Logic

Luka Ilić

October 8, 2018

Contents

1 Propositional logic

1.1 Truth functions

Convention: In this course we write T for true and F for false.

Definition. 1.1. The language of propositional logic consists of following symbols: *propositional variables* denoted (mostly) by $p, q, \dots$ or $p_1, p_2, \dots, q_1, q_2, \dots$ and the *connectives* $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$.

Definition. 1.2. A *propositional formula* is a string of symbols obatained in the following way

1. Any variable is a formula

2. If ϕ and ψ are formulas then so are $(\phi \wedge \psi), (\phi \vee \psi), (\neg \phi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)$
3. Any formula is obtained in this way.

Definition. 1.3. A *truth function* of n variables is a function

$$f : \{T, F\}^n \rightarrow \{T, F\} \quad .$$

Exercise. How many functions are there for n variables?

Definition. 1.4. Suppose ϕ is a formula with variables $p_1, \dots, p_n$ then we obtain a truth function $F_\phi : \{T, F\}^n \rightarrow \{T, F\}$ whose value at $(x_1, \dots, x_n) \quad x_i \in \{T, F\}$ is the truth value of ϕ when p_i has value x_i. The function F_ϕ is the *truth function of* ϕ.

Example. What is the truth function of

$$(((p \rightarrow q) \wedge (q \rightarrow (\neg p))) \rightarrow (\neg p)) \quad ?$$

Definition. 1.5. A propositional formula ϕ whose truth function F_ϕ is always true is called *tautology.* Say that formulas ϕ, ψ are *logically equivalent* (l.e.) if they have the same truth function.

Remark. ϕ, ψ are l.e. iff $(\phi \leftrightarrow \psi)$ is a tautology. Also, suppose that we got some formula ϕ with variables $p_1, \dots, p_n$ and $\phi_1, \dots, \phi_n$ are formulas with variables $q_1, \dots, q_r$. For each $i \leq n$ substitute ϕ in place of p_i in ϕ. Then the result is a formula ψ and if ϕ is a tautology, then so is ψ.

Proof. The first statement is easy. For the second remark that

$$F_\psi(q_1, \dots, q_r) = F_\phi(F_{\phi_1}(q_1, \dots, q_r), \dots, F_{\phi_n}(q_1, \dots, q_r))$$

by the induction on the number of connectives in ϕ. □

Example.

1. $(p_1 \wedge (p_2 \wedge p_3))$ is l.e. to $((p_1 \wedge p_2) \wedge p_3)$,
2. same with $\vee$,
3. $(p_1 \vee (p_2 \wedge p_3))$ is l.e. to $((p_1 \vee p_2) \wedge (p_1 \vee p_3))$

4. similar the other way around.
5. etc.

Remark. Note that by the remark above, we can boost these equivalences by substituting formulas for the variables.

Definition. 1.6. Say that a set of connectives is *adequate* if for evry $n \geq 1$, every truth function of n variables is the truth function of some formula which involves only connectives from the set and variables $p_1, \dots, p_n$.

Theorem. 1.7. *The set* $\{\neg, \wedge, \vee\}$ *is adequate.*

Proof. Let $G : \{T, F\}^n \to \{T, F\}$

1. $G(v) = F$ for all $v \in \{T, F\}$. Take ϕ to be $(p_1 \wedge (\neg p_1))$ then $G = F_\phi$
2. (*Disjunctive Normal Form* List the $v \in \{T, F\}^n$ with $G(v) = T$ as $v_1, \dots, v_r$. Write $v_i = (v_{i1}, \dots, v_{in})$ Define
$$q_{ij} = \begin{cases} p_j & \text{if } v_{ij} = T \\ (\neg p_j) & \text{if } v_{ij} = F \end{cases}$$
So q_{ij} has value T iff p_j has value v_{ij}. Let ψ_i be
$$(q_{i1}, \dots, q_{in})$$
Then $F_{\psi_i}(v) = T$ iff each q_{ij} has value T iff $v = v_i$.

   Let θ be $(\phi_1 \vee, \dots, \vee \phi_r)$. Then $F_\theta(v) = T$ iff $F_{\psi_i}(v) = T$ for some i which is equivalent to $v = v_i$ for some $i \leq r$. Thus $F_\theta(v) = T$ iff $G(v) = T$ i.e. $F_\theta = G$. As θ was constructed using only $\neg, \vee, \wedge$ the statement follows.

□

Definition. 1.8. A formula in the form as θ in the proof above (1.7) is said to be in *disjunctive normal form (dnf).*

Remark. Apart from the very intuitive and useful dnf, having a small adequate set at our disposal is useful for the following reason. It shortens induction proofs over the structure of formulas by a considerable amount, as the reader will surely realize soon.

Corollary. 1.9. *Suppose χ is a formula which truth function is not always false. Then χ is l.e. to a formula in dnf.*

Proof. Take $G = F_\chi$ and apply the second case from the proof above. □

Example. For
$$\chi : \quad ((p_1 \to p_2) \to (\neg p_2))$$
the truth function $F_\chi(v)$ is true, precisely when $v = \{T, F\}$ or $v = \{F, F\}$. Hence the dnf is:
$$((p_1 \wedge (\neg p_2)) \vee ((\neg p_1) \wedge (\neg p_2))).$$

Corollary. 1.10. *The following sets of connectives are adequate:*

1. $\neg, \vee$

2. $\neg, \wedge$

3. $\neg, \rightarrow$.

Proof. 1. By 1.7 we just need to show, that $\wedge$ can be expressed using $\neg, \vee$. $(p \wedge q)$ is l.e. to $(\neg((\neg p) \vee (\neg q))$.

2. similar to the approach above. $(p \vee q)$ is l.e. to $(\neg((\neg p) \wedge (\neg q))$.

3. Due to the cases above, it suffices to express either $\wedge$ or $\vee$ using $\neg, \rightarrow$. $(p \vee q)$ is l.e. to $((\neg p) \rightarrow q)$.

$\square$

Example. Some sets of connectives that are not adequate are:

1. $\wedge, \vee$

2. $\neg, \leftrightarrow$

Proof. 1. If ϕ is build using $\wedge, \vee$ then $F_\phi(T, \ldots, T) = T$ as proven by induction over number of connectives.

2. exercise

$\square$

Example. The *NOR* connective $\downarrow$ has truth table:

p	q	$(p \downarrow q)$
T	T	F
T	F	F
F	T	F
F	F	T

It is adequate on its own.

(exercise - express $\neg, \wedge$)

1.2 A formal system for propositional logic

Idea: Try to generate all tautologies from certain basic assumptions (axioms) using appropriate deduction rules.

Definition. 1.11. This is important!

A *formal deduction system* Σ has the following ingredients:

1. An *alphabet* A of symbols ($A \neq \emptyset$

2. a non empty set $\mathcal{J}$ of the set of all finite sequences ('strings') of the elements of A: the *formulas* of Σ.

3. A subset $\mathcal{A} \subseteq \mathcal{J}$ called the *axioms* of Σ.

4. A collection of *deduction rules.*

Definition. 1.12. A *proof* in Σ us a finite sequence of formulas in $\mathcal{J}$

$$\phi_1, \dots, \phi_n$$

such that each ϕ_i is either an axiom *or* is obtained from $phi_1, \dots, \phi_{i-1}$ using one of the deduction rules. The last (or any) formula in a proof is a *theorem* of Σ. Write $\vdash_\Sigma \phi$ for 'ϕ is a theorem of Σ'.

Remark.

1. If $\phi \in \mathcal{A}$ then $\vdash_\Sigma \phi$
2. We should have an algorithm to test whether a string of symbols really is a formula and whether it is an axiom. Then someone who is willing to follow an algorithm precisely (computer) should be able to generate all possible proofs in σ and check whether something is a proof. (Say Σ is *recursive* in this case

Definition. 1.13. The formal system L for propositional logic has:

- **Alphabet**: variables $p_1, p_2, p_3 \dots$ connectives $\neg, \rightarrow$ punctuation)(.
- **Formulas**: as defined in 1.2 and will be called *L-formulas*.
- **Axioms**: Suppose ϕ, ψ, χ are L-formulas. The following are axioms of L:
  - A1 $(\phi \rightarrow (\psi \rightarrow \phi))$
  - A2 $((\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\psi \rightarrow \chi)))$
  - A3 : $(((\neg\psi) \rightarrow (\neg\phi)) \rightarrow (\phi \rightarrow \psi))$
- **deduction rule**: *Modus Ponens (MP)* from *phi* $(phi \rightarrow \psi)$ deduce ψ.

Example. Suppose ϕ is an L-formula. Then $\vdash_L (\phi \rightarrow \phi$. A proof in L could be as follows:

1. $(\phi \rightarrow ((\phi \rightarrow \phi) \rightarrow \phi))$
2. $(\chi \rightarrow (\phi \rightarrow ((\phi \rightarrow \phi) \rightarrow \phi)))$
3. $((\phi \rightarrow (\phi \rightarrow \phi)) \rightarrow (\phi \rightarrow \phi))$
4. $(\phi \rightarrow (\phi \rightarrow \phi))$
5. $(\phi \rightarrow \phi$.