Chapter 2

Propositional Logic

2.1 Syntax

Consider a finite, non-empty signature Σ of propositional variables, the "alphabet" of propositional logic. In addition to the alphabet "propositional connectives" are further building blocks composing the sentences (formulas) of the language and auxiliary symbols such as parentheses enable disambiguation.

Definition 2.1.1 (Propositional Formula)**.** The set PROP(Σ) of *propositional formulas* over a signature Σ is inductively defined by:

PROP(Σ)	Comment
$\bot$	connective $\bot$ denotes "false"
$\top$	connective $\top$ denotes "true"
P	for any propositional variable $P \in \Sigma$
$(\neg\phi)$	connective $\neg$ denotes "negation"
$(\phi \land \psi)$	connective $\land$ denotes "conjunction"
$(\phi \lor \psi)$	connective $\lor$ denotes "disjunction"
$(\phi \rightarrow \psi)$	connective $\rightarrow$ denotes "implication"
$(\phi \leftrightarrow \psi)$	connective $\leftrightarrow$ denotes "equivalence"

where $\phi, \psi \in$ PROP(Σ).

The above definition is an abbreviation for setting PROP(Σ) to be the language of a context free grammar PROP(Σ) $= L((N, T, P, S))$ (see Definition 1.3.9) where $N = \{\phi, \psi\}$, $T = \Sigma \cup \{(,)\} \cup \{\bot, \top, \neg, \land, \lor, \rightarrow, \leftrightarrow\}$ with rules $S \Rightarrow \bot \mid \top \mid (\phi \land \psi) \mid (\phi \lor \psi) \mid (\phi \leftrightarrow \psi)$ and $S \Rightarrow P$ for every $P \in \Sigma$.

As a notational convention we assume that $\neg$ binds strongest and we omit outermost parenthesis. So $\neg P \lor Q$ is actually a shorthand for $((\neg P) \lor Q)$. For all other logical connectives we will explicitly put parenthesis when needed. From the semantics we will see that $\land$ and $\lor$ are associative and commutative. Therefore instead of $((P \land Q) \land R)$ we simply write $P \land Q \land R$.

Definition 2.1.2 (Atom, Literal)**.** A propositional formula P is called an *atom*. It is also called a *(positive) literal* and its negation $\neg P$ is called a *(negative) literal*. If L is a literal, then $\neg L = P$ if $L = \neg P$ and $\neg L = \neg P$ if $L = P$. Literals are denoted by letters L, K. The literals P and $\neg P$ are called *complementary*.

Automated reasoning is very much formula manipulation. In order to precisely represent the manipulation of a formula, we introduce positions.

Definition 2.1.3 (Position)**.** A *position* is a word over $\mathbb{N}$. The set of positions of a formula ϕ is inductively defined by

$$\begin{array}{rcl} \mathrm{pos}(\phi) & := & \{\epsilon\} \text{ if } \phi \in \{\top, \bot\} \text{ or } \phi \in \Sigma \\ \mathrm{pos}(\neg\phi) & := & \{\epsilon\} \cup \{1p \mid p \in \mathrm{pos}(\phi)\} \\ \mathrm{pos}(\phi \circ \psi) & := & \{\epsilon\} \cup \{1p \mid p \in pos(\phi)\} \cup \{2p \mid p \in \mathrm{pos}(\psi)\} \end{array}$$

where $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

The prefix order $\leq$ on positions is defined by $p \leq q$ if there is some p' such that $pp' = q$. Note that the prefix order is partial, e.g., the positions 12 and 21 are not comparable, they are "parallel", see below. By $<$ we denote the strict part of $\leq$, i.e., $p < q$ if $p \leq q$ but not $q \leq p$. By $\parallel$ we denote incomparable positions, i.e., $p \parallel q$ if neither $p \leq q$, nor $q \leq p$. Then we say that p is *above* q if $p \leq q$, p is *strictly above* q if $p < q$, and p and q are *parallel* if $p \parallel q$.

The *size* of a formula ϕ is given by the cardinality of $\mathrm{pos}(\phi)$: $|\phi| := |\mathrm{pos}(\phi)|$. The *subformula* of ϕ at position $p \in \mathrm{pos}(\phi)$ is recursively defined by $\phi|_\epsilon := \phi$, $\neg\phi|_{1p} := \phi|_p$, and $(\phi_1 \circ \phi_2)|_{ip} := \phi_i|_p$ where $i \in \{1, 2\}$, $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Finally, the *replacement* of a subformula at position $p \in \mathrm{pos}(\phi)$ by a formula ψ is recursively defined by $\phi[\psi]_\epsilon := \psi$ and $(\phi_1 \circ \phi_2)[\psi]_{ip} := \phi_i[\psi]_p$ where $i \in \{1, 2\}$, $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Example 2.1.4. The set of positions for the formula $\phi = (A \wedge B) \rightarrow (A \vee B)$ is $\mathrm{pos}(\phi) = \{\epsilon, 1, 11, 12, 2, 21, 22\}$. The subformula at position 22 is B, $\phi|_{22} = B$ and replacing this formula by $A \leftrightarrow B$ results in $\phi[A \leftrightarrow B]_{22} = (A \wedge B) \rightarrow (A \vee (A \leftrightarrow B))$.

A further prerequisite for efficient formula manipulation is notion of the *polarity* of a subformula of ϕ at position p. The polarity considers the number of "negations" starting from ϕ at ϵ down to p. It is 1 for an even number along the path, -1 for an odd number and 0 if there is at least one equivalence connective along the path.

Definition 2.1.5 (Polarity)**.** The *polarity* of a subformula of ϕ at position p is inductively defined by

$$\begin{array}{rcl} \mathrm{pol}(\phi, \epsilon) & := & 1 \\ \mathrm{pol}(\neg\phi, 1p) & := & -\mathrm{pol}(\phi, p) \\ \mathrm{pol}(\phi_1 \circ \phi_2, ip) & := & \mathrm{pol}(\phi_i, p) \text{ if } \circ \in \{\wedge, \vee\} \\ \mathrm{pol}(\phi_1 \rightarrow \phi_2, 1p) & := & -\mathrm{pol}(\phi_1, p) \\ \mathrm{pol}(\phi_1 \rightarrow \phi_2, 2p) & := & \mathrm{pol}(\phi_2, p) \\ \mathrm{pol}(\phi_1 \leftrightarrow \phi_2, ip) & := & 0 \end{array}$$