

# Curso de Lógica Matemática

Cristo Daniel Alvarado

15 de febrero de 2024

# Índice general

<b>0. Introducción</b>	<b>2</b>
0.1. Temario . . . . .	2
0.2. Conectivas Lógicas . . . . .	2
<b>1. Lógica Proposicional</b>	<b>4</b>
1.1. Alfabeto . . . . .	4
1.2. Modeos o Estructuras . . . . .	5

# Capítulo 0

## Introducción

### 0.1. Temario

Los siguientes temas se verán a lo largo del curso:

1. Lógica (Teoría de Modelos).
  - 1.1) Lógica proposicional.
  - 1.2) Lógica de primer orden.
2. Teoría de la Computabilidad.
3. Teoría de Conjuntos.

Y la bibliografía para el curso es la siguiente:

- Enderton, 'Introducción matemática a la lógica'.
- Enderton, 'Teoría de la computabilidad'.
- Copi, 'Lógica Simbólica' o 'Computability Theory'.
- Rebeca Weber 'Computability Theory'.

### 0.2. Conectivas Lógicas

La disyunción ( $\vee$ ), conjunción ( $\wedge$ ), negación ( $\neg$ ), implicación ( $\Rightarrow$ ) y si y sólo si ( $\iff$ ) son las conectivas lógicas usadas usualmente.

(Se habló un poco de una cosa llamada forma normal disyuntiva).

A  $\{\wedge, \vee, \neg\}$  se le conoce como un conjunto completo de conectivas lógicas. Nos podemos quedar simplemente con conjuntos completos de disyuntivas con solo dos elementos, a saber:  $\{\wedge, \neg\}$  y  $\{\vee, \neg\}$ , ya que  $P \vee Q$  es  $\neg(\neg P \wedge \neg Q)$ . (de forma similar a lo otro  $P \wedge Q$  es  $\neg(\neg P \vee \neg Q)$ ).

También  $\{\Rightarrow, \neg\}$  es otro conjunto completo de conectivas lógicas, ya que  $P \wedge Q$  es  $\neg(P \Rightarrow \neg Q)$ .

Y,  $\{|\}$  es un conjunto completo, donde  $|$  es llamado la **barra de Scheffel**, que tiene la siguiente tabla de verdad.

$P$	$Q$	$P Q$
$V$	$V$	$F$
$V$	$F$	$V$
$F$	$V$	$V$
$F$	$F$	$V$

con este, se tiene un conjunto completo de conectivas lógicas.

Como muchas veces se usan conectivas de este tipo:

$$(P \Rightarrow \neg Q) \Rightarrow ((P \Rightarrow R) \wedge \neg(Q \Rightarrow S) \wedge T)$$

al ser muy largas, a veces es más conveniente escribirlas en forma Polaca. De esta forma, lo anterior quedaría de la siguiente manera:

$$\Rightarrow \Rightarrow P \neg Q \wedge \wedge P R \neg \Rightarrow Q S T$$

Ahora empezamos con el estudio formal de la lógica.

# Capítulo 1

## Lógica Proposicional

### 1.1. Alfabeto

El alfabeto de la lógica proposicional es un conjunto que consta de dos tipos de símbolos:

1. **Variables**, denotadas por  $p_1, p_2, \dots, p_n, \dots$  (a lo más una cantidad numerable). Estas representan proposiciones o enunciados (tengo un paraguas, me caí de las escaleras, no tengo café en la cafetera, etc...).
2. **Conectivas**, como  $\Rightarrow$  y  $\neg$ .

Aceptamos la existencia de estas cosas (pues, al menos debemos aceptar la existencia de algo).

Se van a trabajar con sucesiones finitas de símbolos del alfabeto descrito anteriormente. Ahora necesitaremos especificar que tipos de sucesiones van a servirnos para tener un significado formal.

#### Definición 1.1.1

En el conjunto de sucesiones finitas de símbolos del alfabeto, definimos una **fórmula bien formada** (abreviada como **FBF**) como sigue:

1. Cada variable es una **FBF**.
2. Si  $\varphi, \psi$  son **FBF**, entonces  $\neg\varphi$  y  $\Rightarrow \varphi\psi$  también lo son.

#### Observación 1.1.1

Recordar que usamos la notación Polaca en la definición anterior.

A continuación unos ejemplos:

#### Ejemplo 1.1.1

$p_{17}, p_{54}$  y  $\Rightarrow p_2 p_{25}$  son FBF. Las primeras dos son llamadas **variables aisladas**. También lo es  $\neg \Rightarrow p_2 p_{25}$  (en este ejemplo, los  $p_i$  son variables).

Pero, por ejemplo  $\Rightarrow \neg p_1 p_2 p_3$  y  $\Rightarrow p_4$  no son FBF.

Viendo el ejemplo anterior, notamos que el operador  $\Rightarrow$  es binario (solo usa dos entradas) y  $\neg$  es unario (solo una entrada). Por lo cual, añadir o no demás variables a los operadores dentro de la fórmula, hace que la fórmula ya no sea una FBF.

#### Observación 1.1.2

Eventualmente se va a sustituir la notación Polaca por la normal, para que se pueda leer la FBF y el proceso no sea robotizado.

Definiremos ahora más conectivas lógicas para poder trabajar más cómodamente.

### Definición 1.1.2

Se definirán tres conectivas lógicas adicionales.

1. Se define la **disyunción**  $\varphi \vee \psi$  como  $\Rightarrow \neg\psi\varphi$  (en notación Polaca).
2. Se define la **conjunción**  $\varphi \wedge \psi$  como  $\neg(\neg\psi \vee \neg\varphi)$ .
3. Se define el **si sólo si**  $\psi \iff \varphi$  como  $(\psi \Rightarrow \varphi) \wedge (\varphi \Rightarrow \psi)$ .

## 1.2. Modeos o Estructuras

En el fondo, queremos que las FBF sean cosas verdaderas o falsas. Un Modelo o Estructura es algo que le va a dar significado a las FBF. De alguna manera va a ser una forma de asignarle el valor de verdadero o falso a cada una de las variables.

### Definición 1.2.1

Un **Modelo o Estructura** de la lógica proposicional es una función  $m : \text{Var} \rightarrow \{V, F\}$ , donde  $\text{Var}$  denota al conjunto de símbolos que son variables. Básicamente estamos diciendo que hay variables que son verdaderas y otras que son falsas.

### Teorema 1.2.1

Para todo modelo  $m$ , existe una única extensión  $\bar{m} : \text{FBF} \rightarrow \{V, F\}$ , donde  $\text{FBF}$  denota al conjunto de las fórmulas bien formadas, tal que  $\bar{m}(\neg\varphi) = V \iff \bar{m}(\varphi) = F$  y  $\bar{m}(\neg\varphi\psi) = F \iff \bar{m}(\varphi) = V$  y  $\bar{m}(\psi) = F$ .

### Definición 1.2.2

Sea  $m$  un modelo,  $\varphi$  una fórmula y  $\Sigma$  un conjunto de fórmulas. Definimos que

1.  $m \models \varphi$  ( $m$  satisface  $\varphi$ ) si  $\bar{m}(\varphi) = V$ .
2.  $m \models \Sigma$  si  $m \models \varphi$  para cada  $\varphi$  elemento de  $\Sigma$ .

### Ejemplo 1.2.1

Sea  $m$  un modelo tal que  $m(p_1) = V$  y  $m(p_i) = F$ , para todo  $i \geq 2$ . En este caso  $m \not\models p_1 p_3$ , pero  $m \models \neg p_5$ .

### Definición 1.2.3

Decimos que una fórmula  $\varphi$  es:

1. **Satisfacible** si existe un modelo  $m$  tal que  $m \models \varphi$ .
2. **Contradictoria** si todo modelo cumple que  $m \not\models \varphi$ .
3. **Una tautología** si todo modelo  $m$  cumple que  $m \models \varphi$ .

### Ejemplo 1.2.2

Tomemos de ejemplo  $a \Rightarrow p_1 p_2$ . cualquier modelo que haga a  $p_1$  y  $p_2$  verdaderas, o ambas falsas satisfacen la FBF,  $p_1, \neg \Rightarrow p_1 p_2$  o  $\neg(p_1 \Rightarrow \neg p_1)$ . Por lo cual, esta fórmula es satisfacible.

En cambio,  $\neg(p_1 \Rightarrow p_1)$  es contradictoria y, por ende  $p_1 \Rightarrow p_1$  y  $\neg p_1 \Rightarrow \neg p_1$  son tautologías.

### Definición 1.2.4

Sea  $\Sigma$  un conjunto de fórmulas. Decimos que  $\Sigma$  es

1. **Satisfacible** si existe un modelo  $m$  tal que  $m \models \Sigma$ .
2. **Contradictoria** si todo modelo cumple que  $m \not\models \Sigma$ .
3. **Una tautología** si todo modelo  $m$  cumple que  $m \models \Sigma$ .

### Ejemplo 1.2.3

El conjunto de fórmulas  $\Sigma = \{\Rightarrow p_1 p_2, p_1, \neg p_2\}$  no es satisfacible (en este caso, es contradictorio).

### Observación 1.2.1

Se tiene lo siguiente:

1. Una tautología  $\Rightarrow$  satisfacible.
2.  $\varphi$  es satisfacible  $\iff \neg\varphi$  es una contradicción.
3. Satisfacible es lo mismo que no contradictoria.

### Definición 1.2.5

Si  $\Sigma$  es un conjunto de FBF y  $\varphi$  es alguna otra fórmula, entonces decimos que  $\varphi$  es **consecuencia lógica** de  $\Sigma$ , o que  $\Sigma$  **implica lógicamente** a  $\varphi$ , escrito como  $\Sigma \models \varphi$ , si para todo modelo  $m$  tal que  $m \models \Sigma$  se tiene que  $m \models \varphi$ .

### Ejemplo 1.2.4

El conjunto de FBF  $\{\Rightarrow p_1 p_2, p_1\} \models p_2$ .

### Observación 1.2.2

Se tiene lo siguiente:

1. Un conjunto de FBF  $\Sigma \not\models \varphi$  si y sólo si  $\Sigma \cup \{\neg\varphi\}$  es satisfacible.
2. Además, un conjunto de FBF  $\Sigma \models \varphi$  si y sólo si  $\Sigma \cup \{\neg\varphi\}$  no es satisfacible.

---

### Lema 1.2.1

Sea  $\Sigma$  un conjunto de fórmulas y sean  $\text{Var}(\Sigma)$  el conjunto de las variables  $p_i$  que aparecen en las fórmulas de  $\Sigma$ . Si  $m_1$  y  $m_2$  son dos modelos tales que

$$m_1|_{\text{Var}(\Sigma)} = m_2|_{\text{Var}(\Sigma)}$$

entonces,  $\overline{m_1}|_{\Sigma} = \overline{m_2}|_{\Sigma}$ . En particular, para cada fórmula  $\varphi$  que sea elemento de  $\Sigma$ , entonces

$m_1 \models \varphi$  si y sólo si  $m_2 \models \varphi$ , más aún  $m_1 \models \Sigma$  si y sólo si  $m_2 \models \Sigma$ .

---

### **Demostración:**

Sin pérdida de generalidad,  $\Sigma$  es cerrado bajo subformulas.

Procederemos por inducción sobre  $\varphi \in \Sigma$ , demostraremos que  $\overline{m_1}(\varphi) = \overline{m_2}(\varphi)$ . Si  $\varphi$  coincide con algún  $p_i$ , entonces  $p_i \in \text{Var}(\Sigma)$  y, por tanto

$$\overline{m_1}(p_i) = m_1(p_i) = m_2(p_i) = \overline{m_2}(p_i)$$

Ahora hacemos el paso inductivo.

1. Tenemos el caso en que  $\varphi$  es de la forma  $\neg\psi$  y suponemos que  $\overline{m_1}(\psi) = \overline{m_2}(\psi)$ . Se tiene que  $\overline{m_1}(\neg\psi) = F \iff \overline{m_1}(\psi) = V \iff \overline{m_2}(\psi) = V \iff \overline{m_2}(\neg\psi) = F$ . Por lo tanto,  $\overline{m_1}(\psi) = \overline{m_2}(\psi)$ . El caso en que sea verdadero es análogo.
2. Tenemos el caso en que  $\varphi$  es de la forma  $\Rightarrow \varphi_1\psi$  y, supongamos que  $\overline{m_1}(\varphi_1) = \overline{m_2}(\varphi_1)$  y  $\overline{m_1}(\psi) = \overline{m_2}(\psi)$ . Se tiene que  $\overline{m_1}(\Rightarrow \varphi_1\psi) = F \iff \overline{m_1}(\varphi_1) = V$  y  $\overline{m_1}(\psi) = F \iff$  (por hipótesis de inducción)  $\overline{m_2}(\varphi_1) = V$  y  $\overline{m_2}(\psi) = F \iff \overline{m_2}(\Rightarrow \varphi_1\psi) = F$ . El caso en que sean verdaderas es análogo. Por tanto,  $\overline{m_1}(\Rightarrow \varphi_1\psi) = \overline{m_2}(\Rightarrow \varphi_1\psi)$ .

Lo cual completa el paso inductivo. ■

---

### **Corolario 1.2.1**

Si  $\Sigma$  es un conjunto finito de fórmulas, entonces se puede verificar 'Mecánicamente' si es el caso, que  $\Sigma \models \varphi$ .

---

El procedimiento para verificar el modelo, se hace mediante la tabla de verdad de las variables y las FBF de  $\Sigma$ .

### **Definición 1.2.6**

Decimos que un conjunto de fórmulas bien formadas  $\Sigma$  es **finitamente satisfacible** si cualquier subconjunto finito  $\Delta \subseteq \Sigma$  es satisfacible.

---

### **Teorema 1.2.2** (Teorema de Compacidad de Gödel)

Si  $\Sigma$  es un conjunto (arbitrario) de fórmulas tal que  $\Sigma \models \varphi$ , entonces existe un  $\Delta \subseteq \Sigma$  finito tal que  $\Delta \models \varphi$ .

---

El teorema que Gödel probó originalmente fue este:

---

### **Teorema 1.2.3** (Teorema de Gödel)

Un conjunto de fórmulas  $\Sigma$  es satisfacible si y sólo si es finitamente satisfacible.

---

Veamos por qué el teorema de Gödel implica el teorema de compacidad de Gödel. Se tiene que  $\Sigma \not\models \varphi \iff$  existe un modelo  $m$  tal que  $m \models \Sigma \cup \{\neg\varphi\}$ . Es decir, si y sólo si  $\Sigma \cup \{\neg\varphi\}$  es satisfacible, es decir que es finitamente satisfacible (por el teorema de Gödel), es decir que para todo  $\Delta \subseteq \Sigma$  finito se cumple que

$$\Delta \cup \{\neg\varphi\}$$

es satisfacible. Y esto sucede si y sólo si para todo  $\Delta \subseteq \Sigma$  finito existe  $m$  tal que  $m \models \Delta \cup \{\neg\varphi\}$ , si y sólo si para todo  $\Delta \subseteq \Sigma$  finito  $\Delta \not\models \varphi$ , con lo cual

$$\Sigma \not\models \varphi \iff \Delta \not\models \varphi$$

para todo  $\Delta \subseteq \Sigma$  finito, que es el teorema de compacidad en su forma contrapositiva.



---

**Lema 1.2.2**

Sea  $\Sigma$  un conjunto finitamente satisfacible, y sea  $\varphi$  cualquier fórmula, entonces o bien  $\Sigma \cup \{\varphi\}$  es finitamente satisfacible o  $\Sigma \cup \{\neg\varphi\}$  lo es.

---

**Demostración:**

Supongamos que no, es decir que tanto  $\Sigma \cup \{\varphi\}$  como  $\Sigma \cup \{\neg\varphi\}$  no son finitamente satisfacibles, por lo cual existen  $\Delta_1, \Delta_2 \subseteq \Sigma$  finitos tales que  $\Delta_1 \cup \{\varphi\}$  y  $\Delta_2 \cup \{\neg\varphi\}$  no son satisfacibles. Entonces  $\Delta_1 \cup \Delta_2$  no puede ser satisfacible. ■