Paradigmas de Programación

Lógica de primer orden

1er cuatrimestre de 2024

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Lógica **proposicional**

Permite razonar acerca de proposiciones.
Ejemplo: **Llueve** $\vee \neg$ **Llueve**

Lógica de **primer orden**

Permite razonar acerca de elementos sobre los que se predica.
Ejemplo:

$$\forall X.\,(\textbf{EsPar}(X) \Rightarrow \neg\textbf{EsPar}(\mathtt{succ}(X)))$$

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.

Lenguajes de primer orden

Definición

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

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

Términos de primer orden

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

Definición

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

$$t \quad ::= \quad X \quad | \quad \mathtt{f}(t_1, \ldots, t_n)$$

donde:

X denota una variable

$\mathtt{f}$ denota un símbolo de función de aridad n

Términos de primer orden

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

$$\underbrace{0^0 \quad \mathtt{succ}^1 \quad +^2 \quad *^2}_{\text{símbolos de función}} \qquad \underbrace{=^2 \quad <^2}_{\text{símbolos de predicado}}$$

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

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

Los símbolos de función de aridad 0 se llaman constantes.

Nota. Usamos notación infija como conveniencia.

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

Fórmulas de primer orden

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

σ	$::=$	$\mathbf{P}(t_1, \ldots, t_n)$	**fórmula atómica**
	$\mid$	$\bot$	contradicción
	$\mid$	$\sigma \Rightarrow \sigma$	implicación
	$\mid$	$\sigma \wedge \sigma$	conjunción
	$\mid$	$\sigma \vee \sigma$	disyunción
	$\mid$	$\neg\sigma$	negación
	$\mid$	$\forall X.\, \sigma$	**cuantificación universal**
	$\mid$	$\exists X.\, \sigma$	**cuantificación existencial**

$\mathbf{P}$ denota un símbolo de predicado de aridad n.

Los cuantificadores ligan una variable X.

Fórmulas de primer orden

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

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

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

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

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

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

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 σ por el término t, evitando la captura de variables.

Ejemplo

Sean:

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

entonces:

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

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 $\mathtt{f}$ de aridad n:

$$I(\mathtt{f}) : M^n \to M$$

- Para cada símbolo de predicado $\mathbf{P}$ de aridad n:

$$I(\mathbf{P}) \subseteq M^n$$

Estructuras de primer orden

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

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

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

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

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

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

Estructuras de primer orden

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

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

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

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

$$\begin{array}{ll} I(0) & = \varnothing \\ I(\mathtt{succ})(A) & = \{1 + x \mid x \in A\} \\ I(+)(A,B) & = A \cup B \\ I(*)(A,B) & = A \cap B \end{array} \qquad \begin{array}{l} (A,B) \in I(=) \iff A = B \\ (A,B) \in I(<) \iff A \subseteq B \end{array}$$

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:

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

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:

$$\mathbf{a}(\mathtt{f}(t_1, \ldots, t_n)) \quad = \quad I(\mathtt{f})(\mathbf{a}(t_1), \ldots, \mathbf{a}(t_n))$$

Interpretación de fórmulas

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

Definimos una relación de **satisfacción** $\mathbf{a} \vDash_{\mathcal{M}} \sigma$.
"La asignación $\mathbf{a}$ (bajo la estructura $\mathcal{M}$) satisface la fórmula σ".

$\mathbf{a} \vDash_{\mathcal{M}} \mathbf{P}(t_1, \ldots, t_n)$	sii	$(\mathbf{a}(t_1), \ldots, \mathbf{a}(t_n)) \in I(\mathbf{P})$
$\mathbf{a} \vDash_{\mathcal{M}} \sigma \wedge \tau$	sii	$\mathbf{a} \vDash_{\mathcal{M}} \sigma$ y $\mathbf{a} \vDash_{\mathcal{M}} \tau$
$\mathbf{a} \vDash_{\mathcal{M}} \sigma \vee \tau$	sii	$\mathbf{a} \vDash_{\mathcal{M}} \sigma$ o $\mathbf{a} \vDash_{\mathcal{M}} \tau$
$\mathbf{a} \vDash_{\mathcal{M}} \sigma \Rightarrow \tau$	sii	$\mathbf{a} \nvDash_{\mathcal{M}} \sigma$ o $\mathbf{a} \vDash_{\mathcal{M}} \tau$
$\mathbf{a} \vDash_{\mathcal{M}} \neg\sigma$	sii	$\mathbf{a} \nvDash_{\mathcal{M}} \sigma$
$\mathbf{a} \nvDash_{\mathcal{M}} \bot$		
$\mathbf{a} \vDash_{\mathcal{M}} \forall X.\, \sigma$	sii	$\mathbf{a}[X \mapsto m] \vDash_{\mathcal{M}} \sigma$ para todo $m \in M$
$\mathbf{a} \vDash_{\mathcal{M}} \exists X.\, \sigma$	sii	$\mathbf{a}[X \mapsto m] \vDash_{\mathcal{M}} \sigma$ para algún $m \in M$
$\mathbf{a} \vDash_{\mathcal{M}} \sigma \clubsuit \tau$	sii	$\mathbf{a} \vDash_{\mathcal{M}} \sigma$ brócoli $\mathbf{a} \vDash_{\mathcal{M}} \tau$

(Chiste robado de J.-Y. Girard)

Validez y satisfactibilidad

Decimos que una fórmula σ es:

VÁLIDA si $\mathbf{a} \vDash_{\mathcal{M}} \sigma$ para toda $\mathcal{M}, \mathbf{a}$	SATISFACTIBLE si $\mathbf{a} \vDash_{\mathcal{M}} \sigma$ para alguna $\mathcal{M}, \mathbf{a}$
INVÁLIDA si $\mathbf{a} \nvDash_{\mathcal{M}} \sigma$ para alguna $\mathcal{M}, \mathbf{a}$	INSATISFACTIBLE si $\mathbf{a} \nvDash_{\mathcal{M}} \sigma$ para toda $\mathcal{M}, \mathbf{a}$

Observaciones

σ es VÁLIDA	sii	σ no es INVÁLIDA
σ es SATISFACTIBLE	sii	σ no es INSATISFACTIBLE
σ es VÁLIDA	sii	$\neg\sigma$ es INSATISFACTIBLE
σ es SATISFACTIBLE	sii	$\neg\sigma$ es INVÁLIDA

Ejemplos de validez y satisfactibilidad

Ejemplo

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

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

El problema de la decisión

Querríamos un algoritmo que resuelva el siguiente problema:

Entrada: una fórmula σ.
Salida: un booleano que indica si σ es válida.

No es posible dar un algoritmo que cumpla dicha especificación.

Deducción natural

La deducción natural proposicional se extiende a primer orden.

Igual que antes:

1. Un **contexto** Γ 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	$\bot$E		
Lógica clásica	$\neg\neg$E		
Cuantificación universal	$\forall$I	$\forall$E	
Cuantificación existencial	$\exists$I	$\exists$E	

Cuantificación universal

Regla de eliminación

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

Regla de introducción

$$\frac{\Gamma \vdash \sigma \qquad X \notin \mathrm{fv}(\Gamma)}{\Gamma \vdash \forall X.\, \sigma}\,\forall\mathrm{I}$$

Cuantificación universal

Ejemplo

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X))}\,\mathrm{AX}}{\forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\mathtt{f}(X)) \wedge \mathbf{Q}(\mathtt{f}(X))}\,\forall\mathrm{E}}{\forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \mathbf{P}(\mathtt{f}(X))}\,\wedge\mathrm{E}_1}{\forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X)) \vdash \forall X.\,\mathbf{P}(\mathtt{f}(X))}\,\forall\mathrm{I}}{\vdash \forall X.\,(\mathbf{P}(X) \wedge \mathbf{Q}(X)) \Rightarrow \forall X.\,\mathbf{P}(\mathtt{f}(X))}\,\Rightarrow\mathrm{I}$$

Cuantificación universal

Ejemplo

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{P}(X), \forall X.\,\forall Y.\,\mathbf{Q}(X,Y) \vdash \forall Z.\,\forall Y.\,\mathbf{Q}(Z,Y)}\,\mathrm{AX}}{\mathbf{P}(X), \forall X.\,\forall Y.\,\mathbf{Q}(X,Y) \vdash \forall Y.\,\mathbf{Q}(Z,Y)}\,\forall\mathrm{E}}{\mathbf{P}(X), \forall X.\,\forall Y.\,\mathbf{Q}(X,Y) \vdash \mathbf{Q}(Z,Y)}\,\forall\mathrm{E}}{\mathbf{P}(X), \forall X.\,\forall Y.\,\mathbf{Q}(X,Y) \vdash \forall\ \ Z.\,\mathbf{Q}(\ \ Z,Y)}\,\forall\mathrm{I}}{\mathbf{P}(X), \forall X.\,\forall Y.\,\mathbf{Q}(X,Y) \vdash \forall Y.\,\forall X.\,\mathbf{Q}(X,Y)}\,\forall\mathrm{I}$$

Cuantificación universal

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

Ejemplo — aplicación incorrecta de la regla $\forall\mathrm{I}$

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

Cuantificación existencial

Regla de introducción

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

Regla de eliminación

$$\frac{\Gamma \vdash \exists X.\, \sigma \qquad \Gamma, \sigma \vdash \tau \qquad X \notin \mathrm{fv}(\Gamma, \tau)}{\Gamma \vdash \tau}\exists_{\mathrm{E}}$$

Cuantificación existencial

Ejemplo

$$\dfrac{\dfrac{\dfrac{}{\sigma, \mathbf{P}(\mathtt{f}(X)) \vdash \mathbf{P}(\mathtt{f}(X))}\mathrm{AX}}{\sigma, \mathbf{P}(\mathtt{f}(X)) \vdash \mathbf{P}(\mathtt{f}(X)) \vee \mathbf{Q}(\mathtt{f}(X))}\vee_{\mathrm{I}_1}}{\sigma, \mathbf{P}(\mathtt{f}(X)) \vdash \exists X.\, (\mathbf{P}(X) \vee \mathbf{Q}(X))}\exists_{\mathrm{I}}$$

$$\dfrac{\dfrac{}{\sigma \vdash \sigma}\mathrm{AX} \qquad \sigma, \mathbf{P}(\mathtt{f}(X)) \vdash \exists X.\, (\mathbf{P}(X) \vee \mathbf{Q}(X))}{\sigma \vdash \exists X.\, (\mathbf{P}(X) \vee \mathbf{Q}(X))}\exists_{\mathrm{E}}$$

$$\dfrac{\sigma \vdash \exists X.\, (\mathbf{P}(X) \vee \mathbf{Q}(X))}{\vdash \exists X.\, \mathbf{P}(\mathtt{f}(X)) \Rightarrow \exists X.\, (\mathbf{P}(X) \vee \mathbf{Q}(X))}\Rightarrow_{\mathrm{I}}$$

$\sigma :\equiv \exists X.\, \mathbf{P}(\mathtt{f}(X))$

Cuantificación existencial

Ejemplo

$$\dfrac{\dfrac{}{\sigma \vdash \sigma}\mathrm{AX} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\sigma, \mathbf{P}(W, W), \mathbf{Q}(X) \vdash \mathbf{P}(W, W)}\mathrm{AX}}{\sigma, \mathbf{P}(W, W) \vdash \mathbf{Q}(X) \Rightarrow \mathbf{P}(W, W)}\Rightarrow_{\mathrm{I}}}{\sigma, \mathbf{P}(W, W) \vdash \exists Z.\, (\mathbf{Q}(X) \Rightarrow \mathbf{P}(W, Z))}\exists_{\mathrm{I}}}{\sigma, \mathbf{P}(W, W) \vdash \exists Y.\, \exists Z.\, (\mathbf{Q}(X) \Rightarrow \mathbf{P}(Y, Z))}\exists_{\mathrm{I}}}{\exists\ W.\, \mathbf{P}(\ W,\ W) \vdash \exists Y.\, \exists Z.\, (\mathbf{Q}(X) \Rightarrow \mathbf{P}(Y, Z))}\exists_{\mathrm{E}}$$

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

Cuantificación existencial

Para pensar

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

Modelos

Una *sentencia* es una fórmula σ sin variables libres.
Una *teoría de primer orden* es un conjunto de sentencias.

Definición — consistencia

Una teoría $\mathcal{T}$ es *consistente* si $\mathcal{T} \nvdash \bot$.

Definición — modelo

Una estructura $\mathcal{M} = (M, I)$ es un *modelo* de una teoría $\mathcal{T}$ si vale $\vDash_{\mathcal{M}} \sigma$ para toda fórmula $\sigma \in \mathcal{T}$.
(La asignación es irrelevante pues σ 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 σ, son equivalentes:

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

Corolario

Dada una fórmula σ, son equivalentes:

1. $\vdash \neg\sigma$ es derivable.
2. σ es insatisfactible.

Algoritmo de unificación

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

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

$$\{\mathtt{f}(t_1, \ldots, t_n) \stackrel{?}{=} \mathtt{f}(s_1, \ldots, s_n)\} \cup E \xrightarrow{\texttt{Decompose}} \{t_1 \stackrel{?}{=} s_1, \ldots, t_n \stackrel{?}{=} s_n\} \cup E$$

$$\{t \stackrel{?}{=} X\} \cup E \xrightarrow{\texttt{Swap}} \{X \stackrel{?}{=} t\} \cup E \quad \text{si } t \text{ no es una variable}$$

$$\{X \stackrel{?}{=} t\} \cup E \xrightarrow{\texttt{Elim}_{\{X:=t\}}} E\{X{:=}t\} \quad \text{si } X \notin \mathrm{fv}(t)$$

$$\{\mathtt{f}(t_1, \ldots, t_n) \stackrel{?}{=} \mathtt{g}(s_1, \ldots, s_m)\} \cup E \xrightarrow{\texttt{Clash}} \texttt{falla} \quad \text{si } \mathtt{f} \neq \mathtt{g}$$

$$\{X \stackrel{?}{=} t\} \cup E \xrightarrow{\texttt{Occurs-Check}} \texttt{falla} \quad \text{si } X \neq t \text{ y } X \in \mathrm{fv}(t)$$

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`

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

Lema — corrección de la regla `Swap`

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

Lema — corrección de la regla `Decompose`

$\mathbf{S}$ m.g.u. de $\{t_1 \stackrel{?}{=} s_1, \ldots, t_n \stackrel{?}{=} s_n\} \cup E$
$\implies \mathbf{S}$ m.g.u. de $\{\mathtt{f}(t_1, \ldots, t_n) \stackrel{?}{=} \mathtt{f}(s_1, \ldots, s_n)\} \cup E$.

Lema — corrección de la regla `Elim`

$\mathbf{S}$ m.g.u. de $E\{X := t\}$ y $X \notin \mathrm{fv}(t)$
$\implies \mathbf{S} \circ \{X := t\}$ m.g.u. de E.

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 $E_0 \to_{\mathbf{S}_1} E_1 \to_{\mathbf{S}_n} E_2 \to \ldots \to_{\mathbf{S}_n} E_n = \varnothing$.
Veamos que $\mathbf{S}_n \circ \ldots \circ \mathbf{S}_1$ es un m.g.u. de E.
Por inducción en n:

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

$$E_0 \to_{\mathbf{S}_1} E_1 \qquad E_1 \to_{\mathbf{S}_2} \ldots \to_{\mathbf{S}_n} E_n = \varnothing$$

   Por HI, $\mathbf{S}_n \circ \ldots \circ \mathbf{S}_2$ es un m.g.u. de E_1.
   Aplicando alguno de los lemas anteriores, se concluye que
   $\mathbf{S}_n \circ \ldots \circ \mathbf{S}_2 \circ \mathbf{S}_1$ 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".