

Discrete Mathematics: Chapter 1

Steven Schmatz, University of Michigan: College of Engineering

September 1, 2014

stevenschmatz@gmail.com

Chapter 1: The Foundations: Logic and Proofs

- A *proof* is a correct mathematical argument.
- A *theorem* is a mathematical statement that is proven to be true.

1.1: Propositional Logic

Propositions

Logic gives us rules to distinguish between valid and invalid mathematical arguments. A *proposition* is a declarative sentence that is either true or false, but not both. *Propositional variables* are variables that represent propositions, such as p, q, r, s, \dots , just as letters are used to denote numerical variables.

- $\neg p$ represents the *negation of p* .
- $p \wedge q$ represents the *logical conjunction of p and q* , essentially an **and** statement in traditional programming languages.
- $p \vee q$ represents the *logical disjunction of p and q* , essentially an **or** statement in traditional programming languages.

The *or* statement can be in two forms: the *inclusive or*, or the *exclusive or*. The inclusive or is a disjunction when **at least one** of the propositions is true. The exclusive or, $p \oplus q$, is a disjunction which is true only when one of the two propositions is true.

Conditional statements

Let p and q be propositions. p is known as the *hypothesis* (or *antecedent* or *premise*), while q is known as the *conclusion* (or *consequence*).

The *conditional statement* (also known as an implication) $p \rightarrow q$ is the proposition: if p , then q . This statement evaluates to true in all cases except when p is true and q is false.

If you think of p to be a strict subset of q , then whatever is evaluated true in p should also be true in q . Hence, if p is evaluated true, so should q be evaluated true. If q is *not* true, then this statement is impossible and $p \rightarrow q$ is evaluated false.

Converse, contrapositive, and inverse

- The *converse* of a proposition $p \rightarrow q$ is $q \rightarrow p$. The converse never has the same truth value as the proposition for all possible truth values of p and q .

- The *contrapositive* of a proposition $p \rightarrow q$ is $\neg q \rightarrow \neg p$. A contrapositive statement **always** has the same truth statement as the proposition.
- The *inverse* of a proposition $p \rightarrow q$ is $\neg p \rightarrow \neg q$. The inverse never has the same truth value as the proposition for all possible truth values of p and q .

Two compound statements are called *equivalent* when they have all the same truth values for all values of p and q .

Propositional logic is different from if-then statements both in the English language and in computer programming:

- In English, the statement ‘If Juan gets a smartphone, then $2 + 3 = 6$ ’ is very obviously false. However, in propositional logic, this conditional statement would be completely valid.
- In computer programming, if-then statements take the form of **if p then S** , where p is a propositional statement and S is a segment of code.

Mathematical propositions are much more general than in either of these two cases.

Biconditionals

Biconditional statements, $p \leftrightarrow q$, also known as “ p if and only if q ”, are only true when both p and q are of the same truth value. This can also be expressed as the following: the statement is only true when $p \rightarrow q$ and $q \rightarrow p$.

Compound statements

With these four logical connectives – conjunctions, disjunctions, conditional statements, and biconditional statements – one can create complex logical statements, such as the one below:

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

To correctly interpret these statements, one needs the order of precedence.

1. The negation operator \neg .
2. The conjunction operator \wedge .
3. The disjunction operator \vee .
4. The implication operator \rightarrow .
5. The biconditional operator \leftrightarrow .

Logic and binary operations

A *bit*, or “binary digit”, is a value which may either be 0 or 1. A variable that stores a bit is known as a *boolean variable*. All of the logical operators can be applied to Boolean variables.

1.2: Applications of propositional logic

Logic has many important applications to sciences and mathematics:

- Design of artificial intelligence
- Design of electrical circuits and programming languages
- Verifying correctness of programs
- Automated theorem proving

Translating English sentences

English is an ambiguous language. To resolve these ambiguities, we can translate English into logic (using assumptions).

”You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.”

$$(r \wedge s) \rightarrow \neg q$$

System specifications

In the translation from natural language to logic and vice versa, it is important that there is no contradiction evident in multiple statements. For example,

”The diagnostic message is stored in the buffer or it is retransmitted.” “The diagnostic message is not stored in the buffer.” “If the diagnostic message is stored in the buffer, then it is retransmitted.”

We can call p the situation where “the diagnostic message is stored in the buffer” is true, and q where the situation “the diagnostic message is retransmitted” is true. From this, we can infer that $p \vee q, \neg p, p \rightarrow q$.

Also logic puzzles and logic circuits.

1.3: Propositional Equivalences

Often, in mathematical arguments, mathematicians replace compound propositions with other compound propositions with equivalent truth values. A *tautology* is a compound proposition that is always true, no matter what the truth values of the propositional variables that compose the statement. On the other hand, a *contradiction* is a statement that is unconditionally false. A compound proposition that is neither a tautology nor a contradiction is a *contingency*.

Logical equivalence

If two statements have the same truth value in all statements, they are said to be *logically equivalent*. This is only true when $p \leftrightarrow q = \top$. The notation $p \equiv q$ is used to represent logical equivalence.

The *De Morgan laws* state:

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

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

There are many laws governing compound propositions. Another law is the associative law – with logical conjunctions or disjunctions, there is no distinction between ordering operations.