

Lógica

Mauro Polenta Mora

CLASE 13 - 07/05/2025

Sintáxis de la lógica de predicados

Definición (estructura)

Una estructura es una secuencia ordenada:

$$\mathcal{M} = \langle U, R_1, \dots, R_n, F_1, \dots, F_m, \{C_i \mid 1 \leq i \leq k\} \rangle$$

Tal que:

- U es un conjunto no vacío (notación: $U = |\mathcal{M}|$)
- R_1, \dots, R_n son relaciones sobre U ($n \geq 0$)
- F_1, \dots, F_m son funciones en U ($m \geq 0$)
- $\{C_i \mid 1 \leq i \leq k\}$ son elementos distinguidos de U

Observación: Todo esto corresponde a la idea intuitiva que teníamos en la clase anterior.

Ejemplos

- $\langle \mathbb{N}, Par, \leq, +, *, 0, 1 \rangle$ son los naturales.
- $\langle \mathbb{Z}, +, -, 0 \rangle$ son los enteros.

Definición (tipo de similaridad)

Dada una estructura determinada, por ejemplo:

$$\langle U, R_1, \dots, R_n, F_1, \dots, F_m, \{C_i \mid 1 \leq i \leq k\} \rangle$$

Decimos que tiene la siguiente secuencia como tipo de similaridad:

$$\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$$

Donde:

- $R_i \subseteq U^{r_i}$ ($1 \leq i \leq n$ y $r_i \geq 0$), es decir r_1, \dots, r_n representan la “aridad” de las relaciones R_i .

- $F_j : U^{a_j} \rightarrow U$ ($1 \leq j \leq n$ y $a_j \geq 0$), es decir que a_1, \dots, a_m representan la cantidad de parámetros que recibe cada función F_i .
- k es el número de constantes.

Ejemplos

- $\langle \mathbb{N}, Par, \leq, +, *, 0, 1 \rangle$ tiene tipo $\langle 1, 2; 2, 2; 2 \rangle$
- $\langle \mathbb{Z}, +, -, 0 \rangle$ tiene tipo $\langle -, 2, 1; 1 \rangle$

Definición (alfabeto de primer orden)

El alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$ para un lenguaje de primer orden consta de los siguientes símbolos: - Símbolos de relación: $P_1, \dots, P_n, ='$ - Símbolos de función: f_1, \dots, f_m - Símbolos de constantes c_i tal que $1 \leq i \leq k$ - Variables: x_1, x_2, x_3, \dots - Conectivos: $\rightarrow, \leftrightarrow, \neg, \wedge, \vee, \perp$ - Cuantificadores: \forall, \exists - Auxiliares: $(,)$

Definición (términos)

Sea A el alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$. El conjunto $TERM_A$ de los términos del lenguaje de primer orden con alfabeto A se define inductivamente por: 1. $x_i \in TERM_A$ ($i \in \mathbb{N}$) 2. $c_i \in TERM_A$ ($1 \leq i \leq k$) 3. Si $t_1, \dots, t_{a_i} \in TERM_A$ entonces $f_i(t_1, \dots, t_{a_i}) \in TERM_A$

Definición (fórmulas)

Sea A el alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$. El conjunto $FORM_A$ de las fórmulas del lenguaje de primer orden con alfabeto A se define inductivamente por:

1. $\perp \in FORM_A$
2. Si $t_1, \dots, t_{r_i} \in TERM_A$, entonces $P_i(t_1, \dots, t_{r_i}) \in FORM_A$
3. Si $t_1, t_2 \in TERM_A$, entonces $t_1 =' t_2 \in FORM_A$
4. Si $\alpha, \beta \in FORM_A$, entonces $(\alpha \square \beta) \in FORM_A$
5. Si $\alpha \in FORM_A$, entonces $(\neg \alpha) \in FORM_A$
6. Si $\alpha \in FORM_A$, entonces $((\forall x_i) \alpha), ((\exists x_i) \alpha) \in FORM_A$

Ejemplos

Sea A el alfabeto de tipo $\langle 1, 2; 1, 2; 2 \rangle$.

1. ¿ $f_2(c_1, x_4) \in FORM_A$? VERDADERO, pues f_2 es una función que toma dos parámetros y la constante c_1 existe.
2. ¿ $f_1(c_1, x_4) \in FORM_A$? FALSO, pues f_1 solo toma un parámetro.
3. ¿ $((\forall x_1)P_2(f_1(x_1), c_1)) \rightarrow ((\exists x_2)P_1(x_2))$? VERDADERO, pues todas las funciones y predicados usados cumplen con las reglas marcadas por el tipo de similaridad.
4. ¿ $((\exists x_2)f_2(x_1, c_2)) \in FORM_A$? FALSO, pues $f_2(x_1, c_2) \notin FORM_A$, por lo que esto no respetaría la regla 6 de construcción.
5. ¿ $((\forall x_1)P_1(x_1, c_1)) \in FORM_A$? FALSO, pues P_1 solo toma un parámetro.
6. ¿ $((\exists x_1)(\exists x_2)(\exists x_3)P_3(x_1, x_2, x_3)) \in FORM_A$? FALSO, pues P_3 ni siquiera existe en este tipo de similaridad.

Reglas de parentización

- Las reglas de precedencia de conectivos son las mismas que para *PROP*.
- Los conectivos de igual precedencia se asocian a la derecha (igual que *PROP*).
- Cuantificadores: el \forall y \exists tienen igual precedencia que el \neg .

Atención: No confundir las siguientes fórmulas.

- $(\forall x)(\alpha \rightarrow \beta)$ y $(\forall x)\alpha \rightarrow \beta$
- $(\exists x)(\alpha \rightarrow \beta)$ y $(\exists x)\alpha \rightarrow \beta$

Conjuntos importantes

Sea A el alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$.

Definición (Var)

Var es el conjunto de las variables de A : $\{x_i \mid i \in \mathbb{N}\}$

Definición ($Const_A$)

$Const_A$ es el conjunto de los símbolos de constante de A : $\{c_i \mid 1 \leq i \leq k\}$

Definición (fórmulas atómicas AT_A)

AT_A es el conjunto de fórmulas de $FORM_A$ que se obtienen con las cláusulas base:
 $(\perp, P_j(t_1, \dots, t_{r_j}, t_i = 't_j))$

PIP para $TERM_A$

Sea A el alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$.

(H) Sea P una propiedad sobre $TERM_A$. Si se cumple:

1. $P(x)$ para todo $x \in Var$
2. $P(c)$ para todo $c \in Const_A$
3. Si $P(t_1), \dots, P(t_{a_i})$, entonces $P(f_i(t_1, \dots, t_{a_i}))$ para todo $i \in \{1, \dots, m\}$

(T) Entonces se cumple $(\forall t \in TERM_A)P(t)$

PIP para $FORM_A$

Sea A el alfabeto de tipo $\langle r_1, \dots, r_n; a_1, \dots, a_m; k \rangle$.

(H) Sea P una propiedad sobre $TERM_A$. Si se cumple:

1. $P(\alpha)$ para todo $\alpha \in AT_A$
2. Si $P(\alpha)$ y $P(\beta)$, entonces $P(\alpha \square \beta)$ donde $\square \in \{\rightarrow, \leftrightarrow, \wedge, \vee\}$
3. Si $P(\alpha)$ entonces $P(\neg \alpha)$
4. Si $P(\alpha)$ entonces $P((\forall x)\alpha)$ y $P((\exists x)\alpha)$ para todo $x \in Var$

(T) Entonces se cumple $(\forall \alpha \in FORM_A)P(\alpha)$

ERP simplificado para $TERM_A$

Una función está bien definida para $TERM_A$ cuando tenemos una definición inductiva libre y tenemos lo siguiente:

1. $F : TERM_A \rightarrow B$
2. $F(t) = H_b(t)$ si $t \in Var \cup Const_A$
3. $F(f_i(t_1, \dots, t_{a_i})) = H_i(t_1, F(t_1), \dots, t_{a_i}, F(t_{a_i}))$

ERP simplificado para $FORM_A$

1. $F : FORM_A \rightarrow B$
2. $F(\alpha) = H_{AT}(\alpha)$ si $\alpha \in AT_A$
3. $F(\alpha \Box \beta) = H_{\Box}(\alpha, F(\alpha), \beta, F(\beta))$
4. $F(\neg \alpha) = H_{\neg}(\alpha, F(\alpha))$
5. $F((\forall x)\alpha) = H_{\forall}(x, \alpha, F(\alpha))$
6. $F((\exists x)\alpha) = H_{\exists}(x, \alpha, F(\alpha))$

Observación

Notemos que es exactamente la misma idea que para Lógica Proposicional, los cambios en este caso son mínimos.