

# Lógica computacional

Tema: Lógica de primer orden: Introducción y sintaxis.

Facultad de Ciencias  
Universidad Nacional Autónoma de México

Material desarrollado bajo el proyecto UNAM-PAPIME PE102117.



$x_1, x_2, \dots, x_n$

$c_1, c_2, \dots, c_n$

$f(t_1, t_2, \dots, t_n)$

TÉRMINOS

$\top, \perp$

$P(t_1, \dots, t_n)$

$\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi$

$\varphi \rightarrow \psi, \varphi \leftrightarrow \psi$

$\exists x\varphi, \forall y\varphi$

FÓRMULAS

## Términos

Los **términos** del lenguaje son aquellas expresiones que representan objetos, elementos o individuos en el universo del discurso y se generan con la siguiente gramática:

$$t ::= x \mid c \mid f(t_1, \dots, t_m)$$

Al conjunto de términos en lenguaje de la Lógica de Primer Orden lo denotaremos con TERM.

## Fórmulas

El conjunto de expresiones atómicas se denotará con  $ATOM$  y está formado por:

- Las constantes lógicas  $\perp, \top$ .
- Las expresiones de la forma:  $P_1(t_1, \dots, t_n)$  donde  $t_1, \dots, t_n$  son términos.
- Las expresiones de la forma  $t_1 = t_2$ , si el lenguaje cuenta con igualdad.

Los términos se construyen bajo la siguiente gramática:

$$ATOM ::= \perp \mid \top \mid P(t_1, \dots, t_m) \mid t_1 = t_2$$

## Fórmulas

El conjunto FORM de fórmulas compuestas, llamadas usualmente *fórmulas*, se define recursivamente como sigue:

- Si  $\varphi \in \text{ATOM}$  entonces  $\varphi \in \text{FORM}$ . Es decir, toda fórmula atómica es una fórmula.
- Si  $\varphi \in \text{FORM}$  entonces  $(\neg\varphi) \in \text{FORM}$ .
- Si  $\varphi, \psi \in \text{FORM}$  entonces  $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi) \in \text{FORM}$ .
- Si  $\varphi \in \text{FORM}$  y  $x \in \text{Var}$  entonces  $(\forall x\varphi), (\exists x\varphi) \in \text{FORM}$ .

La gramática correspondiente es:

$$\begin{aligned} F &::= \text{ATOM} \mid (\neg F) \mid (F \star F) \mid (\forall x F) \mid (\exists x F) \\ \star &::= \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow \end{aligned}$$

**Convención:** Los cuantificadores se aplican a la *mínima* expresión sintácticamente posible delante del cuantificador.

De manera que

$$\begin{array}{ll} \forall x \varphi \rightarrow \psi & \text{es } (\forall x \varphi) \rightarrow \psi \\ \exists y \varphi \wedge \forall w \psi \rightarrow \chi & \text{es } (\exists y \varphi) \wedge (\forall w \psi) \rightarrow \chi \end{array}$$

## Definición recursiva de funciones sobre términos

Para definir una función  $h : \text{TERM} \rightarrow A$ , basta definir  $h$  como sigue:

- Definir  $h(x)$  para  $x \in \text{Var}$ .
- Definir  $h(c)$  para cada constante  $c \in \mathcal{C}$ .
- Suponiendo que  $h(t_1), \dots, h(t_n)$  están definidas, definir  $h(f(t_1, \dots, t_n))$  para cada símbolo de función  $f \in \mathcal{F}$  de índice  $n$ .

## Definición recursiva de funciones sobre fórmulas

Para definir una función  $h : \text{FORM} \rightarrow A$ , basta definir  $h$  como sigue:

- Definir  $h$  para cada fórmula atómica, es decir, definir  $h(\perp)$ ,  $h(\top)$ ,  $h(P(t_1, \dots, t_n))$  y  $h(t_1 = t_2)$  si el lenguaje tiene igualdad.
- Suponiendo definidas  $h(\varphi)$  y  $h(\psi)$ , definir a partir de ellas a  $h(\neg\varphi)$ ,  $h(\varphi \vee \psi)$ ,  $h(\varphi \wedge \psi)$ ,  $h(\varphi \rightarrow \psi)$ ,  $h(\varphi \leftrightarrow \psi)$ ,  $h(\forall x\varphi)$  y  $h(\exists x\varphi)$ .



## Principio de inducción estructural para términos

Sea  $\mathcal{P}$  una propiedad acerca de términos. Para demostrar que  $\mathcal{P}$  es válida para todos los términos, basta seguir los siguientes pasos:

- Caso base: mostrar que
  - $\mathcal{P}$  es válida para  $x$ , con  $x \in \text{Var}$ .
  - $\mathcal{P}$  es válida para  $c$ , con  $c \in \mathcal{C}$ .
- Hipótesis de inducción: suponer  $\mathcal{P}$  para cualesquiera  $t_1, \dots, t_n \in \text{TERM}$ .
- Paso inductivo: usando la Hipótesis de inducción mostrar que
  - $f(t_1, \dots, t_n)$  cumple  $\mathcal{P}$ , donde  $f \in \mathcal{F}$  es un símbolo de función de índice  $n$ .

## Principio de inducción estructural para fórmulas

Sea  $\mathcal{P}$  una propiedad acerca de fórmulas. Para probar que toda fórmula  $\varphi \in \text{FORM}$  tiene la propiedad  $\mathcal{P}$  basta seguir los siguientes pasos:

- Caso base: mostrar que toda fórmula atómica tiene la propiedad  $\mathcal{P}$ .
- Hipótesis de inducción: suponer que  $\varphi$  y  $\psi$  cumplen  $\mathcal{P}$ .
- Paso inductivo: mostrar usando la Hipótesis de inducción que
  - 1  $(\neg\varphi)$  también cumple  $\mathcal{P}$ .
  - 2  $(\varphi \star \psi)$  tiene la propiedad  $\mathcal{P}$ , donde  $\star \in \{\rightarrow, \wedge, \vee, \leftrightarrow\}$
  - 3  $\forall x\varphi$  y  $\exists x\varphi$  cumplen  $\mathcal{P}$ .

## Ligado y alcance

Dada una cuantificación  $\forall x\varphi$  o  $\exists x\varphi$ , la presencia de  $x$  en  $\forall x$  o  $\exists x$  es la variable que *liga* el cuantificador correspondiente; mientras que la fórmula  $\varphi$  es el **alcance**, ámbito o radio del cuantificador.

Una presencia de la variable  $x$  en la fórmula  $\varphi$  está **ligada** si figura en el alcance de un cuantificador y éste es el más cercano a  $x$ .

Si una presencia de la variable  $x$  no es ligada, decimos que es **libre**.

## Sustitución sobre términos

La aplicación de una sustitución  $[\vec{x} := \vec{t}]$  a un término  $r$ , denotada  $r[\vec{x} := \vec{t}]$ , se define como el término obtenido al reemplazar **simultáneamente** todas las presencias de  $x_i$  en  $r$  por  $t_i$ . Este proceso se define recursivamente como sigue:

$$x_i[\vec{x} := \vec{t}] = t_i \quad 1 \leq i \leq n$$

$$z[\vec{x} := \vec{t}] = z \quad \text{si } z \neq x_i \quad 1 \leq i \leq n$$

$$c[\vec{x} := \vec{t}] = c \quad \text{si } c \in \mathcal{C}, \text{ es decir, } c \text{ constante}$$

$$f(t_1, \dots, t_m)[\vec{x} := \vec{t}] = f(t_1[\vec{x} := \vec{t}], \dots, t_m[\vec{x} := \vec{t}]) \quad \text{con } f^{(m)} \in \mathcal{F}.$$

Debido a la presencia de variables libres y ligadas, la aplicación de una *sustitución textual* a una fórmula puede llevar a situaciones problemáticas, por ejemplo:

- Generar expresiones que no son fórmulas:

$$(\forall x P(y, fx))[x, y := gy, z] = \forall gy P(z, fgy))$$

La expresión de la derecha no es una fórmula.

# Sustitución

- Captura de variables: Consideremos la siguiente aplicación de sustitución

$$\begin{aligned}\forall x(\exists y(x \neq y))[x := y] & \rightarrow \\ & \rightarrow \exists y((x \neq y))[x := y] \\ & \rightarrow (x \neq y)[x := y] \\ & \rightarrow y \neq y\end{aligned}$$

$$\forall x(\exists y(x \neq y))[x := y] = \forall x(\exists y(y \neq y))$$

Para solucionar lo anterior, vamos a preferir un método utilizado en teoría de lenguajes de programación: *la aplicación de una sustitución a una fórmula se define renombrando variables ligadas de manera que siempre podremos obtener una sustitución admisible.*

## Sustitución sobre fórmulas

La aplicación de una sustitución a una fórmula  $\varphi[\vec{x} := \vec{t}]$  se define **recursivamente** como sigue:

$$\begin{aligned}\perp[\vec{x} := \vec{t}] &= \perp \\ \top[\vec{x} := \vec{t}] &= \top \\ P(t_1, \dots, t_m)[\vec{x} := \vec{t}] &= P(t_1[\vec{x} := \vec{t}], \dots, t_m[\vec{x} := \vec{t}]) \\ (t_1 = t_2)[\vec{x} := \vec{t}] &= t_1[\vec{x} := \vec{t}] = t_2[\vec{x} := \vec{t}] \\ (\neg\varphi)[\vec{x} := \vec{t}] &= \neg(\varphi[\vec{x} := \vec{t}]) \\ (\varphi \star \psi)[\vec{x} := \vec{t}] &= (\varphi[\vec{x} := \vec{t}] \star \psi[\vec{x} := \vec{t}]) \\ (\forall y\varphi)[\vec{x} := \vec{t}] &= \forall y(\varphi[\vec{x} := \vec{t}]) \text{ si } y \notin \vec{x} \cup \text{Var}(\vec{t}) \\ (\exists y\varphi)[\vec{x} := \vec{t}] &= \exists y(\varphi[\vec{x} := \vec{t}]) \text{ si } y \notin \vec{x} \cup \text{Var}(\vec{t})\end{aligned}$$

La definición de sustitución en fórmulas cuenta con una restricción aparente en el caso de los cuantificadores, por ejemplo, la sustitución

$$\forall x(Q(x) \rightarrow R(z, x))[z := f(x)]$$

no está definida, puesto que  $x$  figura en  $f(x)$  es decir  $x \in \text{Var}(f(x))$ , con lo que no se cumple la condición necesaria para aplicar la sustitución.

Esta restricción desaparece al notar que los nombres de las variables ligadas no importan: por ejemplo, las fórmulas  $\forall xP(x)$  y  $\forall yP(y)$  significan exactamente lo mismo.



Por lo tanto, convenimos en identificar fórmulas que sólo difieren en sus variables ligadas, esto se hace formalmente mediante la llamada relación de  $\alpha$ -equivalencia definida como sigue:

## Alfa Equivalencia

Decimos que dos fórmulas  $\varphi_1$ ,  $\varphi_2$  son  $\alpha$ -equivalentes lo cual escribimos  $\varphi_1 \sim_\alpha \varphi_2$  si y sólo si  $\varphi_1$  y  $\varphi_2$  difieren únicamente en los nombres de sus variables ligadas.

Las siguientes expresiones son  $\alpha$ -equivalentes.

$$\forall x P(x, y) \rightarrow \exists y R(x, y, z) \sim_{\alpha} \forall w P(w, y) \rightarrow \exists v R(x, v, z)$$

$$\forall w P(w, y) \rightarrow \exists v R(x, v, z) \sim_{\alpha} \forall z P(z, y) \rightarrow \exists u R(x, u, z)$$