Lógica de Predicados, Sintaxis

Dante Zanarini

LCC

16 de Septiembre de 2015

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_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?

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 secuente 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 secuente 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:

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

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

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

(B) Un conjunto $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ de símbolos de predicados, también acompañados por su aridad (utilizaremos $\mathrm{ar}(P_i)$ para denotar la aridad de P_i)

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

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

(B) Un conjunto $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ de símbolos de predicados, también acompañados por su aridad (utilizaremos $\mathrm{ar}(P_i)$ para denotar la aridad de P_i)

(C) Un conjunto infinito $\mathrm{Var} = \{x_0, x_1, \ldots\}$ de variables

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

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

(B) Un conjunto $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ de símbolos de predicados, también acompañados por su aridad (utilizaremos $\mathrm{ar}(P_i)$ para denotar la aridad de P_i)

(C) Un conjunto infinito $\mathrm{Var} = \{x_0, x_1, \ldots\}$ de variables

(D) Conectivos, $C = \{\neg, \wedge, \vee, \rightarrow, \perp, \forall, \exists\}$

(E) Símbolos auxiliares, $A = \{\,(\,,\,)\,\}$

Alfabeto

El alfabeto está compuesto por los siguientes símbolos:

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

(B) Un conjunto $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ de símbolos de predicados, también acompañados por su aridad (utilizaremos $\mathrm{ar}(P_i)$ para denotar la aridad de P_i)

(C) Un conjunto infinito $\mathrm{Var} = \{x_0, x_1, \ldots\}$ de variables

(D) Conectivos, $C = \{\neg, \wedge, \vee, \to, \perp, \forall, \exists\}$

(E) Símbolos auxiliares, $A = \{\, (\ ,\ )\, \}$

- Si $\mathrm{ar}(f_i) = 0$, decimos que f_i es una constante
- Si $\mathrm{ar}(P_i) = 0$, 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* $i \in \mathbb{N}$, $x_i \in \mathrm{TERM}$

(2) *Si* $\mathrm{ar}(f_i) = 0$, *entonces* $f_i \in \mathrm{TERM}$

(3) *Si* $\mathrm{ar}(f_i) = n > 0$ *y* $t_1, t_2, \ldots, t_n \in \mathrm{TERM}$, *entonces* $f_i(t_1, t_2, \ldots, t_n) \in \mathrm{TERM}$

Lenguaje de Términos

Ejemplo

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

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

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

$$c \in \mathrm{TERM}$$

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

Lenguaje de Fórmulas

Definición ($\textsc{Form}$)

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

(i) $\bot \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

Lenguaje de Fórmulas

Definición ($\textsc{Form}$)

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

(i) $\perp \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$, *entonces* $P_i \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

Lenguaje de Fórmulas

Definición ($\textsc{Form}$)

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

(i) $\perp \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$, *entonces* $P_i \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = n > 0$, *y* $t_1, t_2, \ldots, t_n \in \mathrm{TERM}$, *entonces* $P_i(t_1, t_2, \ldots, t_n) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

Lenguaje de Fórmulas

Definición (Form)

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

(i) $\perp \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$*, entonces* $P_i \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = n > 0$*, y* $t_1, t_2, \ldots, t_n \in \mathrm{TERM}$*, entonces* $P_i(t_1, t_2, \ldots, t_n) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(ii) *Si* $\phi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$*, entonces* $(\neg\phi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

Lenguaje de Fórmulas

Definición ($\textsc{Form}$)

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

(i) $\bot \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$, *entonces* $P_i \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = n > 0$, *y* $t_1, t_2, \ldots, t_n \in \mathrm{TERM}$, *entonces* $P_i(t_1, t_2, \ldots, t_n) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(ii) *Si* $\phi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$, *entonces* $(\neg\phi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(iii) *Si* $\phi, \psi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$, *entonces* $(\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

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) $\bot \in \text{FORM}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$, *entonces* $P_i \in \text{FORM}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = n > 0$, *y* $t_1, t_2, \ldots, t_n \in \text{TERM}$, *entonces* $P_i(t_1, t_2, \ldots, t_n) \in \text{FORM}_{(\mathcal{F},\mathcal{P})}$

(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* $(\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi) \in \text{FORM}_{(\mathcal{F},\mathcal{P})}$

(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 ($\textsc{Form}$)

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

(i) $\bot \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = 0$, *entonces* $P_i \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$
Si $\mathrm{ar}(P_i) = n > 0$, *y* $t_1, t_2, \ldots, t_n \in \mathrm{TERM}$, *entonces* $P_i(t_1, t_2, \ldots, t_n) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(ii) *Si* $\phi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$, *entonces* $(\neg\phi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(iii) *Si* $\phi, \psi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$, *entonces* $(\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

(iv) *Si* $x_i \in \mathrm{Var}$ *y* $\phi \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$, *entonces* $(\forall x_i \phi), (\exists x_i \phi) \in \textsc{Form}_{(\mathcal{F},\mathcal{P})}$

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 $\ldots, w, x, y, z$ al conjunto de variables

Lenguaje de fórmulas

Ejemplo

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

Lenguaje de fórmulas

Ejemplo

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

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

Algunas fórmulas:

- $L(e, i(e))$
- $(i(x_1) \doteq i(e)) \to (x_1 \doteq e)$
- $(\forall x_1((\neg(x_1 \doteq e)) \to L(e, x_1)))$
- $(\exists x_3 L(x_3, e)) \to (e \doteq i(x_3))$

Variables libres

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

$$
\begin{array}{lcll}
FV & : & \text{FORM} \to 2^{\mathrm{Var}} & \\
FV(\bot) & = & \emptyset & \\
FV(P_i) & = & \emptyset & \text{si } \mathrm{ar}(P_i) = 0 \\
FV(P_i(t_1, \ldots, t_n)) & = & \bigcup_{i=1}^{n} FV_T(t_i) & \text{si } \mathrm{ar}(P_i) = n > 0 \\
FV(\neg\phi) & = & FV(\phi) & \\
FV(\phi \square \psi) & = & FV(\phi) \cup FV(\psi) & \\
FV(\forall x_i \phi) & = & FV(\phi) - \{x_i\} & \\
FV(\exists x_i \phi) & = & FV(\phi) - \{x_i\} &
\end{array}
$$

Ejercicios:

- Definir $FV_T : \mathrm{TERM} \to 2^{\mathrm{Var}}$, que calcula el conjunto de variables libres de un término
- Definir $BV : \text{FORM} \to 2^{\mathrm{Var}}$, 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
- $\mathrm{SENT}_{(\mathcal{F},\mathcal{P})}$ es el conjunto de sentencias sobre una signatura, y
- $\mathrm{TERM}^{C}_{\mathcal{F}}$ es el conjunto de términos cerrados

Sustitución para Términos

Definición

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

$$\begin{array}{llcl}
(\mathit{vars}) & x_j[t/x_i] & = & \begin{cases} x_j & \mathit{si}\ i \neq j \\ t & \mathit{si}\ i = j \end{cases} \\
(\mathit{ctes}) & c[t/x_i] & = & c \\
(\mathit{func}) & f(t_1, \ldots t_n)[t/x_i] & = & f(t_1[t/x_i], \ldots t_n[t/x_i])
\end{array}$$

Sustitución para Fórmulas

Definición

Sean $t \in \mathrm{TERM}$ *y* $\phi \in \textsc{Form}$, *definimos* $\phi[t/x_i]$ *por recursión en* ϕ*:*

$$\begin{array}{lcll}
\perp[t/x_i] & = & \perp & \\
P_i[t/x_i] & = & P_i & (\mathrm{ar}(P_i) = 0)
\end{array}$$

Sustitución para Fórmulas

Definición

Sean $t \in \mathrm{TERM}$ *y* $\phi \in \textsc{Form}$, *definimos* $\phi[t/x_i]$ *por recursión en* ϕ*:*

$$
\begin{array}{lcll}
\bot[t/x_i] & = & \bot & \\
P_i[t/x_i] & = & P_i & (\mathrm{ar}(P_i) = 0) \\
P_i(t_1, \ldots, t_n)[t/x_i] & = & P_i(t_1[t/x_i], \ldots t_n[t/x_i]) &
\end{array}
$$

Sustitución para Fórmulas

Definición

Sean $t \in \mathrm{TERM}$ *y* $\phi \in \textsc{Form}$, *definimos* $\phi[t/x_i]$ *por recursión en* ϕ*:*

$$
\begin{array}{lcll}
\bot[t/x_i] & = & \bot & \\
P_i[t/x_i] & = & P_i & (\mathrm{ar}(P_i) = 0) \\
P_i(t_1, \ldots, t_n)[t/x_i] & = & P_i(t_1[t/x_i], \ldots t_n[t/x_i]) & \\
 & & & \\
(\neg\phi)[t/x_i] & = & \neg(\phi[t/x_i]) & \\
(\phi\Box\psi)[t/x_i] & = & \phi[t/x_i]\Box\psi[t/x_i] &
\end{array}
$$

Sustitución para Fórmulas

Definición

Sean $t \in \mathrm{TERM}$ *y* $\phi \in \textsc{Form}$, *definimos* $\phi[t/x_i]$ *por recursión en* ϕ:

$$
\begin{array}{lcll}
\bot[t/x_i] & = & \bot & \\
P_i[t/x_i] & = & P_i & (\mathrm{ar}(P_i) = 0) \\
P_i(t_1, \ldots, t_n)[t/x_i] & = & P_i(t_1[t/x_i], \ldots t_n[t/x_i]) & \\
 & & & \\
(\neg\phi)[t/x_i] & = & \neg(\phi[t/x_i]) & \\
(\phi\Box\psi)[t/x_i] & = & \phi[t/x_i]\Box\psi[t/x_i] & \\
 & & & \\
(\forall x_j\phi)[t/x_i] & = & \begin{cases} (\forall x_j\phi) & si\ i = j \\ (\forall x_j\phi[t/x_i]) & si\ i \neq j \end{cases} &
\end{array}
$$

Sustitución para Fórmulas

Definición

Sean $t \in \mathrm{TERM}$ *y* $\phi \in \textsc{Form}$, *definimos* $\phi[t/x_i]$ *por recursión en* ϕ*:*

$$
\begin{array}{lcll}
\bot[t/x_i] & = & \bot & \\
P_i[t/x_i] & = & P_i & (\mathrm{ar}(P_i) = 0) \\
P_i(t_1, \ldots, t_n)[t/x_i] & = & P_i(t_1[t/x_i], \ldots t_n[t/x_i]) & \\
 & & & \\
(\neg\phi)[t/x_i] & = & \neg(\phi[t/x_i]) & \\
(\phi\Box\psi)[t/x_i] & = & \phi[t/x_i]\Box\psi[t/x_i] & \\
 & & & \\
(\forall x_j\phi)[t/x_i] & = & \begin{cases} (\forall x_j\phi) & si\ i = j \\ (\forall x_j\phi[t/x_i]) & si\ i \neq j \end{cases} & \\
 & & & \\
(\exists x_j\phi)[t/x_i] & = & \begin{cases} (\exists x_j\phi) & si\ i = j \\ (\exists x_j\phi[t/x_i]) & si\ i \neq j \end{cases} &
\end{array}
$$

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. $\phi \equiv \phi_1 \square \phi_2$ *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
   - $y \notin FV(t)$