Typical problems

- How many ways are there to choose a valid password on a computer system?
- What is the probability of winning a lottery?
- How can one encrypt a message so that no unintended recipient can read it?
- What is the shortest path between two cities using a transportation system?
- How can a list of integers be sorted so that the integers are in increasing order? How many steps are required to do such a sorting? How do we know that our procedure always works?
- How can a circuit that adds two integers be designed?
- How many valid Internet addresses are there?

In other words, most problems are concerned with *counting*, relations between finite, or countable, collections of objects, analyzing processes which take finitely many steps or have finitely many outcomes.

Propositions

Definition

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

Example

- "Washington, DC is the capital of the United States", "London is the capital of Australia", "My iPad has 64GB of internal storage", "$2 + 2 = 4$", "$3 \times 5 = 17$" are propositions. The first and the fourth ones are true, the second and the last one are false. The 3rd one is either true or false, depending on a particular iPad.
- "What time is it now?" and "Answer this question" are not propositions since these are not declarative sentences. "$x + y = z$" is not a proposition because it can be both true (for example, if $x = 1$, $y = 2$, $z = 3$) and false (if $x = 3$, $y = 2$, $z = 1$).

If a proposition is true, its *truth value* is True (denoted T). If a proposition is false, its *truth value* is False (denoted F).

Example

The truth value of the proposition $2^2 = 4$ is True, while the truth value of the proposition $3^2 = 5$ is False.

Propositions are also called *logical variables*. We will denote them mostly by letters p and q, sometimes with indices (p_1, p_2, $\ldots$).

A logical variable can take only *two* values: T or F (True or False).

Knight-and-Knave Puzzles

R. Smullyan created many logical puzzles about an island that has two kinds of inhabitants: *knights*, who always tell the truth, and *knaves*, who always lie.

Thus, if we know the type of a person stating a proposition, we know the truth value of that proposition.

Also, if we know the truth value of a proposition, we know the type of the person who stated it.

Example

You met two inhabitants of the island, A and B. What are they if A says "B is a knight" and B says "The two of us are of opposite types"?

Let a and b be the propositions stated by A and B, respectively.

- If $a = T$ then A is a knight (because A is telling the truth) and also B is a knight (because $a = T$). But then $b = F$, as both of them are of the same type. On the other hand, since B is the knight, we must have $b = T$. This is impossible since b is a proposition and so cannot be true and false at the same time.
- If $a = F$ then B is a knave and so $b = F$. This means that A and B are of the same kind, that is, are both knaves. This of course agrees with $a = F$. Thus, A and B are both knaves is the only solution.

In the following example we introduce the 3rd kind, the *normals*, who can either tell the truth or lie. This means that if a normal makes a proposition we do not know its truth value.

Example

Three inhabitants (A, B and C) of the island met. No two of them are of the same type.
A looked at B and said: "B is a normal".
B replied: "C is not a knave".
And C declared: "A never lies".
What are A, B and C?

Let a (respectively b, c) be the proposition stated by A (respectively, B, C).

- Suppose that $a = T$. Then B is a normal (because $a = T$) and A is a knight (because $a = T$ and we already know who is the normal). This forces C to be the knave. Then $b = F$ (which is possible, as a normal can lie), but $c = T$ which is impossible as C is the knave. Thus, $a = T$ cannot occur.
- Suppose that $a = F$. Then $c = F$ (A is not telling the truth), and so neither A nor C is a knight. This forces B to be the knight. Then $b = T$ and so C is a normal (which agrees with $c = F$) and A is a knave (which agrees with $a = F$). Thus, we found the only solution: A is the knave, B is the knight and C is the normal.

Negation

Definition

Let p be a proposition. The *negation* of p (denoted $\neg p$) is the proposition *it is not the case that* p

Example

- Let p be the proposition “Sacramento is the largest city in California”. Then $\neg p$ is the proposition “It is not the case that Sacramento is the largest city in California” or simply “Sacramento is not the largest city of California”. The truth value of p is F and the truth value of $\neg p$ is T.
- Let p be the proposition $2 + 2 = 4$. Then $\neg p$ is “It is not the case that $2 + 2 = 4$” or “$2 + 2 \neq 4$”. Clearly, the truth value of p is T and the truth value of $\neg p$ is F.

Example

Let p be the proposition "Jane uses the latest version of Skype". Then $\neg p$ is "Jane does not use the latest version of Skype". Note that the truth value of $\neg p$ is T if Jane uses an older version of Skype, but also if Jane does not use Skype at all. So, it would be a mistake to say that $\neg p$ is "Jane uses an obsolete version of Skype".

Negation is our first example of *logical operators*. A logical operator can be determined by its *truth table*. For the negation, the truth table is

p	$\neg p$
T	F
F	T

Logical *AND* or Conjunction

Definition

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

Example

- Let p be "John speaks English" and q be "John speaks Spanish". Then $p \wedge q$ is "John speaks English and Spanish". It's truth value is T if John can speak both languages and is F otherwise.
- Let p be "Marsha's computer has 16GB of RAM" and q be "Marsha's computer has more than 10GB free hard drive space". Then $p \wedge q$ is "Marsha's computer has 16GB of RAM and more than 10GB free hard drive space".

In computer languages conjunction is usually denoted by *AND*, & or &&.

Example

- Let p be "Whales are mammals" and let q be "Pelicans are mammals". Then $p \wedge q$ is "Whales and pelicans are mammals" and its truth value is F.

The truth table for the conjunction of two propositions is

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

OR and *XOR*

There are two kinds of “or” we can use.

Example

- The proposition “EE114 or STAT155 are prerequisites for CS 172” means that a student who took one of these classes (EE114, STAT155) or both, can enroll into CS 172. This is an example of *inclusive OR*
- The proposition “Credit is awarded for only one of CS 229 or CS 229V” is an example of *exclusive OR*

Definition

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

Example

- Let p be "A Dell PC can run Windows" and q be "A Dell PC can run Linux". Then $p \vee q$ is "A Dell PC can run Windows or Linux" and its truth value is T. If we replace Linux by iOS then $p \vee q$ becomes "A Dell PC can run Windows or iOS" and its truth value is still T even though the truth value of q is now F.
- Let p be "Whales are mammals" and q be "Whales are fish". Then $p \vee q$ is "Whales are mammals or fish". The truth value of this statement is T, even though one of propositions p, q is manifestly false, and, moreover, only one of p and q can be true.

In computer languages disjunction is usually denoted by OR, $|$ or $||$.

Definition

Let p and q be propositions. Then the *exclusive or* (xor) of p and q, denoted $p \oplus q$, is the proposition which is true if exactly one of p and q is true.

The truth tables for $\vee$ and $\oplus$ are shown below

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

Conditional statements

Definition

Let p and q be propositions. A conditional statement, denoted $p \to q$, is the proposition "if p then q". p is called the hypothesis, or premise, and q is called the conclusion. $p \to q$ is false if p is true and q is false and is true otherwise.

Other forms of conditional statements: "if p, q", "p implies q", "q follows from p", "a sufficient condition of q is p", "a necessary condition for p is q", "p only if q", "q when p" and - perhaps the most confusing - "q unless $\neg p$", etc

The truth table for $p \to q$

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

Example

- A standard election pledge is "If I am elected, I will lower taxes". This is a conditional statement $p \to q$ where p is "I am elected" and q is "I will lower taxes". It is only false if the politician gets elected but does not lower taxes. Otherwise, from the point of view of formal logic, it is true.
- A syllabus says "In order to get an A, you need the score of at least 85%". This is a conditional statement $p \to q$ with p being "Your score is at least 85%" and q being "you get an A". This statement is true except when your score was 85 or higher but you did not get an A.
- Consider the proposition "We will go to the beach unless it rains". It can be reformulated as "If it does not rain then we will go to the beach". This is an example of a conditional statement in the form "q unless $\neg p$".

A conditional statement in proposition logic can look like "If Mary has a little lamb then $2+2=5$". This does not make much sense in everyday language, but it is a perfectly legitimate conditional statement - which is false if Mary does have a little lamb and is true otherwise.

In most computer languages the construction **if** x **then** y means something different: it means that y is executed only if the proposition x is true. For example, consider the code (in C)

```
int x=0;
if(x>0){x=x+1;}
```

The value of x will remain 0 since the assignment $x=x+1$ (that is, x is assigned its value plus one) is executed only if $x>0$ which was not the case.

Converse, contrapositive and inverse

Definition

Let p, q be propositions and consider the conditional statement $p \to q$.

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

Example

Consider the conditional statement “If you take two classes next quarter then you are able to graduate this year”.

- The *premise* p is “You take two classes next quarter” and the *conclusion* q is “You are able to graduate this year”.
- The *converse* $q \to p$ is “If you are able to graduate this year, you take two classes next quarter”,
- The *contrapositive* is “If you are not able to graduate this year, you do not take two classes next quarter”
- The *inverse* is “If you do not take two classes next quarter, you are not able to graduate this year”.

Let us compare truth tables of a conditional statement, its converse, contrapositive and inverse.

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

Note that $p \to q$ and its contrapositive have has the same truth table.

Note also that the converse and the inverse of $p \to q$ have the same truth table. This is because the inverse of $p \to q$ is the same as the contrapositive of the converse (exercise)

Definition

A *compound proposition* is a proposition obtained from one or more propositions by applying logic operators.

Definition

When two compound propositions have the same truth value for all truth values of logical variables they are called *equivalent*.

We saw that $p \to q$ is equivalent to its contrapositive and that the converse of $p \to q$ is equivalent to its inverse.

Biconditionals

Definition

Let p and q be propositions. The proposition "p if and only if q" is called the biconditional statement and is denoted by $p \leftrightarrow q$. It is true if p and q have the same truth value and false otherwise.

Example

"You can take the train if and only if you have a ticket" is a biconditional statement.
Another form of a biconditional statement is "p is necessary and sufficient for q", for instance "To be able to graduate it is necessary and sufficient to fulfill all requirements of the program".

We claim that $p \leftrightarrow q$ is equivalent to $(p \to q) \wedge (q \to p)$ and also to $\neg(p \oplus q)$. Indeed, let us compare their truth tables

p	q	$p \leftrightarrow q$	$p \oplus q$	$\neg(p \oplus q)$	$p \to q$	$q \to p$	$(p \to q) \wedge (q \to p)$
T	T	T	F	T	T	T	T
T	F	F	T	F	F	T	F
F	T	F	T	F	T	F	F
F	F	T	F	T	T	T	T

In other words, we need not introduce $\leftrightarrow$ as a separate operator, as it can be expressed in terms of other operators.

Later we will see that all logical operators can be expressed in terms of $\neg$ and $\wedge$ (or in terms of $\neg$ and $\vee$)

Truth tables of compound propositions

Example

Let p and q be propositions. Find the truth table for the proposition $(\neg p \wedge q) \to (p \vee \neg q)$.

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

We have

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

So, our complicated statement is equivalent to $q \to p$.

Precedence of logical operations

To avoid using too many parenthesis, the following precedence order is established:

$$
\begin{array}{c|c}
\neg & 1 \\
\hline
\wedge & 2 \\
\vee & 3 \\
\hline
\rightarrow & 4 \\
\leftrightarrow & 5
\end{array}
$$

So, proposition $\neg p \wedge q \vee \neg r$ is equivalent to $((\neg p) \wedge q) \vee (\neg r)$.

More examples

Example

Let p, q, r be the following propositions
p: "You have the flu" q: "You miss the final" r: "You pass the course"
Express the proposition $(p \to \neg r) \vee (q \to \neg r)$ as a sentence.

$p \to \neg r$ reads "If you have the flu, you do not pass the course", while $q \to \neg r$ reads "if you miss the final, you do not pass the course".

We can, of course, write "If you have the flu you do not pass the course OR if you miss the final you do not pass the course".

On the other hand, $(p \to \neg r) \vee (q \to \neg r)$ is false if only if $p \to \neg r$ is false and $q \to \neg r$ is false. This happens if and only if r is true and both p and q are true. So, the compound statement $(p \to \neg r) \vee (q \to \neg r)$ is equivalent to $\neg(p \wedge q \wedge r)$. This is not very convenient for our purposes (although correct). A better way is to observe that $(p \wedge q) \to \neg r$ is also equivalent to our statement. Indeed, it is false if and only if p, q and r are all true. So we can write "If you have the flu and miss the final then you do not pass the course".

Bit operations

Bit (**B**inary dig**it**) is a symbol which can take two values: 1 and 0. A bit string is a sequence of bits. The smallest unit of computer memory which can be addressed is called a *byte*, and it was standardized to mean 8 bits.

Logical operators can be applied to bit strings if we agree that T is represented by 1 and F is represented by 0.

Example

$$\begin{array}{rr} & 10110011 \\ \wedge & \\ & 01011010 \\ \hline & 00010010 \end{array} \qquad \begin{array}{rr} & 10110011 \\ \vee & \\ & 01011010 \\ \hline & 11111011 \end{array} \qquad \begin{array}{rr} & 10110011 \\ \oplus & \\ & 01011010 \\ \hline & 11101001 \end{array}$$

If we regard these bit strings as bytes representing numbers (non-negative integers) this reads $179 \wedge 90 = 18$, $179 \vee 90 = 251$, $179 \oplus 90 = 233$, or in hexadecimals $B3 \wedge 5A = 12$, $B3 \wedge 5A = FB$, $B3 \oplus 5A = E9$.