Logic of Compound Statements

Venkata Khandrika

Outline

Topic

Subtopic

1. Logical Form and Logical Equivalence
   - Overview
   - Propositions
   - Compound Propositions

Arguments, Form, and Conclusions

An **argument** is a sequence of statements aimed at proving the truth of an assertion. The assertion at the end of the sequence is called the **conclusion**, and the preceding statements are called **premises**.

In logic, the form of an argument is distinguished from its content. This means that logical analysis won't help you determine the truth of an argument's content, but it will help you analyze an argument's form to determine whether the truth of the conclusion follows necessarily from the truth of the premises.

Arguments, Form, and Conclusions

Consider the following two arguments.

Argument 1

If $\underbrace{\text{the bell rings}}_{p}$ or $\underbrace{\text{the flag drops}}_{q}$, then $\underbrace{\text{the race is over}}_{r}$.

$\therefore$ If $\underbrace{\text{the race is not over}}_{\text{not } r}$, then $\underbrace{\text{the bell hasn't rung}}_{\text{not } p}$ and $\underbrace{\text{the flag hasn't dropped}}_{\text{not } q}$.

Argument 2

If $\underbrace{x = 2}_{p}$ or $\underbrace{x = -2}_{q}$, then $\underbrace{x^2 = 4}_{r}$.

$\therefore$ If $\underbrace{x^2 \neq 4}_{\text{not } r}$, then $\underbrace{x \neq 2}_{\text{not } p}$ and $\underbrace{x \neq -2}_{\text{not } q}$.

Arguments, Form, and Conclusions

They have very different content but their logical form is the same. To help make this clear, we use letters like p, q, and r to represent component sentences. We let the expression "not p" refer to the sentence "It is not the case that p is true" or "p is false".

The two arguments have a common *logical form* of:

$$\begin{array}{l} \text{If } p \text{ or } q\text{, then } r. \\ \therefore \text{ If not } r\text{, then not } p \text{ and not } q. \end{array}$$

Subtopic

1. Logical Form and Logical Equivalence
   - Overview
   - Propositions
   - Compound Propositions

Proposition

Definition

A **proposition** is a sentence that is either true or false but not both.

Example

For example, "$2 + 2 = 4$" and "$2 + 2 = 5$" are both propositions, the first because it is true and the second because it is false.

Definition

A **theorem** is a proposition that can be shown to be true.

Example

For example, Pythagoras' theorem "$a^2 + b^2 = c^2$" can be shown to be true in a multitude of ways.

Propositions

Here are some examples of sentences that are *not* propositions:

(1) $x^2 + 2 = 11$
(2) $x + y > 0$

(1) is not a proposition because its truth or falsity depends on the value of x. For some values of x, it is true, whereas for other values it is false.

(2) is not a proposition because its truth or falsity depends on the values of x and y. For instance, when $x = -1$ and $y = 2$ it is true, whereas when $x = -1$ and $y = 1$ it is false.

Subtopic

1. Logical Form and Logical Equivalence
   - Overview
   - Propositions
   - Compound Propositions

Compound Propositions

We have seen that propositions are sentences that are either true or false but not both.

Naturally, we would next like to build more complex sentences. We now introduce three symbols that are used to build more complicated logical expressions out of simpler ones.

Negation

The symbol $\neg$ denotes "not". Given a proposition p, the sentence $\neg p$ is read "not p" or "it is not the case that p is true" or "p is false."

More formally, it called the **negation** of p. It can be written in many different ways including: $\sim p$, $\bar{p}$, $!p$, or even p'.

Disjunction

The symbol $\vee$ denotes "or". Given propositions p and q, the sentence $p \vee q$ is read "p or q."

More formally, $p \vee q$ is called the **disjunction** of p and q. This can also be written as: $p + q$, $p \mid q$, and $p \mid\mid q$.

Programming languages often use $p \mid\mid q$.

Conjunction

The symbol $\wedge$ denotes "and". Given propositions p and q, the sentence $p \wedge q$ is read "p and q."

More formally, $p \wedge q$ is called the **conjunction** of p and q. This can also be written as: $p \times q$, $p \ \&\ q$, $p \ \&\&\ q$.

Programming languages often use $p \ \&\&\ q$.

Topic

References

- Epp, Susanna S. (2019). Discrete Mathematics with Applications (5th ed.). Cengage Learning. ISBN: 9781337694193.