently used. They will be defined below. Moreover, we show that they can be expressed in terms of conjunction, disjunction, negation. At the same time, they are interested in themselves.

The use of the connective ***or*** in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, in an inclusive way. Sometimes, we use *or* in an exclusive sense. Namely, **exclusive or** is used to connect the propositions p and q, in the proposition **"*p* or *q* (but not both)"**. This proposition is true when p is true and q is false, and when p is false and q is true. It is false when both p and q are false and when both are true.

Definition 4. 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. Equivalently, the truth table for the exclusive or of two propositions is displayed in Table 4. ■

EXAMPLE 8. We are using the *exclusive or* when we say "Students who have taken Calculus or Computer Science, but not both, can enroll in this class." Here, we mean that students who have taken both Calculus and a Computer Science course cannot take the class. Only those who have taken exactly one of the two courses can take the class.
Similarly, when a menu at a restaurant states, "Coffee or tea comes with dinner", the restaurant always means that customers can have either a cup of coffee or a cup of tea, but not both. ■

Table 4. $p\oplus q$

Row 0	p	q	$p\oplus q$
Row 1	T	T	F
Row 2	T	F	T
Row 3	F	T	T
Row 4	F	F	F

Table 5. $p\to q$

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

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, that is, true when both p and q are true and when p is false (no matter what truth value q has). In the conditional statement $p\to q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*). Equivalently, the truth table for the ***conditional statement*** is displayed in Table 5. ■

The proposition $p\to q$ is called a conditional statement because $p\to q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract.

EXAMPLE 9. Consider a statement that a professor might make:

"If you get 100% on the final, then you will get an A."

If you manage to get a 100% on the final, then you would expect to receive an A. If you do not get 100% you may or may not receive an A depending on other factors. However, if you do get 100%, but the professor does not give you an A, you will feel cheated. ■

The well-known Rolle's Theorem from Calculus is a very good interpretation of conditional statement rules (Table 5).

EXAMPLE. (Rolle's Theorem=RT). Suppose that a real-valued function y=f(x), satisfies the following conditions:

(a) f is continuous at every point of the closed interval [a, b]

(b) f is differentiable at every point of its interior (a, b).

(c) f(a)=f(b)

Then (conclusion)

(d) there is at least one number t∈(a, b) at which f′(t)=0.

Now we interpret **RT** using rules of logic. Introduce the propositions p and q as following:

p="f satisfies all three requirements (a), (b), and (c)" in other words, p is compound proposition: p=(a) **AND** (b) **AND** (c);

q="there is at least one number t∈(a,b) at which f′(t)=0".

Analyze row by row Table 5 (Definition of p→q) for the proposed Example:

Row 1. Let p=T, q=T. Then **RT** asserts that IT CAN BE PROVED that p implies q, in other words, **RT** says: "Yes, it is TRUE that under p we get q, or, equivalently, the proposition "If p Then q" has truth value TRUE". Hence, the truth value of conditional statement "If p Then q" (or p→q) is T, and therefore, **RT** confirms Row 1 of Table 5.

Row 2. Let p=T, q=F. But **RT** asserts that p implies q, therefore under p=TRUE the conclusion q cannot be FALSE, it must be TRUE (this is exactly **RT**). Therefore, the proposition "If p Then q" (or p→q) in this case has the truth value FALSE". Hence, **RT** confirms Row 2 of Table 5.

Row 3. Let p=F, q=T. Now p=F means that at least one of three components of p [(a) or (b) or (c)] is FALSE. Clear that under violation of (a) or (b) or (c) it is easy to provide examples when there exists a point t∈(a, b) at which f′(t)=0. Therefore, q=T **is possible** despite of p=F and so the compound proposition "If p Then q" (or p→q) in this case has the truth value TRUE". Hence, Row 3 of Table 5 is also confirmed.

Row 4. Let p=F, q=F. Again p=F means that at least one of three components of p [(a) or (b) or (c)] is FALSE. We can provide lots of counterexamples when violation of (a) or (b) or (c) implies the nonexistence of seeking point t. Therefore q=F **is possible** if p=F and so the proposition "If p Then q" (or p→q) in this case has also the truth value TRUE". Hence, Row 4 of Table 5 is also confirmed. ■

Next example illustrates possible ways to translate conditional statements from formal logic to corresponding statements in English.

EXAMPLE 10. Let p="Aynur learns discrete structures" and q="Aynur will find a good job." Express the conditional statement p→q as a statement in English.

Solution: From the definition of conditional statement, we see that p→q represents the following statement in English:

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

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

- "Aynur will find a good job when she learns discrete structures."

- "For Aynur to get a good job, it is sufficient for her to learn discrete structures." ■

In mathematical reasoning, we consider conditional statements of a more general sort than we use in any human language. The mathematical concept of a conditional statement is independent of a cause-and-effect relationship between hypothesis and conclusion. **Our definition of a conditional statement (Definition 5) specifies its truth values; it is not based on English or any other human language usage. Propositional language is an artificial language; we parallel English usage to make it easy to use and remember.**

EXAMPLE 11. The conditional statement "If Aynur 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 Aynur has a smartphone, then 2+3=6" is true if Aynur does not have a smartphone, even though 2+3=6 is false. ■

We would not use last conditional statements (Example 11) in natural language, because there is no relationship between the hypothesis and the conclusion in either statement.

Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express p→q. In different sources you may encounter most if not all the following ways to express conditional statement:

"if p, then q"	"p implies q"
"if p, q"	"p only if q"
"p is sufficient for q"	"a sufficient condition for q is p"
"q if p"	"q whenever p"
"q when p"	"q is necessary for p"
"a necessary condition for p is q"	"q follows from p"
"q unless ¬p"	

Of the various ways to express the conditional statement p→q, the two that seem to cause the most confusion are "p only if q" and "q unless ¬p." Consequently, we will provide some guidance for clearing up this confusion. Namely:

- "p only if q" says that p cannot be true when q is not true. That is, the proposition "p only if q" is false if p is true, but q is false. When p is false, q may be either true or false, because the statement says nothing about the truth value of q. Hence the proposition "p only if q" has the same truth table as the conditional statement p→q
- "q unless ¬p" means that if ¬p is false, then q must be true, equivalently, if p is true then q must be true (the same as Rolle's Theorem). That is, the proposition "q unless ¬p" is false when p is true and q is false, but it is true otherwise. Consequently, "q unless ¬p" and p→q always have the same truth value.

We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

Definition 6. Let p and q be propositions. The ***biconditional statement*** p↔q is the proposition "p if and only if q." The biconditional statement p↔q is true when p and q have the same truth values, and is false otherwise. The truth table for the ***biconditional statement*** is displayed in Table 6. ■

Table 6. p↔q

p	q	p↔q
T	T	T
T	F	F
F	T	F
F	F	T

Note that the biconditional statement $p \leftrightarrow q$ is true when both the conditional statements $p \rightarrow q$ and $q \rightarrow p$ are true and is false otherwise. That is why we use the words "if and only if" to express this logical connective and why it is symbolically written by combining the symbols $\rightarrow$ and $\leftarrow$.

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

- "p is necessary and sufficient for q"
- "if p then q, and conversely"
- "p iff q."

The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation "iff" for "if and only if."

Precedence of Logical Operations

We can construct compound propositions using the negation operation and the binary logical operations defined so far. We generally use parentheses to specify the order in which logical operations in a compound proposition are to be applied. For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$. However, in order to avoid an excessive number of parentheses, we sometimes adopt an order of precedence for the logical connectives. We specify that the negation is applied before all other logical operations. This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q, namely, $(\neg p) \wedge q$.

Table 7. Precedence of Logical Operations

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

Table 7 shows the precedence of operations. Nevertheless, we will frequently use the parenthesis to make the formulae more understandable and readable.

Section 3. Truth Tables of Compound Propositions

Let P(p, q, ...) denote a compound proposition, that is an expression constructed from logical (=propositional) variables p, q, . . ., which take on the values TRUE (T) or FALSE (F), and the logical connectives (operations). The main property of a proposition P(p, q, ...) is that its truth value, as we mentioned earlier, depends exclusively upon the truth values of its variables, that is, the truth value of a compound proposition is known once the truth values of all variables are known. A simple concise way to show this relationship is through a **truth table**. We describe a way to obtain such a truth table as following:

- 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 12. Construct the truth table of the compound proposition

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

Solution: Because this truth table involves two propositional variables p and q, there are four rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q, respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table 8 below. ■

Table 8. 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

Tautologies and Contradictions

Definition 7. A compound proposition P(p, q, . ..) is called a ***tautology*** if it contains only T in the last column of its truth table or, in other words, if P is true for any truth values of its variables. Analogously, a compound proposition P(p, q, . . .) is called a ***contradiction*** if it contains only F in the last column of its truth table or, in other words, if it is false for any truth values of its variables. ■

EXAMPLE 13. Show that

- the proposition "p **or** not p," that is, $p \vee \neg p$, is a tautology, and
- the proposition "p **and** not p", that is, $p \wedge \neg p$, is a contradiction.

Solution. This is verified by looking at their truth tables 9 and 10. ■

Table 9. $p \vee \neg p$, is a tautology

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

Table 10. $p \wedge \neg p$, is a contradiction

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

Note that the negation of a tautology is a contradiction since it is always false, and the negation of a contradiction is a tautology since it is always true.

Now let P(p, q, ...) be a tautology, and let P_1(p, q, ...), P_2(p, q, ...), ... be any propositions. Since P(p, q, ...) does not depend upon the particular truth values of its variables p, q, ..., we can substitute P_1 for p, P_2 for q, ... in the tautology P(p, q, ...) and still have a tautology. In other words:

Theorem 1. (Principle of Substitution): If P(p, q, ...) is a tautology, then P(P_1, P_2, ...) is a tautology for any propositions P_1, P_2, ...

Proof. Clear.

Logical Equivalences of Propositions

Definition 8. Two propositions P(p, q, ...) and Q(p, q, ...) are said to be logically equivalent, or simply equivalent or equal, denoted by

$$P(p, q, \ldots) \equiv Q(p, q, \ldots)$$

if they have identical truth tables, or, equivalently, if $P \leftrightarrow Q$ is a tautology . ■

EXAMPLE 14. It is easy to check that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$. Therefore, they are logically equivalent or equal, $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$. ■

EXAMPLE 15. The truth tables of $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are the same (tables 11 and 12), that is, both propositions are false in the first row (p=T, q=T) and true in the other three rows. Accordingly, we can write

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

In other words, the propositions are logically equivalent. ■

Table 11. $\neg(p \wedge q)$

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

Table 12. $\neg p \vee \neg q$

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

Important Note. In Example 16 below we show that all three new operations, defined above, $p \oplus q$, $p \rightarrow q$, and $p \leftrightarrow q$, can be written as compound propositions by using logical variables and conjunction, disjunction, negation. **Moreover, we will see (Definition 9 and Theorem 2) that any propositional function P(p, q, ...) is logically equivalent to a compound proposition involving only logical variables (p, q, …) and conjunction, disjunction, negation.**

This is a remarkable result: two binary operations and one unary operation are enough to represent arbitrary propositional function P(p, q, ...) of any finite number of logical variables p, q, … as a compound proposition constructed from logical variables p, q, ..., and the logical operations conjunction, disjunction, negation.

EXAMPLE 16. Express $p \oplus q$, $p \rightarrow q$, and $p \leftrightarrow q$ as compound propositions by using logical variables and operations conjunction, disjunction, and negation.

Solution: It easy to check logical equivalences below by constructing truth tables of left and right sides of each expression.

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

$p \rightarrow q \equiv \neg p \vee q$

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

Other Forms of Conditional Statements: Converse, Contrapositive, and Inverse.

Now we form new conditional statements starting with a conditional statement $p \rightarrow q$. There are three related conditional statements that occur so often that they have special names.

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

Truth tables of conditional statements are in Table 13 below.

Table 13. Truth tables for conditional statements

p	q	$\neg p$	$\neg q$	Conditional $p \rightarrow q$	Converse $q \rightarrow p$	Inverse $\neg p \rightarrow \neg q$	Contrapositive $\neg q \rightarrow \neg p$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

From Table 13 clear that Conditional and Contrapositive statements are logically equivalent. The same is true for Converse and Inverse statements, that is:

- $p \rightarrow q \equiv \neg q \rightarrow \neg p$
- $q \rightarrow p \equiv \neg p \rightarrow \neg q$

EXAMPLE 17. What are the contrapositive, the converse, and the inverse of the conditional statement "The home team wins whenever it is raining?"

Solution: Because "q whenever p" is one of the ways to express the conditional statement p→q, the original statement can be rewritten as "If it is raining, then the home team wins."

Consequently,

- the contrapositive of this conditional statement is "If the home team does not win, then it is not raining."
- The converse is "If the home team wins, then it is raining."
- The inverse is "If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement. Inverse and converse statements are also equivalent each other.

Section 4. Algebra of Propositions. Duality.

Probably, the reader has already noticed a certain analogy (and even identity, in some sense) between operations on sets (Lectures 1-2) and operations on propositions. This is no coincidence because both the algebra of sets and the algebra of propositions, which we are going to define now, are examples of a more general structure called Boolean algebra (in honor of the English mathematician George Boole).

Definition 9. Algebra of Propositions, say $\mathscr{P}$, is an algebraic structure which contains 6 elements (6-tuple):

a) A family, say $\mathfrak{P}$, of propositions, including the compound proposition **T** that is always true (tautology) and the compound proposition **F** that is always false (contradiction);

b) Two binary operations $\mathfrak{P}\times\mathfrak{P}\to\mathfrak{P}$
   a. Disjunction $(p, q)\longmapsto(p\vee q)$, and
   b. Conjunction $(p, q)\longmapsto(p\wedge q)$

c) Unary operation $\mathfrak{P}\to\mathfrak{P}$, negation, $p\longmapsto\neg p$

Denotation for Algebra of Propositions: $\mathscr{P}$= <$\mathfrak{P}$, **F**, **T**, ∨, ∧, ¬ > ■

Note 1. In Definition 9 we use only three operations ∨, ∧, and ¬. At the same time in previous sections we introduced three more operations. Clear that it is possible to define more and more unary, binary operations. The question arises: why we do not use in Definition 9 some other set of operations**. It turns out that in Propositional Algebra any propositional function of one or many variables can be written as a compound proposition which uses only variables and operations ∨, ∧, and ¬.** This is a fundamental fact which is called **functionally completeness** of the system of functions ∨, ∧, and ¬. More detail:

Definition 10. A collection of logical operations is called **functionally complete** if every compound proposition is logically equivalent to a compound proposition involving only these logical operations. ■

Theorem 2. The collection consisting of three operations: ¬, ∧, and ∨ form a functionally complete collection of logical operators.

Proof. A proof will be offered when studying Boolean Algebras, special case of which is an Algebra of Propositions. ■

Algebra $\mathscr{P}$ satisfies various laws (identities) which are listed in Table 14 below. In fact, we formally state this as:

Theorem 3. Algebra $\mathscr{P}$ satisfies the laws in Table 14.

Table 14. Laws of the Algebra of Propositions

Idempotent laws:	(1a) $p \vee p \equiv p$	(1b) $p \wedge p \equiv p$
Associative laws:	(2a) $(p \vee q) \vee r \equiv p \vee (q \vee r)$	(2b) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws:	(3a) $p \vee q \equiv q \vee p$	(3b) $p \wedge q \equiv q \wedge p$
Distributive laws:	(4a) $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	(4b) $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws:	(5a) $p \vee \mathbf{F} \equiv p$	(5b) $p \wedge \mathbf{T} \equiv p$
	(6a) $p \vee \mathbf{T} \equiv \mathbf{T}$	(6b) $p \wedge \mathbf{F} \equiv \mathbf{F}$
Involution laws:	(7) $\neg(\neg p) \equiv P$	
Complement laws:	(8a) $p \vee \neg p \equiv \mathbf{T}$	(8b) $p \wedge \neg p \equiv \mathbf{F}$
	(9a) $\neg \mathbf{T} \equiv \mathbf{F}$	(9b) $\neg \mathbf{F} \equiv \mathbf{T}$
DeMorgan's laws:	(10a) $\neg(p \vee q) \equiv \neg p \wedge \neg q$	(10b) $\neg(p \wedge q) \equiv \neg p \vee \neg q$

Proof. Exercise. (Hint. All equivalences can be established by comparing the related truth tables of right and left sides in each expression). ■

Remark. The associative law (2a) for disjunction shows that the expression $p \vee q \vee r$ is well defined, in the sense that it does not matter whether we first take the disjunction of p with q and then the disjunction of $p \vee q$ with r, or if we first take the disjunction of q and r and then take the disjunction of p with $q \vee r$. Similarly, the expression $p \wedge q \wedge r$ is well defined. By extending this reasoning, it follows that $p_1 \vee p_2 \vee \cdots \vee p_n$ and $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ are well defined whenever $p_1, p_2, \ldots, p_n$ are propositions.

Furthermore, note that De Morgan's laws extend to

$$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n)$$

and

$$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n).$$ ■

Note 2. We observe obvious and very important similarity between Laws of Algebra of Sets (Table 1, Lectures 1-2) and Laws of Algebra of Propositions (Table 14, current Lectures) by replacing corresponding elements in Table 1 from Lectures 1-2 as it is shown in Table 15 below:

Table 15 Correspondence of elements of Algebras $\mathcal{A}$ and $\mathscr{P}$

Algebra of Sets $\mathcal{A}$	**Algebra of Propositions $\mathscr{P}$**
Sets (A, B, C, …)	Propositions (p, q, r, …)
Sets (**U**, ∅)	Propositions (**T**, **F**)
Operations ∪ and ∩	Operations ∨ and ∧
Complement Operation ′	Negation Operation ¬

As in the case of Algebra $\mathcal{A}$ the duality principle is valid for the algebra $\mathscr{P}$. Suppose E is an equation in Algebra $\mathscr{P}$. The dual E^* of E is the equation obtained by replacing each occurrence of ∨, ∧, **T** and **F** in E by ∧, ∨, **F**, and **T**, respectively. *Principle of duality*: if any equation E is an identity then its dual E^* is also an identity.

We also display some useful equivalences for compound propositions involving conditional statements (Table 16) and biconditional statements (Table 17). The reader is asked to verify the equivalences directly, by comparing of truth tables, or by using of equivalences in Table 14.

Table 16

Logical Equivalences Involving Conditional Statements
(1) $p \to q \equiv \neg q \to \neg p$
(2) $p \lor q \equiv \neg p \to q$
(3) $p \land q \equiv \neg(p \to \neg q)$
(4) $\neg(p \to q) \equiv p \land \neg q$
(5) $(p \to q) \land (p \to r) \equiv p \to (q \land r)$
(6) $(p \to r) \land (q \to r) \equiv (p \lor q) \to r$
(7) $(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$
(8) $(p \to r) \lor (q \to r) \equiv (p \land q) \to r$

Table 17

Logical Equivalences Involving Biconditional Statements.
(9) $p \leftrightarrow q \equiv (p \to q) \land (q \to p)$
(10) $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
(11) $p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$
(12) $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$
(13) $\neg(p \leftrightarrow q) \equiv \neg p \leftrightarrow q$

EXAMPLE 18. Use De Morgan's laws to express the negations of

- "Farid has a cellphone and he has a laptop computer" and
- "Farid will go to the concert or Araz will go to the concert."

Solution.

- Let p=" Farid has a cellphone" and q=" Farid has a laptop computer." Then "Farid has a cellphone and he has a laptop computer" can be represented by $p \land q$. By the first of De Morgan's laws, $\neg(p \land q)$ is equivalent to $\neg p \lor \neg q$. Consequently, we can express the negation of original proposition as "Farid does not have a cellphone or he does not have a laptop computer."
- Let r="Farid will go to the concert" and s="Araz will go to the concert." Then "Farid will go to the concert or Araz will go to the concert" can be represented by $r \lor s$. By De Morgan's law, $\neg(r \lor s)$ is equivalent to $\neg r \land \neg s$. Consequently, we can express the negation of original proposition as "Farid will not go to the concert and Araz will not go to the concert." ■

The logical equivalences in from Example 16, Tables 14, 16, and 17, as well as any others that have been established, can be used to construct new logical equivalences. The reason for this is that a proposition in a compound proposition can be replaced by a compound proposition that is logically equivalent to it without changing the truth value of the original compound proposition. This technique is illustrated in Examples below, where we also use the fact that **if p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent** (so called transitivity property of logical equivalence).

EXAMPLE 19. Show that $\neg(p \to q)$ and $p \land \neg q$ are logically equivalent by developing a series of logical equivalences.

Solution. We could use a truth table (definition of logical equivalence) to show that these compound propositions are equivalent. Indeed, it would not be hard to do so. However, we want to illustrate how to use logical identities that we already know to establish new logical identities, something that is of practical importance for establishing equivalences of compound propositions with a large number of variables.

$\neg(p \to q) \equiv \neg(\neg p \lor q)=$ by Example 16

$\equiv \neg(\neg p) \land \neg q$ by the second De Morgan law

$\equiv p \land \neg q$ by the double negation law ■

EXAMPLE 20. Show that $\neg(p \lor (\neg p \land q))$ and $\neg p \land \neg q$ are logically equivalent by developing a series of logical equivalences.

Solution. We will use one of the equivalences in Table 14 at a time, starting with $\neg(p \lor (\neg p \land q))$ and ending with $\neg p \land \neg q$. (Note: we could also easily establish this equivalence using a truth table.) We have the following equivalences.

$\neg(p \lor (\neg p \land q)) \equiv \neg p \land \neg(\neg p \land q)$	by the second De Morgan law
$\equiv \neg p \land [\neg(\neg p) \lor \neg q]$	by the first De Morgan law
$\equiv \neg p \land (p \lor \neg q)$	by the double negation law
$\equiv (\neg p \land p) \lor (\neg p \land \neg q)$	by the second distributive law
$\equiv \mathbf{F} \lor (\neg p \land \neg q)$	because $\neg p \land p \equiv \mathbf{F}$
$\equiv (\neg p \land \neg q) \lor \mathbf{F}$	by the commutative law for disjunction
$\equiv \neg p \land \neg q$	by the identity law for **F** ■

EXAMPLE 21. Show that $(p \land q) \to (p \lor q)$ is a tautology.

Solution. To show that this proposition is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to **T**. (Note: This could also be done using a truth table.)

$(p \land q) \to (p \lor q) \equiv \neg(p \land q) \lor (p \lor q)$	by Example 16
$\equiv (\neg p \lor \neg q) \lor (p \lor q)$	by the first De Morgan law
$\equiv (\neg p \lor p) \lor (\neg q \lor q)$	by the associative and commutative laws
$\equiv \mathbf{T} \lor \mathbf{T}$	by Example 13
$\equiv \mathbf{T}$	by the domination law ■

Section 5. Logic and Bit Operations

Computers present 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 truth values. We will use:

bit 1 to represent true (T)

bit 0 to represent false (F).

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. Regarding terminology: clear that terms "**Boolean variable**" and **"logical (propositional) variable" are used to denote the same object.**

Computer bit operations exactly correspond to the logical connectives. By replacing true by a one and false by a zero in the truth tables for the operators $\land$, $\lor$, and $\oplus$, the tables for the corresponding bit operations are obtained (see Table 18). We will also use the notation OR, AND, and XOR for the operators $\lor$, $\land$, and $\oplus$, as is done in various programming languages.

Table 18. Table for the Bit Operations OR, AND, and XOR.

x	y	$x \lor y$	$x \land y$	$x \oplus y$
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

Information is often represented using **bit strings, which are lists of zeros and ones.** When this is done, operations on the bit strings can be used to manipulate this information.

Definition 11. A **bit string** is a sequence of zero or more bits. The **length** of a string u, L(u), is the number of bits in the string. ■

We can extend bit operations to bit strings as following.

Definition 12. The **bitwise OR, bitwise AND,** and **bitwise XOR** of two strings u and v of the same length n is defined as a new bit string w with the same length n, i-th elements of which is defined, as the OR, AND, and XOR operation over corresponding i-th bits of u and v. We use the symbols $\vee$, $\wedge$, and $\oplus$ to represent the bitwise OR, bitwise AND, and bitwise XOR operations, respectively.

EXAMPLE 21. Find the bitwise OR, AND, XOR of the bit strings 10101011101 and 11101010001.

Solution. The bitwise OR, bitwise AND, and bitwise XOR of these strings are obtained by taking the OR, AND, and XOR of the corresponding bits, respectively. This yields

101 0101 1101

<u>111 0101 0001</u>

111 0101 1101 bitwise OR,

101 0101 0001 bitwise AND

010 0000 1100 bitwise XOR ■

PART 2. PREDICATES AND QUANTIFIERS

Section 0. Introduction

In propositional logic, we are interested to know the truth values of propositions rather than their real content. Therefore, **propositional logic cannot adequately express the meaning of all statements in mathematics and in natural language.** For example, we cannot use the rules of propositional logic to conclude from the proposition

"CS1 is under attack by an intruder," where CS1 is a computer on the university network,

the truth of proposition

"There is a computer on the university network that is under attack by an intruder."

In Part 2 we introduce a more powerful type of logic called **predicate logic** and will see how predicate logic can be used to express the meaning of a wide range of statements in mathematics and computer science.

We first introduce the concept of a **predicate (=propositional function)**. Afterward, we will introduce the notion of **quantifiers**, which enable us to reason with statements that assert that a **certain property holds for all objects of a certain type** and with statements that assert **the existence of an object with a particular property**.

Section 1. Predicates (Propositional Functions)

Let us start, as usually, by examples to bring a light to the topic. Then we will provide strong mathematical definitions.

Consider four statements involving variables, such as:

(1) "**x**<10"
(2) "computer **x** is under attack by an intruder"
(3) "**x**+**y**=5"
(4) "**x**×**y**=**z**"

These statements are neither true nor false, that is, **they are not propositions** until the values of the variables are not specified. Now we are discussing the ways **how to turn** these statements into **propositions (which are true or false)**.

Note that in relations (1) and (2) we have one variable, x; in (3) – two variables x, y; in (4) – 3 variables, x, y, z.

First, we discuss the case (1) and provide the appropriate definition of unary predicate (=single variable propositional function) and then define more general n-ary predicate and provide examples.

Case (1). The statement "x is less than 10" has two parts. The first part, the variable x, is the **subject** of the statement. The second part—the **predicate**, "is less than 10"—refers to a property that the subject of the statement can have. We denote the statement "x is less than 10" by P(x), P denotes the **predicate** "is less than 10" 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 propositional function P(x) becomes a proposition and has a truth value.**

Definition 13. Let A be a set. A **propositional function** (or **unary predicate**) **defined on A** is an expression P(x) which has the property that P(a) is ***true or false*** for each $a \in A$. In other words, **an unary predicate on A is a function P: $A \to \{0, 1\}$,** here 0 and 1 play the roles of False and True respectively.

That is, P(x) becomes a proposition (with a truth value) whenever any element $a \in A$ is substituted for the variable x.

The set A is called the **domain** of P(x), and the set T_P of all elements of A for which P(a) is true is called the **truth set** of P(x). In other words,

$T_P=\{x|x\in A, P(x) \text{ is true}\}$ or $T_P=\{x|P(x)\}$ ■

In general, for the case of n variables we have the following definition.

Definition 14. Let $A_1, A_2,\ldots, A_n$ are arbitrary sets, and $A=A_1\times A_2\times\ldots\times A_n$ be the Cartesian product. A **propositional function** (or **n-place predicate,** or **n-ary predicate**) **defined on A** is an *n*-variable function $P:A\to\{0, 1\}$. In other words, an n-place predicate is an expression $P(x_1, x_2, ..., x_n)$ which has the property: $P(a_1, a_2, ..., a_n)$ is ***true or false*** for each $(a_1, a_2, ..., a_n)\in A$.

The set A is called the **domain** of $P(x_1, x_2, ..., x_n)$, and the set T_P of all elements of A for which $P(a_1, a_2, ..., a_n)$ is true is called the **truth set** of $P(x_1, x_2, ..., x_n)$. In other words,

$T_P=\{(x_1, x_2, ..., x_n) \mid (x_1, x_2, ..., x_n)\in A, P(x_1, x_2, ..., x_n) \text{ is true}\}$ or $T_P=\{(x_1, x_2, ..., x_n)|P(x_1, x_2, ..., x_n)\}$ ■

We continue to discuss cases 1-4.

EXAMPLE 22. Let P(x) denote the **propositional function** "x<10." What are the truth values of **propositions** P(5) and P(12)?

Solution. We obtain the proposition P(5) by setting x=5 in propositional function "x<10." Hence, P(5), which is the proposition "5<10" is true. However, P(12), which is the proposition "12<10" is false. ■

Now we apply similar reasoning to the case (2).

EXAMPLE 23. Let P(x) denote the propositional function "Computer x is under attack by an intruder." Here **x** is the subject, "Computer … is under attack by an intruder" is the predicate (propositional function). Suppose that of the computers on campus, only CS1 and CS2 are currently under attack by intruders. What are truth values of P(CS1), P(CS2), and P(CS3)?

Solution: We obtain the proposition P(CS1) by setting x=CS1 in the statement "Computer x is under attack by an intruder." Because CS1 is on the list of computers currently under attack, we conclude that P(CS1) is true. Similarly, P(CS2) is true but CS3 is on the list of computers under attack, therefore P(CS3) is false. ■

Consider now case (3) with 2 variables: “**x**+**y**=5”. We denote this statement by Q(x,y)=“**x**+**y**=5”, where x and y are variables and Q is the binary predicate or binary propositional function. When values are assigned to the variables x and y, the predicate Q(x, y) becomes the proposition which has a truth value.

EXAMPLE 24. Let Q(x, y) denote the statement “**x**+**y**=5” What are the truth values of the propositions Q(3, 2) and Q(3, 1)?

Solution: To obtain Q(3, 2), set x=3 and y=2 in Q(x, y). Hence, Q(3, 2) is the proposition “3+2=5,” which is true. The statement Q(3, 1) is the proposition “3+1=5,” which is false. ■

Finally, we consider case 4. We can let R(x, y, z) denote the statement “**x**×**y**=**z**.” When values are assigned to the variables x, y, and z this statement has a truth value.

EXAMPLE 25. What are the truth values of the propositions R(3, 2, 6) and R(2, 4, 6)?

Solution. The proposition R(3, 2, 6) is obtained by setting x=3, y=2, and z=6 in the function R(x, y, z). We see that R(3, 2, 6) is the statement “3×2=6,” which is true. Also note that R(2, 4, 6), which is the statement “2×4=6” is false. ■

EXAMPLE 26. Consider the statement in programming as below

if x>0 **then** x:=x+1

When this statement is encountered in a program, the value of the variable x at that point in the execution of the program is inserted into P(x), which is “x>0.” If P(x) is true for this value of x, the assignment statement x:=x+1 is executed, so the value of x is increased by 1. If P(x) is false for this value of x, the assignment statement is not executed, so the value of x is not changed. ■

Predicates are also used to establish the correctness of computer programs, that is, to show that computer programs always produce the desired output when given valid input. Note that unless the correctness of a computer program is established, no amount of testing can show that it produces the desired output for all input values, unless every input value is tested.

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**. We use predicates to describe both preconditions and postconditions (Example 27).

EXAMPLE 27. 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 values before we run the program. So, for this precondition we can use the predicate P(x, y)=“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)=“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 proposition “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 proposition "x=b and y=a" is true. ■

Remark: All three basic logical operations $\wedge$, $\vee$, $\neg$ can be properly defined for propositional functions (predicates) as following.

1. The expression $(\neg P)(x)$ denoted as $\neg P(x)$ has the obvious meaning:
   "$\neg P(x)$ is true when P(x) is false, and vice versa"
2. Similarly, $P(x)\wedge Q(x)$, read "P(x) and Q(x)," is defined by:
   $(P\wedge Q)(x)=P(x)\wedge Q(x)$ is true when both P(x) and Q(x) are true"
3. Similarly, $P(x)\vee Q(x)$, read "P(x) or Q(x)," is defined by:
   $(P\vee Q)(x)=P(x)\vee Q(x)$ is true when at least P(x) or Q(x) is true"

Thus, in terms of truth sets:

(i) $T_{\neg P}=(T_P)'$, that is, truth set of $(\neg P)(x)$ is the complement of truth set of P(x).

(ii) $T_{(P\wedge Q)}=T_P\cap T_Q$, that is, truth set of $P(x)\wedge Q(x)$ is the intersection of truth sets of P(x) and Q(x).

(iii) $T_{(P\vee Q)}=T_P\cup T_Q$, that is, truth set of $P(x)\vee Q(x)$ is the union of truth sets of P(x) and Q(x).

One can also show that the laws for propositions also hold for propositional functions. For example, we have De Morgan's laws:

$\neg(P(x)\wedge Q(x))\equiv\neg P(x)\vee\neg Q(x)$ and $\neg(P(x)\vee Q(x))\equiv\neg P(x)\wedge\neg Q(x)$ ■

Section 2. Quantifiers

When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value.

There is another important way, called quantification, to turn a propositional function into a proposition. **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.

EXAMPLE 28. Find the truth set for each propositional function P(x) defined on the set N of positive integers in cases (a), (b) and (c) below.

Solution.

(a) Let P(x) be "x-10>5". Its truth set is {16, 17, 18, ...} consisting of all integers greater than 15.

(b) Let P(x) be "x+10<5". Its truth set is the empty set ∅. That is, P(x) is not true for any positive integer.

(c) Let P(x) be "x+10>10." Its truth set is N. That is, P(x) is true for every element in N. ■

Remark: *The above example shows that if P(x) is a propositional function defined on a set A then P(x) could be true for all $x\in A$, for some $x\in A$, or for no $x\in A$.*

We will focus on two types of quantification here:

- universal quantification, which tells us that a predicate is true for every element under consideration (=for all values of a variable over a **domain)**, and
- existential quantification, which tells us that there is one or more element under consideration for which the predicate is true.

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

Universal Quantifier.

Let A be a set and P: A→{0, 1} be a propositional function (= unary predicate). Consider the following statement S=**"P(x) is true, or P(x)=1, for all values of x in the domain A",** Clear that for S two cases are possible (but not both).

Case 1: Yes, it is true, S="T", which means that "P(x) is true for all values of x in the domain".

Case 2: No, it is false, S= "F" which means that it is not a case that "P(x) is true for all values of x in the domain", that is, there exist a value $a \in A$ such that P(a)=0 or P(a)="False".

Thus, our statement S="P(x) is true for all values of x in the domain" is the proposition with truth value True or False.

Observation: A propositional function P: A→{0, 1} itself is not a proposition but the statement S="P(x) is true for all values of x in the domain" is always the proposition.

We summarize our observations as the Definition 15 below.

Definition 15. Let P(x) be a predicate (=propositional function) defined on a set (domain) A. That is, P: A→{0, 1}.

The universal quantification of P(x) is the new proposition S which is defined as:

S="**P(x) is true for all values of x in the domain**" or, in denotations,

S=∀xP(x) (1)

Here the symbol ∀ which reads "for all" or "for every" is called the **universal quantifier**. We read ∀xP(x) as "for all xP(x)" or "for every xP(x)."

An element for which P(x) is false is called **a counterexample** of ∀xP(x). ■

Note 3. The meaning of the universal quantification of P(x) changes when we change the domain. The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined. ■

Remember that the truth value of ∀xP(x) *depends on the domain!*

The expression P(x) by itself is a predicate (or propositional function) and therefore has no truth value. However, ∀xP(x), that is, P(x) preceded by the quantifier ∀, does have a truth value.

Theorem 4. The proposition (1) is equivalent to the proposition

$$T_P=\{x \mid x\in A,\ P(x) \text{ is true}\}=A \qquad (2)$$

that is, that the truth set of P(x) is the entire set A.

Proof. Clear, from definition of T_P. ■

EXAMPLE 29. The next two examples are based on the equivalence between (1) and (2). Domain is N.

(a) The proposition ∀nP(n), where P(n) is the predicate "n+4>3", is true since $T_P=\{n \mid n+4>3\}=\{1, 2, ...\}=N$.

(b) The proposition ∀nP(n), where P(n) is the predicate "n+2>8", is false since $T_P=\{n \mid n+2>8\}=$ $=\{7, 8, ...\}\neq N$. In other words, there is at least one element in N for which $n+2\leq 8$, say n=5. Therefore, n=5 is a counterexample for ∀nP(n). ■

EXAMPLE 30.

(a) Let **R,** set of real numbers, is the domain of a propositional function P(x), here P(x) is "$|x|\neq 0$". Then the proposition ∀xP(x) is false, since 0 is a counterexample, that is, $|0|\neq 0$ is not true.

(b) Let **R,** set of real numbers, is the domain of a propositional function P(x), here P(x) is "$x^2 \geq x$". The proposition $\forall xP(x)$ is not true since, for example, ½ is a counterexample. Specifically, $(1/2)^2 \geq 1/2$ is not true, that is, $(1/2)^2 < 1/2$. ■

Existential Quantifier.

Let A be a set and P: A→{0, 1} be a propositional function (= unary predicate). Consider the following statement S=**"There exists x∈A such that P(x) is a true"**. Clear that for S two cases are possible (but not both):

Case 1: Yes, it is true, S="T", which means that "There exist a value a∈A such that P(a)=1 or P(a)="True".

Case 2: No, it is false, S= "F", which means that it is not a case that "There exist a value a∈A such that P(a)=1 or P(a)="True", that is, for all values x∈A P(x)=0 or P(x)="False".

Thus, our statement S="**There exists x∈A such that P(x) is a true**" is the proposition with truth value True or False (but not both).

Observation: A propositional function P: A→{0, 1} itself is not a proposition but the statement S="There exists x∈A such that P(x) is a true" is always the proposition.

We summarize our observations as the Definition 16 below.

Definition 16. Let P(x) be a propositional function defined on a set A. That is, P: A→{0, 1}.
The existential quantification of P(x) is the new proposition S which is defined as:

S="**There exists x∈A such that P(x) is a true**" or, in denotations,

$$S=\exists x\mathbf{P(x)} \tag{3}$$

Here the symbol

$$\exists$$

which reads "there exists" or "for some" or "for at least one" is called the **existential quantifier.** ■

Note 4. A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning.

Remember that the truth value of $\exists xP(x)$ *depends on the domain!*

The expression P(x) by itself is a predicate (or propositional function) and therefore has no truth value. However, ∃xP(x), that is, P(x) preceded by the quantifier ∃, does have a truth value.

Theorem 5. The proposition (3) is equivalent to the proposition

$$T_P=\{x \mid x\in A, P(x)\}\neq\varnothing. \tag{4}$$

i.e., that the truth set of P(x) is not empty.

Proof. Clear, by definition T_P. ■

EXAMPLE 31. The next two examples are based on the equivalence between (3) and (4). Domain is N.

(a) Let P(n) is "n+5<10". Then the proposition $\exists nP(n)$ is true since $T_P=\{n|n+5<10\}=\{1,2,3,4\}\neq\varnothing$;

(b) Let P(n) is "n+10<5". Then the proposition $\exists nP(n)$ is false since $T_P=\{n \mid n+10<5\}=\varnothing$. ■

The meanings of the universal and existential quantifiers which we observed in Examples 29-31 are summarized in the Table 19 below.

Table 19. Meanings of Quantifiers.

Proposition	When True?	When False?
$\forall xP(x)$	P(x) is true for every x (in domain)).	There is an x (in domain) for which P(x) is false.
$\exists xP(x)$	There is an x (in domain) for which P(x) is true.	P(x) is false for every x (in domain).

The observations below provide useful ways to find truth values for quantified statements. ***The method is to represent quantified statement as compound proposition written with only disjunction, conjunction, and negation.***

Namely, when all elements in the domain can be listed - say, $x_1, x_2, ..., x_n$ – then:

Rule 1. The existential quantification $\exists xP(x)$ is the same as the disjunction

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

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

Rule 2. The universal quantification $\forall xP(x)$ is the same as the conjunction

$$P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \ldots, P(x_n)$ are all true.

EXAMPLE 32.

(a) What is the truth value of $\exists xP(x)$, where P(x) is the statement "$2x^3 < (-1000)$" and the domain is the set {-10, -9, …, -6}?

Solution. The proposition $\exists xP(x)$ is the same as the disjunction

$$P(-10) \vee P(-9) \vee P(-8) \vee P(-7) \vee P(-6)$$

because the domain consists of the integers: -10, -9, -8, -7, -6.

Since P(-10), which is the statement "$2(-10)^3 < (-1000)$", is true, it follows that $\exists xP(x)$ is true.

(b) What is the truth value of $\forall xP(x)$, where P(x) is the statement "$2x^3 < (-1000)$" and the domain is the set {-10, -9, …, -6}?

Solution. The proposition $\forall xP(x)$ is the same as the disjunction

$$P(-10) \wedge P(-9) \wedge P(-8) \wedge P(-7) \wedge P(-6)$$

because the domain consists of the integers: -10, -9, -8, -7, -6.

Since P(-7), which is the statement "$2(-7)^3 < (-1000)$", is false, it follows that $\forall xP(x)$ is false. ■

Precedence of Quantifiers

The quantifiers $\forall$ and $\exists$ have higher precedence than all logical operators of 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 $\forall x(P(x) \vee Q(x))$.

Logical Equivalences Involving Quantifiers

We extend the notion of logical equivalences to expressions involving predicates and quantifiers.

Definition 17. Statements S and T involving predicates and quantifiers are logically equivalent, denoted S≡T, if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain is used for the variables in these propositional functions. ■

Next Theorem illustrates how to show that two statements involving predicates and quantifiers are logically equivalent.

Theorem 6. Prove the following equivalences:

(a) $\forall x(P(x) \wedge Q(x)) \equiv \forall xP(x) \wedge \forall xQ(x)$
(b) $\exists x(P(x) \vee Q(x)) \equiv \exists xP(x) \vee \exists xQ(x)$

The propositional functions used have the same domains.

In other words, we can distribute:

(a) ***a universal quantifier over a conjunction*;**

(b) ***an existential quantifier over a disjunction*.**

Proof. We prove here only the statement (a). We must show that the propositions on both sides of (a) always take the same truth value, no matter what the predicates P and Q are, and no matter which domain is used.

Suppose we have predicates P and Q, with a common domain. We can show

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

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 vice versa: 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. It means that for every *a* from the domain, the proposition $P(a) \wedge Q(a)$ is true. Hence, $P(a)$ is true and $Q(a)$ is true for every *a* from the domain. Therefore, 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.

Conversely, 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. It follows that for all *a* from the domain $P(a) \wedge Q(a)$ is true. It follows that $\forall x(P(x) \wedge Q(x))$ is true. We can now conclude that

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

<u>Note 5</u>. *We cannot distribute a universal quantifier over a disjunction, nor can we distribute an existential quantifier over a conjunction* (see exercises).

The Uniqueness Quantifier.

We have introduced universal and existential quantifiers. These are the most important quantifiers in mathematics and computer science. However, there is no limitation on the number of different quantifiers we can define, such as "there are exactly two," "there are no more than three," "there are at least 100," and so on. Of these other quantifiers, the one that is most often seen is **the uniqueness quantifier**, denoted by $\exists!$

The notation $\exists!xP(x)$ states "There exists a **unique** x such that P(x) is true." (Other phrases for uniqueness quantification include "there is exactly one" and "there is one and only one").

For instance, $\exists!x(x-1=0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x-1=0$. This is a true statement, as $x=1$ is the unique real number such that $x-1=0$. Observe that we can use quantifiers and propositional logic to express uniqueness so the uniqueness quantifier can be avoided (see Exercise 52 in Section 1.5 KR textbook).

Generally, it is best to stick with existential and universal quantifiers so that rules of inference for these quantifiers can be used.

Negation of Quantified Statements

EXAMPLE 33. Consider the statement: “All Computer Science majors are male”. Its negation reads:

“It is not the case that all Computer Science majors are male” or, equivalently,
“There exists at least one Computer Science major who is a female (not male)”

Symbolically, using CS to denote the set of Computer Science majors, the above can be written as

$$\neg(\forall x \in CS)(x \text{ is male}) \equiv (\exists x \in CS)\ (x \text{ is not male})$$

or, when P(x) denotes “x is male,”

$$\neg(\forall x \in CS)P(x) \equiv (\exists x \in CS)\neg P(x) \qquad \text{or}$$

$$\neg\forall x P(x) \equiv \exists x \neg P(x)$$ ■

The statement $\neg\forall x P(x) \equiv \exists x \neg P(x)$ from Example 34 is true **for any propositional function (predicate) P(x) with an arbitrary domain A**. That is:

Theorem 7 (DeMorgan): $\neg\forall x P(x) \equiv \exists x \neg P(x)$.

In other words, the following two propositions are equivalent:

(1) It is not true that, for all x from domain the proposition P(x) is true.

(2) There exists an element x from domain such that the P(x) is false.

Proof. To show that $\neg\forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent first note that $\neg\forall x P(x)$ is true if and only if $\forall x P(x)$ is false. Next, note that $\forall x P(x)$ is false if and only if there is an element x in the domain for which P(x) is false. This holds if and only if there is an element x in the domain for which $\neg P(x)$ is true. Finally, note that there is an element x in the domain for which $\neg P(x)$ is true if and only if $\exists x \neg P(x)$ is true. Putting these steps together, we can conclude that $\neg\forall x P(x)$ is true if and only if $\exists x \neg P(x)$ is true. It follows that $\neg\forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent. ■

There is an analogous theorem for the negation of a proposition which contains the existential quantifier.

Theorem 8. (DeMorgan): $\neg\exists x P(x) \equiv \forall x \neg P(x)$.

That is, the following two propositions are equivalent:

(1) It is not true that for some x, P(x) is true.

(2) For all x, P(x) is false.

Proof. Again, no matter what P(x) is and what the domain is.

To show that $\neg\exists x P(x)$ and $\forall x \neg P(x)$ are logically equivalent first note that $\neg\exists x P(x)$ is true if and only if $\exists x P(x)$ is false. This is true if and only if no x exists in the domain for which P(x) is true. Next, note that no x exists in the domain for which P(x) is true if and only if P(x) is false for every x in the domain. Finally, note that P(x) is false for every x in the domain if and only if $\neg P(x)$ is true for all x in the domain, which holds if and only if $\forall x \neg P(x)$ is true. Putting these steps together, we see that $\neg\exists x P(x)$ is true if and only if $\forall x \neg P(x)$ is true. We conclude that $\neg\exists x P(x)$ and $\forall x \neg P(x)$ are logically equivalent. ■

Thus, we have the following rule:

$\forall$ is changed to $\exists$ and $\exists$ is changed to $\forall$ as the negation symbol $\neg$ passes through the proposition from left to right.

The rules for negations for quantifiers are summarized in Table 20.

Table 20. De Morgan's Laws for Quantifiers.

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

EXAMPLE 34. Show that $\neg\forall x(P(x)\rightarrow Q(x))$ and $\exists x(P(x)\wedge\neg Q(x))$ are logically equivalent.

Solution. By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x)\rightarrow Q(x))$ and $\exists x(\neg(P(x)\rightarrow Q(x)))$ are logically equivalent. By the logical equivalence (4) in Table 16, we know that $\neg(P(x)\rightarrow Q(x))$ and $P(x)\wedge\neg Q(x)$ are logically equivalent for every x. Because we can substitute one logically equivalent expression for another in a logical equivalence, it follows that $\neg\forall x(P(x)\rightarrow Q(x))$ and $\exists x(P(x)\wedge\neg Q(x))$ are logically equivalent. ■

Translating Statements from English into Logical Expressions and vice versa.

Translating sentences in English (or other natural languages) into logical expressions by using propositional functions, quantifications, and logical connectives is a crucial task in mathematics, logic programming, artificial intelligence, software engineering, and many other disciplines. In Examples 5, 6, 7, 10, 11 we translated some sentences from English into logical expressions. Translations in those examples, did not require predicates and quantifiers. Translating from English to logical expressions becomes even more complex when quantifiers are needed. Furthermore, **there can be many ways to translate a particular sentence**.

Examples below illustrate how to translate sentences from English into logical expressions using the existential and universal quantifiers and logical connectives. **When using propositional functions and quantifiers the structure of the domain of a propositional function is extremely important. Based on this observation, depending on the domain, we provide different translations of the same statement into logical expressions**.

Note. We should always adopt the simplest approach that is adequate for use in subsequent reasoning.

EXAMPLE 35. Express the statement "Every student in the class Discrete Structures has studied the course Calculus I" using predicates and quantifiers.

Solution. As we highlighted above there can be many ways to translate a particular sentence depending on domain of predicates used.

Approach 1. Let C(x) is the propositional function defined as C(x)="x has studied Calculus I" with domain D_1={all students of the class Discrete Structures}, $x\in D_1$. Then the given statement can be written as following logical expression

$$\forall xC(x)$$

Approach 2. Let D_2={all people}. If we change the domain of C(x) to consist all people (D_2), we will need to express our statemen as:

"For every person x, if person x is a student in Discrete Structures class then x has studied Calculus I". (*)

Let S(x) be the propositional function, defined as S(x)= "x is in the class Discrete Structures with domain $D(S)=D_2$. Clear that $D_1\subset D_2$ and if $x\in D_1$ then truth value of S(x) is True otherwise it is False. Therefore, our statement (*) can be expressed as

$$\forall x(S(x)\rightarrow C(x)).$$

Caution! Our statement cannot be expressed as $\forall x(S(x) \wedge C(x))$ because this statement says that all people are students in the class Discrete Structures and have studied Calculus I!

Note. When we are interested in the background of people in subjects besides Calculus I, we may prefer to use the two-variable quantifier Q(x, y) defined as Q(x, y)="student x has studied subject y." Then we would replace C(x) by Q(x, Calculus I) in both approaches to obtain

$$\forall x Q(x, \text{Calculus I}) \text{ or } \forall x(S(x) \rightarrow Q(x, \text{Calculus I}))$$

in Approaches 1 and 2 respectively ■

EXAMPLE 36. Express the statements

(a) "Some student in the class Discrete Structures has visited Shusha" and

(b) "Every student in the class Discrete Structures has visited either Kalbajar or Shusha"

using predicates and quantifiers.

Solution.

Define the predicates Sh(x) and K(x) as following:

Sh(x)="x has visited Shusha"

K(x)="x has visited Kalbajar".

(a) The statement "Some student in the class Discrete Structures has visited Shusha" means that

"There is a student in the class Discrete Structures with the property that the student has visited Shusha".

Case 1a. Assume that as a domain we use the set D_1={all students in the class Discrete Structures}. Then with domain D_1 we can translate the statement in (a) as

$$\exists x Sh(x).$$

Case 2a. If we are interested in people other than those in the class Discrete Structures, we look at the statement a little differently. Let D_2 be a new domain such that $D_1 \subset D_2$. Our statement can be expressed as

"There is a person x having the properties that x is a student in the class Discrete Structures and x has visited Shusha".

In this case, we introduce S(x) to represent "x is a student in the class Discrete Structures". Clear that the truth values of S(x) are

$S(x) \equiv T$ if $x \in D_1$ and $S(x) \equiv F$ if $x \in D_2 - D_1$. Our solution becomes

$$\exists x(S(x) \wedge Sh(x))$$

because the statement is that there is a person x who is a student in this class and who has visited Shusha.

Caution! Our statement cannot be expressed as $\exists x(S(x) \rightarrow Sh(x))$, which is true when there is someone not in the class Discrete Structures because, in that case, for such a person x, $S(x) \rightarrow Sh(x)$ becomes either $F \rightarrow T$ or $F \rightarrow F$, both of which are true for the conditional statement.

(b) Similarly, the statement "Every student in the class Discrete Structures has visited either Kalbajar or Shusha" means that

"For every x in the class Discrete Structures, x has the property that x has visited Shusha or x has visited Kalbajar".

(Note that we are assuming the inclusive, rather than the exclusive, OR here.)

Case 1b. Assume that as a domain we use the set D_1={all students in the class Discrete Structures}. Then with domain D_1, we can translate the statement in (b) as

$$\forall x(Sh(x) \lor K(x)).$$

Case 2b. If we are interested in people other than those in the class Discrete Structures, we look at the statement a little differently. Let D_2 be a new domain such that $D_1 \subset D_2$. Our statement in (b) can be expressed as

"For every person x, if x is a student in the class Discrete Structures, then x has visited Shusha or x has visited Kalbajar"

In this case, the statement in (b) can be expressed (by using quantifiers, predicates, and logical connectives) as

$$\forall x(S(x) \rightarrow (C(x) \lor M(x))).$$

Note. Instead of using Sh(x) and K(x) to represent that x has visited Shusha and x has visited Kalbajar, respectively, we could use a two-place predicate V(x, y) to represent "x has visited city y" In this case, V (x, Shusha) and V (x, Kalbajar) would have the same meaning as Sh(x) and K(x) and could replace them in our answers. If we are working with many statements that involve people visiting different cities, we might prefer to use this two-variable approach.

Otherwise, for simplicity, we would stick with the one-variable predicates Sh(x) and K(x) ■

EXAMPLE 37. Let P(x) be the statement "x has GPA>3.5" and let Q(x) be the statement "x knows the computer language Prolog". Express each of sentences below in terms of P(x), Q(x), quantifiers, and logical connectives. The domain for quantifiers consists of all students at SITE of ADA University.

a) There is a student at your school who has GPA>3.5 and who knows Prolog.

b) There is a student at your school who has GPA>3.5 but who does not know Prolog.

c) Every student at your school either has GPA>3.5 or knows Prolog (inclusive OR).

d) No student at your school has GPA>3.5 or knows Prolog.

Solution.

a) $\exists x(P(x) \land Q(x))$ b) $\exists x(P(x) \land \neg Q(x))$

c) $\forall x(P(x) \lor Q(x))$ d) $\forall x \neg(P(x) \lor Q(x)) = \forall x(\neg P(x) \lor \neg Q(x))$ (De Morgan Law ■

EXAMPLE 38. Let P(x) be the statement "x spends more than 5 hours every weekday in class", where the domain for x consists of all students of a school.
Express each of these quantifications in English.

a) $\exists xP(x)$ b) $\forall xP(x)$ c) $\exists x\neg P(x)$ d) $\forall x\neg P(x)$

Solution. According to meanings of quantifiers from Table 19 we have:

a) There is a student who spends more than 5 hours every weekday in class.
b) Every student spends more than 5 hours every weekday in class.
c) There is a student who does not spend more than 5 hours every weekday in class.
d) No student spends more than 5 hours every weekday in class. ■

Next two examples come from Lewis Carroll book "Symbolic Logic" (see also KR textbook, p.50-51).

EXAMPLE 39. Consider these statements. The first two are called *premises* and the third is called the *conclusion*. The entire set is called an *argument*.

"All lions are fierce."

"Some lions do not drink coffee."

"Some fierce creatures do not drink coffee."

Assuming that, the domain consists of all creatures, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, and $R(x)$. Here,

$P(x)$="x is a lion",

$Q(x)$="x drinks coffee",

$R(x)$="x is fierce".

Solution. We can express these statements as:

$$\forall x(P(x)\to Q(x)).$$
$$\exists x(P(x)\wedge\neg R(x)).$$
$$\exists x(Q(x)\wedge\neg R(x)).$$

Note. The *second statement* cannot be written as $\exists x(P(x)\to\neg R(x))$. The reason is that $P(x)\to\neg R(x)$ is true whenever x is not a lion, so that $\exists x(P(x)\to\neg R(x))$ is true as long as there is at least one creature that is not a lion, even if every lion drinks coffee. Similarly, the third statement cannot be written as $\exists x(Q(x)\to\neg R(x))$. ■

EXAMPLE 40. Consider these statements, of which *the first three are premises* and the *fourth is a valid conclusion.*

"All hummingbirds are richly colored."

"No large birds live on honey."

"Birds that do not live on honey are dull in color."

"Hummingbirds are small."

Let

$P(x)$="x is a hummingbird"

$Q(x)$="x is large"

$R(x)$="x lives on honey"

$S(x)$="x is richly colored".

Assuming that, the *domain each predicate above consists of all birds*, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

Solution. Assuming that "small" is the same as "not large" and that "dull in color" is the same as "not richly colored" we can express the statements in the argument as

$$\forall x(P(x)\to S(x))$$
$$\neg\exists x(Q(x)\wedge R(x))$$
$$\forall x(\neg R(x)\to\neg S(x))$$
$$\forall x(P(x)\to\neg Q(x))$$

Exercise. To show that the fourth statement is a valid conclusion of the first three, we need to use rules of inference (See Section 1.6. in KR as a self-study).

Section 3. Nested Quantifiers

In Section 2 we introduced existential and universal quantifiers as ways **to get propositions from propositional functions of one logical variable** using quantifications (Definitions 15 and 16).

Besides, we showed how the quantifiers can be used to represent mathematical statements and explained how they can be used to translate English sentences into logical expressions.

In Section 3 we **extend the idea of quantification to turn propositional functions of many variables into propositions**.

Let us start with functions of two variables. Let P: X×Y→{0, 1} be a propositional function of two logical variables x and y with domains X and Y, respectively. How to turn the function P into a proposition?

Basic Principle: A propositional function P: X×Y→{0, 1} preceded **by a quantifier for each variable**, for example,

$$\forall x \exists y P(x, y) \text{ or } \exists x \forall y P(x, y)$$

denotes a proposition and has a truth value True or False.

Clear that we have four possible combinations in this case

1. $\forall x \exists y$ P(x, y) – for every x∈X there exists y∈Y such that the proposition P(x, y) is true;
2. $\forall x \forall y$ P(x, y) – for every x∈X and for every y∈Y the proposition P(x, y) is true;
3. $\exists x \exists y$ P(x, y) – there exist x∈X and y∈Y such that the proposition P(x, y) is true;
4. $\exists x \forall y$ P(x, y) – there exists x∈X such that for every y∈Y the proposition P(x, y) is true.

TABLE 21. 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.

Examples below show that the order of quantifiers is crucially important.

EXAMPLE 41. Let X=Y={1, 2, 3, ..., 9} and let P: X×Y→{0, 1} be the propositional function of two logical variables *x* and *y*, which is defined by the predicate P=**"*x*+*y*=10".**

(a) The following is a proposition since there is a quantifier for each variable:
$\forall x \exists y P(x, y)$, that is, "For every *x*, there exists a *y* such that *x*+*y*=10".
Truth value of this proposition is **True**. For example, if *x*=1, let *y*=9; if *x*=2, let *y*=8, and so on.

(b) The following is also a proposition:
$\exists y \forall x P(x, y)$, that is, "There exists a ***y*** such that, for every *x*, we have *x*+*y*=10"
No such y exists; hence this proposition is **False**. ■

__Note.__ The only difference between (a) and (b) in Example 38 is the order of the quantifiers. Thus, a **different ordering of the quantifiers may yield different truth values for the same predicate.** We note that, when translating such quantified statements into English, the expression "such that" frequently follows "there exists".

EXAMPLE 42. Determine the truth value of each of the following propositions which are produced in examples (a), (b), and (c) from the respective propositional functions with domain A={1, 2, 3}:

(a) $\exists x \forall y P(x, y)$="$x^2$<y+1";

(b) $\forall x \exists y P(x, y)$="$x^2+y^2$<12";

(c) $\forall x \forall y P(x, y)$=”$x^2+y^2<12$”.

Solution.

(a) True. For if x=1, then 1, 2, and 3 are all solutions to 1<y+1.

(b) True. For each x_0, let y=1; then $x_0^2+1<12$ is a true proposition.

(c) False. For if $x_0=2$ and $y_0=3$, then $x_0^2+y_0^2<12$ is not a true proposition. ■

Negating Quantified Predicates with more than One Variable.

Quantified predicates with more than one variable may be negated by successively applying Theorems 6 and 7 (see also Table 20). Thus, we have the following rule:

Each ∀ is changed to ∃ and each ∃ is changed to ∀ as the negation symbol ¬ passes through the proposition from left to right.

For example,

$$\neg[\forall x \exists y \exists z P(x, y, z)] \equiv \exists x \neg[\exists y \exists z P(x, y, z)] \equiv \exists x \forall y \neg[\exists z P(x, y, z)] \equiv \exists x \forall y \forall z \neg P(x, y, z)$$

EXAMPLE 43. Negate each of the following propositions:

(a) $\exists x \forall y P(x, y)$; (b) $\exists x \forall y P(x, y)$; (c) $\exists y \exists x \forall z P(x, y, z)$.

Solution.

Use De Morgan’s Laws for Quantifiers from Table 20 (or Theorem 7 and 8), that is:

$\neg \forall x P(x) \equiv \exists x \neg P(x)$ and $\neg \exists x P(x) \equiv \forall x \neg P(x)$:

We obtain:

(a) $\neg(\exists x \forall y P(x, y)) \equiv \forall x \exists y \neg P(x, y)$

(b) $\neg(\forall x \forall y P(x, y)) \equiv \exists x \exists y \neg P(x, y)$

(c) $\neg(\exists y \exists x \forall z P(x, y, z)) \equiv \forall y \forall x \exists z \neg P(x, y, z)$ ■

EXERCISES. SET 1 (Solved Problems)

Propositions; Truth Tables; Compound Propositions.

1.1. Let p= "It is cold" and let q= "It is raining". Give a simple verbal sentence which describes each of the following statements: (a) $\neg p$; (b) $p \wedge q$; (c) $p \vee q$; (d) $q \vee \neg p$.

Solution. In each case, translate $\wedge$, $\vee$, and $\neg$ to read "and," "or," and "not," respectively, and then simplify the English sentence.

(a) It is not cold.
(b) It is cold and raining.
(c) It is cold or it is raining.
(d) It is raining or it is not cold.

1.2. Verify that the proposition $p \vee \neg(p \wedge q)$ is a tautology.

Solution. Construct the truth table of $p \vee \neg(p \wedge q)$. Since the truth value of $p \vee \neg(p \wedge q)$ is T for all values of p and q, the proposition is a tautology.

1.3. Show that the propositions $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are logically equivalent (De Morgan's Law).

Solution. Since the truth tables for $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are the identical, the propositions $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are logically equivalent and we can write $\neg(p \wedge q) \equiv \neg p \vee \neg q$.

1.4. Use the laws in Table 14 to show that $\neg(p \vee q) \vee (\neg p \wedge q) \equiv \neg p$.

Solution.

Statement	Reason
(1) $\neg(p \vee q) \vee (\neg p \wedge q) \equiv (\neg p \wedge \neg q) \vee (\neg p \wedge q)$	De Morgan's law
(2) $\equiv \neg p \wedge (\neg q \vee q)$	Distributive law
(3) $\equiv \neg p \wedge T$	Complement law
(4) $\equiv \neg p$	Identity law

1.5. Use De Morgan's laws to find the negation of each of the following statements.

(a)Jan is rich and happy.
(b) Akif will bicycle or run tomorrow.
(c) Aynur walks or takes the bus to class.
(d) Ibrahim is smart and hard working.

Solution.

(**a**). Let p="Jan is rich" and q="Jan is happy". Then $(p \wedge q)$ means "Jan is rich and happy". By De Morgan's law we find that $\neg(p \wedge q) = \neg p \vee \neg q$. Thus, a simple verbal sentence which describes negation of statement (a) is "Jan is not rich, or Jan is not happy".

Similarly, for other cases we obtain:

(b) Akif will not bicycle tomorrow, and Akif will not run tomorrow.
(c) Aynur does not walk to class, and Aynur does not take the bus to class.
(d) Ibrahim is not smart, or Ibrahim is not hard working.

Conditional Statements

1.6. Rewrite the following statements without using the conditional:

(a)If it is cold, he wears a hat.
(b) If productivity increases, then wages rise.

Solution. Recall that $p \rightarrow q \equiv \neg p \vee q$. Hence,

(a) It is not cold or he wears a hat.

(b) Productivity does not increase or wages rise.

1.7. Determine the contrapositive of each statement:

(a) If Erik is a school teacher, then he is poor.

(b) Only if Marc studies will he pass the test.

Solution.

(a) The contrapositive of $p \to q$ is $\neg q \to \neg p$. Hence the contrapositive follows: "If Erik is not poor, then he is not a school teacher".

(b) The statement is equivalent to: "If Marc passes the test, then he studied". Thus, its contrapositive is: "If Marc does not study, then he will not pass the test".

1.8. Write the negation of each statement as simply as possible:

(a) If she works, she will earn money.
(b) He swims if and only if the water is warm.
(c) If it snows, then they do not drive the car.

Solution.

(a) Note that $\neg(p \to q) \equiv p \wedge \neg q$; hence the negation of the statement is:

"She works and she will not earn money".

(b) Note that $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q \equiv \neg p \leftrightarrow q$; hence the negation of the statement is either of the following:

"He swims if and only if the water is not warm" or
"He does not swim if and only if the water is warm."

(c) Note that $\neg(p \to \neg q) \equiv p \wedge \neg\neg q \equiv p \wedge q$. Hence the negation of the statement is:

"It snows and they drive the car".

Truth values of Propositions defined as values of Propositional Functions (=Predicates) or as a or quantifications of Propositional Functions

1.9. Let P(x) be the predicate (propositional function) "$x=x^2$" If the domain consists of the **integers (=Z)**, what are **truth values** of the following propositions?

a) P(0) b) P(1) c) P(2)
d) P(−1) e) $\exists x P(x)$ f) $\forall x P(x)$

Solution.

a) $0=0^2 \Rightarrow P(0)=$ "T" b) $1=1^2 \Rightarrow P(1)=$ "T" c) $2\neq 2^2 \Rightarrow P(2)=$ "F"
d) $(-1)\neq(-1)^2 \Rightarrow P(-1)=$ "F" e) $1=1^2 \Rightarrow \exists x P(x)=$ "T" f) $2\neq 2^2 \Rightarrow \forall x P(x))=$ "F"

1.10. Determine the truth value of each propositions below if the domain of all predicates is the set of all integers **(=Z)**.

a) $\forall n(n+1>n)$ b) $\exists n(2n=3n)$
c) $\exists n(n=-n)$ d) $\forall n(3n\leq 4n)$

Solution.

a) As for arbitrary integer n we have $n+1>n$ so $\forall n(n+1>n)=$ "T"
b) If n=0 then $2(0)=3(0) \Rightarrow \exists n(2n=3n)=$ "T"
c) If n=0 then $0=-0 \Rightarrow \exists n(n=-n)=$ "T"
d) If n<0 then $3n>4n \Rightarrow \forall n(3n\leq 4n))=$ "F"

Quantified Propositions as compound propositions written only by disjunction, conjunction, and negation.

1.11. Suppose that the domain of the propositional function P(x) is the set: DomP(x)={0, 1, 2, 3, 4} Write out propositions below using disjunctions, conjunctions, and negations.

a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\exists x \neg P(x)$
d) $\forall x \neg P(x)$ e) $\neg \exists x P(x)$ f) $\neg \forall x P(x)$
g) $\forall x((x\neq 3) \to P(x)) \vee \exists x \neg P(x)$

Solution. Apply Rule 1 and Rule 2 after Table 19. We have:

a) $P(0)\vee P(1)\vee P(2)\vee P(3)\vee P(4)$
b) $P(0)\wedge P(1)\wedge P(2)\wedge P(3)\wedge P(4)$
c) $\neg P(0)\vee\neg P(1)\vee\neg P(2)\vee\neg P(3)\vee\neg P(4)$
d) $\neg P(0)\wedge\neg P(1)\wedge\neg P(2)\wedge\neg P(3)\wedge\neg P(4)$
e) $\neg(P(0)\vee P(1)\vee P(2)\vee P(3)\vee P(4))$
f) $\neg(P(0)\wedge P(1)\wedge P(2)\wedge P(3)\wedge P(4))$
g) $(P(0)\wedge P(1)\wedge P(2)\wedge P(4))\vee(\neg P(0)\vee\neg P(1)\vee\neg P(2)\vee\neg P(3)\vee\neg P(4))$

Translating Statements from English into Logical Expressions and vice versa.

1.12. Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives. Domain is all people.

a) No one is perfect.
b) Not everyone is perfect.
c) All your friends are perfect.
d) At least one of your friends is perfect.
e) Everyone is your friend and is perfect.
f) Not everybody is your friend or someone is not perfect.

Solution. Let P(x) be "x is perfect"; F(x) be "x is your friend"; and let the domain be all people. Then:

a) $\forall x\neg P(x)$ b) $\neg\forall xP(x)$ c) $\forall x(F(x)\rightarrow P(x))$

d) $\exists x(F(x)\wedge P(x))$ e) $\forall x(F(x)\wedge P(x))$ or $(\forall xF(x))\wedge(\forall xP(x))$ (Theorem 6)

f) $(\neg\forall xF(x))\vee(\exists x\neg P(x))$

1.13. Each of the statements below

(a) Someone in your class can speak Turkish.
(b) Everyone in your class is friendly.
(c) There is a person in your class who was not born in Ganja.
(d) A student in your class has been in a movie.
(e) No student in your class has taken a course in logic programming.

translate in two ways into logical expressions using predicates, quantifiers, and logical connectives.

(*Way 1*) Domain consist of the students in your class.

(*Way 2*) Domain consists of all people.

Solution. Let

C(x)="x is in your class";

T(x)="x can speak Turkish"

F(x)="x is friendly"

G(x)="x was born in Ganja"

M(x)="x has been in a movie"

L(x)="x has taken a course in logic programming"

Way 1. The seeking expressions are

(a). $\exists xT(x)$

(b). $\forall xF(x)$

(c). $\exists x \neg G(x)$

(d). $\exists x M(x)$

(e). $\forall x \neg L(x)$

<u>*Way 2.*</u> The seeking expressions are

(a). $\exists x(C(x) \wedge T(x))$

(b). $\forall x(C(x) \rightarrow F(x))$

(c). $\exists x(C(x) \wedge \neg G(x))$

(d). $\exists x(C(x) \wedge M(x))$,

(e). $\forall x(C(x) \rightarrow \neg L(x))$,

Propositional functions of many variables. Nested quantifiers.

1.14. Let A = {1, 2, . . . , 9, 10}. Consider each of the following expressions. If it is a proposition, then determine its truth value. If it is a propositional function, determine its truth set.

(a) $(\forall x \in A)(\exists y \in A)(x+y<14)$

(b) $(\forall y \in A)(x+y<14)$

(c) $(\forall x \in A)(\forall y \in A)(x+y<14)$

(d) $(\exists y \in A)(x+y<14)$

Solution. Let

(a) The open sentence in two variables is preceded by two quantifiers; hence it is a proposition. Moreover, the proposition is true.

(b) The open sentence is preceded by one quantifier; hence it is a propositional function of the other variable.

Note that for every $y \in A$, $x_0+y<14$ if and only if $x_0=1, 2$, or 3. Hence the truth set is {1,2,3}.

(c) It is a statement and it is false: if $x_0=8$ and $y_0=9$, then $x_0+y_0<14$ is not true.

(d) It is a propositional function in x. The truth set is A itself.

EXERCISES. SET 2

Propositions; Truth Tables; Compound Propositions.

2.1. For each of the statements (a), (b), and (c) below:

(a) If the teacher is absent, then some students do not complete their homework.

(b) All the students completed their homework and the teacher is present.

(c) Some of the students did not complete their homework or the teacher is absent.

Do the following tasks

1) Introduce appropriate propositions to write statement as compound propositions
2) Write statement as compound proposition using logical operations
3) Negate the compound proposition
4) Write out the negated compound proposition in English

Truth values of Propositions defined as values of Propositional Functions (=Predicates) or as quantifications of Propositional Functions

2.2. Let Q(x) be the propositional function “x+1>2x”. If the domain consists of all integers, what are truth values of propositions below?

a) Q(0) b) Q(−1) c) Q(1)

d) $\exists xQ(x)$ e) $\forall xQ(x)$ f) $\exists x\neg Q(x)$ g) $\forall x\neg Q(x)$

2.3. Determine the truth value of each proposition below if the domain consists of all real numbers.

a) $\exists x(x^3=-1)$ b) $\exists x(x^4<x^2)$

c) $\forall x((-x)^2=x^2)$ d) $\forall x(2x>x)$

2.4. Determine the truth value of each proposition below if the domain consists of all real numbers.

a) $\exists x(x^2=2)$ b) $\exists x(x^2=-1)$

c) $\forall x(x^2+2\geq 1)$ d) $\forall x(x^2\neq x)$

2.5. Let A={1, 2, 3, 4, 5}.

Subtask 1. Determine the truth value of each of the following propositions:

(a) $(\exists x \in A)(x + 3 = 10)$ (c) $(\exists x \in A)(x + 3 < 5)$

(b) $(\forall x \in A)(x + 3 < 10)$ (d) $(\forall x \in A)(x + 3 \leq 7)$

Subtask 2. Negate each proposition in Subtask 1

2.6. As mentioned in the text, the notation $\exists !xP(x)$ denotes "There exists a unique x such that P(x) is true." If the domain consists of *all integers*, what are the truth values of these propositions?

a) $\exists !x(x>1)$ b) $\exists !x(x^2=1)$

c) $\exists !x(x+3=2x)$ d) $\exists !x(x=x+1)$

2.7. What are the truth values of these propositions?

a) $\exists !xP(x) \rightarrow \exists xP(x)$

b) $\forall xP(x) \rightarrow \exists !xP(x)$

c) $\exists !x\neg P(x)\rightarrow\neg\forall xP(x)$

Quantified Propositions as compound propositions written by only disjunction, conjunction, and negation.

2.8. Suppose that the domain of the propositional function P(x) is: DomP(x)={-2, -1, 0, 1, 2}. Write out propositions below using disjunctions, conjunctions, and negations.

a) $\exists xP(x)$ b) $\forall xP(x)$ c) $\exists x\neg P(x)$

d) $\forall x\neg P(x)$ e) $\neg\exists xP(x)$ f) $\neg\forall xP(x)$

2.9. Suppose that the domain of the propositional function P(x) is: DomP(x)={-5, -3, -1, 1, 3, 5}. Write out propositions below using only disjunctions, conjunctions, and negations.

a) $\exists xP(x)$ b) $\forall xP(x)$ c) $\forall x((x\neq 1)\rightarrow P(x))$

d) $\exists x((x\geq 0)\wedge P(x))$ e) $\exists x(\neg P(x))\wedge\forall x((x<0)\rightarrow P(x))$

2.10. Assume that for the given propositional function P(x) the domain D={1, 2, 3}. Write out $\exists !xP(x)$, in terms of negations, conjunctions, and disjunctions.

Translating Statements from English into Logical Expressions and vice versa.

Note. Keep in mind that Logical Expressions consist of propositional functions, logical connectives, and quantifiers.

2.11. Let G(x) be the statement "x has visited Ganja," where the domain consists of the students in your school. Express each of these quantifications in English.

a) $\exists xG(x)$ b) $\forall xG(x)$ c) $\neg\exists xG(x)$

d) $\exists x \neg G(x)$ e) $\neg \forall x G(x)$ f) $\forall x \neg G(x)$

2.12. Let G(x) be the propositional function "x has GPA>3", where the domain consists of the students in your school. Express each of these quantifications in English.

a) $\exists x G(x)$ b) $\forall x G(x)$ c) $\neg \exists x G(x)$
d) $\exists x \neg G(x)$ e) $\neg \forall x G(x)$ f) $\forall x \neg G(x)$

2.13. Translate these statements into English, where R(x) is "x is a student of the School of IT and Engineering (SITE)" and H(x) is "x takes Discrete structures course" and the domain consists of all ADA University students.

a) $\forall x(R(x) \rightarrow H(x))$ b) $\forall x(R(x) \wedge H(x))$
c) $\exists x(R(x) \rightarrow H(x))$ d) $\exists x(R(x) \wedge H(x))$

2.14. Assume that C(x), D(x), and L(x) are the following functions:

C(x) - "x completed Calculus I course";
D(x) - "x completed Discrete Structures course";
S(x) - "x completed Statistics course"

Assume also that the domain of all functions above consists of *all students in your class*. Express each of statements below in terms of C(x), D(x), S(x), quantifiers, and logical connectives.

a) A student in your class passed Calculus I, Discrete Structures, and Statistics.
b) All students in your class passed Calculus I, Discrete Structures, and Statistics.
c) Some student in your class passed Calculus I, Discrete Structures, but not Statistics.
d) No student in your class passed Calculus I, Discrete Structures, and Statistics.
e) For each of the three courses, Calculus I, Discrete Structures, and Statistics, there is a student in your class who passed the course.

2.15. Each of the statements below

a) Everyone in your class has a cellular phone.
b) Somebody in your class has seen a foreign movie.
c) There is a person in your class who cannot swim.
d) All students in your class can knows the Rolle's Theorem .
e) Some student in your class does not want to continue his education in Master level.

translate in two ways into logical expressions using predicates, quantifiers, and logical connectives.

(*Way 1*) Domain consist of the students in your class.

(*Way 2*) Domain consists of all people.

2.16. Show that $\forall x P(x) \vee \forall x Q(x)$ and $\forall x(P(x) \vee Q(x))$ are not logically equivalent.

2.17. Show that $\exists x P(x) \wedge \exists x Q(x)$ and $\exists x(P(x) \wedge Q(x))$ are not logically equivalent.