CS 0441
Lecture 1: Mathematical Logic

KP. Wang

Course information

Instructor Kunpeng Wang

Course section CS 0441

Email kunpeng.wang@scupi.cn

Textbook Discrete Mathematics and Its Applications (7th Edition), by Kenneth H. Rosen.

Overview

1. Sections 1.1 & 1.2
2. Mathematical logic
3. Propositional logic
4. Logic operators
   - Negation
   - Conjunction
   - Disjunction
   - Exclusive or
   - Implication

Overview

Mathematical logic

- Mathematical logic is the discipline that mathematicians invented in the late nineteenth and early twentieth centuries so they could stop talking nonsense. It's the most powerful tool we have for reasoning about things that we can't really comprehend, which makes it a perfect tool for Computer Science.

Propositional logic

- Let us start with an introduction to the basic building blocks of logic-**propositions**. 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

Example

Which of the following sentences are propositions?

1. Beijing is the capital of China.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.
5. What time is it?
6. Read this carefully.
7. $x + 1 = 2$.
8. $x + y = z$.

Note sentences 1-4 are propositions while sentences 5-8 are not since we cannot determine they are true or false. Also, propositions 1 and 3 are true, whereas 2 and 4 are false.

Propositional variables

- We use letters to denote propositional variables (or statement variables), that is, variables that represent propositions. The conventional letters used for propositional variables are $p, q, r, s, \ldots$.
- If a proposition is true, we can denote it by T. Otherwise, we denote the proposition by F. Note T and F are called **truth values**.
- The study of propositions is **propositional logic** or **propositional calculus**, which was mostly invented in ancient Greece.

Compound propositions

- However, we cannot deduce much from a proposition. Hence, we now introduce how to produce new propositions from the original ones.
- Following the mathematician George Boole, we generate **compound propositions** formed from existing propositions using **logical operators**.

Logic operators

- The following are basic logic operators: **negation** ($\neg$), **conjunction** ($\wedge$), **disjunction** ($\vee$), **exclusive or** ($\oplus$) and **implication** ($\rightarrow$).

Negation

Definition

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

Remark

- The essence of negation is to find the opposite of the predicate in a proposition.

Example 2

Example

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

Solution

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

The truth values after negation

- The negation will alter the truth value of the original proposition. We will display using the following table, called **Truth Table**.

p	$\neg p$
T	F
F	T

Table: The Truth Table for the Negation of a Proposition.

Conjunction

Definition

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p and q" . The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

- Note that in logic the word "but" sometimes is used instead of "and" in a conjunction. For example, the statement "The sun is shining, but it is raining" is another way of saying "The sun is shining and it is raining."
- In natural language, there is a subtle difference in meaning between "and" and "but"; we will not be concerned with this nuance here.

The truth values after conjunction

- We list the truth values for conjunction in the following table.

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

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

Example 3

Example

Find the conjunction of the propositions p and q where p is the proposition "Rebecca's PC has more than 16 GB free hard disk space" and q is the proposition "The processor in Rebecca's PC runs faster than 1 GHz."

Solution

The conjunction of these propositions, $p \wedge q$, is the proposition "Rebecca's PC has more than 16 GB free hard disk space, and the processor in Rebecca's PC runs faster than 1 GHz." This conjunction can be expressed more simply as "Rebecca's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz." For this conjunction to be true, both conditions given must be true. It is false, when one or both of these conditions are false.

Disjunction

Definition

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

Remark

- The conjunction and disjunction are called **connectives**.
- The use of the connective or in a disjunction corresponds to one of the two ways the word or is used in English, namely, as an inclusive or. A disjunction is true when at least one of the two propositions is true.

The truth values after disjunction

- The following table displays the truth table for $p \vee q$.

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

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

Example 4

Example

What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 3?

Solution

The disjunction of p and q, $p \vee q$, *is the proposition "Rebecca's PC has at least 16 GB free hard disk space, or the processor in Rebecca's PC runs faster than* $1GHz$*."*

Exclusive or

- The above disjunction refers to the inclusive or, namely, $p \vee q$ includes p, q as well as both p and q. Next we will introduce the other type of the connective **or**, that is, the **exclusive or**.
- The exclusive or excludes the both scenario. When the exclusive or is used, we mean that it is either p or q, but not both.

Formally, we can define it as

Definition

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.

- The truth values of the exclusive or is given in the following

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

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

Example 5

Example

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.

Remark

- In practice, we will also use the notation OR, AND, and XOR for the operators $\vee$, $\wedge$, and $\oplus$, as is done in various programming languages.

Implication

Definition

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. In the conditional statement $p \to q, p$ is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

- Normally, we call the conditional statement **implication**.

- The following ways can be used to express the implication in natural language:

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

- The truth table for the conditional statement $p \to q$ is shown as follows.

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

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

Example 6

Example

Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job." Express the statement $p \to q$ as a statement in English.

Solution

"If Maria 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:
"Maria will find a good job when she learns discrete mathematics."
"For Maria to get a good job, it is sufficient for her to learn discrete mathematics."
and
"Maria will find a good job unless she does not learn discrete mathematics."

Example 7

Example

What is the value of the variable x after the statement
if $2+2=4$ then $x := x+1$
if $x=0$ before this statement is encountered?
(The symbol $:=$ stands for assignment. The statement $x := x+1$ means the assignment of the value of $x+1$ to x.)

Solution

Because $2 + 2 = 4$ *is true, the assignment statement* $x := x + 1$ *is executed. Hence,* x *has the value* $0 + 1 = 1$ *after this statement is encountered.*

Converse, Inverse and Contrapositive

- The implication can be further extended via finding its **converse**, **inverse** and **contrapositive** .

Converse

- The converse of $p \to q$ is $q \to p$, *e.g.*, the converse of "If I am human then I am a mammal" is "If I am a mammal then I am human."

Inverse

- The inverse of $p \to q$ is $\neg p \to \neg q$. So the inverse of "If you take CS 0441, you will surely die" is "If you do not take CS 0441, you will not surely die."
- There is often no connection between the truth of an implication and the truth of its inverse: "If I am human then I am a mammal" does not have the same truth-value as "If I am not human then I am not a mammal," barring some over-the-top ecological disaster.

Contrapositive

- The contrapositive of $p \to q$ is $\neg q \to \neg p$; it is logically equivalent to the original implication. For example, the contrapositive of "If I am human then I am a mammal" is "If I am not a mammal then I am not human".

Truth values of the extended implication

- Using the truth tables of negation and implication, we can show that neither the converse, $q \rightarrow p$, nor the inverse, $\neg p \rightarrow \neg q$, has the same truth value as $p \rightarrow q$ for all possible truth values of p and q.

- Similarly, we can see that $p \to q$ always has **the same truth value** as $\neg q \to \neg p$.

Equivalent logic

- When two compound propositions always have the same truth value we call them **equivalent**, so that a conditional statement and its contrapositive are equivalent.
- The converse and the inverse of a conditional statement are also equivalent.

Example 8

Example

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 $p \rightarrow q$, 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.

Biconditionals

- In the last we introduce the combination of expressions that expresses two propositions have the same truth value.

Definition

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition " p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

Truth values of the biconditional

- The truth value of the biconditional is easy to see in the following

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

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

- To express the biconditional statement $p \leftrightarrow q$, we can use the abbreviation "iff" for "if and only if."
- Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \to q) \wedge (q \to p)$.

Example 9

Example

Let p be the statement "You can take the flight," and let q be the statement "You buy a ticket." Then $p \leftrightarrow q$ is the statement "You can take the flight if and only if you buy a ticket."

Truth tables of compound propositions

- We will summarize the topics via discussing the truth values of the compound propositions.
- Similar with numerical operations, for example, addition, subtraction, multiplication and division, it is natural to discuss the precedence of the logic operations in compound forms.

- The following table displays the precedence levels of the logic operators.

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

Table: Precedence of Logical Operators.

- For example, $(\neg p \vee q \wedge r \rightarrow s \leftrightarrow t)$ is interpreted as $((((\neg p) \vee (q \wedge r)) \rightarrow s) \leftrightarrow t)$.
- Both OR and AND are associative, so $(p \vee q \vee r)$ is the same as so $(p \vee q \vee r)$ is the same as $((p \vee q) \vee r)$ and as $(p \vee (q \vee r))$, and similarly $(p \wedge q \wedge r)$ is the same as $((p \wedge q) \wedge r)$ and as $(p \wedge (q \wedge r))$.

Remark

- Note that this convention is not universal: many mathematicians give AND and OR equal precedence, so that the meaning of $p \wedge q \vee r$ is ambiguous without parentheses. There are good arguments for either convention.
- There does not seem to be a standard convention for the precedence of XOR, since logicians don't use it much. There are plausible arguments for putting XOR in between AND and OR, but it's probably safest just to use parentheses.

Remark

- Implication is not associative, although the convention is that it binds "to the right," so that $a \to b \to c$ is read as $a \to (b \to c)$. Except for type theorists and Haskell programmers, few people ever remember this, so it is usually safest to put in the parentheses.

Example 10

Example

Construct the truth table of the compound proposition

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

Solution

We compute the truth values in what follows.

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

Translating English sentences

- Now let us use the following examples to see how to translate English in propositional logic.

Example 11

Example

Translate the following sentence into a logical expression.
"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

Solution

Note we are given a compound propositions. We first will denote each proposition by a certain variable. In particular, we let a, c*, and* f *represent "You can access the Internet from campus," "You are a computer science major," and "You are a freshman," respectively. Noting that "only if" is one way a conditional statement can be expressed, this sentence can be represented as*

$$a \to (c \vee \neg f).$$

Example 12

Example

Rewrite the following sentence into a logical expression.
"You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."

Solution

Similarly, let q, r, and s represent "You can ride the roller coaster," "You are under 4 feet tall," and "You are older than 16 years old," respectively. Then the sentence can be translated to

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

System specifications

- Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems.
- System specifications should be consistent, that is, they should not contain conflicting requirements that could be used to derive a contradiction. When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

Example 13

Example

Determine whether these system specifications are consistent:
"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."

Solution

To determine whether these specifications are consistent, we first express them using logical expressions. Let p denote "The diagnostic message is stored in the buffer" and let q denote "The diagnostic message is retransmitted." The specifications can then be written as $p \vee q, \neg p$, and $p \to q$. An assignment of truth values that makes all three specifications true must have p false to make $\neg p$ true. Because we want $p \vee q$ to be true but p must be false, q must be true. Because $p \to q$ is true when p is false and q is true, we conclude that these specifications are consistent, because they are all true when p is false and q is true.
Also, we can test the specification via the truth table by examining all the truth values.

Logic and bit operations

- The binary truth values T and F can be naturally linked to the computer information **bits**.
- A bit is a symbol with two possible values, namely, 0 (zero) and 1 (one). A bit can be used to represent a truth value, because there are two truth values, namely, true and false. As is customarily done, we will use a 1 bit to represent true and a 0 bit to represent false. That is, 1 represents T (true), 0 represents F (false).

- Using the numerical 0 and 1 and replacing the T and F values in the truth table, we can represent the truth tables as follows.

p	$\neg p$
0	1
1	0

- The rest of logical operations is

p	q	$p \vee q$	$p \oplus q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	0	0	0	1	1
0	1	1	1	0	1	0
1	0	1	1	0	0	0
1	1	1	0	1	1	1

Bit strings

- With more bits in a sequence, we arrive at a bit string.

Definition

A bit string is a sequence of zero or more bits. The **length** of this string is the number of bits in the string.

- For example, 101010011 is a bit string of length nine.

- We can extend bit operations to bit strings.
- We define the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of two strings of the same length to be the strings that have as their bits the *OR*, *AND*, and *XOR* of the corresponding bits in the two strings, respectively.
- Note the same symbols $\vee, \wedge$, and $\oplus$ can be used to represent the bitwise *OR*, bitwise *AND*, and bitwise *XOR* operations, respectively.

Example 14

Example

Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit strings 0110110110 and 1100011101.

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 gives us

01	*1011*	*0110*	
11	*0001*	*1101*	
11	*1011*	*1111*	*bitwise OR*
01	*0001*	*0100*	*bitwise AND*
10	*1010*	*1011*	*bitwise XOR*