

## syn.1 Introduction

pl:syn:int:  
sec

Propositional logic deals with **formulas** that are built from **propositional variables** using the propositional connectives  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ , and  $\leftrightarrow$ . Intuitively, a **propositional variable**  $p$  stands for a sentence or proposition that is true or false. Whenever the “truth value” of the **propositional variable in a formula** are determined, so is the truth value of any **formulas** formed from them using propositional connectives. We say that propositional logic is *truth functional*, because its semantics is given by functions of truth values. In particular, in propositional logic we leave out of consideration any further determination of truth and falsity, e.g., whether something is necessarily true rather than just contingently true, or whether something is known to be true, or whether something is true now rather than was true or will be true. We only consider two truth values true ( $\mathbb{T}$ ) and false ( $\mathbb{F}$ ), and so exclude from discussion the possibility that a statement may be neither true nor false, or only half true. We also concentrate only on connectives where the truth value of a **formula** built from them is completely determined by the truth values of its parts (and not, say, on its meaning). In particular, whether the truth value of conditionals in English is truth functional in this sense is contentious. The material conditional  $\rightarrow$  is; other logics deal with conditionals that are not truth functional.

In order to develop the theory and metatheory of truth-functional propositional logic, we must first define the syntax and semantics of its expressions. We will describe one way of constructing **formulas** from **propositional variables** using the connectives. Alternative definitions are possible. Other systems will chose different symbols, will select different sets of connectives as primitive, will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of **formulas** *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

Giving the meaning of expressions is the domain of semantics. The central concept in semantics for propositional logic is that of satisfaction in a **valuation**. A **valuation**  $\mathbf{v}$  assigns truth values  $\mathbb{T}$ ,  $\mathbb{F}$  to the **propositional variables**. Any **valuation** determines a truth value  $\bar{\mathbf{v}}(\varphi)$  for any **formula**  $\varphi$ . A **formula** is satisfied in a **valuation**  $\mathbf{v}$  iff  $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ —we write this as  $\mathbf{v} \models \varphi$ . This relation can also be defined by induction on the structure of  $\varphi$ , using the truth functions for the logical connectives to define, say, satisfaction of  $\varphi \wedge \psi$  in terms of satisfaction (or not) of  $\varphi$  and  $\psi$ .

On the basis of the satisfaction relation  $\mathbf{v} \models \varphi$  for sentences we can then define the basic semantic notions of tautology, entailment, and satisfiability. A **formula** is a tautology,  $\models \varphi$ , if every **valuation** satisfies it, i.e.,  $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$  for any  $\mathbf{v}$ . It is entailed by a set of **formulas**,  $I \models \varphi$ , if every **valuation** that satisfies all the **formulas** in  $I$  also satisfies  $\varphi$ . And a set of **formulas** is satisfiable if some **valuation** satisfies all **formulas** in it at the same time. Because **formulas**

are inductively defined, and satisfaction is in turn defined by induction on the structure of **formulas**, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

## **Photo Credits**

## **Bibliography**