Lógica proposicional: Sintaxis

Lógica

Contenidos

- Sintaxis de la lógica proposicional

Alfabeto $\Sigma_{\mathtt{PROP}}$

Def.1.1.1. $\Sigma_{\mathtt{PROP}}$

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

- el conjunto de las letras proposicionales: $P := \{p_0, p_1, p_2, ...\}$
- el conjunto de los conectivos: $C := C_0 \cup C_1 \cup C_2$, con $C_0 := \{\bot\}, C_1 := \{\neg\}, C_2 := \{\wedge, \vee, \to, \leftrightarrow\}$
- el conjunto de símbolos auxiliares: $A := \{), (\}$

PROP

Def.1.1.2. PROP

El lenguaje PROP $\subseteq \Sigma^*_{\mathrm{PROP}}$ está definido inductivamente por

i. Si $p \in P$, entonces $p \in \mathrm{PROP}$
ii. $\bot \in \mathrm{PROP}$
iii. Si $\alpha \in \mathrm{PROP}, \beta \in \mathrm{PROP}$, entonces
   - $(\alpha \wedge \beta) \in \mathrm{PROP}$
   - $(\alpha \vee \beta) \in \mathrm{PROP}$
   - $(\alpha \to \beta) \in \mathrm{PROP}$
   - $(\alpha \leftrightarrow \beta) \in \mathrm{PROP}$
iv. Si $\alpha \in \mathrm{PROP}$, entonces $(\neg\alpha) \in \mathrm{PROP}$.

PROP

Fórmulas Proposicionales

Son las palabras de PROP.

Fórmulas Atómicas

Son los elementos del conjunto $\mathrm{AT} = P \cup \{\bot\}$. Son precisamente las palabras formadas por las reglas básicas (i y ii).

Notación: Metavariables

Usamos $p, q, r, p', \ldots$ para las letras proposicionales.
Usamos $\alpha, \beta, \varphi, \psi, \ldots$ para las formulas proposicionales.
Usamos $\Gamma, \Delta, \ldots$ para los conjuntos de formulas proposicionales.

Prop: Ejemplos y Contrajemplos de Fórmulas

Algunas palabras de $\Sigma^*_{\texttt{PROP}}$ que están en $\texttt{PROP}$

- p_0
- $(p_1 \to p_3)$
- $\bot$
- $((p_1 \to p_2) \vee (\bot \wedge (\neg p_5)))$

Algunas palabras de $\Sigma^*_{\texttt{PROP}}$ que no están en $\texttt{PROP}$

- (p_0)
- $(p_1 \to)$
- $p_1 \to \bot$

Teo.1.1.3 PIP para PROP

Hipótesis

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

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

Tesis

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

Esquema de recursión primitiva para PROP(informal)

Sea B un conjunto cualquiera. Entonces, para definir una única función $F : \mathtt{PROP} \to B$ basta un conjunto de ecuaciones como el siguiente:

i $F(p) = \ldots$

ii $F(\bot) = \ldots$

iii $F((\alpha \wedge \beta)) = \ldots F(\alpha) \ldots \alpha \ldots F(\beta) \ldots \beta \ldots$

iv $F((\alpha \vee \beta)) = \ldots F(\alpha) \ldots \alpha \ldots F(\beta) \ldots \beta \ldots$

v $F((\alpha \to \beta)) = \ldots F(\alpha) \ldots \alpha \ldots F(\beta) \ldots \beta \ldots$

vi $F((\alpha \leftrightarrow \beta)) = \ldots F(\alpha) \ldots \alpha \ldots F(\beta) \ldots \beta \ldots$

vii $F((\neg\alpha)) = \ldots F(\alpha) \ldots \alpha \ldots$

Formalización del ERP para PROP

ERP para PROP: Tesis

Existe una única función $F : \texttt{PROP} \to B$ tal que

i $F(\alpha) = H_{\mathrm{AT}}(\alpha)$, con $\alpha \in \mathrm{AT}$

ii $F((\alpha * \beta)) = H_*(\alpha, F(\alpha), \beta, F(\beta))$, con $* \in C_2$

iii $F((\neg\alpha)) = H_\neg(\alpha, F(\alpha))$

$\mathrm{LARGO} : \mathsf{PROP} \to \mathbb{N}$

Versión 1

- i $\mathrm{LARGO}(\varphi) = 1$, para cada $\varphi \in \mathrm{AT}$
- ii $\mathrm{LARGO}((\alpha * \beta)) = 3 + \mathrm{LARGO}(\alpha) + \mathrm{LARGO}(\beta)$
- iii $\mathrm{LARGO}((\neg\alpha)) = 3 + \mathrm{LARGO}(\alpha)$

Versión 2: $H_{\mathrm{AT}} : \mathrm{AT} \to \mathbb{N}$

$$H_{\mathrm{AT}}(\varphi) := 1$$

Versión 2: $H_* : \mathsf{PROP} \times \mathbb{N} \times \mathsf{PROP} \times \mathbb{N} \to \mathbb{N}$

$$H_*(\varphi, n, \psi, m) := 3 + n + m$$

Versión 2: $H_\neg : \mathsf{PROP} \times \mathbb{N} \to \mathbb{N}$

$$H_\neg(\varphi, n) := 3 + n$$

$\textsc{Atoms} : \mathrm{PROP} \to 2^{\mathrm{AT}}$

Versión 1

i $\textsc{Atoms}(\varphi) = \{\varphi\}$, para cada $\varphi \in \mathrm{AT}$

ii $\textsc{Atoms}((\alpha * \beta)) = \textsc{Atoms}(\alpha) \cup \textsc{Atoms}(\beta)$

iii $\textsc{Atoms}((\neg\alpha)) = \textsc{Atoms}(\alpha)$

Versión 2: $H_{\mathrm{AT}} : \mathrm{AT} \to 2^{\mathrm{AT}}$

$$H_{\mathrm{AT}}(\varphi) := \{\varphi\}$$

Versión 2: $H_* : \mathrm{PROP} \times 2^{\mathrm{AT}} \times \mathrm{PROP} \times 2^{\mathrm{AT}} \to 2^{\mathrm{AT}}$

$$H_*(\varphi, A, \psi, B) := A \cup B$$

Versión 2: $H_\neg : \mathrm{PROP} \times 2^{\mathrm{AT}} \to 2^{\mathrm{AT}}$

$$H_\neg(\varphi, A) := A$$

Árboles etiquetados y ordenados

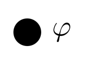
$\mathcal{T}(\mathcal{L})$

Consideramos conocido el lenguaje $\mathcal{T}(\mathcal{L})$ de los árboles etiquetados con palabras de algún lenguaje $\mathcal{L}$.

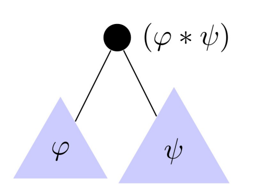
Propiedades

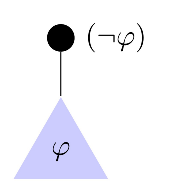
- Cada nodo tiene a lo más un padre. Si no tiene padre, es *la raíz* del árbol.
- Cada nodo tiene un primer hijo, un segundo hijo, etc... ordenados de izquierda a derecha. Si no tiene hijos, es *una hoja* del árbol.
- A cada nodo se le etiqueta con una palabra de $\mathcal{L}$.

$\text{Árbol} : \mathrm{PROP} \to \mathcal{T}(\mathrm{PROP})$ (versión 1)

i $\text{Árbol}(\varphi) =$ 

, para cada $\varphi \in \mathrm{AT}$

ii $\text{Árbol}((\varphi * \psi)) =$ 

iii $\text{Árbol}((\neg\varphi)) =$ 

$\text{Árbol} : \mathrm{PROP} \to \mathcal{T}(\mathrm{PROP})$ (versión 2)

$H_{\mathrm{AT}} : \mathrm{AT} \to \mathcal{T}(\mathrm{PROP})$

$$H_{\mathrm{AT}}(\varphi) := \ldots$$

$H_* : \mathrm{PROP} \times \mathcal{T}(\mathrm{PROP}) \times \mathrm{PROP} \times \mathcal{T}(\mathrm{PROP}) \to \mathcal{T}(\mathrm{PROP})$

$$H_*(\varphi, t_1, \psi, t_2) := \ldots$$

$H_\neg : \mathrm{PROP} \times \mathcal{T}(\mathrm{PROP}) \to \mathcal{T}(\mathrm{PROP})$

$$H_\neg(\varphi, t) := \ldots$$

$\textsc{Rango} : \mathsf{PROP} \to \mathbb{N}$ (versión 1)

i $\textsc{Rango}(\varphi) = 0$, para cada $\varphi \in \mathrm{AT}$

ii $\textsc{Rango}((\alpha * \beta)) =$
$1 + \max\{\textsc{Rango}(\alpha), \textsc{Rango}(\beta)\}$

iii $\textsc{Rango}((\neg\alpha)) = 1 + \textsc{Rango}(\alpha)$

$\textsc{Rango} : \mathsf{PROP} \to \mathbb{N}$ (versión 2)

$H_{\mathrm{AT}} : \mathrm{AT} \to \mathbb{N}$

$$H_{\mathrm{AT}}(a) := 0$$

$H_{*} : \mathsf{PROP} \times \mathbb{N} \times \mathsf{PROP} \times \mathbb{N} \to \mathbb{N}$

$$H_{*}(\varphi, n, \psi, m) := 1 + \max\{n, m\}$$

$H_{\neg} : \mathsf{PROP} \times \mathbb{N} \to \mathbb{N}$

$$H_{\neg}(\varphi, n) := 1 + n$$

Sustitución.

$\_[\_/\_] : \mathrm{PROP} \times \mathrm{PROP} \times P \to \mathrm{PROP}$

i $\bot[\varphi/p] = \bot$

ii $q[\varphi/p] = \begin{cases} \varphi \text{ si } p = q \\ q \text{ si } p \neq q \end{cases}$

iii $(\alpha * \beta)[\varphi/p] = (\alpha[\varphi/p] * \beta[\varphi/p])$

iv $(\neg\alpha)[\varphi/p] = (\neg\alpha[\varphi/p])$

$\_[\_/\_] : \mathsf{PROP} \times \mathsf{PROP} \times P \to \mathsf{PROP}$

$H_{\mathrm{AT}} : \mathrm{AT} \times \mathsf{PROP} \times P \to \mathsf{PROP}$

$$H_{\mathrm{AT}}(\alpha, \varphi, p) := \text{si } p = \alpha \text{ entonces } \varphi \text{ sino } \alpha$$

$H_{*} : \mathsf{PROP} \times \mathsf{PROP} \times \mathsf{PROP} \times \mathsf{PROP} \times \mathsf{PROP} \times P \to \mathsf{PROP}$

$$H_{*}(\alpha, \alpha', \beta, \beta', \varphi, p) := (\alpha' * \beta')$$

$H_{\neg} : \mathsf{PROP} \times \mathsf{PROP} \times \mathsf{PROP} \times P \to \mathsf{PROP}$

$$H_{\neg}(\alpha, \alpha', \varphi, p) := (\neg\alpha')$$

Def.1.1.4.a Secuencia de formación

Una secuencia $\alpha_0, \alpha_1, \ldots \alpha_n$ de palabras de $\Sigma^*_{\mathtt{PROP}}$ es una *secuencia de formación para* α_n si y solamente si para todo $k \leq n$ se cumple

- $\alpha_k \in \mathrm{AT}$, o
- $\alpha_k = (\alpha_i * \alpha_j)$, con $* \in C_2, i < k, j < k$, o
- $\alpha_k = (\neg \alpha_i)$, con $i < k$.

Def.1.1.4.b Subfórmula

Una fórmula $\varphi \in \mathrm{PROP}$ es subfórmula de $\alpha \in \mathrm{PROP}$ si y solamente si se cumple

- $\alpha = \varphi$, o
- $\alpha = (\varphi_1 * \varphi_2)$, con $* \in C_2, \varphi_1 \in \mathrm{PROP}, \varphi_2 \in \mathrm{PROP}$, y φ es subfórmula de φ_1, o
- $\alpha = (\varphi_1 * \varphi_2)$, con $* \in C_2, \varphi_1 \in \mathrm{PROP}, \varphi_2 \in \mathrm{PROP}$, y φ es subfórmula de φ_2, o
- $\alpha = (\neg\varphi_1)$, con $\varphi_1 \in \mathrm{PROP}$, y φ es subfórmula de φ_1.

Def.1.1.7 Conjunto de subfórmulas

$\textsc{Sub} : \mathtt{PROP} \to 2^{\mathtt{PROP}}$

i $\textsc{Sub}(\varphi) = \{\varphi\}$, para cada $\varphi \in \mathrm{AT}$

ii $\textsc{Sub}((\alpha * \beta)) = \{(\alpha * \beta)\} \cup \textsc{Sub}(\alpha) \cup \textsc{Sub}(\beta)$

iii $\textsc{Sub}((\neg\alpha)) = \{(\neg\alpha)\} \cup \textsc{Sub}(\alpha)$

Def.1.1.4.b vs Def.1.1.7

Teorema

$(\bar{\forall}\alpha, \varphi \in \mathtt{PROP})$

φ es subfórmula de α si y solamente si $\varphi \in \textsc{Sub}(\alpha)$

El mismo teorema

$(\bar{\forall}\alpha \in \mathtt{PROP})$

$\textsc{Sub}(\alpha) = \{\varphi \in \mathtt{PROP} : \varphi \text{ es subfórmula de } \alpha\}$

Opciones

¿Qué mecanismo de prueba podemos usar?

¿Inducción?

¿Cómo?

Inducción en PROP (φ)

Propiedad a probar

$$\mathcal{P}(\alpha) := (\bar{\forall}\varphi)(\varphi \text{ es subfórmula de } \alpha \Leftrightarrow \varphi \in \textsc{Sub}(\alpha))$$

Caso BASE

T) $\mathcal{P}(\alpha)$ con $\alpha \in \mathrm{AT}$

Dem.

(Directo)

Suponemos

φ es subfórmula de α

$\Rightarrow$ (Def.1.1.4.b, $\alpha \in \mathrm{AT}$)

$\alpha = \varphi$

$\Rightarrow$ (Def.1.1.7)

$\varphi \in \textsc{Sub}(\alpha)$

■

(Recíproco)

Suponemos

$\varphi \in \textsc{Sub}(\alpha)$

$\Rightarrow$ (...)

$\alpha = \varphi$

$\Rightarrow$ (...)

φ es subfórmula de α

■

Inducción en PROP (φ)

Propiedad a probar

$\mathcal{P}(\alpha) := (\bar{\forall}\varphi)(\varphi$ es subfórmula de $\alpha \Leftrightarrow \varphi \in \textsc{Sub}(\alpha))$

Caso INDUCTIVO. 2

H) $\alpha = (\neg\beta)$ **T)** $\mathcal{P}(\beta) \Rightarrow \mathcal{P}(\alpha)$

Dem.

φ es subfórmula de α

$\Rightarrow$ (Def.1.1.4.b, $\alpha = (\neg\beta)$)

$\alpha = \varphi$ o φ es subfórmula de β

Caso 1 $\alpha = \varphi$. Por Def.1.1.7, $\varphi \in \textsc{Sub}(\alpha)$.

Caso 2 φ es subfórmula de β. Por HI, $\varphi \in \textsc{Sub}(\beta)$.
Por Def.1.1.7, $\varphi \in \textsc{Sub}(\alpha)$.

■

...

¿Qué probamos? ¿Qué falta probar?

- Probamos el caso BASE: $\mathcal{P}(\alpha)$, cuando $\alpha \in \mathrm{AT}$. Falta justificar la mitad de la prueba.
- Probamos el caso IND2: $\mathcal{P}(\alpha) \Rightarrow \mathcal{P}((\neg\alpha))$. Falta probar el recíproco de la propiedad para α.
- Falta probar el caso IND1: $\mathcal{P}(\alpha_1)$ y $\mathcal{P}(\alpha_2) \Rightarrow \mathcal{P}((\alpha_1 * \alpha_2))$.

Luego de realizar las justificaciones y pruebas faltantes, habremos verificado que se cumple la hipótesis del PIP para $\mathcal{P}$, y podremos aplicar el PIP para concluír el teorema.

Aplicación del PIP

El PIP es el (meta)teorema que justifica
$(\bar{\forall}\alpha \in \texttt{PROP})\mathcal{P}(\alpha)$

Teorema

$(\bar{\forall}\alpha, \varphi \in \texttt{PROP})$

φ es subfórmula de α si y solamente si $\varphi \in \textsc{Sub}(\alpha)$

Demostración

En las diapositivas anteriores se probó (faltan partes de la prueba) que la propiedad $\mathcal{P}$ cumple con las hipótesis del PIP de $\texttt{PROP}$, por lo que se deduce que la propiedad se cumple para todo elemento de $\texttt{PROP}$.

PROP y secuencias de formación

Teorema 1.1.5

$\mathtt{PROP}$ es el conjunto de todas palabras de $\Sigma^*_{\mathtt{PROP}}$ que tienen secuencia de formación.

Corolario

Sea $\mathcal{P}$ una propiedad de $\Sigma^*_{\mathtt{PROP}}$. Para demostrar que:

para todo $\alpha \in$ *PROP se cumple* $\mathcal{P}(\alpha)$

podemos hacer la prueba:

- por inducción primitiva en α
- por inducción en la longitud de la secuencia de formación de α

Convenciones sintácticas

- Omitimos los paréntesis exteriores de una fórmula, y los que rodean a $\neg$.
- Los conectivos $\wedge$ y $\vee$ tienen la misma precedencia.
- Los conectivos $\rightarrow$ y $\leftrightarrow$ tienen la misma precedencia.
- El conectivo $\neg$ tiene la mayor precedencia de todos los conectivos.
- Los conectivos $\rightarrow$ y $\leftrightarrow$ tienen la menor precedencia de todos los conectivos.
- Los conectivos de igual precedencia se asocian a la derecha.

Ejemplos

- $\neg\neg p_1$ abrevia a $(\neg(\neg p_1))$
- $p_1 \to p_2 \leftrightarrow p_3$ abrevia a $(p_1 \to (p_2 \leftrightarrow p_3))$
- $p_2 \wedge \bot \vee \neg(p_3 \to p_1)$ abrevia a $(p_2 \wedge (\bot \vee (\neg(p_3 \to p_1))))$
- $p_1 \to p_2 \wedge \bot \vee \neg p_3$ abrevia a $(p_1 \to (p_2 \wedge (\bot \vee (\neg p_3))))$