Computational Logic

Nico Mexis

October 18, 2019

Contents

§1: What is logic?

TODO Rest

§2: Propositional Logic

<u>Syntax</u>:

(a) Atomic formulas are propositions: $A_0, A_1, \dots$ or $A, B, \dots$

(b) A formula is obtained by repeatedly applying the following rules:

   (1) An atomic formula is a formula
   (2) Given a formula F, also $\neg F$ is a formula ("not F")
   (3) Given two formulas F, G, also $F \wedge G$ and $F \vee G$ are formulas

<u>Semantics</u>:

(a) The set of truth values is $\{0,1\}$, where 0 is $FALSE$ and 1 is $TRUE$

(b) Let M be a set of atomic formulas. A map $\alpha : M \to \{0,1\}$ is called a <u>truth assignment</u>

(c) Let $\hat{M}$ be the set of all formulas in which only propositions of M appear. Then we define $\hat{\alpha} : \hat{M} \to \{0,1\}$ recursively as follows:

   (1) If $A \in M$, then we let $\hat{\alpha}(A) = \alpha(A)$
   (2) If $\alpha(F)$ is defined, then we let $\hat{\alpha}(\neg F) = 1 - \hat{\alpha}(F)$
   (3) Given formulas F, G for which $\hat{\alpha}(F), \hat{\alpha}(G)$ have been defined, we let
   $\hat{\alpha}(F \wedge G) = \begin{cases} 1 & \text{if } \hat{\alpha}(F) = \hat{\alpha}(G) = 1 \\ 0 & \text{otherwise} \end{cases}$ and $\hat{\alpha}(F \vee G) = \begin{cases} 1 & \text{if } \hat{\alpha}(F) = 1 \text{ or } \hat{\alpha}(G) = 1 \text{ or both} \\ 0 & \text{if } \hat{\alpha}(F) = \hat{\alpha}(G) = 1 \end{cases}$

F "if" G	$\mathrel{\widehat{=}} G \Rightarrow F$
F "only if" G	$\mathrel{\widehat{=}} F \Rightarrow G$
F "if and only if" G	$\mathrel{\widehat{=}} F \Leftrightarrow G$

Let F be a (propositional logic) formula, M a set of propositions and $\alpha : M \to \{0,1\}$ a truth assignment.

(a) The formula F <u>fits</u> with α or α is <u>suitable</u> for F if in F only the propositions from M appear.

(b) If $\alpha(F) = 1$, then F is called a <u>model</u> for α. We write $\alpha \models F$.

(c) Given a set of formulas $\mathcal{F}$, we write $\alpha \models \mathcal{F}$ if $\alpha \models F$ for every $F \in \mathcal{F}$.

(d) We say that F is <u>satisfiable</u> if there exists a truth assignment α, which is suitable for F and if $\alpha(F) = 1$. Otherwise, we say that F is <u>unsatisfiable</u>.

(e) A formula F is called a <u>tautology</u> (or <u>valid</u>) if $\alpha(F) = 1$ for every suitable truth assignment α.

A formula F is a tautology if and only if $\neg F$ is unsatisfiable.
Two formulas F, G are called (semantically) equivalent, if for all truth assignments α, which are suitable for both F and G, we have $\alpha(F) = \alpha(G)$. Notation: $F \equiv G$

The Fundamental Equivalences: Let F, G, H be formulas.

(a) $F \wedge F \equiv F$ and $F \vee F \equiv F$ (idempotency)

(b) $F \wedge G \equiv G \wedge F$ and $F \vee G \equiv G \vee F$ (commutativity)

(c) $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$ and
$(F \vee G) \vee H \equiv F \vee (G \vee H)$ (associativity)
Hence we write $F_1 \wedge \cdots \wedge F_n$ or $F_1 \vee \cdots \vee F_n$.

(d) $F \wedge (F \vee G) \equiv F$ and $F \vee (F \wedge G) \equiv F$ (absorption)

(e) $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ and
$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$ (distributive law)

(f) $\neg\neg F \equiv F$

(g) $\neg(F \wedge G) \equiv \neg F \vee \neg G$ and
$\neg(F \vee G) \equiv \neg F \wedge \neg G$ (de Morgan's rules)

(h) If F is a tautology, then $F \vee G \equiv F$ and $F \wedge G \equiv G$

(i) If F is unsatisfiable, then $F \vee G \equiv G$ and $F \wedge G \equiv F$

Substitution Theorem:
Let F_1, F_2 be two equivalent formulas.
Let G be a formula, which contains F_1 as a subformula.
Let $\tilde{G}$ be the formula obtained by replacing F_1 in G by F_2.
Then we have $G \equiv \tilde{G}$.

(a) A literal is an atomic formula or the negation of an atomic formula (A_i or $\neg A_i$)

(b) A formula F is said to be in conjunctive normal form (CNF), if it is of the form

$$F = (L_{11} \vee L_{12} \vee \cdots \vee L_{1n_1}) \wedge \cdots \wedge (L_{k1} \vee L_{k2} \vee \cdots \vee L_{kn_k})$$

where the L_{ij} are literals ("F is a conjunction of disjunctions of literals").

(c) We say that F is in disjunctive normal form (DNF) if

$$F = (L_{11} \wedge L_{12} \wedge \cdots \wedge L_{1n_1}) \vee \cdots \vee (L_{k1} \wedge L_{k2} \wedge \cdots \wedge L_{kn_k})$$

with literals L_{ij}.

Algorithm:
Let F be a formula. Consider the following sequence of instructions:

(1) Replace all occurrences of "$\Rightarrow$" and "$\Leftrightarrow$" by their definition

(2) Replace each subformula of the form $\neg\neg G$ by G.

(3) Replace in F every subformula of the form $\neg(G \vee H)$ by $\neg G \wedge \neg H$. If a subformula $\neg\neg K$ results, apply Step (2).

(4) Replace in F every subformula of the form $\neg(G \wedge H)$ by $\neg G \vee \neg H$. If a subformula $\neg\neg K$ results, apply Step (2).

(5) Repeat (3) and (4) as often as possible.

(6) Replace in F every subformula of the form $G \vee (H \wedge K)$ by $(G \vee H) \wedge (G \vee K)$

(7) Replace in F every subformula of the form $(G \wedge H) \vee K$ by $(G \vee K) \wedge (H \vee K)$

(8) Repeat (6) and (7) as often as possible. Then return F and stop.

This is an algorithm, which returns a formula $\tilde{F}$ in CNF, such that $\tilde{F} \equiv F$.