Curso de Lógica Matemática

Cristo Daniel Alvarado

20 de febrero de 2024

Índice general

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 ($\wedge$), conjunción ($\vee$), 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 $\neg(P \Rightarrow \neq Q)$.

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 $p_1, p_2, ..., p_n, ...$ (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 φ, ψ 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 $\Rightarrow p_2p_{25}$ son FBF. Las primeras dos son llamadas **variables aisladas**. También lo es $\neg \Rightarrow p_2p_{25}$ (en este ejemplo, los p_i son variables).

Pero, por ejemplo $\Rightarrow \neg p_1p_2p_3$ 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 opeadores 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 lógicos 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 $m : \mathrm{Var} \to \{V, F\}$, donde 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 $\overline{m} : \mathrm{FBF} \to \{V, F\}$, donde FBF denota al conjunto de las fórmulas bien formadas, tal que $\overline{m}(\neg\varphi) = V \iff \overline{m}(\varphi) = F$ y $\overline{m}(\neg\varphi\psi) = F \iff \overline{m}(\varphi) = V$ y $\overline{m}(\psi) = F$.

Definición 1.2.2
Sea m un modelo, φ una fórmula y Σ un cojunto de fórmulas. Definimos que

1. $m \vDash \varphi$ (m satisface φ) si $\overline{m}(\varphi) = V$.
2. $m \vDash \Sigma$ si $m \vDash \varphi$ para cada φ elemento de Σ.

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 $m \nvDash \Rightarrow p_1p_3$, pero $m \vDash \neg p_5$.

Definición 1.2.3
Decimos que una fórmula φ es:

1. **Satisfacible** si existe un modelo m tal que $m \vDash \varphi$.
2. **Contradictoria** si todo modelo cumple que $m \nvDash \varphi$.
3. **Una tautología** si todo modelo m cumple que $m \vDash \varphi$.

Ejemplo 1.2.2
Tomemos de ejemplo a $\Rightarrow p_1 p_2$. cualquier modelo que haga a p_1 y p_3 verdaderas, o ambas falsas satisfacen la FBF, p_1, $\neg \Rightarrow p_1 p_3$ 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 Σ un conjunto de fórmulas. Decimos que Σ es

1. **Satisfacible** si existe un modelo m tal que $m \vDash \Sigma$.
2. **Contradictoria** si todo modelo cumple que $m \nvDash \Sigma$.
3. **Una tautología** si todo modelo m cumple que $m \vDash \Sigma$.

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. φ es satisfacible $\iff$ $\neg\varphi$ es una contradicción.
3. Satisfacible es lo mismo que no contradictoria.

Definición 1.2.5
Si Σ es un conjunto de FBF y φ es alguna otra fórmula, entonces decimos que φ es **consecuencia lógica** de Σ, o que Σ **implica lógicamente** a φ, escrito como $\Sigma \vDash \varphi$, si para todo modelo m tal que $m \vDash \Sigma$ se tiene que $m \vDash \varphi$.

Ejemplo 1.2.4
El conjunto de FBF $\{\Rightarrow p_1 p_2, p_1\} \vDash p_2$.

Observación 1.2.2
Se tiene lo siguiente:

1. Un conjunto de FBF $\Sigma \nvDash \varphi$ si y sólo si $\Sigma \cup \{\neg\varphi\}$ es satisfacible.
2. Además, un conjunto de FBF $\Sigma \vDash \varphi$ si y sólo si $\Sigma \cup \{\neg\varphi\}$ no es satisfacible.

Lema 1.2.1
Sea Σ un conjunto de fórmulas y sean $\mathrm{Var}(\Sigma)$ el conjunto de las variables p_i que aparecen en las fórmulas de Σ. Si m_1 y m_2 son dos modelos tales que

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

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

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

Demostración:

Sin pérdida de generalidad, Σ es cerrado bajo subformulas.

Procederemos por inducción sobre $\varphi \in \Sigma$, demostraremos que $\overline{m_1}(\varphi) = \overline{m_2}(\varphi)$. Si φ 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 φ 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 φ es de la forma $\Rightarrow \varphi_1\psi$ y, supontemos 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 Σ es un conjunto finito de fórmulas, entonces se puede verificar 'Mecánicamente' si es el caso, que $\Sigma \vDash \varphi$.

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

Definición 1.2.6
Decimos que un conjunto de fórmulas bien formadas Σ es **finitamente satisfacible** si cualquier subconjunto finito $\Delta \subseteq \Sigma$ es satisfacible.

Teorema 1.2.2 (Teorema de Compacidad de Gödel)
Si Σ es un conjunto (arbitrario) de fórmulas tal que $\Sigma \vDash \varphi$, entonces existe un $\Delta \subseteq \Sigma$ finito tal que $\Delta \vDash \varphi$.

El teorema que Gödel probó originalmente fue este:

Teorema 1.2.3 (Teorema de Gödel)
Un conjunto de fórmulas Σ 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 $\Sigma \nvDash \varphi \iff$ existe un modelo m tal que $m \vDash \Sigma \cup \{\neg\varphi\}$. 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 $m \vDash \Delta \cup \{\neg\varphi\}$, si y sólo si para todo $\Delta \subseteq \Sigma$ finito $\Delta \nvDash \varphi$, con lo cual

$$\Sigma \nvDash \varphi \iff \Delta \nvDash \varphi$$

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

Lema 1.2.2
Sea Σ un conjunto finitamente satisfacible, y sea φ 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, pues si m es un modelo tal que $m \vDash \Delta_1 \cup \Delta_2$, entonces $m \vDash \varphi$ contradice el hecho de que $\Delta_1 \cup \{\varphi\}$ es no satisfacible y si $m \vDash \neg\varphi$ contradice el hecho de que $\Delta_2 \cup \{\neg\varphi\}$ no es satisfacible, siendo $\Delta_1 \cup \Delta_2 \subseteq \Sigma$, se contradice el hecho de que Σ es finitamente satisfacible$\#_c$. Luego se tiene el resultado. ■

Ahora procederemos a probar el teorema de Gödel.

Demostración:
Se probará la doble implicación:

$\Rightarrow$): Es inmediato.

$\Leftarrow$): Sean $\varphi_1, \varphi_2, ...$ una enumeración 'efectiva' de todas las fórmulas (checar la observación). Recursivamente, definimos conjuntos de fórmulas $\Sigma_0 \subseteq \Sigma_1 \subseteq \cdots$ tales que $\Sigma_0 = \Sigma$, y

1. Cada Σ_n es finitamente satisfacible.

2. Para cada $n \in \mathbb{N}$, o bien $\varphi_n \in \Sigma_{n+1}$ o bien $\neg\varphi_n \in \Sigma_{n+1}$

en este contexto, definimos:

$$\Sigma_{n+1} = \begin{cases} \Sigma_n \cup \{\varphi_n\} & \text{si este conjunto es finitamente satisfacible} \\ \Sigma_n \cup \{\neg\varphi_n\} & \text{en caso contrario} \end{cases}$$

Esta definición es consistente con la recursión por el lema anterior.

Ahora, definimos $\Sigma_\infty = \bigcup_{n\in\mathbb{N}} \Sigma_n$. Analicemos a este conjunto.

1. Σ_∞ **es finitamente satisfacible**. En efecto, sea $\Delta \subseteq \Sigma$ un subconjunto finito, entonces existe $n \in \mathbb{N}$ tal que $\Delta \subseteq \Sigma_n$, luego como Σ_n es finitamente satisfacible, Δ es satisfacible. Por lo cual Σ_∞ es finitamente satisfacible.

2. **Para cada fórmula ψ o bien $\psi \in \Sigma_\infty$ ó $\neg\psi \in \Sigma_\infty$ y no ambas.** Esto es inmediato con la enumeración efectiva de todas las fórmulas bien formadas.

3. Σ_∞ **es maximal finitamente satisfacible**.

Sea $m : \text{Var}(\Sigma_\infty) \to \{V, F\}$, dado por $m(p_n) = V$ si y sólo si $p_n \in \Sigma_\infty$.

Se probará el siguiente lema:

Lema 1.2.3
Para cualquier fórmula ψ, $\overline{m}(\psi) = V$ si y sólo si $\psi \in \Sigma_\infty$ y $\overline{m}(\psi) = F$ si y sólo si $\neg\psi \in \Sigma_\infty$.

Demostración:
Procederemos por inducción sobre ψ.

- El caso base es inmediato por definición.

- $\overline{m}(\neg\psi) = V \iff \overline{m}(\psi) = F \iff \psi \notin \Sigma_\infty \iff \neg\psi \in \Sigma_\infty$.

- $\overline{m}(\Rightarrow \xi\psi) = F \iff \overline{m}(\xi) = F$ y $\overline{m}(\psi) = V \iff \neg\xi, \psi \in \Sigma_\infty$ si y sólo si $\Rightarrow \psi\xi \notin \Sigma_\infty$ (esto es cierto por la maximalidad de Σ_∞ al ser finitamente satisfacible).

por inducción se tiene lo deseado. ■

En conclusión, el modelo definido cumple que $m \vDash \psi$ si y sólo si $\psi \in \Sigma_\infty$. En particular, $m \vDash \Sigma$, y Σ es satisfacible.

■

Observación 1.2.3
Tuplas. Considere los números naturales. Podemos establecer una biyección entre las tuplas finitas de números naturales junto con el cero, y los números naturales, de esta forma:

Si $n \in \mathbb{N}$, por el TFA podemos expresar a $n = q_1^{\alpha_1} \cdot ... \cdot q_m^{\alpha_m}$. Establecemos la biyección dada como sigue: $n \mapsto (\alpha_1, ..., \alpha_{m-1}, \alpha_m - 1)$. De esta forma podemos enumerar algo con tuplas. Lo que Gödel hace es que hace ciertas asignaciones: $\neg = 0$, $\Rightarrow = 1$, $2 = p_1$, $3 = p_2$, etc... Esta enumeración es llamada **enumeración de Gödel**.

Cuando decimos lo de enumeración, nos referimos a esto. Básicamente enumeramos a todas las fórmulas bien formadas. Cuando decimos que la enumeración es efectiva, hacemos referencia a que podemos hacerlo de forma mecánica.