

Introducción a la Lógica y la Computación

Mariana Badano Héctor Gramaglia
Pedro Sánchez Terraf Mauricio Tellechea
Guido Ivetta

FaMAF, 15 de septiembre de 2021

- 1 Repaso
- 2 Semántica de la lógica proposicional
 - Asignaciones y valuaciones/semánticas
 - Teorema de Extensión
 - Abreviaciones: Conectivos nuevos
 - La relación de consecuencia y tautologías
 - Lema de Coincidencia
 - Tablas de verdad
- 3 Sustitución
 - La regla de Leibnitz

Tres componentes de la lógica

- **Sintaxis**: qué objetos usamos: **proposiciones** (= “fórmulas proposicionales”, “fórmulas”), cómo se escriben.

Tres componentes de la lógica

- **Sintaxis**: qué objetos usamos: **proposiciones** (= “fórmulas proposicionales”, “fórmulas”), cómo se escriben.
 - **Símbolos/variables proposicionales**: $\mathcal{V} := \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$
 - **Conectivos**: $\perp, \wedge, \vee, \rightarrow$.
 - $At := \{\perp\} \cup \mathcal{V}$; $\Sigma := At \cup \{ \ , (, \wedge, \vee, \rightarrow \}$; $PROP \subseteq \Sigma^*$.

Tres componentes de la lógica

- **Sintaxis**: qué objetos usamos: **proposiciones** (= “fórmulas proposicionales”, “fórmulas”), cómo se escriben.
 - **Símbolos/variables proposicionales**: $\mathcal{V} := \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$
 - **Conectivos**: $\perp, \wedge, \vee, \rightarrow$.
 - $At := \{\perp\} \cup \mathcal{V}$; $\Sigma := At \cup \{ , (, \wedge, \vee, \rightarrow \}$; $PROP \subseteq \Sigma^*$.
- **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** (= “fórmulas proposicionales”, “fórmulas”), cómo se escriben.
 - **Símbolos/variables proposicionales**: $\mathcal{V} := \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$
 - **Conectivos**: $\perp, \wedge, \vee, \rightarrow$.
 - **At** := $\{\perp\} \cup \mathcal{V}$; $\Sigma := At \cup \{ , (, \wedge, \vee, \rightarrow \}$; $PROP \subseteq \Sigma^*$.
- **Semántica**: cómo asignamos *significado* a las proposiciones: **valor de verdad**. Ahora
- **Cálculo**: cómo se **deducen** proposiciones a partir de otras y se obtienen **teoremas**. Después

Tres componentes de la lógica

- **Sintaxis**: qué objetos usamos: **proposiciones** (= “fórmulas proposicionales”, “fórmulas”), cómo se escriben.
 - **Símbolos/variables proposicionales**: $\mathcal{V} := \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$
 - **Conectivos**: $\perp, \wedge, \vee, \rightarrow$.
 - $At := \{\perp\} \cup \mathcal{V}$; $\Sigma := At \cup \{ \text{ }, (, \wedge, \vee, \rightarrow \}$; $PROP \subseteq \Sigma^*$.
- **Semántica**: cómo asignamos *significado* a las proposiciones: **valor de verdad**. Ahora
- **Cálculo**: cómo se **deducen** proposiciones a partir de otras y se obtienen **teoremas**.

Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Para “interpretarlas”, diremos para cada símbolo proposicional en \mathcal{V} si es “falso” (valor 0) o “verdadero” (valor 1).

Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Para “interpretarlas”, diremos para cada símbolo proposicional en \mathcal{V} si es “falso” (valor 0) o “verdadero” (valor 1).

Definición

Una **asignación** será una función $v : \{p_0, p_1, \dots\} \rightarrow \{0, 1\}$.



Universidad
Nacional
de Córdoba



Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Para “interpretarlas”, diremos para cada símbolo proposicional en \mathcal{V} si es “falso” (valor 0) o “verdadero” (valor 1).

Definición

Una **asignación** será una función $v : \{p_0, p_1, \dots\} \rightarrow \{0, 1\}$.

Ahora podemos dar significado a todas las proposiciones.

Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Para “interpretarlas”, diremos para cada símbolo proposicional en \mathcal{V} si es “falso” (valor 0) o “verdadero” (valor 1).

Definición

Una **asignación** será una función $v : \{p_0, p_1, \dots\} \rightarrow \{0, 1\}$.

Ahora podemos dar significado a todas las proposiciones.

Definición

Una función $\llbracket \cdot \rrbracket : PROP \rightarrow \{0, 1\}$ es una **semántica** o **valuación** si:

- 1 $\llbracket \perp \rrbracket = 0$.
- 2 $\llbracket (\varphi \wedge \psi) \rrbracket = \min\{\llbracket \varphi \rrbracket, \llbracket \psi \rrbracket\}$.
- 3 $\llbracket (\varphi \vee \psi) \rrbracket = \max\{\llbracket \varphi \rrbracket, \llbracket \psi \rrbracket\}$.
- 4 $\llbracket (\varphi \rightarrow \psi) \rrbracket = 0$ si y sólo si $\llbracket \varphi \rrbracket = 1$ y $\llbracket \psi \rrbracket = 0$.

Asignaciones y valuaciones/semánticas

Nuestras proposiciones son sólo cadenas de símbolos.

Para “interpretarlas”, diremos para cada símbolo proposicional en \mathcal{V} si es “falso” (valor 0) o “verdadero” (valor 1).

Definición

Una **asignación** será una función $v : \{p_0, p_1, \dots\} \rightarrow \{0, 1\}$.

Ahora podemos dar significado a todas las proposiciones.

Definición

Una función $\llbracket \cdot \rrbracket : PROP \rightarrow \{0, 1\}$ es una **semántica** o **valuación** si:

- 1 $\llbracket \perp \rrbracket = 0$. $\leftarrow \leftarrow$ Develado el misterio de \perp !!
- 2 $\llbracket (\varphi \wedge \psi) \rrbracket = \min\{\llbracket \varphi \rrbracket, \llbracket \psi \rrbracket\}$.
- 3 $\llbracket (\varphi \vee \psi) \rrbracket = \max\{\llbracket \varphi \rrbracket, \llbracket \psi \rrbracket\}$.
- 4 $\llbracket (\varphi \rightarrow \psi) \rrbracket = 0$ si y sólo si $\llbracket \varphi \rrbracket = 1$ y $\llbracket \psi \rrbracket = 0$.

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Existencia: Construimos la semántica $\llbracket \cdot \rrbracket_f$ por recursión en subfórmulas.

$$\boxed{\varphi \in At} \quad \llbracket p_n \rrbracket_f := f(p_n) \text{ para } n \in \mathbb{N}_0 \text{ y } \llbracket \perp \rrbracket_f := 0.$$

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Existencia: Construimos la semántica $\llbracket \cdot \rrbracket_f$ por recursión en subfórmulas.

$$\boxed{\varphi \in At} \quad \llbracket p_n \rrbracket_f := f(p_n) \text{ para } n \in \mathbb{N}_0 \text{ y } \llbracket \perp \rrbracket_f := 0.$$

$$\boxed{(\varphi \wedge \psi)} \quad \llbracket (\varphi \wedge \psi) \rrbracket_f := \min\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Existencia: Construimos la semántica $\llbracket \cdot \rrbracket_f$ por recursión en subfórmulas.

$$\boxed{\varphi \in At} \quad \llbracket p_n \rrbracket_f := f(p_n) \text{ para } n \in \mathbb{N}_0 \text{ y } \llbracket \perp \rrbracket_f := 0.$$

$$\boxed{(\varphi \wedge \psi)} \quad \llbracket (\varphi \wedge \psi) \rrbracket_f := \min\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

$$\boxed{(\varphi \rightarrow \psi)} \quad \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 0 \text{ si } \llbracket \varphi \rrbracket_f = 1 \text{ y } \llbracket \psi \rrbracket_f = 0, \text{ y } \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 1 \text{ en caso contrario.}$$

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Existencia: Construimos la semántica $\llbracket \cdot \rrbracket_f$ por recursión en subfórmulas.

$$\boxed{\varphi \in At} \quad \llbracket p_n \rrbracket_f := f(p_n) \text{ para } n \in \mathbb{N}_0 \text{ y } \llbracket \perp \rrbracket_f := 0.$$

$$\boxed{(\varphi \wedge \psi)} \quad \llbracket (\varphi \wedge \psi) \rrbracket_f := \min\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

$$\boxed{(\varphi \rightarrow \psi)} \quad \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 0 \text{ si } \llbracket \varphi \rrbracket_f = 1 \text{ y } \llbracket \psi \rrbracket_f = 0, \text{ y } \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 1 \text{ en caso contrario.}$$

$$\boxed{(\varphi \vee \psi)} \quad \llbracket (\varphi \vee \psi) \rrbracket_f := \max\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

Extensión de asignaciones

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Existencia: Construimos la semántica $\llbracket \cdot \rrbracket_f$ por recursión en subfórmulas.

$$\boxed{\varphi \in At} \quad \llbracket p_n \rrbracket_f := f(p_n) \text{ para } n \in \mathbb{N}_0 \text{ y } \llbracket \perp \rrbracket_f := 0.$$

H_{At}

$$\boxed{(\varphi \wedge \psi)} \quad \llbracket (\varphi \wedge \psi) \rrbracket_f := \min\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

H_{\wedge}

$$\boxed{(\varphi \rightarrow \psi)} \quad \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 0 \text{ si } \llbracket \varphi \rrbracket_f = 1 \text{ y } \llbracket \psi \rrbracket_f = 0, \text{ y } \llbracket (\varphi \rightarrow \psi) \rrbracket_f := 1 \text{ en caso contrario.}$$

H_{\rightarrow}

$$\boxed{(\varphi \vee \psi)} \quad \llbracket (\varphi \vee \psi) \rrbracket_f := \max\{\llbracket \varphi \rrbracket_f, \llbracket \psi \rrbracket_f\}.$$

H_{\vee}

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Unicidad:

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Unicidad:

Por definición, una valuación debe cumplir con los casos inductivos de la definición recursiva anterior (las H_{\odot}).

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Unicidad:

Por definición, una valuación debe cumplir con los casos inductivos de la definición recursiva anterior (las H_{\odot}).

Y el caso base (la H_{At}) está fijado por la hipótesis y la definición de valuación en \perp .

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Demostración.

Unicidad:

Por definición, una valuación debe cumplir con los casos inductivos de la definición recursiva anterior (las H_{\odot}).

Y el caso base (la H_{At}) está fijado por la hipótesis y la definición de valuación en \perp .

Luego el Teorema de Recursión nos dice que hay a lo sumo una función que satisface todo esto.

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Corolario

$\llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in At \implies \llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in PROP$.

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Corolario

$\llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in At \implies \llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in PROP$.

Demostración.

Por la unicidad en el Teorema de Extensión

Teorema (de Extensión)

Para toda asignación f , existe una única función semántica $\llbracket \cdot \rrbracket_f$ tal que $\llbracket \varphi \rrbracket_f = f(\varphi)$ para toda $\varphi \in \mathcal{V}$.

Corolario

$\llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in At \implies \llbracket \varphi \rrbracket_1 = \llbracket \varphi \rrbracket_2$ para toda $\varphi \in PROP$.

Demostración.

Por la unicidad en el Teorema de Extensión: ambas valuaciones son extensiones de la misma asignación $\llbracket \cdot \rrbracket_1 \upharpoonright \mathcal{V} = \llbracket \cdot \rrbracket_2 \upharpoonright \mathcal{V}$.

Introducimos nueva notación.

Introducimos nueva notación.

Abreviaturas

- $(\neg\varphi)$ denotará $(\varphi \rightarrow \perp)$.
- $(\varphi \leftrightarrow \psi)$ denotará $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$.

Introducimos nueva notación.

Abreviaturas

- $(\neg\varphi)$ denotará $(\varphi \rightarrow \perp)$.
- $(\varphi \leftrightarrow \psi)$ denotará $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$.

Ejercicio

Para toda valuación $\llbracket \cdot \rrbracket$:

- 1 $\llbracket (\neg\varphi) \rrbracket = 1 - \llbracket \varphi \rrbracket$.
- 2 $\llbracket (\varphi \leftrightarrow \psi) \rrbracket = 1 \iff \llbracket \varphi \rrbracket = \llbracket \psi \rrbracket$.

La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

■ v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$.

La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

■ v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)



Universidad
Nacional
de Córdoba



La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

- v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)
- φ es **consecuencia** de $\Gamma \iff$ toda asignación v que valida Γ hace verdadera a φ :

$$\llbracket \Gamma \rrbracket_v = 1 \implies \llbracket \varphi \rrbracket_v = 1$$

La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

- v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)
- φ es **consecuencia** de $\Gamma \iff$ toda asignación v que valida Γ hace verdadera a φ :

$$\llbracket \Gamma \rrbracket_v = 1 \implies \llbracket \varphi \rrbracket_v = 1$$

(**notación:** $\Gamma \models \varphi$)



Universidad
Nacional
de Córdoba



La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

- v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)
- φ es **consecuencia** de $\Gamma \iff$ toda asignación v que valida Γ hace verdadera a φ :

$$\llbracket \Gamma \rrbracket_v = 1 \implies \llbracket \varphi \rrbracket_v = 1$$

(**notación:** $\Gamma \models \varphi$)

- φ es una **tautología** $\iff \llbracket \varphi \rrbracket_v = 1$ para toda asignación v .



Universidad
Nacional
de Córdoba



La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

- v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)
- φ es **consecuencia** de $\Gamma \iff$ toda asignación v que valida Γ hace verdadera a φ :

$$\llbracket \Gamma \rrbracket_v = 1 \implies \llbracket \varphi \rrbracket_v = 1$$

(**notación:** $\Gamma \models \varphi$)

- φ es una **tautología** $\iff \llbracket \varphi \rrbracket_v = 1$ para toda asignación v .
(**notación:** $\models \varphi$)



UNC
Universidad
Nacional
de Córdoba



La relación de consecuencia y tautologías

Sea $\Gamma \subseteq PROP$ y v una asignación.

Definición

- v **valida** $\Gamma \iff$ para toda $\psi \in \Gamma$, $\llbracket \psi \rrbracket_v = 1$. (**notación:** $\llbracket \Gamma \rrbracket_v = 1$)
- φ es **consecuencia** de $\Gamma \iff$ toda asignación v que valida Γ hace verdadera a φ :

$$\llbracket \Gamma \rrbracket_v = 1 \implies \llbracket \varphi \rrbracket_v = 1$$

(**notación:** $\Gamma \models \varphi$)

- φ es una **tautología** $\iff \llbracket \varphi \rrbracket_v = 1$ para toda asignación v .
(**notación:** $\models \varphi$)

Ejercicio

$$\models \varphi \iff \emptyset \models \varphi.$$



UNC

Universidad
Nacional
de Córdoba



Ejemplos

$$1 \models (\varphi \rightarrow \varphi).$$

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.

Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

Ejemplos

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.

Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).



Universidad
Nacional
de Córdoba



Ejemplos

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$



Universidad
Nacional
de Córdoba



1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$

Debemos ver que si v valida $\{\varphi, (\varphi \rightarrow \psi)\}$, entonces $\llbracket \psi \rrbracket_v = 1$

Ejemplos

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$

Debemos ver que si v valida $\{\varphi, (\varphi \rightarrow \psi)\}$, entonces $\llbracket \psi \rrbracket_v = 1$:

$$\llbracket \varphi \rrbracket_v = \llbracket (\varphi \rightarrow \psi) \rrbracket_v = 1 \implies \llbracket \psi \rrbracket_v = 1.$$

Ejemplos

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$

Debemos ver que si v valida $\{\varphi, (\varphi \rightarrow \psi)\}$, entonces $\llbracket \psi \rrbracket_v = 1$:

$$\llbracket \varphi \rrbracket_v = \llbracket (\varphi \rightarrow \psi) \rrbracket_v = 1 \implies \llbracket \psi \rrbracket_v = 1.$$

4 $\not\models p_1$

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$

Debemos ver que si v valida $\{\varphi, (\varphi \rightarrow \psi)\}$, entonces $\llbracket \psi \rrbracket_v = 1$:

$$\llbracket \varphi \rrbracket_v = \llbracket (\varphi \rightarrow \psi) \rrbracket_v = 1 \implies \llbracket \psi \rrbracket_v = 1.$$

4 $\not\models p_1$

Sale negando la definición

Ejemplos

1 $\models (\varphi \rightarrow \varphi).$

Tenemos que ver que para toda asignación v , $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v = 1$.
Equivalentemente, $\llbracket (\varphi \rightarrow \varphi) \rrbracket_v \neq 0$.

2 $\models ((\neg(\neg\varphi)) \rightarrow \varphi)$ (Ejercicio).

3 $\{\varphi, (\varphi \rightarrow \psi)\} \models \psi.$

Debemos ver que si v valida $\{\varphi, (\varphi \rightarrow \psi)\}$, entonces $\llbracket \psi \rrbracket_v = 1$:

$$\llbracket \varphi \rrbracket_v = \llbracket (\varphi \rightarrow \psi) \rrbracket_v = 1 \implies \llbracket \psi \rrbracket_v = 1.$$

4 $\not\models p_1$

Salte negando la definición:

p_1 **no** es una tautología \iff existe alguna v tal que $\llbracket p_1 \rrbracket_v = 0$.



Universidad
Nacional
de Córdoba



Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.



UNC
Universidad
Nacional
de Córdoba



Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\varphi \in At$ Si $\varphi = p_n$, sólo ocurre p_n en φ .



Universidad
Nacional
de Córdoba



Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\boxed{\varphi \in At}$ Si $\varphi = p_n$, sólo ocurre p_n en φ . Luego $\llbracket \varphi \rrbracket_v = v(\varphi) = v'(\varphi) = \llbracket \varphi \rrbracket_{v'}$.

Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\varphi \in At$ Si $\varphi = p_n$, sólo ocurre p_n en φ . Luego $\llbracket \varphi \rrbracket_v = v(\varphi) = v'(\varphi) = \llbracket \varphi \rrbracket_{v'}$. Además, $\llbracket \perp \rrbracket_v = \llbracket \perp \rrbracket_{v'} = 0$ siempre.



Universidad
Nacional
de Córdoba



Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\varphi \in At$ Si $\varphi = p_n$, sólo ocurre p_n en φ . Luego $\llbracket \varphi \rrbracket_v = v(\varphi) = v'(\varphi) = \llbracket \varphi \rrbracket_{v'}$. Además, $\llbracket \perp \rrbracket_v = \llbracket \perp \rrbracket_{v'} = 0$ siempre.

$(\varphi \wedge \psi)$ Supongamos que v y v' coinciden en las variables de $(\varphi \wedge \psi)$

Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\varphi \in At$ Si $\varphi = p_n$, sólo ocurre p_n en φ . Luego $\llbracket \varphi \rrbracket_v = v(\varphi) = v'(\varphi) = \llbracket \varphi \rrbracket_{v'}$. Además, $\llbracket \perp \rrbracket_v = \llbracket \perp \rrbracket_{v'} = 0$ siempre.

$(\varphi \wedge \psi)$ Supongamos que v y v' coinciden en las variables de $(\varphi \wedge \psi)$.
Probamos que $\llbracket (\varphi \wedge \psi) \rrbracket_v = \llbracket (\varphi \wedge \psi) \rrbracket_{v'}$.

Lema de Coincidencia

La verdad de una proposición se determina **localmente**.

Lema (de Coincidencia)

Si $v(p_i) = v'(p_i)$ para todos los p_i que ocurran en φ , entonces $\llbracket \varphi \rrbracket_v = \llbracket \varphi \rrbracket_{v'}$.

Demostración.

$\varphi \in At$ Si $\varphi = p_n$, sólo ocurre p_n en φ . Luego $\llbracket \varphi \rrbracket_v = v(\varphi) = v'(\varphi) = \llbracket \varphi \rrbracket_{v'}$. Además, $\llbracket \perp \rrbracket_v = \llbracket \perp \rrbracket_{v'} = 0$ siempre.

$(\varphi \wedge \psi)$ Supongamos que v y v' coinciden en las variables de $(\varphi \wedge \psi)$. Probamos que $\llbracket (\varphi \wedge \psi) \rrbracket_v = \llbracket (\varphi \wedge \psi) \rrbracket_{v'}$.

$(\varphi \odot \psi)$ El resto de los casos queda como ejercicio.

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Pregunta

¿Cuántas asignaciones posibles hay?

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Pregunta

¿Cuántas asignaciones posibles hay?

→ [Actividad en Aula virtual!](#)

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Pregunta

¿Cuántas asignaciones posibles hay?

→ [Actividad en Aula virtual!](#)

Demasiadas.

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Pregunta

¿Cuántas asignaciones posibles hay?

→ [Actividad en Aula virtual!](#)

Demasiadas.

¿Hay que chequear todas para saber si $\models ((p_0 \wedge p_2) \rightarrow p_2)$?

Recordemos que una asignación es una función de $\mathcal{V} = \{p_0, p_1, \dots, p_n, p_{n+1}, \dots\}$ en $\{0, 1\}$.

Pregunta

¿Cuántas asignaciones posibles hay?

→ [Actividad en Aula virtual!](#)

Demasiadas.

¿Hay que chequear todas para saber si $\models ((p_0 \wedge p_2) \rightarrow p_2)$?

Por el Lema de Coincidencia, **no**.

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots
v_1	1	0	1	1	\dots

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots
v_1	1	0	1	1	\dots
v_2	1	1	0	1	\dots

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots
v_1	1	0	1	1	\dots
v_2	1	1	0	1	\dots
v_3	1	0	1	0	\dots

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots
v_1	1	0	1	1	\dots
v_2	1	1	0	1	\dots
v_3	1	0	1	0	\dots
v_4	0	0	1	1	\dots

Tablas de verdad

	p_0	p_1	p_2	p_3	...
v_1	1	0	1	1	...
v_2	1	1	0	1	...
v_3	1	0	1	0	...
v_4	0	0	1	1	...
v_5	0	1	0	0	...

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots
v_1	1	0	1	1	\dots
v_2	1	1	0	1	\dots
v_3	1	0	1	0	\dots
v_4	0	0	1	1	\dots
v_5	0	1	0	0	\dots
\vdots			\vdots		\ddots

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots		
v_2	1	1	0	1	\dots		
v_3	1	0	1	0	\dots		
v_4	0	0	1	1	\dots		
v_5	0	1	0	0	\dots		
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots		
v_3	1	0	1	0	\dots		
v_4	0	0	1	1	\dots		
v_5	0	1	0	0	\dots		
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots		
v_4	0	0	1	1	\dots		
v_5	0	1	0	0	\dots		
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots	1	1
v_4	0	0	1	1	\dots		
v_5	0	1	0	0	\dots		
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots	1	1
v_4	0	0	1	1	\dots	0	1
v_5	0	1	0	0	\dots		
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots	1	1
v_4	0	0	1	1	\dots	0	1
v_5	0	1	0	0	\dots	0	1
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots	1	1
v_4	0	0	1	1	\dots	0	1
v_5	0	1	0	0	\dots	0	1
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_1	p_2	p_3	\dots	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	0	1	1	\dots	1	1
v_2	1	1	0	1	\dots	0	1
v_3	1	0	1	0	\dots	1	1
v_4	0	0	1	1	\dots	0	1
v_5	0	1	0	0	\dots	0	1
\vdots			\vdots		\ddots		

Tablas de verdad

	p_0	p_2	$(p_0 \wedge p_2)$	$((p_0 \wedge p_2) \rightarrow p_2)$
v_1	1	1	1	1
v_2	1	0	0	1
v_4	0	1	0	1
v_5	0	0	0	1

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\varphi \in At$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

■ $p_1[(p_1 \wedge p_2)/p_3] = p_1$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

- $p_1[(p_1 \wedge p_2)/p_3] = p_1$.
- $p_1[(p_1 \wedge p_2)/p_1] = (p_1 \wedge p_2)$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

- $p_1[(p_1 \wedge p_2)/p_3] = p_1$.
- $p_1[(p_1 \wedge p_2)/p_1] = (p_1 \wedge p_2)$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

- $p_1[(p_1 \wedge p_2)/p_3] = p_1$.
- $p_1[(p_1 \wedge p_2)/p_1] = (p_1 \wedge p_2)$.
- $(p_1 \wedge p_2)[(p_3 \wedge p_4)/p_1] = ((p_3 \wedge p_4) \wedge p_2)$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

- $p_1[(p_1 \wedge p_2)/p_3] = p_1$.
- $p_1[(p_1 \wedge p_2)/p_1] = (p_1 \wedge p_2)$.
- $(\textcolor{red}{p}_1 \wedge p_2)[(\textcolor{blue}{p}_3 \wedge \textcolor{blue}{p}_4)/\textcolor{red}{p}_1] = ((\textcolor{blue}{p}_3 \wedge \textcolor{blue}{p}_4) \wedge p_2)$.

Definición

$\varphi[\psi/p] :=$ **sustitución** del símbolo proposicional p **por** la proposición ψ **en** φ :

$\boxed{\varphi \in At}$ Si $\varphi = p$ entonces $\varphi[\psi/p] := \psi$. Caso contrario, $\varphi[\psi/p] := \varphi$.

$\boxed{(\varphi \odot \chi)}$ $(\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p])$.

Ejemplo

- $p_1[(p_1 \wedge p_2)/p_3] = p_1$.
- $p_1[(p_1 \wedge p_2)/p_1] = (p_1 \wedge p_2)$.
- $(\textcolor{red}{p}_1 \wedge p_2)[(\textcolor{blue}{p}_3 \wedge \textcolor{blue}{p}_4)/\textcolor{red}{p}_1] = ((\textcolor{blue}{p}_3 \wedge \textcolor{blue}{p}_4) \wedge p_2)$.

Ejercicio

- 1 $(\neg\varphi)[\psi/p] = (\neg\varphi[\psi/p])$
- 2 $(\varphi \leftrightarrow \chi)[\psi/p] = (\varphi[\psi/p] \leftrightarrow \chi[\psi/p])$.

La regla de Leibnitz

$$\boxed{\varphi \in At} \quad p[\psi/p] := \psi. \quad \varphi \neq p \implies \varphi[\psi/p] = \varphi.$$

$$\boxed{(\varphi \odot \chi)} \quad (\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p]).$$

La regla de Leibnitz

$$\boxed{\varphi \in At} \quad p[\psi/p] := \psi. \quad \varphi \neq p \implies \varphi[\psi/p] = \varphi.$$

$$\boxed{(\varphi \odot \chi)} \quad (\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p]).$$

Teorema (Regla de Leibnitz)

Si $\models \varphi_1 \leftrightarrow \varphi_2$ entonces $\models \psi[\varphi_1/p] \leftrightarrow \psi[\varphi_2/p]$.



Universidad
Nacional
de Córdoba



La regla de Leibnitz

$$\boxed{\varphi \in At} \quad p[\psi/p] := \psi. \quad \varphi \neq p \implies \varphi[\psi/p] = \varphi.$$

$$\boxed{(\varphi \odot \chi)} \quad (\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p]).$$

Teorema (Regla de Leibnitz)

Si $\models \varphi_1 \leftrightarrow \varphi_2$ entonces $\models \psi[\varphi_1/p] \leftrightarrow \psi[\varphi_2/p]$.

Lema

$\llbracket \varphi_1 \rrbracket = \llbracket \varphi_2 \rrbracket$ implica $\llbracket \psi[\varphi_1/p] \rrbracket = \llbracket \psi[\varphi_2/p] \rrbracket$.



Universidad
Nacional
de Córdoba



La regla de Leibnitz

$$\boxed{\varphi \in At} \quad p[\psi/p] := \psi. \quad \varphi \neq p \implies \varphi[\psi/p] = \varphi.$$

$$\boxed{(\varphi \odot \chi)} \quad (\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p]).$$

Teorema (Regla de Leibnitz)

Si $\models \varphi_1 \leftrightarrow \varphi_2$ entonces $\models \psi[\varphi_1/p] \leftrightarrow \psi[\varphi_2/p]$.

Lema

$\llbracket \varphi_1 \rrbracket = \llbracket \varphi_2 \rrbracket$ implica $\llbracket \psi[\varphi_1/p] \rrbracket = \llbracket \psi[\varphi_2/p] \rrbracket$.

Demostración.

Inducción en ψ .

La regla de Leibnitz

$$\boxed{\varphi \in At} \quad p[\psi/p] := \psi. \quad \varphi \neq p \implies \varphi[\psi/p] = \varphi.$$

$$\boxed{(\varphi \odot \chi)} \quad (\varphi \odot \chi)[\psi/p] := (\varphi[\psi/p] \odot \chi[\psi/p]).$$

Teorema (Regla de Leibnitz)

Si $\models \varphi_1 \leftrightarrow \varphi_2$ entonces $\models \psi[\varphi_1/p] \leftrightarrow \psi[\varphi_2/p]$.

Lema

$\llbracket \varphi_1 \rrbracket = \llbracket \varphi_2 \rrbracket$ implica $\llbracket \psi[\varphi_1/p] \rrbracket = \llbracket \psi[\varphi_2/p] \rrbracket$.

Demostración.

Inducción en ψ . En cada caso, suponemos en antecedente y probamos el consecuente.