

Parte II: Lógica Proposicional

13 de septiembre de 2017

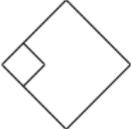
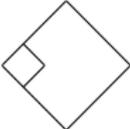
¿Qué es lógica (proposicional)?

- La lógica se entiende, en términos generales, como el *análisis de los razonamientos correctos*.
- A fines del siglo XIX y principios del XX, se despertó un interés por dar bases sólidas a la matemática.
- Para ello se introducen sistemas lógicos en los que las fórmulas y los razonamientos válidos están establecidos sin ninguna referencia a lenguajes naturales.
- De esta manera se puede comprobar la validez de razonamientos por medios puramente sintácticos.

Ejemplo de razonamiento correcto

- Premisa 1: Si P es un poset finito, entonces P tiene al menos un maximal.
- Premisa 2: $(D_{32}, |)$ es un poset finito.
- Conclusión: D_{32} tiene al menos un maximal.

Ejemplo de razonamiento correcto

- Premisa 1: Si un reticulado tiene un sub-reticulado isomorfo a N_5 , entonces no es distributivo.
- Premisa 2:  tiene un reticulado isomorfo a N_5 .
- Conclusión:  no es distributivo.

El patrón de esos ejemplos

- Ambos razonamientos tienen el siguiente esquema:
- Premisa 1: Si P , entonces Q .
- Premisa 2: P
- Conclusión: Q

Ejemplo de razonamiento **incorrecto**

- Si el ADN de una muestra de sangre coincide con el de Santiago Maldonado, entonces la gendarmería está implicada.
- No hay (por el momento) muestras de ADN que coinciden.
- Conclusión (Patricia Bullrich): gendarmería no está implicada.

Sintaxis y Semántica

- Para estudiar matemáticamente los razonamientos válidos debemos saber representar los mismos de manera matemática. Esto es, definir un conjunto de proposiciones, la *sintaxis*.
- Cada proposición la podremos interpretar como cierta o falsa; a esta interpretación la llamamos la *semántica*.
- Lo primero que descubrimos es que los razonamientos demuestran la *validez* de una *afirmación* a partir de la validez de otras siguiendo ciertas *reglas*. Esta noción se formaliza como *pruebas*.

El alfabeto

- Asumimos un conjunto numerable \mathcal{V} de variables proposicionales que representan las afirmaciones más básicas.
- A los elementos de \mathcal{V} los escribiremos simplemente como p_0, p_1, \dots .
- Definimos $At = \mathcal{V} \cup \{\perp\}$, el símbolo \perp representa la afirmación “es falso”. A este conjunto At lo llamamos el conjunto de *átomos*.
- Las proposiciones serán ciertas palabras construidas sobre el alfabeto: $\Sigma = At \cup \{\neg, \vee, \wedge\} \cup \{(,)\}$.

Palabras

- Al conjunto de palabras que se construyen con el alfabeto Σ lo denotamos con Σ^* .
- Ejemplos de palabras sobre Σ son:

$$p_0 \quad \wedge p_0(p_1 \quad (p_{23} \wedge p_9) \quad (\neg \perp)$$

- Ejemplos de cadenas que *NO* son palabras sobre Σ :

$$4 + 0 \quad x \leq y \wedge z \quad A \vee B$$

- Pero no todas las palabras serán proposiciones.

Proposiciones

- Algunos elementos de Σ^* representan proposiciones:

$$p_2 \quad p_{35} \quad (\neg p_{35}) \quad ((\neg p_{35}) \wedge p_2)$$

- ..., pero otras no:

$$\wedge p_0(p_1 \quad p_{23} \wedge \neg p_9 \vee p_2)$$

- En la última podemos descubrir la necesidad de utilizar paréntesis.
- Pero, cómo definir un sub-conjunto $Prop \subseteq \Sigma^*$ que sólo contenga proposiciones?

Proposiciones

- Ahora daremos una definición *inductiva* del conjunto *Prop*.
 - $\phi \in At$ Si $\phi \in At$, entonces $\phi \in Prop$;
 - $(\neg \phi)$ Si $\phi \in Prop$, entonces $(\neg \phi) \in Prop$;
 - $(\phi \vee \psi)$ Si $\phi \in Prop$ y $\psi \in Prop$, entonces $(\phi \vee \psi) \in Prop$.
 - $(\phi \wedge \psi)$ Si $\phi \in Prop$ y $\psi \in Prop$, entonces $(\phi \wedge \psi) \in Prop$.
 - $(\phi \rightarrow \psi)$ Si $\phi \in Prop$ y $\psi \in Prop$, entonces $(\phi \rightarrow \psi) \in Prop$.
- Observación: podemos unificar las tres últimas cláusulas usando una meta-variable \square que puede ser \vee, \wedge , ó \rightarrow :
 - $(\phi \square \psi)$ Si $\phi \in Prop$ y $\psi \in Prop$, entonces $(\phi \square \psi) \in Prop$.

Proposiciones

- ¿Cómo sabemos que esas cláusulas definen un conjunto?
- Podemos definir conjuntos cada vez más grandes:

$$Prop_0 = At$$

$$Prop_1 = Prop_0 \cup \{(\neg\phi) \mid \phi \in Prop_0\} \\ \cup \{(\phi \square \psi) \mid \phi, \psi \in Prop_0\}$$

...

$$Prop_{k+1} = Prop_k \cup \{(\neg\phi) \mid \phi \in Prop_k\} \\ \cup \{(\phi \square \psi) \mid \phi, \psi \in Prop_k\}$$

- El conjunto de proposiciones es la unión de todos esos conjuntos:

$$Prop = \bigcup_{n \in \mathbb{N}} Prop_n$$

Proposiciones como TAD

- El TAD apropiado para las proposiciones está parametrizado por \mathcal{V} :
- Constructores:

$$\perp: Prop$$

$$\text{Symbol}: \mathcal{V} \rightarrow Prop$$

$$\neg: Prop \rightarrow Prop$$

$$\vee: Prop \times Prop \rightarrow Prop$$

$$\wedge: Prop \times Prop \rightarrow Prop$$

$$\rightarrow: Prop \times Prop \rightarrow Prop$$

- Ecuaciones:

Recursión en los naturales

- Si queremos definir una función $f: \mathbb{N} \rightarrow X$ de los naturales en algún conjunto X alcanza con:

$n = 0$ elegir $a \in X$ y definir $f(0) = a$;

$n = k + 1$ definir $f(n)$ en términos de $f(k)$:

$$f(k + 1) = \dots f(k) \dots$$

- Es fácil ver que de esa manera tenemos bien definida la función f .
- Puesto que *Prop* también es un conjunto definido inductivamente, podemos utilizar un esquema semejante.

Recursión en *Prop*

- Supongamos ahora que queremos definir una función $f: Prop \rightarrow X$.
- Ahora tenemos muchos casos base: $\perp, p_0, p_1, \dots, p_{2356}, \dots$ por lo tanto debemos definir:

$$f(\phi) = \begin{cases} x_i & \text{si } \phi = p_i \\ y & \text{si } \phi = \perp \end{cases}$$

- Y tenemos varios casos recursivos, uno para cada conectivo; en el caso de la negación:

$$f((\neg\phi)) = \dots f(\phi) \dots$$

- Para las fórmulas de la forma $(\phi_1 \square \phi_2)$, podemos utilizar la llamada recursiva tanto en ϕ_1 como en ϕ_2 :

$$f((\phi_1 \square \phi_2)) = \dots f(\phi_1) \dots f(\phi_2) \dots$$

Recursión, ejemplos

- Cantidad de conectivos en una proposición:

$$\text{con}(-): \text{Prop} \rightarrow \mathbb{N}$$

$$\text{con}(\phi) = 0 \quad \text{si } \phi \in \text{At}$$

$$\text{con}((\neg \phi)) = \text{con}(\phi) + 1$$

$$\text{con}((\phi_1 \square \phi_2)) = \text{con}(\phi_1) + \text{con}(\phi_2) + 1$$

- Cantidad de símbolos "(" y ")" en una proposición:

$$\text{paren}(-): \text{Prop} \rightarrow \mathbb{N}$$

$$\text{paren}(\phi) = 0 \quad \text{si } \phi \in \text{At}$$

$$\text{paren}((\neg \phi)) = \text{paren}(\phi) + 2$$

$$\text{paren}((\phi_1 \square \phi_2)) = \text{paren}(\phi_1) + \text{paren}(\phi_2) + 2$$

Otros ejemplos de recursión

- Sub-fórmulas:

$$\text{sub}(-): \text{Prop} \rightarrow \mathcal{P}(\text{Prop})$$

$$\text{sub}(\phi) = \{\phi\} \quad \text{si } \phi \in \text{At}$$

$$\text{sub}((\neg \phi)) = \text{sub}(\phi) \cup \{(\neg \phi)\}$$

$$\text{sub}((\phi_1 \square \phi_2)) = \text{sub}(\phi_1) \cup \text{sub}(\phi_2) \cup \{(\phi_1 \square \phi_2)\}$$

- Sustitución del símbolo p_i por la proposición ψ :

$$- [\psi/p_i]: \text{Prop} \rightarrow \text{Prop}$$

$$p_j [\psi/p_i] = \begin{cases} \psi & \text{si } i = j \\ p_j & \text{si } i \neq j \end{cases}$$

$$\perp [\psi/p_i] = \perp$$

$$(\neg \phi) [\psi/p_i] = (\neg \phi [\psi/p_i])$$

$$(\phi_1 \square \phi_2) [\psi/p_i] = (\phi_1 [\psi/p_i] \square \phi_2 [\psi/p_i])$$

Inducción en sub-fórmulas

- Así como podemos definir funciones recursivamente, también podemos probar propiedades sobre $Prop$ utilizando *inducción*.
- Para probar que todo $\phi \in Prop$ satisface un predicado A , entonces alcanza con probar:
 - $\phi \in At$ $A(\phi)$, para todo $\phi \in At$;
 - $(\neg \phi)$ Si $A(\phi)$, entonces $A((\neg \phi))$;
 - $(\phi \square \psi)$ Si $A(\phi)$ y $A(\psi)$, entonces $A((\phi \square \psi))$.
- Notemos que ese principio de inducción es análogo al principio de inducción para los naturales.

Inducción, ejemplo

Teorema

Para toda $\phi \in Prop$, $paren(\phi) = 2 * con(\phi)$.

- Antes de intentar probarlo, enunciemos las hipótesis inductivas que tendremos a nuestra disposición:
- Si $\phi = (\neg\phi')$, entonces la hipótesis inductiva vale para ϕ' , es decir:

$$paren(\phi') = 2 * con(\phi')$$

- Si $\phi = (\phi_1 \square \phi_2)$, ahora disponemos de la hipótesis inductiva tanto sobre ϕ_1 como sobre ϕ_2 :

$$paren(\phi_1) = 2 * con(\phi_1) \quad paren(\phi_2) = 2 * con(\phi_2)$$

Inducción, ejemplo

- Si $\phi \in At$, es fácil

$$paren(\phi) = 0 = 2 * 0 = 2 * con(\phi)$$

- Si $\phi = (\neg \phi')$, utilizando la hipótesis inductiva para ϕ' calculamos:

$$\begin{aligned} & paren((\neg \phi')) \\ = & \{ \text{definición} \} \\ & paren(\phi') + 2 \\ = & \{ \text{por h.i. en } \phi' \} \\ & (2 * con(\phi')) + 2 \\ = & \{ \text{distributividad} \} \\ & 2 * (con(\phi') + 1) \\ = & \{ \text{definición} \} \\ & 2 * con((\neg \phi')) \end{aligned}$$

Inducción, ejemplo

- Si $\phi = (\phi_1 \square \phi_2)$, ahora disponemos de la hip. inductiva sobre ϕ_1 y sobre ϕ_2 .

$$\begin{aligned}
 & \text{paren}((\phi_1 \square \phi_2)) \\
 = & \{ \text{definición} \} \\
 & \text{paren}(\phi_1) + \text{paren}(\phi_2) + 2 \\
 = & \{ \text{por h.i. en ambas sub-fórmulas} \} \\
 & 2 * \text{con}(\phi_1) + 2 * \text{con}(\phi_2) + 2 \\
 = & \{ \text{distributividad} \} \\
 & 2 * (\text{con}(\phi_1) + \text{con}(\phi_2) + 1) \\
 = & \{ \text{definición} \} \\
 & 2 * \text{con}((\phi_1 \square \phi_2))
 \end{aligned}$$