

Mathematical Logic

*A summary of “A Concise Introduction to Mathematical Logic”, W. Rautenberg
Ari Feiglin*

Contents

1	Propositional Logic	2
1.1	Semantics of Propositional Logic	2
	Principle of Formula Induction	2
	Unique Formula Reconstruction Property	3
2	Index	5

1 Propositional Logic

1.1 Semantics of Propositional Logic

Propositional logic is the study of logic removed from interpretation of individual variables and context. I will assume that the reader already has experience with propositional logic, as this is something an undergraduate will cover in one of their first courses. While this subsection will focus mainly on the semantics of propositional logic, we will begin by defining its *syntax*,

1.1.1 Definition

Let PV be an arbitrary set of **propositional variables** (which are regarded as arbitrary symbols). **Propositional formulas** are formulas defined recursively by the following rules,

- (1) Propositional variables in PV are formulas, called **prime** or **atomic** formulas.
- (2) If α and β are formulas, then so are $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, and $\neg\alpha$.

The set of all the formulas constructed in this manner is denoted \mathcal{F} .

We can generalize this definition; instead of utilizing only the symbols \wedge and \vee , we can take a general *logical signature* σ consisting of logical connectives of differing arities. We then recursively define σ -formulas as following: if c is an n -ary logical connective in σ , and $\alpha_1, \dots, \alpha_n$ are formulas, then so is

$$(c\alpha_1, \dots, \alpha_n)$$

Alternatively, if we only consider binary and unary connectives, then if c is a unary connective, we define $c\alpha$ to be a formula, and if \circ is a binary connective, then $(\alpha \circ \beta)$ is a formula. But we don't have much need for such generalizations, as $\{\wedge, \vee, \neg\}$ is complete, in the sense that all connectives can be defined using them. This is a fact we will discuss soon.

We can define other connectives, for example \rightarrow and \leftrightarrow are used as shorthands:

$$(\alpha \rightarrow \beta) := \neg(\alpha \wedge \neg\beta), \quad (\alpha \leftrightarrow \beta) := ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$$

We similarly define symbols for false and true:

$$\perp := (p_1 \wedge \neg p_1), \quad \top = \neg\perp$$

For readability, we will use the following conventions when writing formulas (this is not a change to the definition of a formula, rather conventions for writing them in order to enhance readability)

- (1) We will omit the outermost parentheses when writing formulas, if there are any.
- (2) The order of operations for logical connectives is as follows, from first to last: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
- (3) We associate \rightarrow from the right, meaning $\alpha \rightarrow \beta \rightarrow \gamma$ is to be read as $\alpha \rightarrow (\beta \rightarrow \gamma)$. All other connectives associate from the left, for example $\alpha \wedge \beta \wedge \gamma$ is to be read as $(\alpha \wedge \beta) \wedge \gamma$.
- (4) Instead of writing $\alpha_0 \wedge \alpha_1 \wedge \dots \wedge \alpha_n$, we write $\bigwedge_{i=0}^n \alpha_i$, similar for \vee .

Since formulas are constructed in a recursive manner, most of our proofs about them are inductive.

1.1.2 Principle (Principle of Formula Induction)

Let \mathcal{E} be a property of strings which satisfies the following conditions:

- (1) $\mathcal{E}\pi$ for all prime formulas π ,
- (2) If $\mathcal{E}\alpha$ and $\mathcal{E}\beta$, then $\mathcal{E}(\alpha \wedge \beta)$, $\mathcal{E}(\alpha \vee \beta)$, and $\mathcal{E}\neg\alpha$ for all formulas $\alpha, \beta \in \mathcal{F}$.

Then $\mathcal{E}\varphi$ is true for all formulas φ .

An example of this is that every formula $\varphi \in \mathcal{F}$ is either prime, or of one of the following forms

$$\varphi = \neg\alpha, \quad \varphi = (\alpha \wedge \beta), \quad \varphi = (\alpha \vee \beta)$$

The proof of this is straightforward: let \mathcal{E} be this property. Then trivially, $\mathcal{E}\pi$ for all prime formulas π . And if $\mathcal{E}\alpha$ and $\mathcal{E}\beta$, then of course we have

$$\mathcal{E}\neg\alpha, \quad \mathcal{E}(\alpha \wedge \beta), \quad \mathcal{E}(\alpha \vee \beta)$$

This is the first step in showing the *unique formula reconstruction property*. Let us prove a lemma before proving the property itself,

1.1.3 Lemma

Proper initial segments of formulas are not formulas. Equivalently (by contrapositive), if α and β are formulas and $\alpha\xi = \beta\eta$ for arbitrary strings ξ and η , then $\alpha = \beta$.

Let us prove this by induction on α . If α is a prime formula, suppose that β is not a prime formula, then its first character is either (or \neg , but then $\alpha = ($ or $\alpha = \neg$, in contradiction. Thus β is a prime formula and so $\alpha = \beta$ as they are both a single character. Now if $\alpha = (\alpha_1 \circ \alpha_2)$, then the first character of β must too be (, so β is of the form $(\beta_1 * \beta_2)$. Thus

$$\alpha_1 \circ \alpha_2 \xi = \beta_1 * \beta_2 \eta$$

and so by our inductive assumption, $\alpha_1 = \beta_1$, and so $\circ = *$, and thus $\alpha_2 = \beta_2$ by our inductive assumption again. And so $\alpha = \beta$ as required. The proof for the case that $\alpha = \neg\alpha'$ is similar. ■

1.1.4 Proposition (Unique Formula Reconstruction Property)

Every compound formula $\varphi \in \mathcal{F}$ is of one of the following forms:

$$\varphi = \neg\alpha, \quad \varphi = (\alpha \wedge \beta), \quad \varphi = (\alpha \vee \beta)$$

For some formulas $\alpha, \beta \in \mathcal{F}$.

We have already shown existence. We will now show that this is unique, meaning that φ can be written uniquely as one of these strings. Using the lemma proven above, the proof for uniqueness of the reconstruction property is immediate. For example, if $\varphi = (\alpha_1 \wedge \beta_1)$ then obviously φ cannot be written as $\neg\alpha_2$ since $(\neq \neg$, and if $\varphi = (\alpha_2 \vee \beta_2)$ then by the lemma $\alpha_1 = \alpha_2$, and so we get that $\wedge = \vee$ in contradiction. And finally if $\varphi = (\alpha_2 \wedge \beta_2)$, then again by the lemma, $\alpha_1 = \alpha_2$ and $\beta_1 = \beta_2$ as required. The proof for \neg and \vee are similar. ■

Utilizing formula recursion, we can define functions on formulas. For example,

1.1.5 Definition

For a formula φ , we define $\text{Sf}\varphi$ to be the set of all subformulas of φ . This is done recursively:

$$\begin{aligned} \text{Sf}\pi &= \{\pi\} \text{ for prime formulas } \pi, \\ \text{Sf}\neg\alpha &= \text{Sf}\alpha \cup \{\alpha\}, \quad \text{Sf}(\alpha \circ \beta) = \text{Sf}\alpha \cup \text{Sf}\beta \cup \{(\alpha \circ \beta)\} \text{ for a binary logical connective } \circ \end{aligned}$$

Similarly, we can define the **rank** of a formula φ ,

$$\begin{aligned} \text{rank}\pi &= 0 \text{ for prime formulas } \pi, \\ \text{rank}\neg\alpha &= \text{rank}\alpha + 1, \quad \text{rank}(\alpha \circ \beta) = \max\{\text{rank}\alpha, \text{rank}\beta\} + 1 \text{ for a binary logical connective } \circ \end{aligned}$$

And we can also define the set of variables in φ ,

$$\begin{aligned} \text{Var}\pi &= \{\pi\} \text{ for prime formulas } \pi, \\ \text{Var}\neg\alpha &= \text{Var}\alpha, \quad \text{Var}(\alpha \circ \beta) = \text{Var}\alpha \cup \text{Var}\beta \text{ for a binary logical connective } \circ \end{aligned}$$

In all definitions \circ is either \wedge or \vee .

So now that we have discussed the syntax of propositional logic, it is time to discuss its semantics; how we assign to formulas truth values. Recall the truth tables for \wedge , \vee , and \neg :

α	β	$\alpha \wedge \beta$	α	β	$\alpha \vee \beta$	α	$\neg\alpha$
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		

These define how the logical connectives function as functions on $\{0, 1\}$.

1.1.6 Definition

A **propositional valuation**, or a **propositional model**, is a function

$$w: PV \longrightarrow \{0, 1\}$$

We can extend it to a function $w: PV \longrightarrow \mathcal{F}$ as follows:

$$w(\alpha \wedge \beta) = w\alpha \wedge w\beta, \quad w(\alpha \vee \beta) = w\alpha \vee w\beta, \quad w\neg\alpha = \neg w\alpha$$

Notice that we would need to define, for example, $w(\alpha \rightarrow \beta) = w\alpha \rightarrow w\beta$ had \rightarrow been an element of our logical signature. But since \rightarrow is defined using \wedge and \neg , we must prove this identity:

$$w(\alpha \rightarrow \beta) = w\neg(\alpha \wedge \neg\beta) = \neg w(\alpha \wedge \neg\beta) = \neg(w\alpha \wedge \neg w\beta) = w\alpha \rightarrow w\beta$$

This is of course not a coincidence, but a result of the fact that $\alpha \rightarrow \beta = \neg(\alpha \wedge \neg\beta)$ (where $\alpha, \beta \in \{0, 1\}$). Notice that furthermore,

$$w\top = 1, \quad w\perp = 0$$

1.1.7 Proposition

The valuation of a formula is dependent only on its variables. Meaning if φ is a formula and w and w' are two valuations where $w\pi = w'\pi$ for all $\pi \in \text{Var}\varphi$, then $w\varphi = w'\varphi$.

We will prove this by induction on φ . For prime formulas, this is obvious as $\text{Var}\varphi = \{\varphi\}$ and then $w\varphi = w'\varphi$ by the proposition's assumption. For $\varphi = \alpha \wedge \beta$, we have that

$$w\varphi = w\alpha \wedge w\beta = w'\alpha \wedge w'\beta = w'\varphi$$

where the second equality is our inductive assumption. The proof for $\varphi = \alpha \vee \beta$ and $\varphi = \neg\alpha$ is similar. ■

Let us suppose that $PV = \{p_1, p_2, \dots, p_n, \dots\}$, then we define \mathcal{F}_n to be the set of formulas φ such that $\text{Var}\varphi \subseteq \{p_1, \dots, p_n\}$.

1.1.8 Definition

A **boolean function** is a function

$$f: \{0, 1\}^n \longrightarrow \{0, 1\}$$

for some $n \geq 0$. The set of boolean functions of arity n is denoted \mathbf{B}_n . A formula $\varphi \in \mathcal{F}_n$ **represents** a boolean function $f \in \mathbf{B}_n$ (similarly, f is represented by φ), if for all valuations w ,

$$w\varphi = f(w\vec{p}) \quad (w\vec{p} = (wp_1, \dots, wp_n))$$

So for example, $\alpha = p_1 \wedge p_2$ represents the function $f(p, q) = p \wedge q$. This is since

$$f(wp_1, wp_2) = wp_1 \wedge wp_2 = w(p_1 \wedge p_2) = w\alpha$$

Since valuations of $\varphi \in \mathcal{F}_n$ are defined by their values on p_1, \dots, p_n , φ represents at most a single function f . In fact, it represents the function

$$\varphi^{(n)}(x_1, \dots, x_n) = w\varphi$$

where w is any valuation such that $wp_i = x_i$ (all of these valuations value φ the same). Now, notice that $\mathcal{F}_n \subset \mathcal{F}_{n+1}$ and $\mathbf{B}_n \subset \mathbf{B}_{n+1}$ and so $\varphi \in \mathcal{F}_n$ represents a function in \mathbf{B}_{n+1} as well. But this function is not essentially in \mathbf{B}_n in the sense that its last argument does not impact its value. Formally we say that given a function $f: M^n \longrightarrow M$, we call its i th argument *fictional* if for all $x_1, \dots, x_i, \dots, x_n \in M$ and $x'_i \in M$:

$$f(x_1, \dots, x_i, \dots, x_n) = f(x_1, \dots, x'_i, \dots, x_n)$$

An *essentially n -ary* function is a function with no fictional arguments.

2 Index

Atomic formula: 2

See also: Prime formula

Boolean function: 4

Essentially n -ary function: 4

Model: 3

See also: Valuation

Prime formula: 2

Rank: 3

Valuation: 3