Mathematical Logic and Model Theory

Lectures by Slurp and Sharp

Contents

1 Lecture 1

Sources: A Concise Introduction to Mathematical Logic, Section 1, W. Rautenberg

We begin by defining *what* exactly mathematical logic is. Mathematical logic is a sort of metamathematical study of mathematics itself. It studies what sorts of logical statements we can make, how we can manipulate them, and what we can say about the mathematical objects which satisfy them. But the best way to understand what mathematical logic is, is to actually *do* it! So let's begin.

1.1 Propositional Logic

We begin our discussion with the simplest logic: propositional logic. This logic studies how we can connect propositions together, e.g. using *and*, *or*, *not*, etc. Suppose we wanted to say that "*if* it is cold outside *then* I will wear a coat", how could we go about this mathematically?

We begin with some definitions:

1.1 Definition

A **boolean function** is a function $\{0,1\}^n \longrightarrow \{0,1\}$ for some $n > 0$.

1.2 Definition

A **connective** is a symbol s with an associated boolean function (which will be named with s as well). A set of connectives is called a **logical signature**.

For boolean connectives $\circ$ (connectives whose function accepts two parameters), we can use a *truth table* to define it:

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$

where a_{ij} is defined to be the value of $i \circ j$. So for example, we can define the following connective $\wedge$ as follows:

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

This connective takes two boolean values x and y and checks that both x *and* y are true. For this reason $\wedge$ is called *logical and* or a *conjunction*.

We can also define *logical or* or *disjunction* $\vee$:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

And *logical negation* $\neg$, which is a unary connective: $\neg 0 = 1$ and $\neg 1 = 0$. Finally let us define *logical implication* $\rightarrow$:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Why should false imply false $(0 \rightarrow 0 = 1)$? Well suppose I said "if it rains then it is cloudy", but it is not raining. Is what I said false? Well, not necessarily! This is what is called a *vaccuous truth*.

Now suppose I wanted to string together connectives, like "if it rains then it is cloudy and I should wear a jacket".

1.3 Definition

Suppose we have a global set of variables V, whose elements are simply symbols which we will call *propositional variables*. Now suppose we have a logical signature ℓ, we define the set of **propositional formulas** $\mathcal{F}_\ell$ recursively as follows:

(**1**) If $p \in V$ is a propositional variable then it is a formula: $p \in \mathcal{F}_\ell$. Such formulas are called **prime formulas**.

(**2**) If $\mathsf{s} \in \ell$ is a connective of arity n and $\varphi_1, \ldots, \varphi_n \in \mathcal{F}_\ell$ are formulas, then so too is

$$\mathsf{s}\varphi_1 \cdots \varphi_n \in \mathcal{F}_\ell$$

So for example, if $\ell = \{\wedge, \vee, \neg\}$ and $V = \{p_1, p_2, \ldots\}$ then the following are formulas:

$$p_1, \qquad \wedge p_1 p_2, \qquad \wedge \vee p_1 p_2 \neg p_3, \qquad \neg \wedge \vee p_1 p_2 \wedge p_1 p_3$$

But using prefix notation like this is confusing, so we will adopt the custom that for binary connectives $\circ$, $\circ\varphi\psi$ is instead written as $(\varphi \circ \psi)$. So these formulas become

$$p_1, \qquad (p_1 \wedge p_2), \qquad ((p_1 \vee p_2) \wedge \neg p_3), \qquad \neg((p_1 \vee p_2) \wedge (p_1 \wedge p_3))$$

We will call the signature $\ell = \{\wedge, \vee, \neg\}$ the *standard signature*.

An important thing to keep in mind is that currently formulas are simply special strings. We haven't assigned to them any value yet.

Note that our definition of propositional formulas isn't really all that formal: how can we define a set using itself? Well formally, what we do is we look at the collection of all sets S of strings (over the alphabet $\ell \cup V$) with the properties that (1) $V \subseteq S$, (2) if $\mathsf{s} \in \ell$ has arity n and $\varphi_1, \ldots, \varphi_n \in S$ then $\mathsf{s}\varphi_1 \cdots \varphi_n \in S$. Then we simply define $\mathcal{F}_\ell$ to be the intersection of these sets.

From this definition the following is immediate:

1.4 Lemma (The Principle of Formula Induction)

Let ℓ be a logical signature. Suppose $\mathcal{E}$ is a property of strings (i.e. a subset of the set of all strings over $V \cup \ell$), with the following properties:

(**1**) For every $p \in V$, $\mathcal{E}p$.

(**2**) For every $\mathsf{s} \in \ell$ with arity n, if $\varphi_i \in \mathcal{F}_\ell$ for $i = 1, \ldots, n$ then $\mathcal{E}\mathsf{s}\varphi_1 \cdots \varphi_n$.

Then $\mathcal{E}\varphi$ holds for all formulas $\varphi \in \mathcal{F}_\ell$.

Now how can we be sure that if we have a formula φ, suppose of the form $\wedge\alpha\beta$, it is not simultaneously of the form $\wedge\alpha'\beta'$ for some other $\alpha, \beta \in \mathcal{F}$? We can use formula induction to prove the following:

1.5 Lemma (The Unique Formula Reconstruction Property)

Every compound formula $\varphi \in \mathcal{F}_\ell$ is of the form $\mathsf{s}\varphi_1 \cdots \varphi_n$ for uniquely determined s and φ_i for $i = 1, \ldots, n$.

Proof: s is obviously uniquely determined since it is the first character of φ. Now, we need to prove that if $\varphi_1 \cdots \varphi_n = \psi_1 \cdots \psi_m$ then $n = m$ and $n = m$. To do so, we need to prove the claim that a proper prefix of a formula is not itself a formula.

This uses formula induction: for prime formulas this is trivial. Otherwise, let $\varphi = \mathsf{s}\varphi_1 \cdots \varphi_n$ then a proper prefix of φ is either s which is not a formula, or of the form $\mathsf{s}\varphi_1 \cdots \varphi_i'$ where φ_i' is a prefix of φ_i. In order for $\mathsf{s}\varphi_1 \cdots \varphi_i'$ to be a formula, $\varphi_1 \cdots \varphi_i'$ must be able to be split into n formulas. So suppose $\varphi_1 \cdots \varphi_i' = \psi_1 \cdots \psi_n$, but φ_1 and ψ_1 cannot be prefixes of one another, so $\varphi_1 = \psi_1$. And so on until $i-1$. Then we have $\varphi_i' = \psi_i \cdots \psi_n$, but then if $\varphi_i' \neq \varphi_i$ we have that ψ_i is a proper prefix of φ_i, a contradiction to our inductive assumption. So $\varphi_i = \psi_i \cdots \psi_n$, so we get that $i = n$, contradicting the assumption that the prefix is proper.

Note that in proving this claim, we have proven precisely the unique reconstruction property. ■

Note that the reconstruction property holds true even when our strings use the custom that binary connectives are written as $(\alpha \circ \beta)$.

Using the unique reconstruction property, we can define functions on formulas in a recursive manner. For example, we can define the *substring function*:

$$Sf\pi = \{\pi\} \text{ for prime } \pi, \qquad Sf\mathsf{s}\varphi_1 \cdots \varphi_n = \bigcup_{i=1}^{n} Sf\varphi_i \cup \{\mathsf{s}\varphi_1 \cdots \varphi_n\}$$

So $Sf\varphi$ is precisely all the subformulas of φ. Note that Sf is well-defined precisely because of the unique reconstruction property: a formula cannot satisfy multiple conditions in the definition of Sf at once.
More importantly than this example, we can assign truth to formulas:

1.6 Definition
Let $w\colon V \longrightarrow \{0,1\}$ be a **valuation** – a mapping of truth values to each propositional variable. Then we can extend w to a function $w\colon \mathcal{F}_\ell \longrightarrow \{0,1\}$ by recursion as follows:

(**1**) For π prime, $w\pi$ is already defined ($\pi \in V$).

(**2**) If $\mathsf{s} \in \ell$ and $\varphi_1, \ldots, \varphi_n \in \mathcal{F}_\ell$ then

$$w\mathsf{s}\varphi_1 \cdots \varphi_n = \mathsf{s}(w\varphi_1, \ldots, w\varphi_n)$$

So for example suppose $w(p) = 1$ and $w(q) = 0$ then

$$w(p \wedge q) = 0, \qquad w(p \wedge (q \vee \neg q)) = 1$$

Let $var\varphi$ denote all the variables in V occurring in φ. This can be defined recursively as follows:

$$var\pi = \{\pi\} \text{ for } \pi \text{ prime}, \qquad vars\varphi_1 \cdots \varphi_n = \bigcup_{i=1}^{n} var\varphi_i$$

Then notice that $w\varphi$ is dependent only on w's values on $var\varphi$. That is, we have the following:

1.7 Lemma
If w, w' are two valuations such that $w(p) = w'(p)$ for all $p \in var\varphi$, then $w\varphi = w'\varphi$.

Proof: by formula induction. ■

Now, suppose that for every $n \in \mathbb{N}$ we have $p_n \in V$. Then we can define $\mathcal{F}_\ell^n = \{\varphi \in \mathcal{F}_\ell \mid var\varphi \subseteq \{1, \ldots, n\}\}$, the set of all formulas whose variables are contained in $p_1, \ldots, p_n$.

1.8 Definition
We say that a formula $\varphi \in \mathcal{F}_n^\ell$ **represents** a boolean function $f\colon \{0,1\}^n \longrightarrow \{0,1\}$ if for every valuation w,

$$w\varphi = f(wp_1, \ldots, wp_n)$$

Since $w\varphi$ is dependent only $wp_1, \ldots, wp_n$, f is uniquely determined: define $f(x_1, \ldots, x_n)$ to be $w\varphi$ where $w(p_i) = x_i$ and $w(p)$ arbitrary for all other variables. So we can denote f by $\varphi^{(n)}$ since it is unique.
Notice that implication can be represented by $\neg(p_1 \wedge \neg p_2)$, thus in the standard signature, we can use $(\alpha \to \beta)$ as a standin for $\neg(\alpha \wedge \neg\beta)$. Similarly $\leftrightarrow$ can be represented by $(\alpha \to \beta) \wedge (\beta \to \alpha)$.

1.2 Logical Equivalence

1.1 Definition
Say two formulas α, β are **logically equivalent** if $w\alpha = w\beta$ for every valuation w. Denote this by $\alpha \equiv \beta$.

For example, $\alpha \equiv \neg\neg\alpha$. We can define $\top = p \vee \neg p$ and $\bot = p \wedge \neg p$ for $p \in V$. Notice that $w\top = 1$ for all w, and $w\bot = 0$, so $\top$ and $\bot$ represent truth and false respectively.
The following can be easily verified:

$\alpha \wedge (\beta \wedge \gamma) \equiv (\alpha \wedge \beta) \wedge \gamma$	$\alpha \vee (\beta \vee \gamma) \equiv (\alpha \vee \beta) \vee \gamma$
$\alpha \wedge \beta \equiv \beta \wedge \alpha$	$\alpha \vee \beta \equiv \beta \vee \alpha$
$\alpha \wedge \alpha \equiv \alpha$	$\alpha \vee \alpha \equiv \alpha$
$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$	$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$	$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$

Furthermore, $\alpha \vee \neg\alpha \equiv \top$ and $\alpha \wedge \neg\alpha \equiv \bot$ for all α. For implication, we adopt the custom that it is right associative: that is, $\alpha \to \beta \to \gamma$ means $\alpha \to (\beta \to \gamma)$. Then notice the interesting equivalence:

$$\alpha_1 \to \cdots \to \alpha_n \to \beta \equiv \bigwedge_{i=1}^{n} \alpha_i \to \beta$$

where $\bigwedge_{i=1}^{n} \alpha_i = \alpha_1 \wedge \cdots \wedge \alpha_n$.
Notice that $\equiv$ is an equivalence relation. Furthermore it is a *congruence* (the precise definition of this will be given in a later lecture): for $\mathsf{s} \in \ell$ and $\varphi_1, \ldots, \varphi_n$ and $\psi_1, \ldots, \psi_n$, if $\varphi_i \equiv \psi_i$ for all i then $\mathsf{s}\varphi_1 \cdots \varphi_n \equiv \mathsf{s}\psi_1 \cdots \psi_n$. Thus we get the following:

1.2 Lemma (The Replacement Lemma)

Suppose $\alpha \equiv \alpha'$, and let $\varphi \in \mathcal{F}_\ell$. Let φ' result from φ by replacing one or more instances of α in φ with α'. Then $\varphi \equiv \varphi'$.

This will be proven in more generality in a later lecture.

1.3 Normal Forms

1.1 Definition

Prime formulas and their negations are called **literals**. If α_i are conjunctions of literals, then $\bigvee_{i=1}^{n} \alpha_i$ is called a **disjunctive normal form** (DNF). Similarly if β_i are disjunctions of literals, then $\bigwedge_{i=1}^{n} \beta_i$ is called a **conjunctive normal form** (CNF).

We get the following important theorem.

1.2 Theorem

Every boolean function $f\colon \{0,1\}^n \longrightarrow \{0,1\}$ can be represented by a DNF α_f and a CNF β_f.

Proof: define $p^1 = p$ and $p^0 = \neg p$, then define

$$\alpha_f = \bigvee_{f\bar{x}=1} \bigwedge_{i=1}^{n} p_i^{x_i}$$

Notice that if $w(p_i) = x_i$, then if $f\bar{x} = 1$ then $\bigwedge_{i=1}^{n} p_i^{x_i}$ is in the disjunction, and

$$w \bigwedge_{i=1}^{n} p_i^{x_i} = \bigwedge_{i=1}^{n} w p_i^{x_i} = 1$$

Otherwise if $f\bar{x} = 0$ then for every $f\bar{y} = 1$, there is a $y_i \neq x_i$ and so $w p_i^{y_i} = 0$, so every conjunction is not satisfied, and thus α_f is not.
A similar proof goes for

$$\beta_f = \bigwedge_{f\bar{x}=0} \bigvee_{i=1}^{n} p_i^{\neg x_i}$$

■

Note that by definition if two formulas represent the same boolean function then they are logically equivalent. Thus we have:

1.3 Corollary

Every formula is logically equivalent to a CNF and DNF.

1.4 Definition

A logical signature ℓ is **functional complete** if every boolean function can be represented by a formula in $\mathcal{F}_\ell$.

Since every boolean function can be represented by a DNF and CNF, this means that the standard signature is functional complete. Furtheremore, we know that $\wedge$ can be represented by $\vee$: $\alpha \wedge \beta \equiv \neg(\neg\alpha \vee \neg\beta)$, and vice versa. This means that $\{\wedge, \neg\}$ and $\{\vee, \neg\}$ are both functional complete.