CHAPTER 2

Logic

1. Logic Definitions

1.1. Propositions.

Definition 1.1.1. *A* **proposition** *is a declarative sentence that is either*

true (denoted either T or 1) or

false (denoted either F or 0).

Notation: Variables are used to represent propositions. The most common variables used are p, q, and r.

Discussion

Logic has been studied since the classical Greek period (600-300BC). The Greeks, most notably Thales, were the first to formally analyze the reasoning process. Aristotle (384-322BC), the "father of logic", and many other Greeks searched for universal truths that were irrefutable. A second great period for logic came with the use of symbols to simplify complicated logical arguments. Gottfried Leibniz (1646-1716) began this work at age 14, but failed to provide a workable foundation for symbolic logic. George Boole (1815-1864) is considered the "father of symbolic logic". He developed logic as an abstract mathematical system consisting of defined terms (propositions), operations (conjunction, disjunction, and negation), and rules for using the operations. It is this system that we will study in the first section.

Boole's basic idea was that if simple propositions could be represented by precise symbols, the relation between the propositions could be read as precisely as an algebraic equation. Boole developed an "algebra of logic" in which certain types of reasoning were reduced to manipulations of symbols.

1.2. Examples.

Example 1.2.1. *"Drilling for oil caused dinosaurs to become extinct." is a proposition.*

EXAMPLE 1.2.2. *"Look out!" is not a proposition.*

EXAMPLE 1.2.3. *"How far is it to the next town?" is not a proposition.*

EXAMPLE 1.2.4. *"$x + 2 = 2x$" is not a proposition.*

EXAMPLE 1.2.5. *"$x + 2 = 2x$ when $x = -2$" is a proposition.*

Recall a *proposition* is a declarative sentence that is either true or false. Here are some further examples of propositions:

EXAMPLE 1.2.6. *All cows are brown.*

EXAMPLE 1.2.7. *The Earth is further from the sun than Venus.*

EXAMPLE 1.2.8. *There is life on Mars.*

EXAMPLE 1.2.9. $2 \times 2 = 5$.

Here are some sentences that are not propositions.

EXAMPLE 1.2.10. *"Do you want to go to the movies?" Since a question is not a declarative sentence, it fails to be a proposition.*

EXAMPLE 1.2.11. *"Clean up your room." Likewise, an imperative is not a declarative sentence; hence, fails to be a proposition.*

EXAMPLE 1.2.12. *"$2x = 2 + x$." This is a declarative sentence, but unless x is assigned a value or is otherwise prescribed, the sentence neither true nor false, hence, not a proposition.*

EXAMPLE 1.2.13. *"This sentence is false." What happens if you assume this statement is true? false? This example is called a paradox and is not a proposition, because it is neither true nor false.*

Each proposition can be assigned one of two *truth values*. We use T or 1 for true and use F or 0 for false.

1.3. Logical Operators.

DEFINITION 1.3.1. *Unary Operator* **negation:** *"not p", $\neg p$.*

DEFINITIONS 1.3.1. *Binary Operators*

(a) **conjunction:** *"p and q", $p \wedge q$.*
(b) **disjunction:** *"p or q", $p \vee q$.*
(c) **exclusive or:** *"exactly one of p or q", "p xor q", $p \oplus q$.*
(d) **implication:** *"if p then q", $p \to q$.*
(e) **biconditional:** *"p if and only if q", $p \leftrightarrow q$.*

Discussion

A sentence like "I can jump and skip" can be thought of as a combination of the two sentences "I can jump" and "I can skip." When we analyze arguments or logical expression it is very helpful to break a sentence down to some composition of simpler statements.

We can create *compound propositions* using propositional variables, such as $p, q, r, s, ...$, and *connectives* or *logical operators*. A logical operator is either a *unary* operator, meaning it is applied to only a single proposition; or a *binary* operator, meaning it is applied to two propositions. *Truth tables* are used to exhibit the relationship between the truth values of a compound proposition and the truth values of its component propositions.

1.4. Negation. **Negation Operator**, "not", has symbol $\neg$.

EXAMPLE 1.4.1. *p: This book is interesting.*

$\neg p$ can be read as:

(i.) This book is not interesting.
(ii.) This book is uninteresting.
(iii.) It is not the case that this book is interesting.

Truth Table:

p	$\neg p$
T	F
F	T

Discussion

The *negation* operator the a unary operator which, when applied to a proposition p, changes the truth value of p. That is, the negation of a proposition p, denoted by $\neg p$, is the proposition that is false when p is true and true when p is false. For example, if p is the statement "I understand this", then its negation would be "I do not understand this" or "It is not the case that I understand this". Another notation commonly used for the negation of p is $\sim p$.

Generally, an appropriately inserted "not" or removed "not" is sufficient to negate a simple statement. Negating a compound statement may be a bit more complicated as we will see later on.

1.5. Conjunction. Conjunction Operator, "and", has symbol $\wedge$.

EXAMPLE 1.5.1. *p: This book is interesting. q: I am staying at home.*

$p \wedge q$*: This book is interesting, and I am staying at home.*

Truth Table:

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

Discussion

The *conjunction* operator is the binary operator which, when applied to two propositions p and q, yields the proposition "p and q", denoted $p \wedge q$. The conjunction $p \wedge q$ of p and q is the proposition that is true when both p and q are true and false otherwise.

1.6. Disjunction. Disjunction Operator, inclusive "or", has symbol $\vee$.

EXAMPLE 1.6.1. *p: This book is interesting. q: I am staying at home.*

$p \vee q$*: This book is interesting, or I am staying at home.*

Truth Table:

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

Discussion

The *disjunction* operator is the binary operator which, when applied to two propositions p and q, yields the proposition "p or q", denoted $p \vee q$. The disjunction $p \vee q$ of p and q is the proposition that is true when either p is true, q is true, or *both* are true, and is false otherwise. Thus, the "or" intended here is the *inclusive or.* In fact, the symbol $\vee$ is the abbreviation of the Latin word *vel* for the inclusive "or".

1.7. Exclusive Or. Exclusive Or Operator, "xor", has symbol $\oplus$.

EXAMPLE 1.7.1. *p: This book is interesting. q: I am staying at home.*

$p \oplus q$: *Either this book is interesting, or I am staying at home, but not both.*

Truth Table:

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

Discussion

The *exclusive or* is the binary operator which, when applied to two propositions p and q yields the proposition "p xor q", denoted $p \oplus q$, which is true if exactly one of p or q is true, but not both. It is false if both are true or if both are false.

Many times in our every day language we use "or" in the exclusive sense. In logic, however, we always mean the inclusive or when we simply use "or" as a connective in a proposition. If we mean the exclusive or it must be specified. For example, in a restaurant a menu may say there is a choice of soup or salad with a meal. In logic this would mean that a customer may choose both a soup and salad with their meal. The logical implication of this statement, however, is probably not what is intended. To create a sentence that logically states the intent the menu could say that there is a choice of *either* soup or salad (but not both). The phrase "either . . . or . . . " is normally indicates the exclusive or.

1.8. Implications. Implication Operator, "if...then...", has symbol $\rightarrow$.

EXAMPLE 1.8.1. *p: This book is interesting. q: I am staying at home.*

$p \rightarrow q$: *If this book is interesting, then I am staying at home.*

Truth Table:

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

Equivalent Forms of "If p then q":

- p implies q
- If p, q
- p only if q
- p is a sufficient condition for q
- q if p
- q whenever p
- q is a necessary condition for p

Discussion

The *implication* $p \to q$ is the proposition that is often read "if p then q." "If p then q" is false precisely when p is true but q is false. There are many ways to say this connective in English. You should study the various forms as shown above.

One way to think of the meaning of $p \to q$ is to consider it a contract that says if the first condition is satisfied, then the second will also be satisfied. If the first condition, p, is not satisfied, then the condition of the contract is null and void. In this case, it does not matter if the second condition is satisfied or not, the contract is still upheld.

For example, suppose your friend tells you that if you meet her for lunch, she will give you a book she wants you to read. According to this statement, you would expect her to give you a book if you do go to meet her for lunch. But what if you do not meet her for lunch? She did not say anything about that possible situation, so she would not be breaking any kind of promise if she dropped the book off at your house that night or if she just decided not to give you the book at all. If either of these last two possibilities happens we would still say the implication stated was true, because she did not break her promise.

Exercise 1.8.1. *Which of the following statements are equivalent to "If x is even, then y is odd"? There may be more than one or none.*

(1) *y is odd only if x is even.*
(2) *x is even is sufficient for y to be odd.*
(3) *x is even is necessary for y to be odd.*
(4) *If x is odd, then y is even.*
(5) *x is even and y is even.*
(6) *x is odd or y is odd.*

1.9. Terminology. For the compound statement $p \to q$

- p is called the **premise, hypothesis,** or the **antecedent**.
- q is called the **conclusion** or **consequent**.
- $q \to p$ is the **converse** of $p \to q$.

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

Discussion

We will see later that the converse and the inverse are not equivalent to the original implication, but the contrapositive $\neg q \to \neg p$ is. In other words, $p \to q$ and its contrapositive have the exact same truth values.

1.10. Example.

EXAMPLE 1.10.1. *Implication: If this book is interesting, then I am staying at home.*

- **Converse:** *If I am staying at home, then this book is interesting.*
- **Inverse:** *If this book is not interesting, then I am not staying at home.*
- **Contrapositive:** *If I am not staying at home, then this book is not interesting.*

Discussion

The converse of your friend's promise given above would be "if she gives you a book she wants you to read, then you will meet her for lunch," and the inverse would be "If you do not meet her for lunch, then she will not give you the book." We can see from the discussion about this statement that neither of these are the same as the original promise. The contrapositive of the statement is "if she does not give you the book, then you do not meet her for lunch." This is, in fact, equivalent to the original promise. Think about when would this promise be broken. It should be the exact same situation where the original promise is broken.

EXERCISE 1.10.1. *p is the statement "I will prove this by cases", q is the statement "There are more than 500 cases," and r is the statement "I can find another way."*

1. *State $(\neg r \vee \neg q) \to p$ in simple English.*
2. *State the converse of the statement in part (a) in simple English.*
3. *State the inverse of the statement in part (a) in simple English.*
4. *State the contrapositive of the statement in part (a) in simple English.*

1.11. Biconditional. Biconditional Operator, "if and only if", has symbol $\leftrightarrow$.

EXAMPLE 1.11.1. *p: This book is interesting. q: I am staying at home.*

$p \leftrightarrow q$: *This book is interesting if and only if I am staying at home.*

Truth Table:

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

Discussion

The biconditional statement is equivalent to $(p \to q) \wedge (q \to p)$. In other words, for $p \leftrightarrow q$ to be true we must have both p and q true or both false. The difference between the implication and biconditional operators can often be confusing, because in our every day language we sometimes say an "if...then" statement, $p \to q$, when we actually mean the *biconditional* statement $p \leftrightarrow q$. Consider the statement you may have heard from your mother (or may have said to your children): "If you eat your broccoli, then you may have some ice cream." Following the strict logical meaning of the first statement, the child still may or may not have ice cream even if the broccoli isn't eaten. The "if...then" construction does not indicate what would happen in the case when the hypothesis is not true. The intent of this statement, however, is most likely that the child *must* eat the broccoli in order to get the ice cream.

When we set out to prove a biconditional statement, we often break the proof down into two parts. First we prove the implication $p \to q$, and then we prove the converse $q \to p$.

Another type of "if...then" statement you may have already encountered is the one used in computer languages. In this "if...then" statement, the premise is a condition to be tested, and if it is true then the conclusion is a procedure that will be performed. If the premise is not true, then the procedure will not be performed. Notice this is different from "if...then" in logic. It is actually closer to the biconditional in logic. However, it is not actually a logical statement at all since the "conclusion" is really a list of commands, not a proposition.

1.12. NAND and NOR Operators.

DEFINITION 1.12.1. *The* **NAND Operator***, which has symbol* $|$ *("Sheffer Stroke"), is defined by the truth table*

p	q	$p \vert q$
T	T	F
T	F	T
F	T	T
F	F	T

DEFINITION 1.12.2. *The* **NOR Operator**, *which has symbol* $\downarrow$ *("Peirce Arrow"), is defined by the truth table*

p	q	$p \downarrow q$
T	T	F
T	F	F
F	T	F
F	F	T

Discussion

These two additional operators are very useful as logical gates in a combinatorial circuit, a topic we will discuss later.

1.13. Example.

EXAMPLE 1.13.1. *Write the following statement symbolically, and then make a truth table for the statement. "If I go to the mall or go to the movies, then I will not go to the gym."*

Solution. *Suppose we set*

- $p =$ *I go to the mall*
- $q =$ *I go to the movies*
- $r =$ *I will go to the gym*

The proposition can then be expressed as "If p *or* q*, then not* r*," or* $(p \vee q) \to \neg r$.

p	q	r	$(p \vee q)$	$\neg r$	$(p \vee q) \to \neg r$
T	T	T	T	F	F
T	T	F	T	T	T
T	F	T	T	F	F
T	F	F	T	T	T
F	T	T	T	F	F
F	T	F	T	T	T
F	F	T	F	F	T
F	F	F	F	T	T

Discussion

When building a truth table for a compound proposition, you need a row for every possible combination of T's and F's for the component propositions. Notice if there is only one proposition involved, there are 2 rows. If there are two propositions, there are 4 rows, if there are 3 propositions there are 8 rows.

EXERCISE 1.13.1. *How many rows should a truth table have for a statement involving n different propositions?*

It is not always so clear cut how many columns one needs. If we have only three propositions p, q, and r, you would, in theory, only need four columns: one for each of p, q, and r, and one for the compound proposition under discussion, which is $(p \vee q) \to \neg r$ in this example. In practice, however, you will probably want to have a column for each of the successive intermediate propositions used to build the final one. In this example it is convenient to have a column for $p \vee q$ and a column for $\neg r$, so that the truth value in each row in the column for $(p \vee q) \to \neg r$ is easily supplied from the truth values for $p \vee q$ and $\neg r$ in that row.

Another reason why you should show the intermediate columns in your truth table is for grading purposes. If you make an error in a truth table and do not give this extra information, it will be difficult to evaluate your error and give you partial credit.

EXAMPLE 1.13.2. *Suppose p is the proposition "the apple is delicious" and q is the proposition "I ate the apple." Notice the difference between the two statements below.*

(a) $\neg p \wedge q$ = The apple is not delicious, and I ate the apple.
(b) $\neg(p \wedge q)$ = It is not the case that: the apple is delicious and I ate the apple.

EXERCISE 1.13.2. *Find another way to express Example 1.13.2 Part b without using the phrase "It is not the case."*

EXAMPLE 1.13.3. *Express the proposition "If you work hard and do not get distracted, then you can finish the job" symbolically as a compound proposition in terms of simple propositions and logical operators.*

Set

- p = you work hard
- q = you get distracted
- r = you can finish the job

In terms of p, q, and r, the given proposition can be written

$$(p \wedge \neg q) \to r.$$

The comma in Example 1.13.3 is not necessary to distinguish the order of the operators, but consider the sentence "If the fish is cooked then dinner is ready and I am hungry." Should this sentence be interpreted as $f \to (r \wedge h)$ or $(f \to r) \wedge h$, where f, r, and h are the natural choices for the simple propositions? A comma needs to be inserted in this sentence to make the meaning clear or rearranging the sentence could make the meaning clear.

EXERCISE 1.13.3. *Insert a comma into the sentence "If the fish is cooked then dinner is ready and I am hungry." to make the sentence mean*

(a) $f \to (r \wedge h)$
(b) $(f \to r) \wedge h$

EXAMPLE 1.13.4. *Here we build a truth table for* $p \to (q \to r)$ *and* $(p \wedge q) \to r$. *When creating a table for more than one proposition, we may simply add the necessary columns to a single truth table.*

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

EXERCISE 1.13.4. *Build one truth table for* $f \to (r \wedge h)$ *and* $(f \to r) \wedge h$.

1.14. Bit Strings.

DEFINITION 1.14.1. *A* **bit** *is a 0 or a 1 and a* **bit string** *is a list or string of bits.*

The logical operators can be turned into **bit operators** by thinking of 0 as false and 1 as true. The obvious substitutions then give the table

$\overline{0} = 1$	$\overline{1} = 0$	
$0 \vee 0 = 0$	$0 \wedge 0 = 0$	$0 \oplus 0 = 0$
$0 \vee 1 = 1$	$0 \wedge 1 = 0$	$0 \oplus 1 = 1$
$1 \vee 0 = 1$	$1 \wedge 0 = 0$	$1 \oplus 0 = 1$
$1 \vee 1 = 1$	$1 \wedge 1 = 1$	$1 \oplus 1 = 0$

Discussion

We can define the *bitwise NEGATION* of a string and *bitwise OR, bitwise AND,* and *bitwise XOR* of two bit strings of the same length by applying the logical operators to the corresponding bits in the natural way.

EXAMPLE 1.14.1.

(a) $\overline{11010} = 00101$

(b) $11010 \vee 10001 = 11011$

(c) $11010 \wedge 10001 = 10000$

(d) $11010 \oplus 10001 = 01011$