

# 1 Chapter 9: Deductive Reasoning

What distinguishes deductive reasoning from inductive reasoning is that in inductive arguments, if they are sound, the conclusion is merely supported by the premises, whereas in deductive reasoning, if the argument is sound, then the conclusion is *guaranteed*. Recall that the definition of soundness is that an argument is sound iff it has a high degree of logical strength and the premises are true, and recall the definition of logical strength is that if the premises are true, then they provide some degree of support for the conclusion. In the case of deductive arguments the degree of logical strength is the highest possible, the truth of the premises guarantees the truth of the conclusion. Arguments with this property are called *valid arguments*.

**valid argument:** An argument is valid iff the truth of the premises guarantees the truth of the conclusion.

Thus, valid arguments enable us to derive true statements from true statements.

An example of a deductive argument is the following:

She will order coffee or scotch with her lunch. She won't order scotch, so she will order coffee.

The truth of the first two premises guarantees the truth of the conclusion. You may notice, however, that there is nothing about the particular premises that makes the argument valid. Any argument of the same form

p or q  
not-p  
∴ q

will also be valid. This illustrates that validity is a property of the *form* of the argument, and not its *content*, *i.e.*, validity is independent of the content of the sentences making up the argument. This enables us to examine deductive arguments just by studying their purely formal properties, which we will come to later.

## 1.1 Truth-Functional Statements

We can divide the class of all sentences into two subclasses: simple and complex statements. Complex statements contain other statements as parts and simple statements contain no other statements as parts. we are interested here in the subclass of complex statements called *truth-functional statements*. Truth-functional statements are characterized by the fact that their truth value is determined by the truth values of their simple components.

The truth-functional statements are ones are composed of simple statements joined by *logical operators*: and; either/or; if ... then ...; it is not the case that. The logical operators, or logical connectives, are the following (using the convention that the latin letters *p*, *q*, *r*, ... represent simple sentences):

**Negation (not- $p$ ):** The statement that  $p$  is false is a negation. A negation is true when its component is false, and false when its component statement is true. For example:

Bill Clinton is not the president of the United States.

**Conjunction ( $p$  and  $q$ ):** The statement  $p$  and  $q$  is a conjunction. It is true only when both  $p$  and  $q$  are true and false if either or both of  $p$  or  $q$  are false. For example:

London gets very hot in the summer and very cold in the winter.

**Disjunction ( $p$  or  $q$ ):** The statement  $p$  or  $q$  is a disjunction. It is true when  $p$  is true, when  $q$  is true or when  $p$  and  $q$  are both true, but is false when both  $p$  and  $q$  are false. For example:

You may have soup, salad or both.

**Implication (if  $p$  then  $q$ ):** The statement *if  $p$  then  $q$*  is an implication. It is true when  $p$  and  $q$ , called the antecedent and consequent respectively, are true and false when  $p$  is true but  $q$  is false. It is also considered to be true when the antecedent is false. For example: If you look directly at the sun for too long, then you will go blind.

Note: Though the truth definition for the conditional seems odd and not to agree with the natural language usage of the conditional, it is a result of constructing a truth-functional definition of the conditional. This is the kind of conditional used in mathematical reasoning, which is the model for deductive reasoning in general and it is the only possible truth definition remaining that is independent of the other connectives. Thus, we will proceed with this truth definition for the conditional, but be aware that there is a debate concerning proper logical representation of the conditional from natural language.

Also Note: “because” is not a truth-functional operator. Why?

## 1.2 Formal Languages: Syntax

Given that the validity of deductive arguments derives solely from their form, tools that enable us to examine the formal properties of deductive arguments, abstracting from the particular content of a particular argument, will be useful. Formal logical languages are such a tool. Languages in general can be considered to be made up of two main components: semantics (or meaning) and syntax (or structural rules). We first examine how to form sentences in the language using the rules of syntax (or grammar in the case of natural languages) and then we interpret sentences by introducing a semantics. The semantics of formal languages, as we shall see, is defined in terms of the truth values of sentences. But before we examine this we must learn how to construct sentences in a formal language.

The sentences of our formal language are called *well formed formulas*. They are the sentences that can be constructed from simple sentences using the truth-functional connectives (the logical connectives): and; or; not; if ..., then .... In our formal language we will represent the logical connectives by symbols:

- ‘and’ is represented by  $\&$
- ‘or’ is represented by  $\vee$

- ‘not’ is represented by  $\sim$
- ‘if ..., then ...’ is represented by  $\supset$
- ‘if and only if’ is represented by  $\equiv$

To give an example, the sentence

If you fertilize your lawn and you are not careful, then you will burn the grass or kill your plants.

is represented as

$$(F \& \sim C) \supset (B \vee K),$$

where F is ‘you fertilize your lawn’, C is ‘you are careful’, B is ‘you will burn the grass’ and K is ‘you will kill your plants’.

Now we will consider how to construct the sentences of our formal language (the well-formed formulas or wffs). The basic components of our formal language are a set of basic symbols for simple sentences (A, B, C, *etc.*) and the symbols for the logical connectives ( $\&$ ,  $\vee$ ,  $\sim$ ,  $\supset$ ,  $\equiv$ ). We also have a set of variables which represent sentences (wffs) of our language ( $p$ ,  $q$ ,  $r$ , *etc.*). With this we may construct a collection of well-formed formulas, which will constitute the formal language of propositional logic.

**Well-formed formulas (wffs)** The well-formed formulas are defined recursively as follows:

1. Any simple proposition name (A, B, C, ...) is a wff;
2. For any wff,  $p$ ,  $(\sim p)$  is also a wff;
3. For any wffs,  $p$  and  $q$ ,  $(p \& q)$  is a wff;
4. For any wffs,  $p$  and  $q$ ,  $(p \vee q)$  is a wff;
5. For any wffs,  $p$  and  $q$ ,  $(p \supset q)$  is a wff;
6. For any wffs,  $p$  and  $q$ ,  $(p \equiv q)$  is a wff.