

## CLASE 4 - 25/02/2025

### Lógica proposicional

#### Definición (proposición)

Una oración de la cual podemos decir si es verdadera o falsa

#### Definición (proposición simple)

Veamos ejemplos para definir una proposición simple:

- Ayer llovió en Paysandú
- El Sol gira alrededor de la Tierra
- $2 \times 3 = 3 + 3$
- 3 es primo
- El sucesor de 3 es primo

#### Definición (proposición compuesta)

- Si ayer llovió en Paysandú, entonces el Sol gira alrededor de la Tierra
- El sol gira alrededor de la Tierra o la Tierra gira alrededor del Sol
- $2 \times 3 = 6$  y 6 es impar
- 3 no es primo
- Hay un número natural que es par y es primo
- Todo entero par mayor que 4 es la suma de dos números primos

Estas proposiciones, son básicamente una combinación de proposiciones simples. La forma de combinarlas no es única, depende de que forma las quiero relacionar.

### Razonamientos

Queremos saber si cierta conclusión (una proposición) se desprende de un conjunto de hipótesis (un conjunto de proposiciones).

**Razonamiento válido** Siempre que las hipótesis sean verdaderas, la conclusión también lo será.

**Razonamiento inválido** Si es posible obtener conclusiones falsas, a partir de las hipótesis verdadera.

### ¿Cómo analizar un razonamiento?

Para analizar si un razonamiento es válido o no, tenemos dos enfoques:

1. **Semántico:** Investiga si la verdad de las hipótesis implica la verdad de la conclusión.
2. **Prueba/Sintáctico:** Investiga la existencia de un objeto formal que encadene las hipótesis con la conclusión usando otras proposiciones.

## Traducción a $PROP$

Las proposiciones simples, se traducen como letras de proposición (elementos de  $P$ ), por ejemplo:

- Ayer llovió en Paysandú  $\Rightarrow p_0$
- El Sol gira alrededor de la Tierra  $\Rightarrow p_1$
- $2 \times 3 = 3 + 3 \Rightarrow p_2$
- 6 es primo  $\Rightarrow p_3$
- El sucesor de 3 es primo  $\Rightarrow p_4$

Las proposiciones compuestas, se traducen usando los conectivos:

- Si ayer llovió en Paysandú, entonces el Sol gira alrededor de la Tierra  $\Rightarrow p_0 \rightarrow p_1$
- $2 \times 3 = 6$  y 6 es primo  $\Rightarrow p_2 \wedge p_3$
- 6 no es primo  $\Rightarrow \neg p_3$

Algunas proposiciones no tienen una buena representación en  $PROP$ :

- Hay un número natural que es par y es primo
- Todo entero par mayor que 4 es la suma de dos números primos

Más adelante, trabajaremos con un lenguaje más expresivo para abordar este tipo de problemas.

## Definición (alfabeto $\Sigma_{PROP}$ )

El alfabeto del lenguaje de la lógica proposicional  $\Sigma_{PROP} := P \cup C \cup A$  consiste de:

- el conjunto de las letras proposicionales:  $P := \{p_0, p_1, p_2, \dots\}$
- el conjunto de los conectivos:  $C := C_0 \cup C_1 \cup C_2$  donde:
  - $C_0$  es el conjunto de los conectivos nulos:  $\{\perp\}$ , donde  $\perp$  representa siempre un valor falso
  - $C_1$  es el conjunto de los conectivos unarios:  $\{\neg\}$
  - $C_2$  es el conjunto de los conectivos binarios:  $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- el conjunto de los auxiliares:  $A := \{(\,,\,)\}$

## Definición (lenguaje $PROP$ )

El lenguaje de la lógica proposicional  $PROP \subseteq \Sigma_{PROP}^*$  está definido inductivamente por:

1. Si  $p \in P$ , entonces  $p \in PROP$
2.  $\perp \in PROP$
3. Si  $\alpha, \beta \in PROP$ , entonces:
  - $(\alpha \wedge \beta) \in PROP$
  - $(\alpha \vee \beta) \in PROP$
  - $(\alpha \rightarrow \beta) \in PROP$
  - $(\alpha \leftrightarrow \beta) \in PROP$
4. Si  $\alpha \in PROP$ , entonces  $(\neg \alpha) \in PROP$

## Nomenclatura

- **Fórmulas proposicionales:** son las palabras de  $PROP$

- **Fórmulas atómicas:** son los elementos del conjunto  $AT = P \cup \{\perp\}$ . Son precisamente las palabras formadas por las reglas básicas (i) y (ii) de la definición de  $PROP$ .
- **Metavariables:**
  - Usaremos  $p, q, r, p', \dots$  para las letras proposicionales
  - Usaremos  $\alpha, \beta, \varphi, \psi, \dots$  para las fórmulas proposicionales
  - Usaremos  $\Gamma, \Delta, \dots$  para conjuntos de fórmulas proposicionales

### PIP para $PROP$

Sea  $\mathcal{P}$  una propiedad sobre las palabras de  $PROP$  que cumple:

*BASE1:*  $\mathcal{P}(p)$  para todo  $p \in P$  *BASE2:* Se cumple  $\mathcal{P}(\perp)$  *IND1:* Para todo  $*$   $\in C_2$  y  $\alpha, \beta \in PROP$  que cumplen  $\mathcal{P}(\alpha)$  y  $\mathcal{P}(\beta)$ , se cumple  $\mathcal{P}((\alpha * \beta))$  *IND2:* Para todo  $\alpha \in PROP$  que cumple  $\mathcal{P}(\alpha)$ , se cumple  $\mathcal{P}(\neg\alpha)$

Entonces,  $\mathcal{P}$  se cumple para todas las palabras de  $PROP$ .

### ERP para $PROP$

Sea  $B$  un conjunto, y:

1. una función  $H_{AT} : AT \rightarrow B$ , y
2. para cada conector  $*$   $\in C_2$ , una función  $H_* : PROP \times B \times PROP \times B \rightarrow B$ , y
3. una función  $H_{\neg} : PROP \times B \rightarrow B$

Entonces, existe una única función  $F : PROP \rightarrow B$  tal que:

1.  $F(\alpha) = H_{AT}(\alpha)$  con  $\alpha \in AT$
2.  $F((\alpha * \beta)) = H_*(\alpha, F(\alpha), \beta, F(\beta))$  con  $\alpha, \beta \in PROP$
3.  $F(\neg\alpha) = H_{\neg}(\alpha, F(\alpha))$  con  $\alpha \in PROP$

### Ejemplos de ERP en $PROP$

$LARGO : PROP \rightarrow \mathbb{N}$  Veamos dos formas de definirlo, una más informal pero más usada, y luego la más formal que se adapta a la definición de ERP.

#### 1. Versión 1:

1.  $LARGO(\varphi) = 1$  con  $\varphi \in AT$
2.  $LARGO((\alpha * \beta)) = 3 + LARGO(\alpha) + LARGO(\beta)$  con  $\alpha, \beta \in PROP$
3.  $LARGO(\neg\alpha) = 3 + LARGO(\alpha)$  con  $\alpha \in PROP$

#### 2. Versión 2:

1.  $H_{AT}(\varphi) = 1$  con  $\varphi \in AT$
2.  $H_*(\alpha, n, \beta, m) = 3 + n + m$  con  $\alpha, \beta \in PROP$
3.  $H_{\neg}(\alpha, n) = 3 + n$  con  $\alpha \in PROP$

Observemos que la versión 2 nos define las funciones  $H$ , aparte de ellas tenemos que definir  $LARGO$  usandolas para terminar la definición. Esto es idéntico a como trabajamos los casos de ERP en el tema anterior.

$ATOMS : PROP \rightarrow 2^{AT}$  En este ejemplo solo veremos la forma más directa, se entiende que todos los casos que vemos tienen la forma de ERP.

1.  $ATOMS(\varphi) = \{\varphi\}$  con  $\varphi \in AT$

Propiedad 1 de arbol

Figure 1: Propiedad 1 de arbol

Propiedad 2 de arbol

Figure 2: Propiedad 2 de arbol

2.  $ATOMS((\alpha * \beta)) = ATOMS(\alpha) \cup ATOMS(\beta)$  con  $\alpha, \beta \in PROP$
3.  $ATOMS((\neg \alpha)) = ATOMS(\alpha)$  con  $\alpha \in PROP$

### Árboles etiquetados y ordenados

Consideramos a  $\mathcal{T}(\mathcal{L})$  de los árboles etiquetados con palabras del lenguaje  $\mathcal{L}$ .

#### Propiedades

- Cada nodo tiene como máximo un padre, si no tiene un padre, entonces es la raíz del árbol
- Cada nodo tiene un primer hijo, un segundo hijo, etc..., ordenados de izquierda a derecha. Si no tiene hijos, es una hoja del árbol
- A cada nodo se le etiqueta con una palabra de  $\mathcal{L}$

Ahora veamos otro ejemplo de ERP sobre  $PROP$ , incluyendo este lenguaje de árboles etiquetados:

$RBOL : PROP \rightarrow \mathcal{T}(PROP)$

1.  $\$ÁRBOL( ) = \$$   
con  $\varphi \in AT$
2.  $\$ÁRBOL(( * )) = \$$   
con  $\alpha, \beta \in PROP$
3.  $\$ÁRBOL((\neg )) = \$$   
con  $\alpha \in PROP$

O visto de otra forma:

1.  $H_{AT} : AT \rightarrow \mathcal{T}(PROP)$ 
  - $H_{AT} := \dots$
2.  $H_* : PROP \times \mathcal{T}(PROP) \times PROP \times \mathcal{T}(PROP) \rightarrow \mathcal{T}(PROP)$ 
  - $H_*(\alpha, t_1, \beta, t_2) := \dots$
3.  $H_{\neg} : PROP \times \mathcal{T}(PROP) \rightarrow \mathcal{T}(PROP)$ 
  - $H_{\neg}(\alpha, t) := \dots$

Propiedad 3 de arbol

Figure 3: Propiedad 3 de arbol