

# Introducción a la Lógica y la Computación

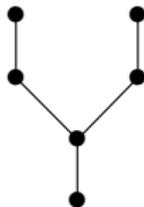
Mariana Badano   Héctor Gramaglia   Pedro Sánchez Terraf  
M. Clara Gorín   Matías Steinberg

FaMAF, 2 de octubre de 2020



# Ejes de Contenidos

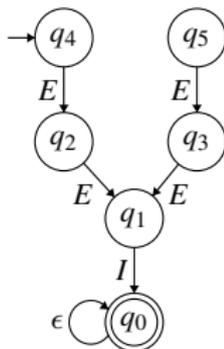
## 1 Estructuras Ordenadas



## 2 Lógica Proposicional

$$\frac{\frac{[\varphi \wedge \psi]_1 \wedge E}{\psi} \quad \frac{[\varphi \wedge \psi]_1 \wedge E}{\varphi}}{\psi \wedge \varphi} \wedge I$$
$$\frac{\psi \wedge \varphi}{\varphi \wedge \psi \rightarrow \psi \wedge \varphi} \rightarrow I_1$$

## 3 Lenguajes y Autómatas



# Parte 2: Lógica Proposicional

- Dirk van Dalen, *Logic and Structure*, 3ra edición (Springer).
- PST, *Apunte de Lógica Proposicional*.

## 1 Componentes de la lógica proposicional

### 2 Sintaxis

- El lenguaje de la lógica
- Inducción y recursión
- Recursión en *PROP*

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

$$\begin{aligned} & \underline{p \Rightarrow q \vee p} \\ \equiv & \{ \text{Definición de } \Rightarrow \} \\ & \underline{p \vee q \vee p} \equiv \underline{q \vee q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee, \text{ Idempotencia } \vee \} \\ & \underline{p \vee q} \equiv \underline{q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee \} \\ & \textit{True} \end{aligned}$$

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

$$\begin{aligned} & \underline{p \Rightarrow q \vee p} \\ \equiv & \{ \text{Definición de } \Rightarrow \} \\ & \underline{p \vee q \vee p} \equiv \underline{q \vee q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee, \text{ Idempotencia } \vee \} \\ & \underline{p \vee q} \equiv \underline{q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee \} \\ & \textit{True} \end{aligned}$$

## Preguntas...

**1** ¿Qué *demuestra* esto?

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

$$\begin{aligned} & \underline{p \Rightarrow q \vee p} \\ \equiv & \{ \text{Definición de } \Rightarrow \} \\ & \underline{p \vee q \vee p} \equiv \underline{q \vee q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee, \text{ Idempotencia } \vee \} \\ & \underline{p \vee q} \equiv \underline{q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee \} \\ & \textit{True} \end{aligned}$$

## Preguntas...

- 1 ¿Qué *demuestra* esto? ¿Qué es **demostrar**?

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

$$\begin{aligned} & \underline{p \Rightarrow q \vee p} \\ \equiv & \{ \text{Definición de } \Rightarrow \} \\ & \underline{p \vee q \vee p} \equiv \underline{q \vee q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee, \text{ Idempotencia } \vee \} \\ & \underline{p \vee q} \equiv \underline{q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee \} \\ & \textit{True} \end{aligned}$$

## Preguntas...

- 1 ¿Qué *demuestra* esto? ¿Qué es **demostrar**?
- 2 ¿Qué es *True*?

# Proposicional en “Introducción a los Algoritmos”

En primer año de la carrera se hacía algo como esto:

$$\begin{aligned} & \underline{p \Rightarrow q \vee p} \\ \equiv & \{ \text{Definición de } \Rightarrow \} \\ & \underline{p \vee q \vee p} \equiv \underline{q \vee q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee, \text{ Idempotencia } \vee \} \\ & \underline{p \vee q} \equiv \underline{q \vee p} \\ \equiv & \{ \text{Conmutativa } \vee \} \\ & \textit{True} \end{aligned}$$

## Preguntas...

- 1 ¿Qué *demuestra* esto? ¿Qué es **demostrar**?
- 2 ¿Qué es *True*?
- 3 ¿Qué son  $p$  y  $q$ ?

# Tres componentes de la lógica

# Tres componentes de la lógica

## Sintaxis

Qué objetos usamos: **proposiciones**, cómo se escriben.

# Tres componentes de la lógica

## Sintaxis

Qué objetos usamos: **proposiciones**, cómo se escriben.

## Semántica

Cómo asignamos *significado* a las proposiciones: **valor de verdad**.

# Tres componentes de la lógica

## Sintaxis

Qué objetos usamos: **proposiciones**, cómo se escriben.

## Semántica

Cómo asignamos *significado* a las proposiciones: **valor de verdad**.

## Cálculo

Cómo se **deducen** proposiciones a partir de otras y se obtienen **teoremas**

# Tres componentes de la lógica

## Sintaxis

Qué objetos usamos: **proposiciones**, cómo se escriben.

## Semántica

Cómo asignamos *significado* a las proposiciones: **valor de verdad**.

## Cálculo

Cómo se **deducen** proposiciones a partir de otras y se obtienen **teoremas**

Estudiaremos especialmente la interrelación entre los dos últimos conceptos.

## Proposiciones

$(p_1 \wedge p_2)$ ,

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1),$

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)),$

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: **cadena de símbolos**

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: **cadena de símbolos**

String

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: **cadena de símbolos**

String

Los símbolos que usaremos:

$\Sigma := \{ \}, (, \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: **cadena de símbolos**

Los símbolos que usaremos:

$\Sigma := \{ \}, (, \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

String



## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \}, (, \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String



## Proposiciones

$$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$$\Sigma := \{ \text{ ), (, } \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}.$$

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

### Ejemplo

$$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}.$$

String



$\Sigma^*$

# Sintaxis: El lenguaje

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \}, (, \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String

$\updownarrow$   
 $\Sigma^*$

## Ejemplo

$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}$ .

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

# Sintaxis: El lenguaje

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \text{ ), (, } \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String

$\updownarrow$   
 $\Sigma^*$

## Ejemplo

$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}$ .

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\boxed{\varphi \in At}$  Para todo  $\varphi \in At, \varphi \in PROP$ .

# Sintaxis: El lenguaje

## Proposiciones

$$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$$\Sigma := \{ \text{ ), (, } \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}.$$

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String



## Ejemplo

$$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}.$$

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\boxed{\varphi \in At}$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$\boxed{(\varphi \rightarrow \psi)}$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \rightarrow \psi)$  está en *PROP*.

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \}, (, \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String



## Ejemplo

$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}$ .

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\boxed{\varphi \in At}$  Para todo  $\varphi \in At, \varphi \in PROP$ .

$\boxed{(\varphi \vee \psi)}$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \vee \psi)$  está en *PROP*.

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \text{ ), (, } \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String



## Ejemplo

$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}$ .

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\boxed{\varphi \in At}$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$\boxed{(\varphi \wedge \psi)}$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \wedge \psi)$  está en *PROP*.

## Proposiciones

$(p_1 \wedge p_2), (p_2 \vee p_1), (p_0 \rightarrow (p_5 \vee p_0)), \dots$

Tomémoslas por lo que son: cadenas de símbolos

Los símbolos que usaremos:

$\Sigma := \{ \text{ ), (, } \wedge, \vee, \rightarrow, \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$ .

Llamaremos **átomos** al subconjunto  $\{ \perp, p_0, p_1, \dots, p_n, p_{n+1}, \dots \}$  de  $\Sigma$ .

String



## Ejemplo

$(p_1 \wedge p_2) = \boxed{(} \boxed{p_1} \boxed{\wedge} \boxed{p_2} \boxed{)}$ .

## Definición

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\boxed{\varphi \in At}$  Para todo  $\varphi \in At, \varphi \in PROP$ .

$\boxed{(\varphi \odot \psi)}$  todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

# Inducción en *PROP*

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

# Inducción en *PROP*

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (inducción en subfórmulas)

Sea *A* un predicado sobre *PROP*. Luego  $A(\varphi)$  es verdadero para toda  $\varphi \in PROP$  si y sólo si:

# Inducción en *PROP*

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (inducción en subfórmulas)

Sea *A* un predicado sobre *PROP*. Luego  $A(\varphi)$  es verdadero para toda  $\varphi \in PROP$  si y sólo si:

$\varphi \in At$  Si  $\varphi$  es atómica,  $A(\varphi)$  vale.

$(\varphi \odot \psi)$  Si  $A(\varphi)$  y  $A(\psi)$  entonces  $A((\varphi \odot \psi))$ .

# Inducción en *PROP*

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (inducción en subfórmulas)

Sea  $A$  un predicado sobre *PROP*. Luego  $A(\varphi)$  es verdadero para toda  $\varphi \in PROP$  si y sólo si:

$\varphi \in At$  Si  $\varphi$  es atómica,  $A(\varphi)$  vale.

$(\varphi \odot \psi)$  Si  $A(\varphi)$  y  $A(\psi)$  entonces  $A((\varphi \odot \psi))$ .

## Demostración.

Sea  $X = \{\varphi \in PROP : A(\varphi)\}$ .

# Inducción en *PROP*

*PROP* es el menor subconjunto de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (inducción en subfórmulas)

Sea *A* un predicado sobre *PROP*. Luego  $A(\varphi)$  es verdadero para toda  $\varphi \in PROP$  si y sólo si:

$\varphi \in At$  Si  $\varphi$  es atómica,  $A(\varphi)$  vale.

$(\varphi \odot \psi)$  Si  $A(\varphi)$  y  $A(\psi)$  entonces  $A((\varphi \odot \psi))$ .

## Demostración.

Sea  $X = \{\varphi \in PROP : A(\varphi)\}$ . Luego  $X \subseteq PROP \subseteq \Sigma^*$ .

# Inducción en *PROP*

*PROP* es el **menor subconjunto** de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (inducción en subfórmulas)

Sea  $A$  un predicado sobre *PROP*. Luego  $A(\varphi)$  es verdadero para toda  $\varphi \in PROP$  si y sólo si:

$\varphi \in At$  Si  $\varphi$  es atómica,  $A(\varphi)$  vale.

$(\varphi \odot \psi)$  Si  $A(\varphi)$  y  $A(\psi)$  entonces  $A((\varphi \odot \psi))$ .

## Demostración.

Sea  $X = \{\varphi \in PROP : A(\varphi)\}$ . Luego  $X \subseteq PROP \subseteq \Sigma^*$ .

Y además  $PROP \subseteq X$  por **minimalidad**.

## Definición

Una sucesión de proposiciones  $\varphi_1, \dots, \varphi_n$  es una **serie de formación** (sdf) de  $\varphi \in PROP$  si  $\varphi_n = \varphi$  y para todo  $i \leq n$ ,  $\varphi_i$  es:

- atómica, o bien
- igual a  $(\varphi_j \odot \varphi_k)$  con  $j, k < i$ .

## Definición

Una sucesión de proposiciones  $\varphi_1, \dots, \varphi_n$  es una **serie de formación** (sdf) de  $\varphi \in PROP$  si  $\varphi_n = \varphi$  y para todo  $i \leq n$ ,  $\varphi_i$  es:

- atómica, o bien
- igual a  $(\varphi_j \odot \varphi_k)$  con  $j, k < i$ .

## Teorema

*Toda  $\varphi \in PROP$  tiene una serie de formación.*

# Inducción en *PROP*

## Definición

Una sucesión de proposiciones  $\varphi_1, \dots, \varphi_n$  es una **serie de formación** (sdf) de  $\varphi \in PROP$  si  $\varphi_n = \varphi$  y para todo  $i \leq n$ ,  $\varphi_i$  es:

- atómica, o bien
- igual a  $(\varphi_j \odot \varphi_k)$  con  $j, k < i$ .

## Teorema

*Toda  $\varphi \in PROP$  tiene una serie de formación.*

## Demostración.

$\varphi \in At$  “ $\varphi$ ” es una sdf de  $\varphi$  (tenemos  $n = 1, \varphi_1 := \varphi$ ).

# Inducción en *PROP*

## Definición

Una sucesión de proposiciones  $\varphi_1, \dots, \varphi_n$  es una **serie de formación** (sdf) de  $\varphi \in PROP$  si  $\varphi_n = \varphi$  y para todo  $i \leq n$ ,  $\varphi_i$  es:

- atómica, o bien
- igual a  $(\varphi_j \odot \varphi_k)$  con  $j, k < i$ .

## Teorema

*Toda  $\varphi \in PROP$  tiene una serie de formación.*

## Demostración.

$\varphi \in At$  “ $\varphi$ ” es una sdf de  $\varphi$  (tenemos  $n = 1$ ,  $\varphi_1 := \varphi$ ).

$(\varphi \odot \psi)$  Por HI,  $\varphi$  y  $\psi$  tienen sdf  $\varphi_1, \dots, \varphi_n (= \varphi)$  y  $\psi_1, \dots, \psi_m (= \psi)$ .

## Definición

Una sucesión de proposiciones  $\varphi_1, \dots, \varphi_n$  es una **serie de formación** (sdf) de  $\varphi \in PROP$  si  $\varphi_n = \varphi$  y para todo  $i \leq n$ ,  $\varphi_i$  es:

- atómica, o bien
- igual a  $(\varphi_j \odot \varphi_k)$  con  $j, k < i$ .

## Teorema

*Toda  $\varphi \in PROP$  tiene una serie de formación.*

## Demostración.

$\varphi \in At$  “ $\varphi$ ” es una sdf de  $\varphi$  (tenemos  $n = 1$ ,  $\varphi_1 := \varphi$ ).

$(\varphi \odot \psi)$  Por HI,  $\varphi$  y  $\psi$  tienen sdf  $\varphi_1, \dots, \varphi_n (= \varphi)$  y  $\psi_1, \dots, \psi_m (= \psi)$ .

Luego  $\varphi_1, \dots, \varphi_n, \psi_1, \dots, \psi_m, (\varphi \odot \psi)$  es sdf de  $(\varphi \odot \psi)$ .

# Recursión en *PROP*

*PROP* es el **menor subconjunto** de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

# Recursión en *PROP*

*PROP* es el **menor subconjunto** de  $\Sigma^*$  que cumple con:

$\varphi \in At$  Para todo  $\varphi \in At$ ,  $\varphi \in PROP$ .

$(\varphi \odot \psi)$  Para todas  $\varphi, \psi$  en *PROP*,  $(\varphi \odot \psi)$  está en *PROP*.

## Teorema (definición por recursión en subfórmulas)

Sea  $A$  un conjunto y supongamos dadas funciones

$H_{At} : At \rightarrow A$  y  $H_{\odot} : A^2 \rightarrow A$  para cada  $\odot$ .

Entonces hay exactamente una función  $F : PROP \rightarrow A$  tal que

$$\begin{cases} F(\varphi) & = H_{At}(\varphi) \text{ para } \varphi \text{ en } At \\ F((\varphi \odot \psi)) & = H_{\odot}(F(\varphi), F(\psi)) \end{cases}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$gr(((p_0 \wedge p_3) \rightarrow p_2)) = \max\{gr((p_0 \wedge p_3)), gr(p_2)\} \quad \text{caso “}\odot\text{”}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$\begin{aligned} gr(((p_0 \wedge p_3) \rightarrow p_2)) &= \max\{gr((p_0 \wedge p_3)), gr(p_2)\} && \text{caso “}\odot\text{”} \\ &= \max\{gr((p_0 \wedge p_3)), 2\} && \text{caso “}At\text{”} \end{aligned}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$\begin{aligned} gr(((p_0 \wedge p_3) \rightarrow p_2)) &= \max\{gr((p_0 \wedge p_3)), gr(p_2)\} && \text{caso “}\odot\text{”} \\ &= \max\{gr((p_0 \wedge p_3)), 2\} && \text{caso “}At\text{”} \\ &= \max\{\max\{gr(p_0), gr(p_3)\}, 2\} && \text{caso “}\odot\text{”} \end{aligned}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$\begin{aligned} gr(((p_0 \wedge p_3) \rightarrow p_2)) &= \max\{gr((p_0 \wedge p_3)), gr(p_2)\} && \text{caso “}\odot\text{”} \\ &= \max\{gr((p_0 \wedge p_3)), 2\} && \text{caso “}At\text{”} \\ &= \max\{\max\{gr(p_0), gr(p_3)\}, 2\} && \text{caso “}\odot\text{”} \\ &= \max\{\max\{0, 3\}, 2\} && \text{caso “}At\text{”} \end{aligned}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$\begin{aligned} gr(((p_0 \wedge p_3) \rightarrow p_2)) &= \max\{gr((p_0 \wedge p_3)), gr(p_2)\} && \text{caso “}\odot\text{”} \\ &= \max\{gr((p_0 \wedge p_3)), 2\} && \text{caso “}At\text{”} \\ &= \max\{\max\{gr(p_0), gr(p_3)\}, 2\} && \text{caso “}\odot\text{”} \\ &= \max\{\max\{0, 3\}, 2\} && \text{caso “}At\text{”} \\ &= \max\{3, 2\} && \text{def de máx} \end{aligned}$$

# Recursión en *PROP*

Ejemplo de definición por recursión:

## Definición

El *grado* de una proposición,  $gr(\cdot)$ , es la función definida de la siguiente manera.

$$\boxed{\varphi \in At} \quad gr(p_n) := n; \quad gr(\perp) := -1.$$

$$\boxed{(\varphi \odot \psi)} \quad gr((\varphi \odot \psi)) := \max\{gr(\varphi), gr(\psi)\}.$$

$$\begin{aligned} gr(((p_0 \wedge p_3) \rightarrow p_2)) &= \max\{gr((p_0 \wedge p_3)), gr(p_2)\} && \text{caso “}\odot\text{”} \\ &= \max\{gr((p_0 \wedge p_3)), 2\} && \text{caso “}At\text{”} \\ &= \max\{\max\{gr(p_0), gr(p_3)\}, 2\} && \text{caso “}\odot\text{”} \\ &= \max\{\max\{0, 3\}, 2\} && \text{caso “}At\text{”} \\ &= \max\{3, 2\} && \text{def de máx} \\ &= 3 && \text{def de máx} \end{aligned}$$