Universidad Complutense de Madrid

Facultad de Ciencias Matemáticas

Lógica Matemática

Curso académico 2019-2020

Autor: Javier López

Versión Febrero 2020

Fe de errores

Este texto está sacado íntegramente de los apuntes que he tomado en clase. Por ello, es más que probable, encontrar en él erratas y errores. Todos ellos pueden ser comunicados a través del foro del campus virtual de la asignatura y serán subsanados lo antes posible o, en su defecto, añadidos a esta lista para que se tengan en cuenta.

- En el *ejemplo III, el apartado* ($\square$) no lo tengo escrito en mis apuntes y por tanto la solución no es de Luis.
- Recomiendo revisar el *ejemplo V, caso 4*.

Índice

1. Lógica de proposiciones

Definición 1. Diremos que una **proposición** es un enunciado que puede ser verdadero o falso. Nunca será una proposición cualquier enunciado que expresa duda o sentimientos. Tampoco lo serán aquellos enunciados que no tengan sentido lógico.

Un ejemplo de lo que no es proposición sería
p ≡ "Juan se cae"
q ≡ "Yo me río"
$\varphi \equiv$ Juan se cae y yo me río

Conectivas lógicas. Son los símbolos que utilizamos para formalizar las proposiciones. Estos son

$$\neg \rightsquigarrow \text{ negación} \quad \wedge \rightsquigarrow \text{ conjunción} \quad \vee \rightsquigarrow \text{ disyunción} \quad \rightarrow \rightsquigarrow \text{ implicación}$$

$$\leftrightarrow \rightsquigarrow \text{ implicación} \quad \bot \rightsquigarrow \text{ falso} \quad \top \rightsquigarrow \text{ cierto}$$

Definición 2. Se denomina **formalizar** una proposición, a escribirla mediante conectivas lógicas.

Ejemplo I: Formalizar las siguientes frases

1. Si llueve se suspende el partido.
2. Solo si llueve se suspende el partido.

tomando como proposiciones p ≡ " Llueve" y q ≡ "se suspende el partido".

(1) $p \rightarrow q$

(2) $q \rightarrow p$

Definición 3. Llamaremos **formula** a una cadena de símbolos.

Definición 4. Denotamos el **conjunto de** todos los **símbolos de proposición** como

$$\text{SP} = \{p, q, \ldots\}$$

que es un conjunto numerable (no necesariamente finito).

Definición 5. Al conjunto formado por SP y las conectivas lógicas se le denomina **alfabeto** y lo denotamos como

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

denotamos por A^* al **conjunto de cadenas de símbolos** de A

$$\text{A}^* = \{\varepsilon, a_1, a_2, \ldots, a_n : a_n \geq 0, a_i \in A, 1 \leq j \leq n\}$$

donde ε es la cadena vacía.

Ejemplo II: Dado el vocabulario $\mathrm{A} = \{a,\, b\}$ su conjunto de cadena de símbolos será el conjunto

$$\mathrm{A}^* = \{\varepsilon,\, a, b, ab, ba, aaa, aab, \ldots\}$$

Definición 6. Dado SP un conjunto de símbolos de proposición, tomamos el alfabeto A_{SP} y definimos PROP_{SP} como el menor subconjunto de A^*_{SP} que verifica

1. $SP \subseteq \mathrm{PROP}_{SP}$
2. Si $\varphi \in \mathrm{PROP}_{SP}$, entonces $(\neg\varphi) \in \mathrm{PROP}_{SP}$
3. Si $\varphi, \psi \in \mathrm{PROP}_{SP}$, entonces $(\varphi \,\square\, \psi) \in \mathrm{PROP}_{SP}$, donde

$$\square \in \{\wedge,\, \vee\, \rightarrow,\, \leftrightarrow\}$$

Veamos como se construye esta definición. Sean

$$P_0 = \mathrm{SP}$$

$$P_{n+1} = P_n \cup \{(\neg\varphi),\, (\varphi \,\square\, \psi) : \square \in \{\wedge,\, \vee\, \rightarrow,\, \leftrightarrow\},\, \varphi, \psi \in P_n\}$$

$$P = \bigcup_{i \geq 0}$$

veamos como P cumple las propiedades 1, 2 y 3 de la definición anterior. De forma trivial se verifica que $\mathrm{PROP}_{SP} \subseteq P$ y nos faltaría por demostrar la inclusión en el otro sentido.

Demostración. Sea $\varphi \in P$ entonces $\exists k$ tal que $\varphi \in P_k$ y aplicamos inducción sobre k para ver que $\varphi \in \mathrm{PROP}_{SP}$.

Para $k = 0$, por la propiedad 1 de la definición se tienen que $\varphi \in \mathrm{PROP}_{SP}$. Para $k \geq 0$

(i) $\varphi \in P_{k-1}$

(ii) $\psi \in P_{k-1}$ tal que $\varphi = (\neg\psi)$

(iii) $\psi_1, \psi_2 \in P_{k-1}$ entonces $\varphi = (\psi_1 \,\square\, \psi_2)$

□

1.1. Inducción estructural

Supongamos que queremos probar una propiedad P que cumpla $P(\varphi), \forall\varphi \in \mathrm{PROP}_{SP}$. Para ello vamos a usar una estructura basada en el *método de inducción* usual sobre $\mathbb{N}$ aplicado sobre las proposiciones. El método tiene la siguiente estructura

(1) Demostrar la **base inductiva**. Lo haremos sobre las atómicas (*i.e.* SP, $\bot, \top$)

(AT) Se cumple $P(\varphi), \forall\varphi \in \mathrm{AT}$.

(2) **Paso inductivo**. Una vez que tenemos la propiedad P probada para el caso base, suponemos la cierta la hipótesis de inducción, es decir que se cumple $P(\varphi)$, y la utilizamos para los dos casos siguientes

($\neg\varphi$) Utilizando la *h.i.* demostraremos que se cumple $P((\neg\varphi))$.

($\square$) Suponemos que φ_1 cumple P y que φ_2 cumple P, es decir, se verifican $P(\varphi_1)$ y $P(\varphi_2)$ y entonces hay que demostrar $P(\varphi_1 \square \varphi_2)$ con $\square \in \{\wedge, \vee\ \rightarrow, \leftrightarrow\}$. Dependiendo de la propiedad que queramos demostrar, podremos, o bien agrupar la conectivas lógicas en un sólo caso, o bien separarlas de forma en casos particulares.

Ejemplo III: Vamos a demostrar por inducción estructural la siguiente propiedad

P: *Toda fórmula tiene el mismo número de paréntesis abiertos y cerrados*

Para ello, vamos a denotar $|\varphi|_($ al número de paréntesis abiertos de φ y, análogamente, denotamos $|\varphi|_)$ al número de paréntesis cerrados de φ.

(AT) Si $\varphi \in SP$ ó $\varphi = \bot$ ó $\varphi = \top$, en cualquiera de los casos no hay paréntesis, luego $|\varphi|_(= |\varphi|_)$ y por tanto se verifica

$$P(\varphi), \forall \varphi \in \text{AT}$$

($\neg\varphi$) Sea $\varphi \in \text{PROP}_{SP}$ tal que se verifica $P(\varphi)$, es decir

$$|\varphi|_(= |\varphi|_)$$

ahora, el $|(\neg\varphi)|_(= |\varphi|_(+ 1$ y analogamente $|(\neg\varphi)|_) = |\varphi|_) + 1$, luego por *h.i.* se tiene que

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

luego se verifica $P(\neg\varphi), \forall \varphi \in \text{PROP}_{SP}$.

($\square$) Sean φ_1, $\varphi_2 \in \text{PROP}_{SP}$, supongamos que

$$\text{Se verifica } P(\varphi_1) \Rightarrow |\varphi_1|_(= |\varphi_1|_)$$

$$\text{Se verifica } P(\varphi_2) \Rightarrow |\varphi_2|_(= |\varphi_2|_)$$

y veamos que ocurre con $P(\varphi_1 \square \varphi_2)$

$$|\varphi_1 \square \varphi_2|_(= |\varphi_1|_(+ |\varphi_2|_(+ 1$$

$$|\varphi_1 \square \varphi_2|_) = |\varphi_1|_) + |\varphi_2|_) + 1$$

y, por *h.i.* se tiene que

$$|\varphi_1 \square \varphi_2|_) = |\varphi_1 \square \varphi_2|_($$

finalizando así la demostración.

Definición 7. Sea A un alfabeto y $\omega \in A^*$ decimos que ω' es **prefijo** de ω si $\exists \omega''$ tal que

$$\omega = \omega'\omega''$$

con, $\omega = a_1 \dots a_n$, entonces $\exists k$, $0 \leq k \leq n$ tal que $\omega' = a_1 \dots a_k$. Diremos que ω' es **prefijo propio** si $\omega' \neq \varepsilon$ y $\omega' \neq \omega$.

Ejemplo IV: Sea $A = \{a, b\}$ y $\omega = aababb$ entonces

- Si $k = 0 \Rightarrow \omega' = \varepsilon$
- Si $k = 1 \Rightarrow \omega' = a$
- Si $k = 2 \Rightarrow \omega' = aa$
- Si $k = 3 \Rightarrow \omega' = aab$
- Si $k = 4 \Rightarrow \omega' = aaba$
- Si $k = 5 \Rightarrow \omega' = aabab$
- Si $k = 5 \Rightarrow \omega' = \omega$

Ejemplo V: Sea φ' prefijo propio de φ, vamos a probar por inducción estructural la propiedad

P: *El número de paréntesis cerrados de φ' es menor que el número de paréntesis abiertos.*

utilizando la notación del ejemplo (III).

(AT) Sea $\varphi \in SP$, supongamos $\varphi = p$ entonces, o bien $\varphi' = \epsilon$ o $\varphi' = p$ luego φ no tiene prefijos propios de modo que se cumple la propiedad. Si $\varphi = \bot$ o $\varphi = \top$ de nuevo $\varphi' = \epsilon$ o $\varphi' = \bot$ o $\varphi' = \top$ que no son prefijos propios, luego φ tampoco tiene prefijos.

$$P(\varphi), \forall \varphi \in \mathrm{AT}$$

($\neg\varphi$) Supongamos que φ' es prefijo propio de $(\neg\varphi)$, y supongamos que todo prefijo propio de φ cumple la propiedad.

(CASO 1) Si $\varphi' = ($, entonces $|\varphi'|_(= 1 > |\varphi'|_) = 0$

(CASO 2) Si $\varphi' = (\neg$, entonces $|\varphi'|_(= 1 > |\varphi'|_) = 0$

(CASO 3) Si $\varphi' = (\neg\varphi''$, siendo φ'' prefijo de φ luego cumple la propiedad, si además le sumamos uno la cumple también.

(CASO 4) Si $\varphi' = (\neg\varphi$, entonces $|\varphi'|_(= 1 = |\varphi'|_) = 0$

($\square$) Supongamos que todo prefijo propio de φ_1, φ_2 cumple la propiedad. Hay que ver entonces, que los prefijos lo cumplen

$$(,\ (\varphi_1,\ \varphi_1,\ \varphi_1\,\square,\ (\varphi_1\,\square\,\varphi_2',\ (\varphi_1\,\square\,\varphi_2$$

Se deja como ejercicio.

1.2. Definiciones recursivas

Supongamos que queremos definir una función

$$H : \mathrm{PROP}_{SP} \to A$$

donde A puede tomar diferentes tipos de conjunto: $\mathbb{N}$, $\mathcal{P}(\mathrm{PROP}_{SP})$,... para hacerlo de forma recursiva, vamos a necesitar las siguientes funciones auxiliares

- Caso atómico

  $$H_{\mathrm{AT}} : \mathrm{AT} \to A$$

  dada por $H(\varphi) = H_{\mathrm{AT}}(\varphi), \forall \varphi \in \mathrm{AT}$

- Paso recursivo: donde diferenciamos entre la negación y las conectivas binarias.

  (i) Negación
  $$H_\neg : A \to A$$
  dada por
  $$H|(\neg\varphi)| = H_\neg|(H(\varphi)|$$

  (ii) Conectivas binarias
  $$H_\square : A \times A \to A$$
  dada por
  $$H|(\varphi_1 \square \varphi_2)| = H_\square|(H(\varphi_1, \varphi_2)|$$

En los casos (i) e (ii) ni la entrada ni la salida de la función está formada por formulas.

Ejemplo VI: Si queremos definir de forma recursiva el número de paréntesis abiertos de una proposición
$$H_(: \mathrm{PROP}_{SP} \to \mathbb{N}$$
se tendría
$$H_{\mathrm{AT}(} : \mathrm{AT} \to \mathbb{N}$$
dada por
$$H_{\mathrm{AT}(}(\varphi) = 0, \forall\varphi \in \mathrm{AT}$$
la negación
$$\begin{array}{rccc} H_{\neg(}: & \mathbb{N} & \to & \mathbb{N} \\ & n & \longmapsto & n+1 \end{array}$$
finalmente, el resto de conectivas
$$\begin{array}{rccc} H_{\square(}: & \mathbb{N}\times\mathbb{N} & \to & \mathbb{N} \\ & (n,\, m) & \longmapsto & n+m+1 \end{array}$$

Escribirlo así es un poco farragoso por lo que, generalmente, escribiremos las funciones haciendo uso del propio concepto de recursión
$$H_(\varphi) = 0, \forall\varphi \in \mathrm{AT}$$
$$H_{\neg(}((\neg\varphi)) = H_(\varphi) + 1$$
$$H_{\square(}(\varphi_1 \square \varphi_2) = H_(\varphi_1) + H_(\varphi_2) + 1$$
además, en la escritura, se suprime la función auxiliar. **Ejemplo VII**: definiremos de forma recursiva el número de apariciones de $\wedge$ en φ y lo denotamos como $|\varphi|^\wedge$.

(AT) $|\varphi|^\wedge = 0, \forall\varphi \in \mathrm{AT}$

($\neg$) $|(\neg\varphi)|^\wedge = |\varphi|^\wedge$

($\square$) Hay que diferenciar dos casos

$$|(\varphi_1 \square \varphi_2)|^\wedge = \begin{cases} |\varphi_1|^\wedge + |\varphi_2|^\wedge & \text{si} \quad \square \in \{\vee, \to, \leftrightarrow\} \\ |\varphi_1|^\wedge + |\varphi_2|^\wedge + 1 & \text{si} \quad \square = \wedge \end{cases}$$

Definición 8. Dadas dos fórmulas φ y ψ decimos que ψ es una **subformula** de φ si una parte de φ formada por símbolos consecutivos es idéntica a ψ.

Ejemplo VIII: dada la formula,

$$\varphi \equiv (p \vee (q \to (\neg r)))$$

observamos que esta formada por las subformulas siguientes

$$p,\, q,\, r\, (\neg r),\, (q \to (\neg r)), \varphi$$

Ejemplo IX: Creamos una función recursiva que devuelva las diferentes subformulas de una formula dada.

$$\mathrm{SUB} : \mathrm{PROP}_{SP} \to \mathcal{P}(\mathrm{PROP}_{SP})$$

dada por

(AT) $\mathrm{SUB}(\varphi) = \{\varphi\}, \forall \varphi \in \mathrm{AT}$

($\neg$) $\mathrm{SUB}((\neg\varphi)) = \{(\neg\varphi)\} \cup \mathrm{SUB}(\varphi)$

($\square$) $\mathrm{SUB}((\varphi_1 \square \varphi_2)) = \mathrm{SUB}(\varphi_1) + \mathrm{SUB}(\varphi_2) \cup \{(\varphi_1 \square \varphi_2)\}$

Proposición 1. *El esquema de definición recursiva da como resultado una única función. Esto es, dadas*

$$H_{AT} : AT \to A$$
$$H_\neg : A \to A$$
$$H_\square : A \times A \to A$$

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

$$H(\varphi) = H_{AT}(\varphi), \forall \varphi \in AT$$
$$H((\neg\varphi)) = H_\neg(H(\varphi))$$
$$H((\varphi_1 \square \varphi_2)) = H_\square(H(\varphi_1, \varphi_2))$$

1.3. Eliminación de paréntesis

Como en la aritmética básica, cuando escribimos una operación podemos utilizar las reglas de prioridad, asociatividad, etc. para escribir el menor número de paréntesis posible. Así por ejemplo en

$$(((2 \cdot 3) + 5) - (3 \cdot 2)) = 2 \cdot 3 + 5 - 3 \cdot 3$$

la idea es, por tanto, dar una serie de reglas para poder hacer lo mismo con nuestras formulas de proposición.

1.3.1. Reglas de eliminación de paréntesis

1. **Elminación de paréntesis externos**. No aportan información.
2. **Prioridad entre conectivas**. En la siguiente lista, las conectivas, aparecen de más prioridad a menos
$$(+) \quad \neg, \wedge, \vee, \to, \leftrightarrow \quad (-)$$
3. **Asociatividad**. Adoptamos el convenio de asociar por la izquierda. De este modo si tenemos $p \to q \to r$ daremos como asociación válida
$$(p \to q) \to r$$
siendo, por tanto, errónea
$$p \to (q \to r)$$
Esto mismo se aplica para $\leftrightarrow$. En los casos de $\vee$ y $\wedge$ no se presenta problema pues son asociativas.

1.4. Valoraciones: Tablas de verdad

Todo lenguaje se subdivide en

1. Sintaxis: reglas de formación de las frases o fórmulas.
2. Semántica: significado de las frases o fórmulas.

de este modo, se conforma un lenguaje formal.

Definición 9. Denotaremos por BOOL al conjunto formado por dos elementos[1]

$$\text{BOOL} = \{\text{Verdadero, Falso}\} = \{\text{V, F}\} = \{\text{T, F}\} = \{1, 0\}$$

con los que vamos a *valorar* las fórmulas.

Para dar un sentido más formal a la idea de *valorar*, vamos a construir una serie de aplicaciones sobre el conjunto BOOL en la siguiente definición.

Definición 10. Diremos que una **valoración** es una aplicación de la forma

$$v: \ \text{SP} \ \to \ \text{BOOL}$$

cuya extensión da lugar a

$$\begin{array}{rccc} \widehat{v}: & \text{PROP}_{SP} & \to & \text{BOOL} \\ & \varphi & \mapsto & \widehat{v}(\varphi) \end{array}$$

que definiremos de forma recursiva

(AT) $\widehat{v}(\top) = V$, $\widehat{v}(\bot) = F$, $\widehat{v}(p) = v(p)$, si $p \in \text{SP}$

($\neg$) $\widehat{v}(\neg) = v_{\neg}(\widehat{v}(\varphi))$

($\square$) $\widehat{v}((\varphi \square \psi)) = v_{\square}(\widehat{v}(\varphi), \widehat{v}(\psi))$ con $\square \in \{\wedge, \vee \to, \leftrightarrow\}$

Estas funciones de valoración dan lugar a las tablas de verdad de cada una de las conectivas lógicas.

[1] Durante este curso elegimos, generalmente, como notación para el conjunto BOOL = {V, F}.

$p \wedge q$

$v_\wedge$	p: **V**	p: **F**
q: **V**	V	F
q: **F**	F	F

$p \vee q$

$v_\vee$	p: **V**	p: **F**
q: **V**	V	V
q: **F**	V	F

$p \to q$

$v_\to$	p: **V**	p: **F**
q: **V**	V	F
q: **F**	V	V

$p \leftrightarrow q$

$v_\leftrightarrow$	p: **V**	p: **F**
q: **V**	V	F
q: **F**	V	V

$\neg p$

$v_\neg$	
p: **V**	F
p: **F**	V

Las aplicaciones son

$$v_\square : \ \text{BOOL} \times \text{BOOL} \ \to \ \text{BOOL}$$

si $\square \in \{\wedge, \vee \ \to, \leftrightarrow\}$ En el caso de la negación

$$v_\neg : \ \text{BOOL} \ \to \ \text{BOOL}$$

Donde encontremos matrices simétricas, podemos decir que ese operador es conmutativo. Los operadores que conmutan son: la disyunción y la conjunción.

Ejemplo X: utilizando las aplicaciones de la definición (10), vamos a valor la fórmula

$$\varphi = (q \vee r) \to (q \to r)$$

partiendo de las valoraciones

$$v(p) = V, \quad v(q) = F, \quad v(r) = V$$

$$\widehat{v} = v_\to(\widehat{v}(q \vee r), \widehat{v}(p \to q)) = v_\to(v_\vee(\widehat{v}(q), \widehat{v}(r)), v_\to(\widehat{v}(p), \widehat{v}(q))) =$$
$$= v_\to(v_\vee(F, V), v_\to(V, F)) = v_\to(V, F) = F$$

Es claro que no es un método muy óptimo si queremos ver todas las posibles valoraciones de las proposiciones que formen parte de la fórmula. Para ello se utiliza la tabla de verdad.

Ejemplo XI: Completar la tabla de verdad de la fórmula φ del ejemplo anterior. (*Se deja como ejercicio).*

Definición 11. Dada v : SP $\to$ BOOL valoración y $\varphi \in \text{PROP}_{SP}$, v **satisface** φ si y sólo si $\widehat{v}(\varphi) = V$; y denotamos $v \models \varphi$. En caso contrario, v no satisface φ, si $\widehat{v}(\varphi) = F$, denotamos $v \not\models \varphi$. Al símbolo $\models$ se le denomina **símbolo de satisficidad**.

1.4.1. Clasificación de fórmulas

Definición 12. Según como sean las valoraciones de una fórmula, podemos clarificarlas en

1. **Satisfactible**. Existe alguna valoración v tal que $v \models \varphi$.
2. **Tautología**. Siempre es cierto, es decir, $\forall v$ se tiene que $v \models \varphi$.
3. **Contingencia**. φ se dice contingencia si es satisfactible, pero no tautología.
4. **Contradicción**. Siempre es falso, es decir, $\forall v$ se tiene que $v \not\models \varphi$.

Observación. Todas las valoraciones posibles que hay en una fórmula viene dado por 2^ρ donde ρ es el número de proposiciones que tiene la fórmula.

Definición 13. Sea $\Phi \subseteq \mathrm{PROP}_{SP}$ un conjuntos de fórmulas, definimos

$$\mathrm{Mod}(\Phi) = \{v / v \models \varphi, \forall \varphi \in \Phi\}$$

entonces diremos que

1. Φ es **satisfactible**, si $\mathrm{Mod}(\Phi) \neq \emptyset$ y denotamos $\mathrm{Sat}(\Phi)$
2. Φ es **insatisfactible**, si $\mathrm{Mod}(\Phi) = \emptyset$ y denotamos $\mathrm{Insat}(\Phi)$
3. $\varphi \in \mathrm{PROP}_{SP}$ es **consecuencia lógica** de Φ si y sólo si $\mathrm{Mod}(\Phi) \subseteq \mathrm{Mod}(\{\varphi\})$, denotamos $\Phi \models \varphi$.

Observación: si partimos de una premisa falsa podemos concluir cualquier cosa. Esto es

$$\mathrm{Insat}(\Phi) \text{ entonce } \Phi \models \varphi$$

además, podemos expresar una tautología como

$$\text{Si } \Phi \neq \emptyset \text{ y } \Phi \models \varphi \text{ entonce } \models \varphi$$

En este sentido se pueden definir todo tipo de propiedades, como las que vienen a continuación.

Proposición 2. *Se tiene que*

1. $\Phi \cup \{\varphi\} \models \psi$ *si y sólo si* $\Phi \models \varphi \to \psi$
2. $\Phi \models \varphi$ *si y sólo si* $\mathit{Insat}(\Phi \cup \{\varphi\})$

Demostración. Vamos a demostrar el apartado (1) (*el (2) queda como ejercicio*. Comenzamos con la implicación hacía la derecha.

Supongamos que $\Phi \cup \{\varphi\} \models \psi$ y hay que ver si $\mathrm{Mod}(\Phi) \subseteq \mathrm{Mod}(\varphi \to \psi)$. Si $v \in \mathrm{Mod}(\Phi)$ entonces

(i) $\widehat{v}(\varphi) = V \Rightarrow v \in \mathrm{Mod}(\varphi) \Rightarrow v \in \mathrm{Mod}(\Phi \cup \{\varphi\} \Rightarrow v \in \mathrm{Mod}(\psi) \Rightarrow \widehat{v}(\psi) = V$ de modo que $\widehat{v}(\varphi \to \psi) = V$ en consecuencia $v \in \mathrm{Mod}(\varphi \to \psi)$.

(ii) $\widehat{v}(\varphi) = F$, entonces $\widehat{v}(\varphi \to \psi) = V \Rightarrow v \in \mathrm{Mod}(\varphi \to \psi)$

Para la implicación en el otro sentido. Si $v \in \mathrm{Mod}(\Phi \cup \{\varphi\})$ entonces

$$\left.\begin{array}{lllll} v \in \mathrm{Mod}(\Phi) & \Rightarrow & v \in \mathrm{Mod}(\varphi \to \psi) & \Rightarrow & \widehat{v}(\varphi \to \psi) = V \\ v \in \mathrm{Mod}(\varphi) & \Rightarrow & \widehat{v}(\varphi) = V & & \end{array}\right\} \Rightarrow \widehat{v}(\psi) = V \Rightarrow v \in \mathrm{Mod}(\psi)$$

□

1.5. Equivalencias lógicas

Acabamos de ver en la *proposición 2* que podemos escribir propiedades acerca de un conjunto de proposiciones. Pero, de momento, los métodos que disponemos para demostrar este tipo de propiedades no siempre son los más óptimos. Veamoslo en con el siguiente ejemplo.

Ejemplo XII: a través de una tabla de verdad, partiendo de

$$\Phi = \{\neg p \wedge q \to r,\ \neg p,\ \neg r\}$$

$$\varphi = \neg q$$

hay que demostrar que

$$\Phi \models \varphi$$

Para la tabla coloreamos en rojo las premisas que deben satisfacer la consecuencia, pintada de azul. Satisfacer implica que la valoración debe ser V en todas las premisas. Esta situación sólo se da para las valoraciones p, q y r todas ellas F (fila señalada con una flecha).

	p	q	r	$\neg p$	$\neg p \wedge q$	$\neg p \wedge q \to r$	$\neg r$	$\neg q$
	V	**V**	**V**	F	F	V	F	F
	V	**V**	**F**	F	F	V	V	F
	V	**F**	**V**	F	F	V	F	V
	V	**F**	**F**	F	F	V	V	V
	F	**V**	**V**	V	V	V	F	F
	F	**V**	**F**	V	V	F	V	F
	F	**F**	**V**	V	F	V	F	V
$\longrightarrow$	**F**	**F**	**F**	V	F	V	V	V

puesto que la consecuencia es también verdadera, hemos demostrado que $\Phi \models \varphi$.

Esta forma no es efectiva y vamos a dar mecanismos para no tener que hacer uso de valoraciones y tablas de verdad.

Definición 14. Sean $\varphi, \psi \in \mathrm{PROP}_{SP}$ son **lógicamente equivalentes** y lo denotamos como $\varphi \sim \psi$, si y sólo si $\forall v, \widehat{v}(\varphi) = \widehat{v}(\psi)$.

Lema 1. *Diremos que $\sim$ es una **relación de equivalencia**; más aún, es una congruencia respecto a dos operadores lógicos (i.e. la relación se respeta con los operadores lógicos).*

$$\left.\begin{array}{ll} \varphi \sim \psi & \varphi_1 \sim \varphi_1' \\ \neg\varphi \sim \neg\psi & \varphi_2' \sim \varphi_2 \end{array}\right\} \Rightarrow \quad \varphi_1 \square \varphi_2 \sim \varphi_1' \square \varphi_2'$$

con $\square \in \{\wedge, \vee\ \to, \leftrightarrow\}$.

1.5.1. Leyes de equivalencia lógica

1. **Conmutatividad**

$$p \wedge q \sim q \wedge p$$
$$p \vee q \sim q \vee p$$

2. **Asociatividad**

$$p \wedge (q \wedge r) \sim (p \wedge q) \wedge r$$
$$p \vee (q \vee r) \sim (p \vee q) \vee r$$

3. **Idempotencia**

$$p \wedge p \sim p$$
$$p \vee p \sim p$$

4. **Distributiva**

$$p \wedge (q \vee r) \sim (p \wedge q) \vee (p \wedge r)$$
$$p \vee (q \wedge r) \sim (p \vee q) \wedge (p \vee r)$$

5. **Absorción**

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

6. **Cero y uno**

$$p \wedge \top \sim p$$
$$p \vee \top \sim \top$$

$$p \wedge \bot \sim \bot$$
$$p \vee \bot \sim p$$

7. **Contradicción**

$$p \wedge (\neg p) \sim \bot$$

8. **Tercio excluido**

$$p \vee (\neg p) \sim \top$$

9. **Doble negación**

$$\neg(\neg p) \sim p$$

10. **Reducción**

$$p \to q \sim \neg p \vee q$$
$$p \leftrightarrow q \sim (p \to q) \wedge (q \to p)$$

11. **De Morgan**

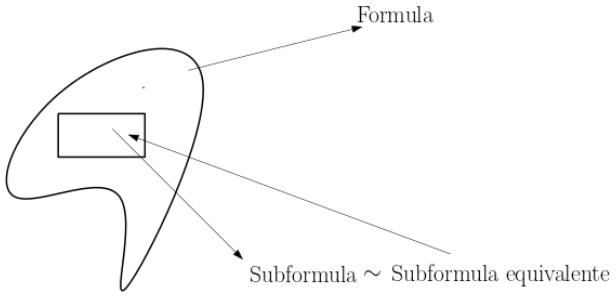
$$\neg(p \wedge q) \sim \neg p \vee \neg q$$
$$\neg(p \vee q) \sim \neg p \wedge \neg q$$

Se deja como ejercicio simplificar las expresiones siguientes mediante las leyes que acabamos de ver

1. $\{\varphi_1, \varphi_2\} \models \psi$
2. $\text{Insat}(\Phi \cup \{\psi\})$
3. $\varphi_1 \wedge \varphi_2 \wedge \neg\psi \sim \bot$

1.6. Sustitución

La idea es tomar una parte de una formula y sustituirla por algo equivalente que sea más manejable. Definamos de forma rigurosa el concepto.



Definición 15. Sean $\varphi, \psi \in \text{PROP}_{SP}$ y $p \in \text{SP}$, se denota

$$\psi[\varphi/p]$$

a sustituir en la fórmula ψ las apariciones de p por φ. Para definir la sustitución utilizamos el método de recursión. De este modo, obtenemos

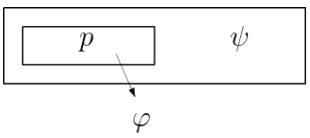
(AT) Para las formulas atómicas se tiene que

- $p[\varphi/p] = \varphi$.
- $q[\varphi/p] = q \quad q \neq p$.
- $\top[\varphi/p] = \top$.
- $\bot[\varphi/p] = \bot$.

$(\neg)$ $(\neg\psi)[\varphi/p] = \neg(\psi[\varphi/p])$.

$(\square)$ $(\psi_1 \square \psi_2)[\varphi/p] = (\psi_1[\varphi/p] \square \psi_2[\varphi/p])$.

De manera esquemática



Teorema 1. *Sean $\varphi, \varphi', \psi \in PROP_{SP}$, $p \in SP$. Si $\varphi \sim \varphi'$, entonces $\psi[\varphi/p] \sim \psi[\varphi'/p]$.*

Notación. Dada

$$f : A \to B$$

se tiene que

$$f[b/a](x) = \begin{cases} f(x) & \text{si} \quad x \neq a \\ b & \text{si} \quad x = a \end{cases}$$

Lema 2. *Sean $\varphi, \psi \in PROP_{SP}$, $p \in SP$ y v valoración. Entonces:*

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

Demostración. Procedemos por inducción estructural sobre ψ. Comenzamos por las atómicas

(AT)
- Si $\psi = p$ entonces, $\psi[\varphi/p] = \varphi$. Luego

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

- Si $\psi = q$ con $q \neq p$ entonces $\psi[\varphi/p] = q$, con lo que $\widehat{v}(\psi[\varphi/p]) = \widehat{v}(q)$, luego

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

- Si $\psi = \top$, $\psi[\varphi/p] = \top$, luego

$$v[\widehat{\widehat{v}(\varphi)/p}](\top) = v(\top) = V$$

- Si $\psi = \bot$, $\psi[\varphi/p] = \bot$, luego

$$v[\widehat{\widehat{v}(\varphi)/p}](\bot) = v(\bot) = F$$

($\neg$) Si $\psi = \neg\psi_1$, entonces $\widehat{v}((\neg\psi_1)[\varphi/p]) = \widehat{v}(\neg(\psi_1[\varphi/p])) = v_\neg(\widehat{v}(\psi_1[\varphi/p]))$ por hipótesis de inducción tenemos que

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

($\square$) Si $\psi = \psi_1 \square \psi_2$, entonces

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

por hipótesis de inducción

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

□

Demostración. (*Teorema 1*).

$$\widehat{v}(\psi[\varphi/p]) = v[\widehat{\widehat{v}(\varphi)/p}](\psi) =_1 v[\widehat{\widehat{v}(\varphi')/p}](\psi) = \widehat{v}(\psi[\varphi'/p])$$

en (1) utilizamos que $\widehat{v}(\varphi) = \widehat{v}(\varphi')$. □

Ejemplo XIII: Sean $\varphi = p \to q$ y $\psi = r \to t$ y tomamos la sustitución $\psi[\varphi/r] = (p \to q) \to t$. Entonces, es indiferente valorar $\psi[\varphi/r]$ en

p	**q**	**r**	**t**
V	F	V	V

que valorarlo, siendo $\widehat{v}(\varphi) = F$, entonces $v[\widehat{v}(\varphi)/r]$

p	**q**	**r**	**t**
V	F	F	V

Lema 3. *Si $\psi_1 \sim \psi_2$ entonces $\psi_1[\varphi/p] \sim \psi_2[\varphi/p]$.*

Demostración. En efecto

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

como $\psi_1 \sim \psi_2$

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

□

Ejemplo XIV: Si queremos ver $\varphi \wedge \psi \sim \psi \wedge \varphi$ basta tomar $\psi_1 = p \wedge \psi$ y $\psi_2 = \psi \wedge p$ y aplicar el lema anterior.

Observación. Esto implica que, gracias al lema de sustitución, las leyes de equivalencia vistas en la sección 1.5.1, es también aplicable a fórmulas completas.

1.7. Completitud funcional

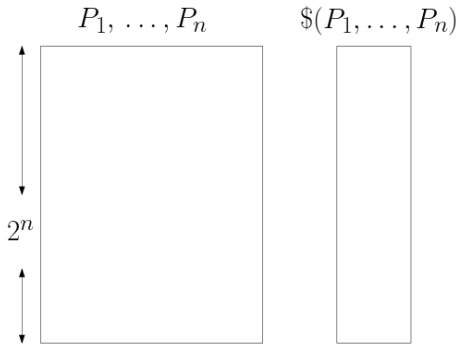
Debido a las leyes de reducción los operadores $\to$, $\leftrightarrow$ se dicen secundarios. También podemos definir nuevos operadores como

Disyunción exclusiva	$\veebar$	$p \veebar q \sim (p \vee \neg q) \vee (\neg p \vee q)$
NAND	$\downarrow$	$p \downarrow q \sim \neg(p \vee q)$
NOR	$\uparrow$	$p \uparrow q \sim \neg(p \wedge q)$

o a través de una tabla de verdad

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

Podríamos construir un operador lógico con tantos argumentos como deseáramos. Pero, vamos a ver que todos ellos se pueden expresar de forma equivalente como conjunción, negación y disyunción.



Definición 16. Un conjunto de conectivas $\mathcal{C}$ es **funcionalmente completo** si cualquier otra conectiva \$ de aridad n se puede expresar con las conectivas de $\mathcal{C}$.

Teorema 2. *Sea $\mathcal{C} = \{\vee, \wedge, \neg\}$ es funcionalmente completo.*

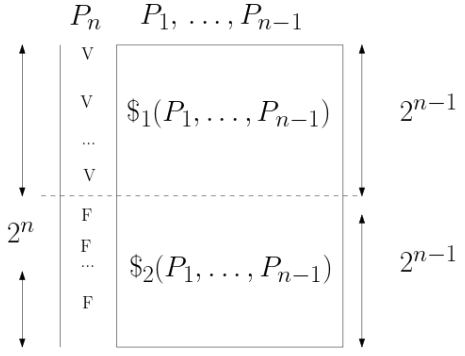
Demostración. Sea \$ conectiva de aridad n y aplicamos inducción sobre n.

Si $n = 1$. Hay cuatro posibles conectivas de aridad 1, veamos sus tablas de verdad

p	$\$_1(p)$	$\$_2(p)$	$\$_3(p)$	$\$_4(p)$
V	V	V	F	F
F	V	F	V	F

Si nos fijamos, $\$_1(p) \sim \top$, $\$_4(p) \sim \bot$, $\$_2(p) \sim p$ y $\$_3(p) \sim \neg p$. Se cumple, por tanto, el caso base.

Si $n > 1$. Se da la situación $\$(P_1, \ldots, P_n) = (P_n \wedge \$_1(P_1, \ldots P_{n-1})) \vee (\neg P_n \wedge \$_2(P_1, \ldots P_{n-1}))$



por hipótesis de inducción existen φ_i, $i = 1, 2$ construidas con las conectivas de $\mathcal{C}$ tales que $\varphi_i \sim \$_i$ con $i = 1, 2$, entonces $\$(P_1, \ldots, P_n) \sim (P_n \wedge \varphi_1) \vee (\neg P_n \wedge \varphi_2)$ □

Se puede demostrar aplicando las leyes de De Morgan que el conjunto $\{\wedge, \neg\}$ también es funcionalmente completo. Otros que también lo son y se deja la comprobación como ejercicio son

$$(1)\,\{\vee, \neg\} \quad (2)\,\{\rightarrow, \bot\}, \quad (3)\,\{\uparrow\} \quad (4)\,\{\downarrow\}$$

1.8. El método de los Tableaux

En proceso de escritura

2. Lógica de primer orden

La idea es extender las posibilidades de la lógica de primer orden. Si pensamos en predicados como

1. *Todos los primos mayores que 2 son impares*
2. *El 3 es un primo mayor que 2*
3. *El 3 es impar*

es claro que no podemos expresarlo con las herramientas de la lógica proposicional.

La lógica de primer orden nos habla de *elementos* de un *conjunto* dado. Empezamos introduciendo algunas herramientas básicas

- **Constantes**: en nuestros predicados anteriores serían el 2 y el 3.
- **Variables**: aquellos elementos del conjunto a los que aplicamos las propiedades. Denotamos por x, y, $z\ldots$
- **Símbolos de función**: para el conjunto de los números naturales, por ejemplo, hablaríamos de la suma $+$, del producto $*$ o de la función sucesor de un número $s(x)$. Gracias a ellas podremos expresar cosas como: $s(+(2,3))$ o en notación habitual $s((2+3))$.

estos objetos nos van a permitir construir los **términos**.

Los símbolos que teníamos hasta ahora son insuficientes, por ello vamos a introducir algunos más

- Símbolo de igualdad: $\doteq$
- Símbolos de predicado: que establecen una propiedad. Por ejemplo, ser impar que podemos denotar como $\mathrm{im}(x)$ o que un elemento sea menor que otro $<(x,y)$. El primer de ellos decimos que tiene aridad 1 (un argumento de entrada) y el segundo tienen aridad 2 (pues necesita dos entradas).
- Cuantificadores: $\forall$ y $\exists$
- Conectivas lógicas: $\{\wedge, \vee\ \rightarrow, \leftrightarrow\}$ y $\top$, $\bot$

Ejemplo XV: A partir de las frases que hemos dado al inicio de la sección, vamos a formalizarlas con los nuevos elementos que hemos introducido.

1. $\forall x\,(\mathrm{pr}(x) \wedge 2 < x \rightarrow \mathrm{im}(x))$
2. $\mathrm{pr}(3) \wedge 2 < 3$
3. $\mathrm{im}(3)$

Definición 17. Una **signatura** S

$$S = \langle \mathrm{Cts}_S, \mathrm{Fn}_S, \mathrm{Pd}_S \rangle$$

donde

- $\mathrm{Cts}_S \equiv$ conjunto de constantes de S.
- $\mathrm{Fn}_S \equiv$ conjunto de símbolos de función con aridad.
- $\mathrm{Pd}_S \equiv$ conjunto de símbolos de predicado con aridad.

Ejemplo XVI: Tomando el conjunto de los números naturales, obtenemos la signatura

$$\mathrm{Nat} = \langle \mathrm{Cts}_{\mathrm{Nat}}, \mathrm{Fn}_{\mathrm{Nat}}, \mathrm{Pd}_{\mathrm{Nat}} \rangle$$

donde

- $\mathrm{Cts}_{\mathrm{Nat}} \equiv \{0, 1, \ldots\}$
- $\mathrm{Fn}_{\mathrm{Nat}} \equiv \{+|_2, s|_1\}$
- $\mathrm{Pd}_{\mathrm{Nat}} \equiv \{\mathrm{im}|_2, < |_1\}$

El número escrito tras la barra indica su aridad.

Consideramos el conjunto de **variables**

$$\mathrm{Var} = \{x, y, z, \ldots\}$$

siempre será numerable.

Definición 18. Dada una signatura S, el **alfabeto**

$$A_S = \mathrm{Cts}_S \cup \mathrm{Fn}_S \cup \mathrm{Pd}_S \cup \mathrm{Var} \cup \{\wedge, \vee \rightarrow, \leftrightarrow\} \cup \{\top, \bot, \forall, \exists, \doteq, (,)\}$$

Definición 19. Dada una signatura S, el conjunto de **términos** de S, TERM_S, es el menor subconjunto X de A_S^* que cumple

- Caso base

  (T1) Si $c \in \mathrm{Cts}_s$ entonces $c \in X$

  (T2) Si $x \in \mathrm{Var}$ entonces $x \in X$

- Paso recursivo

  (T3) $f|_n \in \mathrm{Fn}_s$ y $t_1, \ldots, t_n \in X$ entonces $f \in X$

Ejemplo XVII: En el ejemplo anterior definimos Nat, algunos términos suyos son

$$0, 1, 2, ... \in \mathrm{TERM}_{Nat}$$

$$x, y, z, ... \in \mathrm{TERM}_{Nat}$$

$$s(x), s(1), +(s(x), s(0)), ... \in \mathrm{TERM}_{Nat}$$

Definición 20. Dada una signatura S, el conjunto de fórmulas de primer orden, FORM_s, es el menor subconjunto $X \subseteq A_S^*$ que verifica

- Fórmulas atómicas, que denotamos como FORMAT_S

  (F1) $t_1, t_2 \in \mathrm{TERM}_s$ entonces $t_1 \doteq t_2 \in X$

  (F2) $p \in \mathrm{Pd}_s|_n$ y $t_1, \ldots, t_n \in \mathrm{TERM}_s$ entonces $p(t_1, \ldots, t_n) \in X$

  (F3) $\top, \bot \in X$

(F4) $\varphi_1, \varphi_2 \in X$ entonces $(\varphi_1 \square \varphi_2) \in X$ también $\neg\varphi_i \in X$ con $i = 1, 2$.

(F5) $\varphi \in X$, $x \in \mathrm{Var}$ entonces[2] $\forall x : \varphi$ y $\exists x : \varphi$.

Ejemplo XVIII: Si escribimos

$$\exists x : \varphi \to \psi$$

el existe afecta a todo, en cambio, si escribimos

$$(\exists x : \varphi) \to \psi$$

solo afecta a φ.

[2]El $\forall$ significa que todo elementos de X cumple φ. Por otro lado, el $\exists$ significa que un elemento de X cumple φ