Introduction to Automated Theorem Proving

Tudor Jebelean

October 15, 2009

Contents

Chapter 1

Propositional Logic

1.1 Syntax

Propositional logic is a mathematical model of reasoning with statements composed logically from elementary statements (or *propositions*). The only relevant characteristic of an elementary proposition (like "It rains.") is that it can be true or false. Therefore such a an elementary statement is denoted by a single symbol (*propositional variable*) about which we only know that it can be true or false.

Examples:

$$\underbrace{\text{"It rains."}}_{\text{A}} \qquad \underbrace{\text{"It is sunny."}}_{\text{B}}$$

Using the propositional variables as basic building blocks, we can compose arbitrary complex *formulae* by using the *logical connectives* ($\neg \land \lor \Longrightarrow \Leftrightarrow$ and possibly other).

Examples:

$$\underbrace{\text{"If it rains then it is not sunny.}}_{A \Longrightarrow \neg B}$$

The syntax of propositional logic consists in the definition of the set of all propositional logic formulae, or the language of propositional logic formulae, which will contain formulae like:

$$\begin{array}{c}\mathscr{L}\text{ :Language} \\ \text{with "words" like}\end{array} \begin{cases} A \land B \\ A \land \neg B \\ (\neg A \land B) \Leftrightarrow (A \Rightarrow B) \\ A \land \neg A \end{cases}$$

The language $\mathscr{L}$ is defined over a certain set Σ of symbols: the parantheses, the logical connectives, the logical constants, and an infinite set Θ of propositional variables.

$$\begin{array}{c}\text{Set of "symbols":} \\ \text{"alphabet"}\end{array} \quad \Sigma = \{\neg, \land, \lor, \Longrightarrow, \Leftrightarrow\} \cup \{\mathbb{T}, \mathbb{F}\} \cup \Theta$$

Note: Θ is the set of propositional variables. for instance this could be $\{A, B, C, P, Q, \ldots, A_1, A_2, \ldots\}$. This set Θ is infinite, but enumerable.

Inductive Definition

(0) $\mathbb{T}, \mathbb{F} \in \mathscr{L}$ or one can also write $\{\mathbb{T}, \mathbb{F}\} \subset \mathscr{L}$

(1) if $\underbrace{\vartheta \in \Theta}_{\text{"variable"}}$, then $\underbrace{\vartheta \in \mathscr{L}}_{\text{"word", logical formula}}$ (a propositional variable ϑ "is" also a logical formula)

(2) if $\varphi, \psi \in \mathscr{L}$, then $\underbrace{\underbrace{\neg\varphi}, \underbrace{\varphi\wedge\psi}, \underbrace{\varphi\vee\psi}, \underbrace{\varphi\Longrightarrow\psi}, \underbrace{\varphi\Leftrightarrow\psi} \in \mathscr{L}}_{\text{"are also words in the language"}}$

(3) These are all the formulae.

Note that Σ^* also has the properties **(0), (1)** and **(2)**. However, **[3]** means that $\mathscr{L}$ is the smallest set having these properties. This allows us to use the structural induction principle in order to prove properties of formulae.

Atoms and literals. Formulae consisting in a single propositional variable are called *"atoms"*. Formulae consisting in an atom or a negated atom are called *"literals"*.

Notation: Sometimes we will denote $\neg\varphi$ by $\overline{\varphi}$. Also, if L is a literal, we will denote by $\overline{L}$ the opposite of L (that is $\overline{A}$ if L is A, and A if L is $\overline{A}$).

Exercise: Formulate the grammar for the language of propositional logic.

$$P = \begin{cases} W \to \mathbb{T} \mid \mathbb{F} \mid A \mid B \mid C \\ W \to (\neg W) \mid (W \wedge W) \mid \ldots \end{cases}$$

$$\text{Grammar } G = (\underbrace{\Sigma}_{\text{"alphabet"}}, \underbrace{\Sigma_N}_{\text{"nonterminal symbols"}}, \underbrace{S}_{\text{"nonterminal start symbol"}}, \underbrace{P}_{\text{set of productions}})$$

$$\begin{aligned} \Sigma &= \{\mathbb{T}, \mathbb{F}\} \cup \Theta \cup \{(,), \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\} \\ \Sigma_N &= \{W\} \\ S &= \Sigma_N \end{aligned}$$

1.2 Semantics

Example: Intuitively, the meaning of "$A \wedge B$" is that "this is only true if both A and B are true".

$f_{A\wedge B}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

Table 1.1: Semantic value of $A \wedge B$.

The semantic value (or the meaning) of the formula $A \wedge B$ is the function $f_{A\wedge B} : \mathscr{I}_{\{A,B\}} \to \{\mathbb{T}, \mathbb{F}\}$, where $\mathscr{I}_{\{A,B\}} = \{I : \{A, B\} \to \{\mathbb{T}, \mathbb{F}\}\}$ is the set of all assignments of truth values to the variables A, B.

I is called an *"interpretation"* for the formula $A \wedge B$. $\mathscr{I}_{\{A,B\}}$ is the *"set of interpretations"* for the formula $A \wedge B$.

As syntax is defined as the set $\mathcal{L}$ of all correct formulae, the semantics is defined as the set $\mathcal{S}$ of all possible semantic values:

$$\mathcal{S} = \{\mathscr{I}_V | V \subseteq \Theta\}.$$

The "semantic evaluation function" associates each formula φ from $\mathcal{L}$ to its semantic value f_φ from $\mathcal{S}$. If we denote by Var(φ) the set of propositional variables occuring in φ, then:

$$f_\varphi : \mathcal{I}_{\mathrm{Var}(\varphi)} \to \{\mathbb{T}, F\}, \quad f_\varphi(I) = \langle\varphi\rangle_I,$$

where $\langle\varphi\rangle_I$ is the *"truth value of φ under the interpretation I"*.

$\langle\varphi\rangle_I$ (the truth evaluation of a formula φ under the interpretation I) is defined inductively on the structure of formulae:

$$\begin{array}{rcl}
\langle\mathbb{F}\rangle_I & = & \mathbb{F} \\
\langle\mathbb{T}\rangle_I & = & \mathbb{T} \\
\langle v\rangle_I & = & I(\vartheta), \quad \text{if } v \in \Theta \\
\\
\langle\neg\varphi\rangle_I & = & \mathscr{B}_\neg(\langle\varphi\rangle_I) \\
\langle\varphi \vee \psi\rangle_I & = & \mathscr{B}_\vee(\langle\varphi\rangle_I, \langle\psi\rangle_I) \\
\langle\varphi\wedge\psi\rangle_I & = & \mathscr{B}_\wedge(\langle\varphi\rangle_I, \langle\psi\rangle_I) \\
\dots & &
\end{array}$$

The functions $\mathscr{B}_\neg, \mathscr{B}_\vee, \mathscr{B}_\wedge, \dots$ (boolean evaluation functions) are defined explicitely by truth tables for each logical connective, and they can be seen as the semantic values of the logical connectives.

	$\mathscr{B}_\neg$
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

$\mathscr{B}_\wedge$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\mathscr{B}_\vee$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

$\mathscr{B}_\Rightarrow$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

$\mathscr{B}_\Leftrightarrow$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Table 1.2: The semantics of logical connectives

Example

$$\begin{array}{rcl}
\langle(A \wedge (A \Rightarrow B)) \Rightarrow B\rangle_{\mathcal{I}} & = & \mathscr{B}_\Rightarrow(\langle(A \wedge (A \Rightarrow B))\rangle_{\mathcal{I}}, \langle B\rangle_{\mathcal{I}}) \\
& = & \mathscr{B}_\Rightarrow(\mathscr{B}_\wedge(\langle A\rangle_{\mathcal{I}}, \langle(A \Rightarrow B)\rangle_{\mathcal{I}}), \langle B\rangle_{\mathcal{I}}) \\
& = & \mathscr{B}_\Rightarrow(\mathscr{B}_\wedge(\langle A\rangle_{\mathcal{I}}, \mathscr{B}_\Rightarrow(\langle A\rangle_{\mathcal{I}}, \langle B\rangle_{\mathcal{I}})), \langle B\rangle_{\mathcal{I}}) \\
& = & \mathscr{B}_\Rightarrow\left(\underbrace{\mathscr{B}_\wedge\left(\mathbb{T}, \underbrace{\mathscr{B}_\Rightarrow(\mathbb{T}, \mathbb{F})}_{\mathbb{F}}\right)}_{\mathbb{F}}, \mathbb{F}\right) \\
& = & \mathbb{T}
\end{array}$$