

# **Paradigmas de Programación**

## **Lógica de primer orden**

**2do cuatrimestre de 2025**

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

# Introducción

Sintaxis de la lógica de primer orden

Deducción natural para lógica de primer orden

Semántica de la lógica de primer orden

Unificación de términos

# Introducción

## Lógica proposicional

Permite razonar acerca de proposiciones.

Ejemplo:  $\text{Llueve} \vee \neg \text{Llueve}$

## Lógica de primer orden

Permite razonar acerca de elementos sobre los que se predica.

Ejemplo:

$$\forall X. (\text{EsPar}(X) \Rightarrow \neg \text{EsPar}(\text{succ}(X)))$$

Extiende a la lógica proposicional con términos y cuantificadores.

# ¿Para qué tanta lógica? Yo me anoté en computación...

Conexión estrecha entre lógica de primer orden y computación.

## En sus orígenes históricos

- ▶ Problema de la decisión de Hilbert.

## En la actualidad

- ▶ Computabilidad y complejidad descriptiva.
- ▶ Representación del conocimiento, sistemas multi-agente.
- ▶ Inteligencia artificial, razonamiento automático.
- ▶ Métodos formales, verificación automática.
- ▶ Bases de datos relacionales, lenguajes de consulta.
- ▶ Verificación de hardware.
- ▶ ...
- ▶ **Fundamento de la programación lógica.**

# Programación lógica

## Ideal de la programación declarativa

Los programas deberían asemejarse a especificaciones.

## En particular: programación lógica

- ▶ El usuario escribe una fórmula:

$$\exists X. \mathbf{P}(X)$$

- ▶ El sistema busca satisfacer o refutar la fórmula.
- ▶ En caso de lograr satisfacerla, el sistema produce una salida que verifica la propiedad  $P$  buscada.

Introducción

Sintaxis de la lógica de primer orden

Deducción natural para lógica de primer orden

Semántica de la lógica de primer orden

Unificación de términos

# Lenguajes de primer orden

## Definición

Un **lenguaje de primer orden**  $\mathcal{L}$  está dado por:

1. Un conjunto de **símbolos de función**  $\mathcal{F} = \{f, g, h, \dots\}$ .  
Cada símbolo de función tiene asociada una aridad ( $\geq 0$ ).
2. Un conjunto de **símbolos de predicado**  $\mathcal{P} = \{P, Q, R, \dots\}$ .  
Cada símbolo de predicado tiene asociada una aridad ( $\geq 0$ ).

# Términos de primer orden

Suponemos fijado un lenguaje de primer orden  $\mathcal{L}$  y un conjunto infinito numerable de **variables**  $\mathcal{X} = \{X, Y, Z, \dots\}$ .

## Definición

El conjunto  $\mathcal{T}$  de **términos** se define por la siguiente gramática:

$$t ::= X \quad | \quad f(t_1, \dots, t_n)$$

donde:

$X$  denota una variable

$f$  denota un símbolo de función de aridad  $n$

# Términos de primer orden

Ejemplo — el lenguaje  $\mathcal{L}_{\text{aritmética}}$

$$\begin{array}{cccc} 0^0 & \underbrace{\text{succ}^1}_{\text{símbolos de función}} & +^2 & *^2 \\ & & & \\ & & & \end{array} \qquad \begin{array}{cc} =^2 & <^2 \\ \underbrace{\phantom{...}}_{\text{símbolos de predicado}} & \underbrace{\phantom{...}}_{\text{símbolos de predicado}} \end{array}$$

Ejemplo — términos sobre el lenguaje  $\mathcal{L}_{\text{aritmética}}$

$$+(0, \text{succ}(X)) \qquad *(+(X, Y), Z)$$

Los símbolos de función de aridad 0 se llaman constantes.

**Nota.** Usamos notación infija como conveniencia.

$$0 + \text{succ}(X) \qquad (X + Y) * Z$$

## Fórmulas de primer orden

Recordemos la gramática de las fórmulas en lógica proposicional y extendámosla a lógica de primer orden.

$\sigma ::=$	$P(t_1, \dots, t_n)$	<b>fórmula atómica</b>
	$\perp$	contradicción
	$\sigma \Rightarrow \sigma$	implicación
	$\sigma \wedge \sigma$	conjunción
	$\sigma \vee \sigma$	disyunción
	$\neg \sigma$	negación
	$\forall X. \sigma$	<b>cuantificación universal</b>
	$\exists X. \sigma$	<b>cuantificación existencial</b>

**P** denota un símbolo de predicado de aridad  $n$ .

Los cuantificadores ligan una variable **X**.

# Fórmulas de primer orden

Recordemos — el lenguaje  $\mathcal{L}_{\text{aritmética}}$

$0^0$      $\text{succ}^1$      $+^2$      $*^2$      $=^2$      $<^2$

Ejemplo — fórmulas sobre  $\mathcal{L}_{\text{aritmética}}$

$$\forall X. \exists Y. = (+(X, Y), 0)$$

$$\forall X. \forall Y. (\text{succ}(X) = \text{succ}(Y) \Rightarrow X = Y)$$

$$\forall X. (X < 0 \vee X = 0 \vee 0 < X)$$

## Fórmulas de primer orden

Una ocurrencia de una variable  $X$  en una fórmula está:

- ligada si está bajo el alcance de un cuantificador  $\forall X / \exists X$ ,
- libre si no.

Dos fórmulas que sólo difieren en los nombres de las variables ligadas se consideran iguales.

### Ejemplo

$$\forall X. \exists Y. \mathbf{P}(X, Y) \equiv \forall Y. \exists X. \mathbf{P}(Y, X) \equiv \forall A. \exists B. \mathbf{P}(A, B)$$

## Fórmulas de primer orden

Notamos  $\sigma\{X := t\}$  a la sustitución de las ocurrencias libres de  $X$  en la fórmula  $\sigma$  por el término  $t$ , evitando la captura de variables.

### Ejemplo

Sean:

$$\sigma \equiv \text{succ}(X) = Y \implies \exists Z. X + Z = Y$$

entonces:

$$\sigma\{X := Z * Z\} \equiv \text{succ}(Z * Z) = Y \implies \exists Z'. (Z * Z) + Z' = Y$$

Introducción

Sintaxis de la lógica de primer orden

Deducción natural para lógica de primer orden

Semántica de la lógica de primer orden

Unificación de términos

# Deducción natural

La deducción natural proposicional se extiende a primer orden.

Igual que antes:

1. Un **contexto**  $\Gamma$  es un conjunto finito de fórmulas.
2. Un **secuente** es de la forma  $\Gamma \vdash \sigma$ .

Todas las reglas de deducción natural proposicional siguen vigentes.

Se agregan reglas de introducción y eliminación para  $\forall$  y  $\exists$ .

Axioma	AX		
Conjunción	$\wedge I$	$\wedge E_1$	$\wedge E_2$
Disyunción	$\vee I_1$	$\vee I_2$	$\vee E$
Implicación	$\Rightarrow I$	$\Rightarrow E$	
Negación	$\neg I$	$\neg E$	
Contradicción	$\perp E$		
Lógica clásica	$\neg\neg E$		
Cuantificación universal	$\forall I$	$\forall E$	
Cuantificación existencial	$\exists I$	$\exists E$	

# Cuantificación universal

Regla de eliminación

$$\frac{\Gamma \vdash \forall X. \sigma}{\Gamma \vdash \sigma\{X := t\}} \forall E$$

Regla de introducción

$$\frac{\Gamma \vdash \sigma \quad X \notin \text{fv}(\Gamma)}{\Gamma \vdash \forall X. \sigma} \forall I$$

## Cuantificación universal

## Ejemplo

$\frac{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X))}{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\cos(X)) \wedge \mathbf{Q}(\cos(X))}$	AX
$\frac{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\cos(X)) \wedge \mathbf{Q}(\cos(X))}{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\cos(X))}$	$\wedge E_1$
$\frac{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\cos(X))}{\forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \forall X. \mathbf{P}(\cos(X))}$	$\forall I$
$\frac{}{\vdash \forall X. (\mathbf{P}(X) \wedge \mathbf{Q}(X)) \Rightarrow \forall X. \mathbf{P}(\cos(X))}$	$\Rightarrow I$

# Cuantificación universal

## Ejemplo

$$\frac{\frac{\frac{\frac{\frac{\mathbf{P}(X), \forall X. \forall Y. \mathbf{Q}(X, Y) \vdash \forall Z. \forall Y. \mathbf{Q}(Z, Y)}{\mathbf{P}(X), \forall X. \forall Y. \mathbf{Q}(X, Y) \vdash \forall Y. \mathbf{Q}(Z, Y)} \forall E}{\mathbf{P}(X), \forall X. \forall Y. \mathbf{Q}(X, Y) \vdash \mathbf{Q}(Z, Y)} \forall E}{\mathbf{P}(X), \forall X. \forall Y. \mathbf{Q}(X, Y) \vdash \forall \textcolor{red}{Z}. \mathbf{Q}(\textcolor{red}{Z}, Y)} \forall I}{\mathbf{P}(X), \forall X. \forall Y. \mathbf{Q}(X, Y) \vdash \forall Y. \forall X. \mathbf{Q}(X, Y)} \forall I$$

# Cuantificación universal

¿Por qué se exige que  $X \notin \text{fv}(\Gamma)$  en la regla  $\forall I$ ?

Ejemplo — aplicación incorrecta de la regla  $\forall I$

$$\frac{\mathbf{EsPar}(N) \vdash \mathbf{EsPar}(N)}{\mathbf{EsPar}(N) \vdash \forall N. \mathbf{EsPar}(N)} \Leftarrow \text{Paso de razonamiento inválido}$$

# Cuantificación existencial

Regla de introducción

$$\frac{\Gamma \vdash \sigma\{X := t\}}{\Gamma \vdash \exists X. \sigma} \exists I$$

Regla de eliminación

$$\frac{\Gamma \vdash \exists X. \sigma \quad \Gamma, \sigma \vdash \tau \quad X \notin \text{fv}(\Gamma, \tau)}{\Gamma \vdash \tau} \exists E$$

# Cuantificación existencial

## Ejemplo

$$\frac{\frac{\frac{\frac{\frac{\sigma \vdash \sigma}{\sigma, \mathbf{P}(\cos(X)) \vdash \mathbf{P}(\cos(X))} \text{AX}}{\sigma, \mathbf{P}(\cos(X)) \vdash \mathbf{P}(\cos(X)) \vee \mathbf{Q}(\cos(X))} \text{VI}_1}{\sigma, \mathbf{P}(\cos(X)) \vdash \exists X. (\mathbf{P}(X) \vee \mathbf{Q}(X))} \exists I}{\sigma \vdash \exists X. (\mathbf{P}(X) \vee \mathbf{Q}(X))} \exists E}{\vdash \exists X. \mathbf{P}(\cos(X)) \Rightarrow \exists X. (\mathbf{P}(X) \vee \mathbf{Q}(X))} \Rightarrow I$$

$\sigma := \exists X. \mathbf{P}(\cos(X))$

# Cuantificación existencial

## Ejemplo

$$\frac{\frac{\frac{\frac{\frac{\frac{\sigma, \mathbf{P}(W, W), \mathbf{Q}(X) \vdash \mathbf{P}(W, W)}{\sigma, \mathbf{P}(W, W) \vdash \mathbf{Q}(X) \Rightarrow \mathbf{P}(W, W)} \Rightarrow I}{\sigma, \mathbf{P}(W, W) \vdash \exists Z. (\mathbf{Q}(X) \Rightarrow \mathbf{P}(W, Z))} \exists I}{\sigma \vdash \sigma \quad \sigma, \mathbf{P}(W, W) \vdash \exists Y. \exists Z. (\mathbf{Q}(X) \Rightarrow \mathbf{P}(Y, Z))} \exists E}{\exists \textcolor{red}{W}. \mathbf{P}(\textcolor{red}{W}, \textcolor{red}{W}) \vdash \exists Y. \exists Z. (\mathbf{Q}(X) \Rightarrow \mathbf{P}(Y, Z))}$$

$$\sigma := \exists W. \mathbf{P}(W, W)$$

## Cuantificación existencial

Para pensar

¿Por qué se exige que  $X \notin \text{fv}(\Gamma, \tau)$  en la regla  $\exists E$ ?

Introducción

Sintaxis de la lógica de primer orden

Deducción natural para lógica de primer orden

Semántica de la lógica de primer orden

Unificación de términos

# Estructuras de primer orden

Suponemos fijado un lenguaje de primer orden  $\mathcal{L}$ .

## Definición

Una **estructura de primer orden** es un par  $\mathcal{M} = (M, I)$  donde:

- ▶  $M$  es un conjunto **no vacío**, llamado *universo*.
- ▶  $I$  es una función que le da una interpretación a cada símbolo.
- ▶ Para cada símbolo de función  $f$  de aridad  $n$ :

$$I(f) : M^n \rightarrow M$$

- ▶ Para cada símbolo de predicado  $P$  de aridad  $n$ :

$$I(P) \subseteq M^n$$

# Estructuras de primer orden

Recordemos — el lenguaje  $\mathcal{L}_{\text{aritmética}}$

$$0^0 \quad \text{succ}^1 \quad +^2 \quad *^2 \quad =^2 \quad <^2$$

Ejemplo — una estructura sobre  $\mathcal{L}_{\text{aritmética}}$

$M := \mathbb{N}$  (los elementos son números naturales)

$$\begin{array}{lll} I(0) & = 0 & \\ I(\text{succ})(n) & = n + 1 & (n, m) \in I(=) \iff n = m \\ I(+)(n, m) & = n + m & (n, m) \in I(<) \iff n < m \\ I(*)(n, m) & = n \cdot m & \end{array}$$

Bajo esta estructura, la fórmula  $\forall X. X = X + X$  es falsa.

# Estructuras de primer orden

Recordemos — el lenguaje  $\mathcal{L}_{\text{aritmética}}$

$$0^0 \quad \text{succ}^1 \quad +^2 \quad *^2 \qquad =^2 \quad <^2$$

Ejemplo — otra estructura sobre  $\mathcal{L}_{\text{aritmética}}$

$M := \mathcal{P}(\mathbb{R})$  (los elementos son conjuntos de números reales)

$$\begin{array}{lll} I(0) & = \emptyset \\ I(\text{succ})(A) & = \{1+x \mid x \in A\} & (A, B) \in I(=) \iff A = B \\ I(+)(A, B) & = A \cup B & (A, B) \in I(<) \iff A \subseteq B \\ I(*)(A, B) & = A \cap B \end{array}$$

Bajo esta estructura, la fórmula  $\forall X. X = X + X$  es verdadera.

# Interpretación de términos

Suponemos fijada una estructura de primer orden  $\mathcal{M} = (M, I)$ .

## Definición

Una **asignación** es una función que a cada variable le asigna un elemento del universo:

$$\mathbf{a} : \mathcal{X} \rightarrow M$$

## Definición – interpretación de términos

Cada término  $t \in \mathcal{T}$  se interpreta como un elemento  $\mathbf{a}(t) \in M$ , extendiendo la definición de  $\mathbf{a}$  a términos:

$$\mathbf{a}(f(t_1, \dots, t_n)) = I(f)(\mathbf{a}(t_1), \dots, \mathbf{a}(t_n))$$

## Interpretación de fórmulas

Suponemos fijada una estructura de primer orden  $\mathcal{M} = (M, I)$ .

Definimos una relación de **satisfacción**  $\mathbf{a} \models_{\mathcal{M}} \sigma$ .

“La asignación  $\mathbf{a}$  (bajo la estructura  $\mathcal{M}$ ) satisface la fórmula  $\sigma$ ”.

$$\mathbf{a} \models_{\mathcal{M}} \mathbf{P}(t_1, \dots, t_n) \quad \text{sii} \quad (\mathbf{a}(t_1), \dots, \mathbf{a}(t_n)) \in I(\mathbf{P})$$

$$\mathbf{a} \models_{\mathcal{M}} \sigma \wedge \tau \quad \text{sii} \quad \mathbf{a} \models_{\mathcal{M}} \sigma \text{ y } \mathbf{a} \models_{\mathcal{M}} \tau$$

$$\mathbf{a} \models_{\mathcal{M}} \sigma \vee \tau \quad \text{sii} \quad \mathbf{a} \models_{\mathcal{M}} \sigma \text{ o } \mathbf{a} \models_{\mathcal{M}} \tau$$

$$\mathbf{a} \models_{\mathcal{M}} \sigma \Rightarrow \tau \quad \text{sii} \quad \mathbf{a} \not\models_{\mathcal{M}} \sigma \text{ o } \mathbf{a} \models_{\mathcal{M}} \tau$$

$$\mathbf{a} \models_{\mathcal{M}} \neg \sigma \quad \text{sii} \quad \mathbf{a} \not\models_{\mathcal{M}} \sigma$$

$$\mathbf{a} \not\models_{\mathcal{M}} \perp$$

$$\mathbf{a} \models_{\mathcal{M}} \forall X. \sigma \quad \text{sii} \quad \mathbf{a}[X \mapsto m] \models_{\mathcal{M}} \sigma \text{ para todo } m \in M$$

$$\mathbf{a} \models_{\mathcal{M}} \exists X. \sigma \quad \text{sii} \quad \mathbf{a}[X \mapsto m] \models_{\mathcal{M}} \sigma \text{ para algún } m \in M$$

$$\mathbf{a} \models_{\mathcal{M}} \sigma \clubsuit \tau \quad \text{sii} \quad \mathbf{a} \models_{\mathcal{M}} \sigma \text{ brócoli } \mathbf{a} \models_{\mathcal{M}} \tau$$

(Chiste robado de J.-Y. Girard)

## Validez y satisfactibilidad

Decimos que una fórmula  $\sigma$  es:

VÁLIDA si $\mathbf{a} \models_{\mathcal{M}} \sigma$ para toda $\mathcal{M}, \mathbf{a}$	SATISFACTIBLE si $\mathbf{a} \models_{\mathcal{M}} \sigma$ para alguna $\mathcal{M}, \mathbf{a}$
INVÁLIDA si $\mathbf{a} \not\models_{\mathcal{M}} \sigma$ para alguna $\mathcal{M}, \mathbf{a}$	INSATISFACTIBLE si $\mathbf{a} \not\models_{\mathcal{M}} \sigma$ para toda $\mathcal{M}, \mathbf{a}$

## Observaciones

- |                           |     |                                 |
|---------------------------|-----|---------------------------------|
| $\sigma$ es VÁLIDA        | sii | $\sigma$ no es INVÁLIDA         |
| $\sigma$ es SATISFACTIBLE | sii | $\sigma$ no es INSATISFACTIBLE  |
| $\sigma$ es VÁLIDA        | sii | $\neg\sigma$ es INSATISFACTIBLE |
| $\sigma$ es SATISFACTIBLE | sii | $\neg\sigma$ es INVÁLIDA        |

# Modelos

Una *sentencia* es una fórmula  $\sigma$  sin variables libres.

Una *teoría de primer orden* es un conjunto de sentencias.

## Definición — consistencia

Una teoría  $\mathcal{T}$  es *consistente* si  $\mathcal{T} \not\vdash \perp$ .

## Definición — modelo

Una estructura  $\mathcal{M} = (M, I)$  es un *modelo* de una teoría  $\mathcal{T}$  si vale  $\models_{\mathcal{M}} \sigma$  para toda fórmula  $\sigma \in \mathcal{T}$ .

(La asignación es irrelevante pues  $\sigma$  es cerrada).

# Corrección y completitud

## Teorema (Gödel, 1929)

Dada una teoría  $\mathcal{T}$ , son equivalentes:

1.  $\mathcal{T}$  es consistente.
2.  $\mathcal{T}$  tiene (al menos) un modelo.

## Corolario

Dada una fórmula  $\sigma$ , son equivalentes:

1.  $\vdash \sigma$  es derivable.
2.  $\sigma$  es válida.

## Corolario

Dada una fórmula  $\sigma$ , son equivalentes:

1.  $\vdash \neg\sigma$  es derivable.
2.  $\sigma$  es insatisfacible.

# Ejemplos de validez y satisfactibilidad

## Ejemplo

Determinar si son (in)válidas/(in)satisfactibles:

1.  $\forall X. X = X$  satisfactible e inválida
2.  $\forall X. P(X) \Rightarrow \forall X. P(f(X))$  válida ( $\therefore$  satisfactible)
3.  $\forall X. \neg P(X) \wedge \exists X. P(X)$  insatisfactible ( $\therefore$  inválida)
4.  $\forall X. \exists Y. P(X, Y) \Rightarrow \exists Y. \forall X. P(X, Y)$  satisfactible e inválida
5.  $\forall X. (P(X) \Rightarrow \sigma) \Rightarrow (\exists X. P(X)) \Rightarrow \sigma$  con  $X \notin \text{fv}(\sigma)$  válida

## El problema de la decisión

Querríamos un algoritmo que resuelva el siguiente problema:

Entrada: una fórmula  $\sigma$ .

Salida: un booleano que indica si  $\sigma$  es válida.

**No** es posible dar un algoritmo que cumpla dicha especificación.

Introducción

Sintaxis de la lógica de primer orden

Deducción natural para lógica de primer orden

Semántica de la lógica de primer orden

Unificación de términos

# Algoritmo de unificación

El algoritmo de unificación que conocíamos se adapta a términos de primer orden sólo cambiando la notación:

$$\{X \stackrel{?}{=} X\} \cup E \xrightarrow{\text{Delete}} E$$

$$\{f(t_1, \dots, t_n) \stackrel{?}{=} f(s_1, \dots, s_n)\} \cup E \xrightarrow{\text{Decompose}} \{t_1 \stackrel{?}{=} s_1, \dots, t_n \stackrel{?}{=} s_n\} \cup E$$

$$\{t \stackrel{?}{=} X\} \cup E \xrightarrow{\text{Swap}} \begin{array}{l} \{X \stackrel{?}{=} t\} \cup E \\ \text{si } t \text{ no es una variable} \end{array}$$

$$\{X \stackrel{?}{=} t\} \cup E \xrightarrow{\text{Elim}_{\{X:=t\}}} \begin{array}{l} E\{X:=t\} \\ \text{si } X \notin \text{fv}(t) \end{array}$$

$$\{f(t_1, \dots, t_n) \stackrel{?}{=} g(s_1, \dots, s_m)\} \cup E \xrightarrow{\text{Clash}} \begin{array}{l} \text{falla} \\ \text{si } f \neq g \end{array}$$

$$\{X \stackrel{?}{=} t\} \cup E \xrightarrow{\text{Occurs-Check}} \begin{array}{l} \text{falla} \\ \text{si } X \neq t \text{ y } X \in \text{fv}(t) \end{array}$$

## Terminación del algoritmo de unificación

Dado un conjunto de ecuaciones de unificación  $E$ , definimos:

- ▶  $n_1$ : cantidad de variables distintas en  $E$
- ▶  $n_2$ : tamaño de  $E$ , calculado como  $\sum_{(t \stackrel{?}{=} s) \in E} |t| + |s|$
- ▶  $n_3$ : cantidad de ecuaciones de la forma  $t \stackrel{?}{=} X$  en  $E$

Podemos observar que las reglas que no producen falla achican la tripla  $(n_1, n_2, n_3)$ , de acuerdo con el *orden lexicográfico*:

	$n_1$	$n_2$	$n_3$
Elim	>		
Decompose	=	>	
Delete	$\geq$	>	
Swap	=	=	>

# Corrección del algoritmo de unificación

Recordemos

1. Una **sustitución** es una función  $\mathbf{S}$  que le asocia un término  $\mathbf{S}(X)$  a cada variable  $X$ .
2.  $\mathbf{S}$  es un **unificador** de  $E$  si para cada  $(t \stackrel{?}{=} s) \in E$  se tiene que  $\mathbf{S}(t) = \mathbf{S}(s)$ .
3.  $\mathbf{S}$  es **más general** que  $\mathbf{S}'$  si existe  $\mathbf{T}$  tal que  $\mathbf{S}' = \mathbf{T} \circ \mathbf{S}$ .
4.  $\mathbf{S}$  es un **m.g.u.** de  $E$  si  $\mathbf{S}$  es un unificador de  $E$  y para todo unificador  $\mathbf{S}'$  de  $E$  se tiene que  $\mathbf{S}$  es más general que  $\mathbf{S}'$ .

Técnicamente, nos interesan los m.g.u. **idempotentes**, es decir  $\mathbf{S}(\mathbf{S}(t)) = \mathbf{S}(t)$  para todo término  $t$ .

## Corrección del algoritmo de unificación

Lema — corrección de la regla Delete

$$\mathbf{S} \text{ m.g.u. de } E \implies \mathbf{S} \text{ m.g.u. de } \{X \stackrel{?}{=} X\} \cup E.$$

Lema — corrección de la regla Swap

$$\mathbf{S} \text{ m.g.u. de } \{t \stackrel{?}{=} s\} \cup E \implies \mathbf{S} \text{ m.g.u. de } \{s \stackrel{?}{=} t\} \cup E.$$

Lema — corrección de la regla Decompose

$$\begin{aligned} \mathbf{S} \text{ m.g.u. de } & \{t_1 \stackrel{?}{=} s_1, \dots, t_n \stackrel{?}{=} s_n\} \cup E \\ \implies \mathbf{S} \text{ m.g.u. de } & \{\mathbf{f}(t_1, \dots, t_n) \stackrel{?}{=} \mathbf{f}(s_1, \dots, s_n)\} \cup E. \end{aligned}$$

Lema — corrección de la regla Elim

$$\begin{aligned} \mathbf{S} \text{ m.g.u. de } & E\{X := t\} \text{ y } X \notin \text{fv}(t) \\ \implies \mathbf{S} \circ & \{X := t\} \text{ m.g.u. de } E. \end{aligned}$$

Usar el hecho de que si  $\mathbf{S}(X) = t$  entonces  $\mathbf{S}(s\{X := t\}) = \mathbf{S}(s)$ .

## Corrección del algoritmo de unificación

Probemos la corrección del algoritmo en caso de éxito.

Sea  $E_0 \rightarrow_{S_1} E_1 \rightarrow_{S_n} E_2 \rightarrow \dots \rightarrow_{S_n} E_n = \emptyset$ .

Veamos que  $S_n \circ \dots \circ S_1$  es un m.g.u. de  $E$ .

Por inducción en  $n$ :

1. Si  $n = 0$ , la sustitución identidad es un m.g.u. de  $\emptyset$ .
2. Si  $n > 0$ , se tiene:

$$E_0 \rightarrow_{S_1} E_1 \quad E_1 \rightarrow_{S_2} \dots \rightarrow_{S_n} E_n = \emptyset$$

Por HI,  $S_n \circ \dots \circ S_2$  es un m.g.u. de  $E_1$ .

Aplicando alguno de los lemas anteriores, se concluye que  $S_n \circ \dots \circ S_2 \circ S_1$  es un m.g.u. de  $E_0$ .

## Corrección del algoritmo de unificación

La corrección en caso de falla se prueba de manera similar, con lemas que van “hacia adelante” en lugar de “hacia atrás”.

í í í í í í ? ? ? ? ? ? ?

### Lectura recomendada

#### **Capítulo 2 del libro de van Dalen.**

Dirk van Dalen. *Logic and Structure* (Fifth Edition).  
Springer-Verlag, 2013.