Apuntes de Lógica Matemática

Eduardo González Vaquero
Saúl Rodríguez Martín
Miguel Muñoz Pérez

12 de febrero de 2020

Índice general

Capítulo 1

Lógica proposicional

1.1. Introducción

Antes de comenzar con definiciones rigurosas, conviene dar una breve explicación sobre la motivación de la lógica proposicional, a la que dedicaremos este capítulo. Un aspecto común de los lenguajes naturales (como el español o el chino) es la ambigüedad de ciertas expresiones, lo cual dificulta las intenciones usuales de la ciencia, a saber, la precisión en la comunicación y la distinción clara de aquello que se comunica. De aquí nace una necesidad evidente, que es cubierta por los lenguajes artificiales y, entre ellos, la lógica formal.

Debemos aclarar entonces cuál es nuestro objeto de estudio. Fijémonos en lo siguiente: existen ciertos elementos del lenguaje que no cambian si los traducimos a otro idioma, por ejemplo. Nos referimos a lo que se denomina como *conectivas lógicas*, es decir, 'no' (negación), 'o' (disyunción), 'y' (conjunción), 'si... entonces' (implicación material) y 'si y solo si' (equivalencia). Obviamente, podemos usar múltiples expresiones distintas para referirnos a cada una de ellas, pero su función en el lenguaje natural no cambia. Por esto mismo podemos formalizarlas y estudiarlas desde el punto de vista matemático.

Ahora bien, a fin de sistematizar nuestro lenguaje, introducimos una serie de símbolos que se referirán a las entidades correspondientes del lenguaje natural. Conviene puntualizar de nuevo que estas traducciones, es decir, estas 'asignaciones de significados' *no* son únicas. Esto lo veremos más adelante. Entonces tenemos: $\neg$ (negación), $\vee$ (disyunción), $\wedge$ (conjunción), $\rightarrow$ (implicación material) y $\leftrightarrow$ (equivalencia).

Los otros elementos básicos que constituirán nuestro lenguaje formal son las *proposiciones*. Con el fin de no entrar en cuestiones semánticas que exceden nuestro estudio, establecemos la siguiente:

Definición 1.1. Una proposición es un enunciado que puede ser verdadero o falso.

Como con las conectivas, introducimos una serie de *símbolos de proposición*

para referirnos formalmente a las proposiciones. Denotamos por SP al conjunto de todos estos símbolos.

Así, el enunciado 'Todos los solteros son hombres no casados' es una proposición, mientras que '¿Va a llover mañana?' no lo es. Por otro lado, 'César conquistó las Galias y combatió contra Vercingétorix' admite una descomposición como conjunción de dos proposiciones. Evidentemente, esta reducción a proposiciones más simples tiene que parar eventualmente; así llegamos a lo que denominamos *proposiciones atómicas*. Esto nos permitirá, una vez determinada la verdad o falsedad de estas proposiciones atómicas, hallar la de todas las proposiciones que se compongan de ellas mediante la aplicación de las conectivas lógicas (antiguamente llamadas *proposiciones moleculares*).

Las proposiciones atómicas se representan entonces por los elementos de SP. Añadimos, por motivos que serán aclarados más adelante, dos elementos $\top$ y $\bot$, que se refieren a la *proposición idénticamente verdadera* y a la *proposición idénticamente falsa*, respectivamente[1]. Es decir,

Definición 1.2. Definimos el conjunto de las proposiciones atómicas como:

$$AT := SP \cup \{\top, \bot\}$$

Definición 1.3. (Alfabeto) Dado un conjunto de símbolos de proposiciones SP, se denomina alfabeto a:

$$A_{SP} = SP \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,), \top, \bot\},$$

obtenido añadiendo a SP las conectivas lógicas $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, y los paréntesis, $(,)$.

Definición 1.4. (Cierre de Kleene) Dado un conjunto de símbolos A, definimos el cierre de Kleene de A, A^*, como el conjunto de las concatenaciones de elementos de A junto con ϵ, el *espacio vacío* o en blanco.

Ejemplo 1.5. Dado $A = \{a, b\}$, le corresponde

$$A^* = \{\epsilon, a, b, ab, ba, aab, baa, aba, ...\}$$

Ejemplo 1.6. A^*_{SP} será el conjunto de todas las concatenaciones de elementos de A_{SP}, junto con ϵ.

Una vez que disponemos de los elementos básicos de la sintaxis del lenguaje y sus sucesivas combinaciones, nos interesa definir las expresiones que están 'bien formadas', es decir, queremos excluir de nuestro lenguaje concatenaciones de símbolos como '$)) \leftrightarrow \varphi$'.

Definición 1.7. (Proposiciones posibles) Se define el conjunto de proposiciones posibles, $PROP_{SP}$, como la intersección de todos los subconjuntos de A^*_{SP} que verifican:

[1] En realidad, como veremos, al igual que $\neg, \wedge, \vee, \rightarrow$ y $\leftrightarrow$ son conectivas binarias, $\top$ y $\bot$ pueden considerarse conectivas 0-arias.

1. $SP \subseteq PROP_{SP}$.

2. Si $\phi \in PROP_{SP}$, entonces $(\neg\phi) \in PROP_{SP}$.

3. Si $\phi_1, \phi_2 \in PROP_{SP}$ y $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, entonces $(\phi_1 \square\ \phi_2) \in PROP_{SP}$.

Es fácil comprobar que $PROP_{SP}$ también cumple estas tres propiedades, por tanto decimos que es el mínimo subconjunto que las verifica.

Proposición 1.8. *Se define $P_0 := SP$ y para cada $n \in \mathbb{N}$:*

$$P_n := P_{n-1} \cup \{(\neg\phi) : \phi \in P_{n-1}\} \cup \{(\phi_1 \square\ \phi_2) : \phi_1, \phi_2 \in P_{n-1}, \square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}\}.$$

Entonces $P := \bigcup_{n=0}^{\infty} P_n$ coincide con $PROP_{SP}$.

Demostración. Comenzamos comprobando $P \subseteq PROP_{SP}$. Para ello, comprobamos por inducción que $P_n \subseteq PROP_{SP}\ \forall n \in \mathbb{N}$.

Está claro que $P_0 \subseteq PROP_{SP}$, ya que $PROP_{SP}$ cumple la propiedad 1. Ahora, suponiendo que $P_n \subseteq PROP_{SP}$, por las propiedades 2 y 3, $\{(\neg\phi) : \phi \in P_{n-1}\} \cup \{(\phi_1 \square\ \phi_2) : \phi_1, \phi_2 \in P_{n-1}, \square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}\}$ también estarán contenidos en $PROP_{SP}$. Por tanto, $P_{n+1} \subseteq PROP_{SP}$, y hemos acabado la inducción.

Veamos ahora que $PROP_{SP} \subseteq P$. Para ello nos basta comprobar que P cumple las tres propiedades de la definición 1.7. La propiedad 1 es directa, pasamos a comprobar las propiedades 2 y 3.

Dado $\varphi \in P$, existe $n \in \mathbb{N} \cup \{0\}$ tal que $\varphi \in P_n$ luego $(\neg\varphi) \in P_{n+1}$ por lo que $(\neg\varphi) \in P$.

Sean $\varphi_1, \varphi_2 \in P$ y sea $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Tomamos $n \in \mathbb{N}$ tal que $\varphi_1, \varphi_2 \in P_n$ (para cada uno existe al menos un natural: basta tomar el mayor). Luego $(\phi_1 \square\ \phi_2) \in P_{n+1}$ y por lo tanto también pertenece a P.

□

A partir de ahora, usaremos para mayor brevedad el símbolo $\square$ como intercambiable por $\wedge, \vee, \rightarrow$ o $\leftrightarrow$.

1.2. Inducción estructural y recursión

El método de inducción nos servirá para demostrar que todas las fórmulas cumplen una *propiedad*[2] en dos pasos: demostrar que todas las proposiciones

[2]No hemos definido rigurosamente el concepto de "propiedad", pero esencialmente significa algo que las proposiciones pueden cumplir o no. Una forma rigurosa de definir propiedad consiste en definirla como un subconjunto $P \subseteq PROP_{SP}$, de modo que decimos que una proposición cumple la propiedad P si y solo si pertenece a P.

atómicas cumplen dicha propiedad, y demostrar que si dos proposiciones φ_1, φ_2 cumplen una propiedad, entonces $(\neg\varphi_1)$ y $(\varphi_1 \square \varphi_2)$ también la cumplen.

Proposición 1.9. *(Inducción estructural) Supongamos que queremos probar $P(\varphi)$ para todo $\varphi \in PROP_{SP}$ (siendo P una propiedad cualquiera) entonces basta comprobar que:*

1. *$P(\varphi)$ para toda $\varphi \in AT$.*
2. *Si $P(\varphi)$ entonces $P((\neg\varphi))$.*
3. *Si $P(\varphi_1)$ y $P(\varphi_2)$ entonces $P((\varphi_1 \square \varphi_2))$.*

Demostración. Si una propiedad P cumple estas tres propiedades, entonces el conjunto de proposiciones que cumplen P cumple las tres propiedades de 1.7, por tanto contiene a $PROP_{SP}$.

□

Ejemplo 1.10. Definamos $P(\varphi)$ como la propiedad de $|\varphi|_) = |\varphi|_($ siendo $|\varphi|_)$ la cantidad de) en la fórmula φ y $|\varphi|_($ la cantidad de (. Vamos a comprobar por inducción estructural que todas las proposiciones cumplen esta propiedad:

1. Para cualquier $\varphi \in AT$ se verifica trivialmente $|\varphi|_) = |\varphi|_(= 0$, luego $P(\varphi)$.
2. Supongamos $P(\varphi)$ con $\varphi \in PROP_{SP}$, entonces $|\varphi|_(= |\varphi|_)$ y por tanto:

   $$|(\neg\varphi)|_(= |\varphi|_(+ 1 = |\varphi|_) + 1 = |(\neg\varphi)|_)$$

   luego $P((\neg\varphi))$.
3. Supongamos $P(\varphi), P(\psi)$ con $\varphi, \psi \in PROP_{SP}$, entonces:

   $$|(\varphi\square\psi)|_(= |\varphi|_(+ |\psi|_(+ 1 = |\varphi|_) + |\psi|_) + 1 = |(\varphi\square\psi)|_)$$

   luego $P((\varphi\square\psi))$.

Definición 1.11. (Prefijos) Sea A un alfabeto y sea $w \in A^*$. Decimos que w' es prefijo de w si existe w'' tal que $w = w'w''$, es decir, si $w = w_1w_2...w_n$ entonces existe $0 \leq k \leq n$ tal que $w' = w_1...w_k$.
Decimos además que un prefijo w' es prefijo propio de w si es distinto de w y ϵ.

Ejemplo 1.12. Sea $A := \{a, b\}$ y sea $w := aababb$. Entonces sus prefijos son $\epsilon, a, aa, aab, aaba, aabab, aababb$, correspondientes a $k = 0, 1, 2, 3, 4, 5, 6$, respectivamente, y sus prefijos propios son $a, aa, aab, aaba, aabab$.

Proposición 1.13. *Si φ' es prefijo propio de φ, entonces $|\varphi'|_) < |\varphi'|_($.*

Demostración. Se demuestra por inducción estructural.

1. Si $\varphi \in AT$ no tiene prefijos propios luego se cumple la afirmación.

2. Sea $\varphi \in PROP_{SP}$ cumpliendo la propiedad del enunciado y sea φ' un prefijo propio de $(\neg\varphi)$. Existen cuatro casos:

   a) $\varphi' = ($, entonces $|\varphi'|_(= 1 > 0 = |\varphi'|_)$-

   b) $\varphi' = (\neg$, ocurre lo mismo que antes.

   c) $\varphi' = (\neg\varphi''$ con φ'' prefijo propio de φ entonces por inducción se verifica $|\varphi''|_) < |\varphi''|_($. Ahora bien, $|\varphi'|_(= |\varphi''|_(+ 1$ y $|\varphi'|_) = |\varphi''|_)$ luego:
   $$|\varphi'|_) < |\varphi'|_($$

   d) $\varphi' = (\neg\varphi$, que es directo.

3. Se hace de forma similar al apartado anterior. Basta probar con los prefijos $($, $(\varphi_1'$, $(\varphi_1$, $(\varphi_1\square$, $(\varphi_1\square\varphi_2'$ y $(\varphi_1\square\varphi_2$.

□

En breve nos será útil, dado un conjunto A, poder definir una función $H : PROP_{SP} \to A$ conociendo su restricción a AT, y suponiendo que podemos calcular $H((\neg\varphi))$ y $H((\varphi\square\psi))$ en función de $H(\varphi)$ y $H(\psi)$. Si la función H queda bien definida así, diremos que está *definida de forma recursiva*. La siguiente proposición nos garantiza que podemos definir funciones de esta forma:

Proposición 1.14. *El esquema de definición recursiva da como resultado una única función, es decir, dadas:*

1. $H_{AT} : AT \to A$.
2. $H_\neg : A \to A$.
3. $H_\square : A \times A \to A$.

existe una única función $H : PROP_{SP} \to A$ *tal que:*

1. $H(\varphi) = H_{AT}(\varphi)$, *para toda* $\varphi \in AT$.
2. $H((\neg\varphi)) = H_\neg(H(\varphi))$.
3. $H((\varphi_1\square\varphi_2)) = H_\square(H(\varphi_1), H(\varphi_2))$, *para cada* $\square \in \{\wedge, \vee, \to, \leftrightarrow\}$.

Demostración. Se omite la demostración. □

Ejemplo 1.15. Sea $H : PROP_{SP} \to \mathbb{N}$ la función $|.|_($ definida anteriormente. Podemos definirla de manera recursiva:

1. $H(\varphi) = 0$, para toda $\varphi \in AT$.
2. $H((\neg\varphi)) = H(\varphi) + 1$.
3. $H((\varphi_1\square\varphi_2)) = H(\varphi_1) + H(\varphi_2) + 1$.

Nótese como en ocasiones omitimos las funciones auxiliares y usamos la notación H para referirnos a H, H_{AT}, $H_{\neg}$ y $H_{\Box}$. Este abuso de notación no suele dar lugar a confusión.

Ejercicio 1.16. Definir resursivamente una función que nos lleve cada fórmula al número de veces que $\wedge$ aparece en ella.

Ejemplo 1.17. Podemos definir una función que nos lleve cada fórmula al conjunto de todas sus subfórmulas, es decir, $SUB : PROP_{SP} \to \mathcal{P}(PROP_{SP})$[3] tal que:

1. $SUB(\varphi) = \{\varphi\}$, para toda $\varphi \in AT$.
2. $SUB((\neg\varphi)) = SUB(\varphi) \cup \{(\neg\varphi)\}$.
3. $SUB((\varphi_1 \Box \varphi_2)) = SUB(\varphi_1) \cup SUB(\varphi_2) \cup \{(\varphi_1 \Box \varphi_2)\}$.

1.3. Eliminación de paréntesis

Para conseguir una notación más compacta, podemos establecer una notación que use menos paréntesis cuando nos convenga. Sin embargo, la omisión de paréntesis puede dar lugar a ambigüedades en las fórmulas, por ejemplo:

$$p \to q \to r$$

podría simbolizar dos fórmulas distintas:

- $((p \to q) \to r)$
- $(p \to (q \to r))$

Para evitar este tipo de situaciones, necesitamos las siguientes reglas de omisión de paréntesis:

1. Los paréntesis externos pueden omitirse. Esto no da lugar a ambigüedad.
2. Las conectivas se aplicarán en este orden: $\neg, \wedge, \vee, \to, \leftrightarrow$.
3. Cuando hay varias conectivas del mismo tipo seguidas, se asocia siempre por la izquierda[4].

Algunos ejemplos de aplicación de estas reglas:

$\neg p \vee p$	es	$((\neg p) \vee p)$
$\neg p \wedge q \vee s \to t \leftrightarrow r$	es	$(((((\neg p) \wedge q) \vee s) \to t) \leftrightarrow r)$
$p \to q \to s$	es	$((p \to q) \to s)$
$p \wedge q \leftrightarrow \neg s \to t \vee r$	es	$((p \wedge q) \leftrightarrow ((\neg s) \to (t \vee r)))$
$t \leftrightarrow p \to \neg q \vee p \to s$	es	$(t \leftrightarrow ((p \to ((\neg q) \vee p)) \to s))$

[3] Dado un conjunto X, denotamos por $\mathcal{P}(X)$ a su potencia, es decir, el conjunto de subconjuntos de X.

[4] Cuando veamos semántica, comprobaremos que todas las conectivas salvo $\to$ son asociativas, por tanto, en ese sentido, no importa el orden en que se asocien estas conectivas.

1.4. Asignaciones de verdad

Hasta ahora hemos venido considerando cuestiones sintácticas sobre las fórmulas. Lo que queremos ahora es abordar su semántica, esto es, su comportamiento cuando les damos una interpretación determinada. Nos interesa, por tanto, estudiar qué significa que una proposición se siga de otra independientemente de la traducción que hagamos al lenguaje natural, esto es, la noción de *consecuencia lógica*. Así, por ejemplo, queremos que q se siga de $p \wedge (p \to q)$ independientemente de cómo se traduzcan p y q informalmente. Es decir, exigimos que, sea cual sea la traducción de $p \wedge (p \to q)$, si es verdadera entonces la de q tiene que serlo también. Siguiendo la definición 1.1,

Definición 1.18. Definimos el conjunto de valores de verdad, $Bool = \{V, F\}$, donde V es *la verdad* y F es *la falsedad*[5].

Una vez que disponemos de $Bool$ podemos definir la aplicación que nos lleva cada proposición a su valor de verdad correspondiente. Vamos a definir esta aplicación de forma recursiva, es decir, primero lo haremos para las fórmulas atómicas y después describiremos reglas para deducirla en el resto de fórmulas.

Definición 1.19. (Valoración) Sea el conjunto $Bool = \{V, F\}$. Una valoración o evaluación es una función $v : SP \to Bool$.

Definición 1.20. (Extensión) Se definen las siguientes aplicaciones:

1. $v_\neg : Bool \to Bool$
2. $v_\wedge, v_\vee, v_\to, v_\leftrightarrow : Bool \times Bool \to Bool$.

Definimos estas funciones caso a caso en formato de tabla de verdad:

φ	$v_\neg(\varphi)$
V	F
F	V

φ_1	φ_2	$v_\wedge(\varphi_1, \varphi_2)$	$v_\vee(\varphi_1, \varphi_2)$	$v_\to(\varphi_1, \varphi_2)$	$v_\leftrightarrow(\varphi_1, \varphi_2)$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Dada la valoración $v : SP \to Bool$, definimos recursivamente su *extensión*, $\hat{v} : PROP_{SP} \to Bool$, como:

[5]Nada nos impide considerar casos en que $Bool$ sea infinito numerable, o incluso no numerable, pero la lógica concebida clásicamente es bivalente, esto es, solo consta de dos valores de verdad.

1. $\hat{v}(\bot) = F$.
2. $\hat{v}(\top) = V$.
3. $\hat{v}(\varphi) = v(\varphi)$, para toda $\varphi \in SP$.
4. $\hat{v}((\neg\varphi)) = v_{\neg}(\hat{v}(\varphi))$.
5. $\hat{v}((\varphi_1 \Box \varphi_2)) = v_{\Box}(\hat{v}(\varphi_1), \hat{v}(\varphi_2))$.

Nótese que aquí es donde hemos dado sentido a la definición que dimos de $\top$ y $\bot$ en 1.2.

Definición 1.21. (Satisfacibilidad) Dada $v : SP \to Bool$ y $\varphi \in PROP_{SP}$ se dice que v satisface φ, y se escribe como $v \vDash \varphi$, si y solo si $\hat{v}(\varphi) = V$. Si $\hat{v}(\varphi) = F$, escribimos $v \nvDash \varphi$.

Definición 1.22. φ se dice *satisfacible* si existe alguna valoración v tal que $v \vDash \varphi$. Si $v \vDash \varphi$ para toda valoración v, se dice que φ es una *tautología*. Se dice que φ es una *contingencia* si es satisfacible pero no tautología. φ es *contradicción* si no es satisfacible.

Ejemplo 1.23. La fórmula $p \wedge q \to r \leftrightarrow p \to q \to r$ es contingencia. Invitamos al lector a que realice la tabla de verdad correspondiente.

Definición 1.24. (Conjunto de modelos) Dado $\Phi \subseteq PROP_{SP}$, el conjunto de modelos de Φ se define como:

$$Mod(\Phi) := \{v : SP \to Bool \mid \forall \varphi \in \Phi \; v \vDash \varphi\}$$

Definición 1.25. (Satisfacibilidad) El conjunto $\Phi \subset PROP_{SP}$ se dice satisfacible, y se escribe $Sat(\Phi)$, si $Mod(\Phi) \neq \emptyset$.

Definición 1.26. (Insatisfacibilidad) El conjunto $\Phi \subset PROP_{SP}$ se dice insatisfacible, y se escribe $Insat(\Phi)$, si $Mod(\Phi) = \emptyset$.

Ahora llegamos a la definición del concepto que ha motivado nuestro estudio inicial de la semántica.

Definición 1.27. (Consecuencia lógica) Sea $\varphi \in PROP_{SP}$. Decimos que φ es consecuencia lógica de Φ, $\Phi \vDash \varphi$, si y solo si $Mod(\Phi) \subseteq Mod(\{\varphi\})$.

Esto equivale a decir que toda asignación de verdad que satisface todos los elementos de Φ también satisface φ.

De ahora en adelante, escribimos $Mod(\varphi)$ en vez de $Mod(\{\varphi\})$.

Ejemplo 1.28. Dado Φ tal que $Insat(\Phi)$, se sigue de la anterior definición que $\Phi \vDash \varphi$, para cualquier φ.

Ejercicio 1.29. Demostrar que si $\Phi = \emptyset$ y $\Phi \vDash \varphi$, entonces φ es tautología. (Indicación: ¡los elementos del conjunto vacío verifican cualquier cosa!)

Proposición 1.30. *$\Phi \cup \{\varphi\} \vDash \psi$ si y solo si $\Phi \vDash \varphi \to \psi$.*

Demostración. Veamos que $Mod(\Phi) \subset Mod(\varphi \to \psi)$ suponiendo que $Mod(\Phi \cup \{\varphi\}) \subset Mod(\psi)$. Sea $v \in Mod(\Phi)$. Si $\hat{v}(\varphi) = V$ entonces $v \in Mod(\varphi)$. Por hipótesis, $v \in Mod(\psi)$, luego $\hat{v}(\psi) = V$ y entonces $\hat{v}(\varphi \to \psi) = V$, es decir, $v \in Mod(\varphi \to \psi)$. Si $\hat{v}(\varphi) = F$, $\hat{v}(\varphi \to \psi) = V$ y de nuevo $v \in Mod(\varphi \to \psi)$.

Recíprocamente, si $v \in Mod(\Phi \cup \{\varphi\})$, entonces $\hat{v}(\varphi) = V$. Por hipótesis, al ser $v \in Mod(\Phi)$, $v \in Mod(\varphi \to \psi)$, es decir, $\hat{v}(\varphi \to \psi) = V$, luego $\hat{v}(\psi) = V$ y $v \in Mod(\psi)$. □

Proposición 1.31. *$\Phi \vDash \varphi$ si y solo si $Insat(\Phi \cup \{\neg\varphi\})$*

Demostración. Supongamos que $\Phi \vDash \varphi$, esto es, que $Mod(\Phi) \subset Mod(\varphi)$. Sea $v \in Mod(\Phi \cup \{\neg\varphi\})$. Entonces, por definición, $v \vDash \chi$, para toda $\chi \in \Phi \cup \{\neg\varphi\}$. Luego en especial, $v \vDash \neg\varphi$, lo que contradice que $v \in Mod(\varphi)$ (¿por qué?).

Recíprocamente, si $Mod(\Phi \cup \{\neg\varphi\}) = \emptyset$, sea $v \in Mod(\Phi)$. Entonces $v \notin Mod(\neg\varphi)$, por tanto $\hat{v}(\neg\varphi) = F$, pero como $\hat{v}(\neg\varphi) = v_\neg(\hat{v}(\varphi))$, tenemos que $F = v_\neg(\hat{v}(\varphi))$, es decir, $\hat{v}(\varphi) = V$, así que $v \in Mod(\varphi)$. Como esto lo cumple cualquier $v \in Mod(\Phi)$, tenemos, como queríamos, que $Mod(\Phi) \subseteq Mod(\varphi)$
□

De ahora en adelante, escribiremos $\Phi, \varphi \vDash \psi$ en vez de $\Phi \cup \{\varphi\} \vDash \psi$.

Dada una valoración v, para cualesquiera fórmulas φ_1, φ_2 en que aparecen los símbolos $p_1, \ldots, p_n$ y $q_1, \ldots, q_m$, respectivamente, está claro, por cómo hemos definido las valoraciones, que $\hat{v}(\varphi_1)$ depende de $v(p_1), \ldots, v(p_n)$, y $\hat{v}(\varphi_2)$ depende de $v(q_1), \ldots, v(q_m)$. De modo que para comprobar que φ_2 es consecuencia lógica de φ_1 nos basta con comprobar que, para cada una de las posibles valoraciones de $p_1, \ldots, p_n, q_1, \ldots, q_m$, $\hat{v}(\varphi_2) = V$ si $\hat{v}(\varphi_1) = V$. Este es nuestro primer procedimiento para demostrar equivalencias lógicas entre fórmulas, que resumiremos gráficamente en tablas de verdad.

Ejemplo 1.32. Consideremos $\Phi = \{\neg p \wedge q \to r,\ \neg p,\ \neg r\}$, $\varphi = \neg q$. Para demostrar que $\Phi \vDash \varphi$ basta realizar la tabla de verdad correspondiente y comparar los valores de verdad de las premisas del conjunto Φ con los de φ. Invitamos al lector a que verifique los detalles de esta afirmación.

1.5. Equivalencia lógica y sustitución

El método descrito de las tablas de verdad aumenta de complejidad a medida que lo hace el número de proposiciones que estudiamos. Por ello, investigamos relaciones formales entre las fórmulas que simplifiquen tales cuestiones.

Definición 1.33. (Equivalencia lógica) Sean $\varphi, \psi \in PROP_{SP}$. φ se dice lógicamente equivalente a ψ, $\varphi \sim \psi$, si para toda valoración v, $\hat{v}(\varphi) = \hat{v}(\psi)$.

Proposición 1.34. *$\sim$ es relación de equivalencia. Además, $\sim$ es congruencia respecto de las conectivas lógicas, es decir:*

1. *Si $\varphi \sim \psi$, entonces $\neg\varphi \sim \neg\psi$.*
2. *Si $\varphi_1 \sim \varphi_2$ y $\psi_1 \sim \psi_2$, entonces $\varphi_1 \square \psi_1 \sim \varphi_2 \square \psi_2$.*

Demostración. Que es relación de equivalencia es directo.
Para la segunda parte, solo es necesaria una comprobación directa de los valores de verdad. □

Las siguientes reglas nos ayudarán a demostrar equivalencias entre fórmulas:

Teorema 1.35. *(Leyes de equivalencia lógica)*

1. *Conmutatividad:*
$$p \wedge q \sim q \wedge p, \quad p \vee q \sim q \vee p, \quad p \leftrightarrow q \sim q \leftrightarrow p$$
2. *Asociatividad:*
$$(p \vee q) \vee r \sim p \vee (q \vee r), \quad (p \wedge q) \wedge r \sim p \wedge (q \wedge r)$$
3. *Idempotencia:*
$$p \wedge p \sim p, \quad p \vee p \sim p$$
4. *Distributiva:*
$$p \wedge (q \vee r) \sim (p \wedge q) \vee (p \wedge r), \quad p \vee (q \wedge r) \sim (p \vee q) \wedge (p \vee r)$$
5. *Absorción:*
$$p \wedge (p \vee q) \sim p, \quad p \vee (p \wedge q) \sim p$$
6. *Cero y unidad:*
$$p \wedge \top \sim p, \quad p \vee \top \sim \top, \quad p \wedge \bot \sim \bot, \quad p \vee \bot \sim p$$
7. *Contradicción:*
$$p \wedge \neg p \sim \bot$$
8. *Tercio excluso:*
$$p \vee \neg p \sim \top$$
9. *Negación:*
$$\neg\neg p \sim p$$
10. *Leyes de De Morgan:*
$$\neg(p \wedge q) \sim \neg p \vee \neg q, \quad \neg(p \vee q) \sim \neg p \wedge \neg q$$
11. *Simplificación de los condicionales:*
$$p \to q \sim \neg p \vee q, \quad p \leftrightarrow q \sim (p \to q) \wedge (q \to p)$$

Demostración. Las demostraciones de equivalencia consisten simplemente en comprobar que coinciden las tablas de verdad. Comprobamos las tres primeras:

1. Conmutatividad:

p	q	$p \wedge q$	$q \wedge p$	$p \vee q$	$q \vee p$	$p \leftrightarrow q$	$q \leftrightarrow p$
V	V	V	V	V	V	V	V
V	F	F	F	V	V	F	F
F	V	F	F	V	V	F	F
F	F	F	F	F	F	V	V

2. Asociatividad:

p	q	r	$(p \vee q) \vee r$	$p \vee (q \vee r)$	$(p \wedge q) \wedge r$	$p \wedge (q \wedge r)$
V	V	V	V	V	V	V
V	V	F	V	V	F	F
V	F	V	V	V	F	F
V	F	F	V	V	F	F
F	V	V	V	V	F	F
F	V	F	V	V	F	F
F	F	V	V	V	F	F
F	F	F	F	F	F	F

3. Idempotencia:

p	$p \wedge p$	$p \vee p$
V	V	V
F	F	F

□

Definición 1.36. (Sustitución) Sean $\varphi, \psi \in PROP_{SP}$ y sea $p \in SP$. Llamamos $\psi[\varphi/p]$ a la fórmula que resulta de sustituir en ψ cada aparición de p por φ. Dados φ y p, podemos definir $\psi[\varphi/p]$ recursivamente del siguiente modo:

1. $p[\varphi/p] = \varphi$.
2. $q[\varphi/p] = q$, para cualquier $q \in SP$ distinta de p.
3. $\top[\varphi/p] = \top$.
4. $\bot[\varphi/p] = \bot$.
5. $(\neg\psi)[\varphi/p] = \neg(\psi[\varphi/p])$.
6. $(\psi_1 \square \psi_2)[\varphi/p] = (\psi_1[\varphi/p] \square \psi_2[\varphi/p])$.

Introducimos la siguiente notación que será de gran importancia: dada una función $f : A \to B$ y dados $a \in A$, $b \in B$, denotamos respectivamente por $f[b/a](x)$ a $f(x)$, si $x \neq a$, y a b, si $x = a$.

Ejemplo 1.37. Sea $X \in \{V, F\}$. Si con la notación anterior usamos como función una valoración v, $v[X/p]$ es una valoración de verdad que cumple:

- $v[X/p](q) = v(q)$ si $q \neq p$
- $v[X/p](p) = X$.

Proposición 1.38. *Sean $\varphi, \psi \in PROP_{SP}$, $p \in SP$ y $v : SP \to Bool$ valoración. Entonces:*

$$\hat{v}(\psi[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi)$$

Demostración. Procedemos por inducción estructural. Dividimos las proposiciones atómicas en cuatro casos como en la definición 1.36:

1. Si $\psi = p$ entonces, por definición de sustitución $\psi[\varphi/p] = \varphi$. Por tanto, $\hat{v}(\psi[\varphi/p]) = \hat{v}(\varphi)$ y siguiendo la notación que hemos explicado arriba, $\widehat{v[\hat{v}(\varphi)/p]}(\psi) = \widehat{v[\hat{v}(\varphi)/p]}(p) = v[\hat{v}(\varphi)/p](p) = \hat{v}(\varphi)$.
2. Si $\psi = q \in SP$ y $q \neq p$ entonces $\psi[\varphi/p] = q$, con lo que $\hat{v}(\psi[\varphi/p]) = \hat{v}(q)$ y, como antes, $\widehat{v[\hat{v}(\varphi)/p]}(q) = v[\hat{v}(\varphi)/p](q) = \hat{v}(q)$.
3. Si $\psi = \top$, $\psi[\varphi/p] = \top$, y $\widehat{v[\hat{v}(\varphi)/p]}(\top) = v(\top) = V$.
4. Si $\psi = \bot$, $\psi[\varphi/p] = \bot$, y $\widehat{v[\hat{v}(\varphi)/p]}(\bot) = v(\bot) = F$.
5. Si $\psi = \neg\psi_1$, entonces $\hat{v}((\neg\psi_1)[\varphi/p]) = \hat{v}(\neg(\psi_1[\varphi/p])) = v_\neg(\hat{v}(\psi_1[\varphi/p]))$, por las definiciones de sustitución y extensión de la valoración v. Aplicando la hipótesis de inducción, lo anterior es igual a

   $$v_\neg(\widehat{v[\hat{v}(\varphi)/p]})(\psi_1) = \widehat{v[\hat{v}(\varphi)/p]}(\neg\psi_1).$$

6. Si $\psi = \psi_1 \square \psi_2$, entonces

   $$\hat{v}(\psi[\varphi/p]) = \hat{v}((\psi_1[\varphi/p] \ \square \ \psi_2[\varphi/p])) = v_\square(\hat{v}(\psi_1[\varphi/p]),\ \hat{v}(\psi_2[\varphi/p])).$$

   Aplicando la hipótesis de inducción, esto es igual a

   $$v_\square(\widehat{v[\hat{v}(\varphi)/p]}(\psi_1),\ \widehat{v[\hat{v}(\varphi)/p]}(\psi_2)) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_1 \square \psi_2).$$

□

Lema 1.39. *(De sustitución) Si $\varphi \sim \varphi'$, $\psi[\varphi/p] \sim \psi[\varphi'/p]$.*

Demostración. Sea v valoración. Entonces, al ser $\hat{v}(\varphi) = \hat{v}(\varphi')$, se obtiene que

$$\hat{v}(\psi[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi) = \widehat{v[\hat{v}(\varphi')/p]}(\psi) = \hat{v}(\psi[\varphi'/p])$$

Como esto sucede para cualquier valoración v, $\psi[\varphi/p] \sim \psi[\varphi'/p]$. □

Acabamos de ver que sustituir fórmulas equivalentes en una misma fórmula nos lleva a fórmulas equivalentes. A continuación vemos que sustituir la misma fórmula en fórmulas equivalentes da como resultado fórmulas equivalentes.

Lema 1.40. *(De sustitución) Sean $\varphi, \psi \in PROP_{SP}$, $p \in SP$. Si $\psi_1 \sim \psi_2$, entonces $\psi_1[\varphi/p] \sim \psi_2[\varphi/p]$.*

Demostración. Sea v cualquier valoración. Tenemos que:

$$\hat{v}(\psi_1[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_1) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_2) = \hat{v}(\psi_2[\varphi/p])$$

Lo que nos da el resultado. □

Los dos lemas de sustitución nos permiten extender a todas las proposiciones posibles las equivalencias lógicas que ya conocemos para proposiciones atómicas, tal y como muestra el siguiente:

Ejemplo 1.41. Consideremos las dos fórmulas $\psi_1 = p \vee q$ y $\psi_2 = q \vee p$. Sabemos que ambas son equivalentes, luego si consideramos $\varphi, \chi \in PROP_{SP}$ tenemos que, por el lema 1.40, $\psi_1[\varphi/p]$ y $\psi_2[\varphi/p]$, es decir, $\varphi \vee q$ y $q \vee \varphi$, son equivalentes. Aplicando una vez más 1.40 para sustituir q por χ, obtenemos que $\varphi \vee \chi$ y $\chi \vee \varphi$ son equivalentes.

1.6. Completitud funcional

En 1.20 definimos recursivamente una serie de asignaciones de verdad que correspondían a los símbolos de las conectivas lógicas, $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. Decimos que una conectiva es n-aria si se escribe con n símbolos de proposición, por ejemplo, las conectivas $\wedge, \vee, \rightarrow, \leftrightarrow$ son binarias (2-arias) y la conectiva $\neg$ es 1-aria.

En general, a cualquier tabla de verdad dada por $v_* : Bool^n \rightarrow Bool$ podemos asignarle un símbolo de conectiva n-aria, $*(p_1, \ldots, p_n)$. Según esta definición, podemos interpretar los símbolos $\top, \bot$ como conectivas 0-arias, dadas por las tablas de verdad:

$\top$	$\bot$
V	F

Nótese que debemos distinguir cuidadosamente entre la sintaxis de una fórmula y su semántica, es decir, en $*(p_1, ..., p_n)$ no tenemos una función sobre las proposiciones $p_1, ..., p_n$ sino sobre sus valores de verdad.

Analizando con mayor detenimiento las conectivas binarias, es fácil observar

que existen 16 de ellas distintas, una asociada a cada tabla de verdad[6]. Por ejemplo, podemos definir mediante tablas de verdad las conectivas:

p	q	$p \veebar q$	$p \uparrow q$	$p \downarrow q$
V	V	F	F	F
V	F	V	V	F
F	V	V	V	F
F	F	F	V	V

$\veebar$ se conoce como *disyunción excluyente*, mientras que $\uparrow$ y $\downarrow$ se llaman *NAND* y *NOR*, respectivamente.

Sin embargo, en cierto sentido, estas nuevas conectivas son redundantes, ya que podemos expresarlas en función de nuestras cinco conectivas iniciales:

- $p \veebar q \sim \neg(p \leftrightarrow q)$
- $p \uparrow q \sim \neg(p \wedge q)$
- $p \downarrow q \sim \neg(p \vee q)$

Es natural, por tanto, preguntarse si cualquier conectiva n-aria $*$ será expresable en función de nuestras conectivas, como lo son $\veebar, \downarrow$ y $\uparrow$. Para formalizar esta cuestión, introducimos dos nociones de gran importancia:

Definición 1.42. Una conectiva n-aria $*$ dada por $v_* : Bool^n \to Bool$ se dice expresable en términos de un conjunto de conectivas C si existe una fórmula φ que tiene por conectivas a elementos de C y por símbolos de proposición a $p_1, ..., p_n$, tal que $\varphi \sim *(p_1, ..., p_n)$.

Definición 1.43. (Completitud funcional) Un conjunto de conectivas C se dice funcionalmente completo si cualquier conectiva $*$ dada por $v_* : Bool^n \to Bool$ es expresable en términos de C.

Ahora ya podemos dar respuesta a la pregunta que nos formulamos antes:

Proposición 1.44. $\{\neg, \wedge, \vee\}$ *es funcionalmente completo.*

Demostración. Sea $*$ conectiva n-aria. Procedamos por inducción sobre n.

Si $n = 1$, es fácil comprobar que tenemos cuatro conectivas posibles, y que éstas corresponden a la identidad (es decir, $*(p) \sim p$), $\neg$, $\top \sim p \vee \neg p$ y $\bot \sim p \wedge \neg p$, con lo que se verifica el enunciado.

Supongamos que es válido para $n - 1$ y veámoslo para n. Sea $*(p_1, ..., p_n)$ una conectiva dada por la siguiente tabla de verdad:

[6]Decimos que dos conectivas n-arias son distintas cuando tienen distinta tabla de verdad. Para cada n, hay 2^{2^n} conectivas n-arias distintas.

p_n	p_1	...	p_{n-1}	$*(p_1, ..., p_n)$
V	...	...	...	X(1)
...	...	...	...	...
V	...	...	...	$X(2^{n-1})$
F	...	...	...	$X(2^{n-1}+1)$
...	...	...	...	...
F	...	...	...	$X(2^n)$

donde $X(i)$ denota el valor de verdad correspondiente a la fila i-ésima. De esta forma obtenemos dos tablas de verdad, cada una con 2^{n-1} filas: aquella en la que p_n toma V como valor de verdad (las primeras 2^{n-1} filas de la tabla anterior) y aquella en la que toma el valor F (las filas restantes). Llamamos $*_1(p_1, ..., p_{n-1})$ y $*_2(p_1, ..., p_{n-1})$ a las conectivas inducidas por estas dos tablas de verdad.

Entonces tenemos:

$$*(p_1, ..., p_n) \sim (p_n \wedge *_1(p_1, ..., p_{n-1})) \vee (\neg p_n \wedge *_2(p_1, ..., p_{n-1})).$$

Aplicando la hipótesis de inducción, existen fórmulas φ_1, φ_2 construidas a partir de $\{\neg, \wedge, \vee\}$ tales que $\varphi_1 \sim *_1(p_1, ..., p_{n-1})$ y $\varphi_2 \sim *_2(p_1, ..., p_{n-1})$, con lo que obtenemos que $*(p_1, ..., p_n)$ es expresable en términos de $\{\neg, \wedge, \vee\}$. □

Corolario 1.45. *1.* $\{\neg, \wedge\}$ *es funcionalmente completo.*

2. $\{\neg, \vee\}$ *es funcionalmente completo.*

3. $\{\rightarrow, \bot\}$ *es funcionalmente completo.*

4. $\{\uparrow\}$ *es funcionalmente completo.*

5. $\{\downarrow\}$ *es funcionalmente completo.*

Demostración.

1. Sabemos, por 1.44, que $\{\neg, \wedge, \vee\}$ es funcionalmente completo. Basta ver, entonces, que $\vee$ es expresable en términos de $\neg$ y $\wedge$. Esto es evidente por las leyes de De Morgan (1.35):

$$p \vee q \sim \neg(\neg p \wedge \neg q).$$

2. Análogo al anterior, usando la otra ley de De Morgan:

$$p \wedge q \sim \neg(\neg p \vee \neg q).$$

3. Basta ver que $\neg, \vee$ son expresables en términos de $\rightarrow, \bot$. Esto se sigue de estas afirmaciones que se pueden comprobar con tablas de verdad:

$$\neg p \sim p \rightarrow \bot.$$

$$p \vee q \sim (p \rightarrow \bot) \rightarrow q.$$

4. Expresamos $\neg, \wedge$ en términos de $\uparrow$:

$$\neg p \sim p \uparrow p.$$

$$p \wedge q \sim (p \uparrow q) \uparrow (p \uparrow q).$$

5. Expresamos $\neg, \vee$ en términos de $\downarrow$:

$$\neg p \sim p \downarrow p$$

$$p \vee q \sim (p \downarrow q) \downarrow (p \downarrow q).$$

□