

Definition 1. A **propositional logic language** L consists of the following data:

- (a) a nonempty set of symbols called **propositional variables** (e.g., P, Q, R, \dots),
- (b) a set of **logical connectives** (e.g., $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$).

Alternate names for propositional logic include the following: propositional calculus, sentential logic (operating on sentences), statement logic, and zeroth-order logic.

Remark 2. In practice, the propositional variables are either be upper or lower case letters. If we run out of letters, we can use numerical subscripts (e.g., P_1, P_2, \dots).

Remark 3. For historical reasons, the following five logical connectives are considered canonical in some sense:

Symbol	Formal Name	Informal Name	Alternate Symbols
\neg	negation	“not”	$\sim, !, ', -$
\wedge	conjunction	“and”	$\&, \&\&, \cdot$
\vee	disjunction	“or”	$, , +$
\rightarrow	implication	“implies” or “if-then”	\implies, \supset
\leftrightarrow	biconditional	“if and only if” or “iff”	\iff, \equiv

The alternate symbols may be found in older texts or for logic in other contexts. We will only use the symbols on the left of the above table to refer to the logical connectives proper. Any usage of a symbol on the right means something else. For example, “ \implies ” and “ \iff ” are used in their normal manner of mathematical English.

There are other logical connectives that are used in other contexts (e.g., XOR, NAND, NOR), but we will see later that they can be defined in terms of the five canonical connectives, so they are not strictly necessary. In fact, we don’t even need all five of the canonical connectives, as we can define some in terms of others. Some equivalencies are sketched below and will be treated more formally later.

$$\begin{aligned}
 P \wedge Q &\approx \neg(\neg P \vee \neg Q) & P \vee Q &\approx \neg(\neg P \wedge \neg Q) \\
 P \rightarrow Q &\approx \neg P \vee Q & P \leftrightarrow Q &\approx (P \rightarrow Q) \wedge (Q \rightarrow P)
 \end{aligned}$$

For our purposes, we will take either $\{\neg, \vee\}$ or $\{\neg, \rightarrow\}$ as the minimal set of logical connectives.

Definition 4. Let L be a language of propositional logic. A **formula** of L , also called a **well-formed formula** or **wff**, is defined inductively as follows:

- (a) Every propositional variable is a formula.
- (b) If φ is a formula, then $(\neg\varphi)$ is a formula.
- (c) If φ and ψ are formulas, then $(\varphi \star \psi)$ is a formula, where \star is any binary logical connective, e.g., \vee or \rightarrow .

Any sequence of symbols that is not obtained by one of the above rules is not a formula.

Remark 5. Implicitly, we are allowing the use of parentheses to disambiguate the order of operations. A proper treatment of this aspect of formal languages can be found elsewhere.

Remark 6. One way to characterize the set of formulas is as the smallest set of strings of symbols which contains all the propositional variables and is closed under finite applications of the logical connective operations. Let $\mathcal{V} = \mathcal{V}(L)$ be the set of propositional variables of L . Then the set $\mathcal{F} = \mathcal{F}(L)$ of formulas of L is the set generated by \mathcal{V} under the following **formula building operations**:

$$\begin{aligned}\varepsilon_{\neg}(\varphi) &:= (\neg\varphi), \\ \varepsilon_{\wedge}(\varphi, \psi) &:= (\varphi \wedge \psi), \\ \varepsilon_{\vee}(\varphi, \psi) &:= (\varphi \vee \psi), \\ \varepsilon_{\rightarrow}(\varphi, \psi) &:= (\varphi \rightarrow \psi), \\ \varepsilon_{\leftrightarrow}(\varphi, \psi) &:= (\varphi \leftrightarrow \psi).\end{aligned}$$

In other words, $\mathcal{F} = \overline{\mathcal{V}}$ is the closure of \mathcal{V} under these operations.

Definition 7. Let L be a language of propositional logic. A **truth valuation** on L is a map $v : \mathcal{V} \rightarrow \{T, F\}$ from the set of propositional variables to the set of truth values.

Remark 8. A truth valuation v can be extended to a map $\bar{v} : \mathcal{F} \rightarrow \{T, F\}$ by defining it recursively on formulas of L :

$$\begin{aligned}\bar{v}(\neg\varphi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = F, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \wedge \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = T \text{ and } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \vee \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = T \text{ or } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \rightarrow \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = F \text{ or } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \leftrightarrow \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = \bar{v}(\psi), \\ F & \text{otherwise.} \end{cases}\end{aligned}$$

Displayed below in a truth table format, where the rows correspond to the possible truth values of the relevant propositional variables:

φ	ψ	$\neg\varphi$	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T