

Lógica de Predicados, Sintaxis

Dante Zanarini

LCC

26 de Octubre de 2020

Enriqueciendo el lenguaje

- Consideremos el siguiente razonamiento:
 - *Todos los polinomios son derivables*
 - $p(x) = x^2 + 3x + 9$ es un polinomio
 - $p(x) = x^2 + 3x + 9$ es derivable
- ¿Se puede formalizar en lógica proposicional?

Enriqueciendo el lenguaje

- Consideremos el siguiente razonamiento:

- *Todos los polinomios son derivables*
- $p(x) = x^2 + 3x + 9$ es un polinomio
- $p(x) = x^2 + 3x + 9$ es derivable

p_0

p_1

p_2

- ¿Se puede formalizar en lógica proposicional?

Enriqueciendo el lenguaje

- Consideremos el siguiente razonamiento:

- *Todos los polinomios son derivables* p_0
- $p(x) = x^2 + 3x + 9$ es un polinomio p_1
- $p(x) = x^2 + 3x + 9$ es derivable p_2

- ¿Se puede formalizar en lógica proposicional?
- Sí, deberíamos ver que $p_0, p_1 \vdash p_2$

Enriqueciendo el lenguaje

- Consideremos el siguiente razonamiento:

- *Todos los polinomios son derivables* p_0
- $p(x) = x^2 + 3x + 9$ es un polinomio p_1
- $p(x) = x^2 + 3x + 9$ es derivable p_2

- ¿Se puede formalizar en lógica proposicional?
- Sí, deberíamos ver que $p_0, p_1 \vdash p_2$
- Sin embargo, este secuento no es válido, a pesar que el razonamiento parece serlo

Enriqueciendo el lenguaje

- Consideremos el siguiente razonamiento:

- *Todos los polinomios son derivables* p_0
- $p(x) = x^2 + 3x + 9$ es un polinomio p_1
- $p(x) = x^2 + 3x + 9$ es derivable p_2

- ¿Se puede formalizar en lógica proposicional?
- Sí, deberíamos ver que $p_0, p_1 \vdash p_2$
- Sin embargo, este secuento no es válido, a pesar que el razonamiento parece serlo

Si estamos usando la lógica para identificar los buenos razonamientos, algo está fallando

Lógica de Predicados como Lenguaje Formal

En *Lógica de Predicados* utilizaremos dos lenguajes formales:

- El **Lenguaje de términos**, que describe los objetos con los que trabajamos

Lógica de Predicados como Lenguaje Formal

En *Lógica de Predicados* utilizaremos dos lenguajes formales:

- El **Lenguaje de términos**, que describe los objetos con los que trabajamos
- El **Lenguaje de fórmulas**, que describe *relaciones* entre los objetos de estudio, así como también permite expresar propiedades universales y existenciales sobre ellos

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

- Ⓐ Un conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ de **símbolos de función**, junto con una función $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. Decimos que $\text{ar}(f_i)$ es la *aridad* de f_i

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

- Ⓐ Un conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ de **símbolos de función**, junto con una función $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. Decimos que $\text{ar}(f_i)$ es la *aridad* de f_i
- Ⓑ Un conjunto $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ de **símbolos de predicados**, también acompañados por su aridad (utilizaremos $\text{ar}(P_i)$ para denotar la aridad de P_i)

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

- Ⓐ Un conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ de **símbolos de función**, junto con una función $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. Decimos que $\text{ar}(f_i)$ es la *aridad* de f_i
- Ⓑ Un conjunto $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ de **símbolos de predicados**, también acompañados por su aridad (utilizaremos $\text{ar}(P_i)$ para denotar la aridad de P_i)
- Ⓒ Un conjunto infinito $\text{Var} = \{x_0, x_1, \dots\}$ de **variables**

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

- Ⓐ Un conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ de **símbolos de función**, junto con una función $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. Decimos que $\text{ar}(f_i)$ es la *aridad* de f_i
- Ⓑ Un conjunto $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ de **símbolos de predicados**, también acompañados por su aridad (utilizaremos $\text{ar}(P_i)$ para denotar la aridad de P_i)
- Ⓒ Un conjunto infinito $\text{Var} = \{x_0, x_1, \dots\}$ de **variables**
- Ⓓ Conectivos, $C = \{\neg, \wedge, \vee, \rightarrow, \perp, \forall, \exists\}$
- Ⓔ Símbolos auxiliares, $A = \{ (,) \}$

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

- Ⓐ Un conjunto $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ de **símbolos de función**, junto con una función $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. Decimos que $\text{ar}(f_i)$ es la *aridad* de f_i
- Ⓑ Un conjunto $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ de **símbolos de predicados**, también acompañados por su aridad (utilizaremos $\text{ar}(P_i)$ para denotar la aridad de P_i)
- Ⓒ Un conjunto infinito $\text{Var} = \{x_0, x_1, \dots\}$ de **variables**
- Ⓓ Conectivos, $C = \{\neg, \wedge, \vee, \rightarrow, \perp, \forall, \exists\}$
- Ⓔ Símbolos auxiliares, $A = \{ (,) \}$
 - Si $\text{ar}(f_i) = 0$, decimos que f_i es una **constante**
 - Si $\text{ar}(P_i) = 0$, decimos que P_i es una **proposición**
 - Al par $(\mathcal{F}, \mathcal{P})$ le llamaremos **signatura**

Lenguaje de Términos

Definición (TERM)

El conjunto de términos se define inductivamente por las siguientes reglas:

- ① *Para todo $i \in \mathbb{N}$, $x_i \in \text{TERM}$*
- ② *Si $\text{ar}(f_i) = 0$, entonces $f_i \in \text{TERM}$*
- ③ *Si $\text{ar}(f_i) = n > 0$ y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces $f_i(t_1, t_2, \dots, t_n) \in \text{TERM}$*

Lenguaje de Términos

Ejemplo

Si $\mathcal{F} = \{f, g, h, c\}$ con $\text{ar}(f) = 2$, $\text{ar}(g) = \text{ar}(h) = 1$ y $\text{ar}(c) = 0$,
tenemos

$$f(g(x_5), c) \in \text{TERM}$$

$$h(f(c, g(c))) \in \text{TERM}$$

$$c \in \text{TERM}$$

$$h(g(h(x_{15}))) \in \text{TERM}$$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

❶ $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- ❶ $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- ① $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = n > 0$, y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces
 $P_i(t_1, t_2, \dots, t_n) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- i) $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = n > 0$, y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces $P_i(t_1, t_2, \dots, t_n) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- ii) Si $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\neg \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- i) $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = n > 0$, y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces $P_i(t_1, t_2, \dots, t_n) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- ii) Si $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\neg \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- iii) Si $\phi, \psi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- i) $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = n > 0$, y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces $P_i(t_1, t_2, \dots, t_n) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- ii) Si $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\neg \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- iii) Si $\phi, \psi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- iv) Si $x_i \in \text{Var}$ y $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\forall x_i \phi), (\exists x_i \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Lenguaje de Fórmulas

Definición (FORM)

Sea $(\mathcal{F}, \mathcal{P})$ una *signatura*. El conjunto de fórmulas $\text{FORM}_{(\mathcal{F}, \mathcal{P})}$ se define inductivamente por las siguientes reglas:

- i) $\perp \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = 0$, entonces $P_i \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
Si $\text{ar}(P_i) = n > 0$, y $t_1, t_2, \dots, t_n \in \text{TERM}$, entonces $P_i(t_1, t_2, \dots, t_n) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- ii) Si $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\neg \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- iii) Si $\phi, \psi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$
- iv) Si $x_i \in \text{Var}$ y $\phi \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$, entonces $(\forall x_i \phi), (\exists x_i \phi) \in \text{FORM}_{(\mathcal{F}, \mathcal{P})}$

Convenciones sintácticas

- Cuando podamos, omitiremos los paréntesis más externos en $(\forall x_i \phi)$ y $(\exists x_i \phi)$
- Orden de precedencia de los operadores: $\forall, \exists, \neg, \wedge, \vee, \rightarrow$
- Agregaremos \dots, w, x, y, z al conjunto de variables

Lenguaje de fórmulas

Ejemplo

Sean $\mathcal{F} = \{p, i, e\}$, con $\text{ar}(p) = 2$, $\text{ar}(i) = 1$, $\text{ar}(e) = 0$; y
 $\mathcal{P} = \{L, \dot{=}\}$, con $\text{ar}(L) = \text{ar}(\dot{=}) = 2$

Lenguaje de fórmulas

Ejemplo

Sean $\mathcal{F} = \{p, i, e\}$, con $\text{ar}(p) = 2$, $\text{ar}(i) = 1$, $\text{ar}(e) = 0$; y
 $\mathcal{P} = \{L, \dot{=}\}$, con $\text{ar}(L) = \text{ar}(\dot{=}) = 2$

Algunos términos: $p(e, x_2)$, $p(i(x_1), i(p(x_2, e)))$

Algunas fórmulas:

- $L(e, i(e))$
- $(i(x_1) \dot{=} i(e)) \rightarrow (x_1 \dot{=} e)$
- $(\forall x_1 ((\neg(x_1 \dot{=} e)) \rightarrow L(e, x_1)))$
- $(\exists x_3 L(x_3, e)) \rightarrow (e \dot{=} i(x_3))$

Variables libres

Definimos el conjunto de variables libres de una fórmula ϕ por recursión en ϕ :

$$\begin{array}{lll} FV & : & \text{FORM} \rightarrow 2^{\text{Var}} \\ FV(\perp) & = & \emptyset \\ FV(P_i) & = & \emptyset \quad \text{si } \text{ar}(P_i) = 0 \\ FV(P_i(t_1, \dots, t_n)) & = & \bigcup_{i=1}^n FV_T(t_i) \quad \text{si } \text{ar}(P_i) = n > 0 \\ FV(\neg\phi) & = & FV(\phi) \\ FV(\phi \Box \psi) & = & FV(\phi) \cup FV(\psi) \\ FV(\forall x_i \phi) & = & FV(\phi) - \{x_i\} \\ FV(\exists x_i \phi) & = & FV(\phi) - \{x_i\} \end{array}$$

Ejercicios:

- Definir $FV_T : \text{TERM} \rightarrow 2^{\text{Var}}$, que calcula el conjunto de variables libres de un término
- Definir $BV : \text{FORM} \rightarrow 2^{\text{Var}}$, que determina el conjunto de variables ligadas de una fórmula

Fórmulas cerradas

- Un término t (una fórmula ϕ) se dice cerrado (cerrada) si no tiene variables libres
- A una fórmula cerrada la llamaremos **sentencia**
- $\text{SENT}_{(\mathcal{F}, \mathcal{P})}$ es el conjunto de sentencias sobre una signatura, y
- $\text{TERM}_{\mathcal{F}}^{\mathcal{C}}$ es el conjunto de términos cerrados

Sustitución para Términos

Definición

Sean s, t términos y $x_i \in \text{Var}$. Definimos la sustitución de x_i por t en s por recursión en s :

$$\begin{aligned} (\text{vars}) \quad x_j[t/x_i] &= \begin{cases} x_j & \text{si } i \neq j \\ t & \text{si } i = j \end{cases} \\ (\text{ctes}) \quad c[t/x_i] &= c \\ (\text{func}) \quad f(t_1, \dots, t_n)[t/x_i] &= f(t_1[t/x_i], \dots, t_n[t/x_i]) \end{aligned}$$

Sustitución para Fórmulas

Definición

Sean $t \in \text{TERM}$ y $\phi \in \text{FORM}$, definimos $\phi[t/x_i]$ por recursión en ϕ :

$$\begin{aligned}\perp[t/x_i] &= \perp \\ P_i[t/x_i] &= P_i \quad (\text{ar}(P_i) = 0)\end{aligned}$$

Sustitución para Fórmulas

Definición

Sean $t \in \text{TERM}$ y $\phi \in \text{FORM}$, definimos $\phi[t/x_i]$ por recursión en ϕ :

$$\begin{aligned}\perp[t/x_i] &= \perp \\ P_i[t/x_i] &= P_i & (\text{ar}(P_i) = 0) \\ P_i(t_1, \dots, t_n)[t/x_i] &= P_i(t_1[t/x_i], \dots, t_n[t/x_i])\end{aligned}$$

Sustitución para Fórmulas

Definición

Sean $t \in \text{TERM}$ y $\phi \in \text{FORM}$, definimos $\phi[t/x_i]$ por recursión en ϕ :

$$\begin{aligned}\perp[t/x_i] &= \perp \\ P_i[t/x_i] &= P_i & (\text{ar}(P_i) = 0) \\ P_i(t_1, \dots, t_n)[t/x_i] &= P_i(t_1[t/x_i], \dots, t_n[t/x_i]) \\ (\neg\phi)[t/x_i] &= \neg(\phi[t/x_i]) \\ (\phi \Box \psi)[t/x_i] &= \phi[t/x_i] \Box \psi[t/x_i]\end{aligned}$$

Sustitución para Fórmulas

Definición

Sean $t \in \text{TERM}$ y $\phi \in \text{FORM}$, definimos $\phi[t/x_i]$ por recursión en ϕ :

$$\begin{aligned}\perp[t/x_i] &= \perp \\ P_i[t/x_i] &= P_i & (\text{ar}(P_i) = 0) \\ P_i(t_1, \dots, t_n)[t/x_i] &= P_i(t_1[t/x_i], \dots, t_n[t/x_i]) \\ (\neg\phi)[t/x_i] &= \neg(\phi[t/x_i]) \\ (\phi \Box \psi)[t/x_i] &= \phi[t/x_i] \Box \psi[t/x_i] \\ (\forall x_j \phi)[t/x_i] &= \begin{cases} (\forall x_j \phi) & \text{si } i = j \\ (\forall x_j \phi[t/x_i]) & \text{si } i \neq j \end{cases}\end{aligned}$$

Sustitución para Fórmulas

Definición

Sean $t \in \text{TERM}$ y $\phi \in \text{FORM}$, definimos $\phi[t/x_i]$ por recursión en ϕ :

$$\begin{aligned}\perp[t/x_i] &= \perp \\ P_i[t/x_i] &= P_i & (\text{ar}(P_i) = 0) \\ P_i(t_1, \dots, t_n)[t/x_i] &= P_i(t_1[t/x_i], \dots, t_n[t/x_i])\end{aligned}$$

$$\begin{aligned}(\neg\phi)[t/x_i] &= \neg(\phi[t/x_i]) \\ (\phi\Box\psi)[t/x_i] &= \phi[t/x_i]\Box\psi[t/x_i]\end{aligned}$$

$$(\forall x_j\phi)[t/x_i] = \begin{cases} (\forall x_j\phi) & \text{si } i = j \\ (\forall x_j\phi[t/x_i]) & \text{si } i \neq j \end{cases}$$

$$(\exists x_j\phi)[t/x_i] = \begin{cases} (\exists x_j\phi) & \text{si } i = j \\ (\exists x_j\phi[t/x_i]) & \text{si } i \neq j \end{cases}$$

Captura de Variables

- Un problema de la operación de sustitución, es que puede cambiar el significado de una fórmula
- Por ejemplo,

$$\exists x P(x, y)$$

- Si sustituimos la variable y con $t = x$, obtenemos:

$$(\exists x P(x, y))[x/y] = (\exists x P(x, y)[x/y]) = (\exists x P(x, x))$$

- Este problema se conoce como **captura de variables libres**

Evitando la captura

- Para no alterar el significado (que veremos más adelante) en $\phi[t/x_i]$ de una fórmula, necesitamos que t esté **libre** para x_i en ϕ .

Definición

Un término t está libre para una variable x en una fórmula ϕ sii

- 1 ϕ es atómica
- 2 $\phi \equiv \phi_1 \Box \phi_2$ y t está libre para x en ϕ_1 y ϕ_2 .
- 3 $\phi \equiv \neg \phi_1$ y t está libre para x en ϕ_1
- 4 $\phi \equiv \forall y \phi_1$ y si $x \neq y$, se cumple:
 - ★ t está libre para x en ϕ_1
 - ★ $y \notin FV(t)$