Lógica de predicados. Sintaxis y Propiedades

Lógica 2017

Instituto de Computación

9 de mayo

Estructuras

Def 2.2.1. Estructura

Una *estructura* es una secuencia ordenada

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

tal que:

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

Estructura

Ejemplo

$\langle \mathbb{N}, Par, \leq, +, *, 0, 1 \rangle$	naturales
$\langle \mathbb{N}, < \rangle$	CPO de los naturales
$\langle \mathbb{Z}, +, -, 0 \rangle$	grupo de los enteros

Def 2.2.2 Tipo de similaridad de una estructura

El *tipo de similaridad* de $\langle U, R_1, \ldots, R_n, F_1, \ldots, F_m, \{C_i \mid 1 \leq i \leq k\}\rangle$ es una secuencia

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

tal que:

- $R_i \subseteq U^{r_i}$ $(1 \leq i \leq n$ y $r_i \geq 0)$
- $F_j : U^{a_j} \to U$ $(1 \leq j \leq n$ y $a_j \geq 0)$

Tipo de similaridad de una estructura

Ejemplo

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

Lenguaje de primer orden

Def. Alfabeto de primer orden

El *alfabeto* de tipo $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; k \rangle$ para un lenguaje de primer orden consta de los siguientes símbolos:

- Símbolos de relación: $P_1, P_2, \ldots, P_n, =^{\prime}$
- Símbolos de función: $f_1, f_2, \ldots, f_m$
- Símbolos de constantes: c_i tal que $1 \leq i \leq k$
- Variables: $x_1, x_2, x_3, \ldots$
- Conectivos: $\rightarrow, \leftrightarrow, \neg, \wedge, \vee, \perp$
- Cuantificadores: $\forall, \exists$
- Auxiliares: ( ) ,

Def 2.3.1 Términos

Sea A el alfabeto de tipo $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; k\rangle$. El conjunto $\mathtt{TERM}_A$ de los términos del lenguaje de primer orden con alfabeto A se define inductivamente por:

1. $x_i \in \mathtt{TERM}_A (i \in \mathbb{N})$
2. $c_i \in \mathtt{TERM}_A (1 \leq i \leq k)$
3. si $t_1, \ldots, t_{a_i} \in \mathtt{TERM}_A$ entonces $f_i(t_1, \ldots, t_{a_i}) \in \mathtt{TERM}_A$

Def 2.3.2 Fórmulas

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

1. $\bot \in \mathtt{FORM}_A$
2. si $t_1, \ldots, t_{r_i} \in \mathtt{TERM}_A$ entonces $P_i(t_1, \ldots, t_{r_i}) \in \mathtt{FORM}_A$
3. si $t_1, t_2 \in \mathtt{TERM}_A$ entonces $t_1 =' t_2 \in \mathtt{FORM}_A$
4. si $\alpha, \beta \in \mathtt{FORM}_A$ entonces $(\alpha \square \beta) \in \mathtt{FORM}_A$
5. si $\alpha \in \mathtt{FORM}_A$ entonces $(\neg \alpha) \in \mathtt{FORM}_A$
6. si $\alpha \in \mathtt{FORM}_A$ entonces $((\forall x_i)\alpha), ((\exists x_i)\alpha) \in \mathtt{FORM}_A$

Ejemplo (términos y fórmulas)

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

1. ¿$f_2(c_1, x_4) \in \mathtt{TERM}_A$?
2. ¿$f_1(c_1, x_4) \in \mathtt{TERM}_A$?
3. ¿$((\forall x_1)P_2(f_1(x_1), c_1)) \to ((\exists x_2)P_1(x_2)) \in \mathtt{FORM}_A$?
4. ¿$((\exists x_2)f_2(x_1, c_2)) \in \mathtt{FORM}_A$?
5. ¿$((\forall x_1)P_1(x_1, c_1)) \in \mathtt{FORM}_A$?
6. ¿$((\exists x_1)((\exists x_2)((\exists x_3)P_3(x_1, x_2, x_3)))) \in \mathtt{FORM}_A$?

¡¡¡OJO!!! ¡No confundir símbolo de predicado y símbolo de función!

$$f_2(x_1, c_2) \in \mathtt{TERM} \text{ y } P(x_1) \in \mathtt{FORM}$$

Reglas de parentización

Para simplificar la escritura de las fórmulas omitimos ciertos paréntesis:

- Las reglas de precedencia de conectivos son las mismas que para $\mathtt{PROP}$.
- Conectivos de igual precedencia se asocian a la derecha.
- Cuantificadores: el $\forall$ y el $\exists$ tienen igual precedencia que el $\neg$.

Reglas de parentización

Atención: No confundir las siguientes fórmulas

$$(\forall x)(\alpha \to \beta) \text{ y } (\forall x)\alpha \to \beta$$
$$(\exists x)(\alpha \to \beta) \text{ y } (\exists x)\alpha \to \beta$$

Ejemplo (Parentizar la siguiente expresión)

$$(\forall x)P_1(x) \wedge \bot \to \bot \vee \neg P_1(x)$$
$$((\forall x)P_1(x)) \wedge \bot \to \bot \vee \neg P_1(x)$$
$$(((\forall x)P_1(x)) \wedge \bot) \to \bot \vee \neg P_1(x)$$
$$(((\forall x)P_1(x)) \wedge \bot) \to \bot \vee (\neg P_1(x))$$
$$(((\forall x)P_1(x)) \wedge \bot) \to (\bot \vee (\neg P_1(x)))$$
$$((((\forall x)P_1(x)) \wedge \bot) \to (\bot \vee (\neg P_1(x))))$$

Var, Const, AT

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

Def [Var]

Var es el conjunto de las variables de A ($\{x_i | i \in \mathbb{N}\}$).

Def [Const$_A$]

Const$_A$ es el conjunto de los símbolos de constante de A ($\{c_i | 1 \leq i \leq k\}$).

Def [fórmulas atómicas AT$_A$]

AT$_A$ es el conjunto de fórmulas de $\mathtt{FORM}_A$ que se obtienen con las cláusulas base $(\bot, P_j(t_1, \dots, t_{r_j}), t_1 =' t_2)$.

PIP y ERP en $\mathtt{TERM}$ y $\mathtt{FORM}$

Principio de inducción primitiva para $\mathtt{TERM}$

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

Lema 2.3.3 [principio de inducción para $\mathtt{TERM}_A$]

H) Sea $\mathcal{P}$ una propiedad sobre $\mathtt{TERM}_A$. Si se cumple

1. $\mathcal{P}(x)$ para todo $x \in \mathrm{Var}$
2. $\mathcal{P}(c)$ para todo $c \in \mathrm{Const}_A$
3. si $\mathcal{P}(t_1), \ldots, \mathcal{P}(t_{a_i})$ entonces $\mathcal{P}(f_i(t_1, \ldots, t_{a_i}))$ para todo $i \in \{1, \ldots, m\}$

T) Entonces se cumple $(\bar{\forall} t \in \mathtt{TERM}_A)\mathcal{P}(t)$.

Principio de inducción primitiva para $\texttt{FORM}$

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

Lema 2.3.3 [principio de inducción para $\texttt{FORM}_A$]

H) Sea $\mathcal{P}$ una propiedad sobre $\texttt{FORM}_A$. Si se cumple

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

T) Entonces se cumple $(\bar{\forall}\alpha \in \texttt{FORM}_A)\mathcal{P}(\alpha)$.

Esquema de recursión primitiva para TERM

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

Lema [esquema de recursión primitiva para $\mathtt{TERM}_A$]

H) Sean las siguientes funciones:

- $H_b : \mathrm{Var} \cup \mathrm{Const}_A \to B$
- $H_i : (\mathtt{TERM}_A \times B)^{a_i} \to B$, con $i \in \{1, \dots, m\}$

T) Entonces existe una única función $F : \mathtt{TERM}_A \to B$ tal que:

- $F(t) = H_b(t)$ si $t \in \mathrm{Var} \cup \mathrm{Const}_A$
- $F(f_i(t_1, \dots, t_{a_i})) = H_i(t_1, F(t_1), \dots, t_{a_i}, F(t_{a_i}))$

Esquema de recursión primitiva para FORM

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

Lema [esquema de recursión primitiva para $\mathtt{FORM}_A$]

H) Sean las siguientes funciones:

1. $H_{at} : \mathrm{AT}_A \to B$
2. $H_{\square} : (\mathtt{FORM}_A \times B)^2 \to B \quad (\square \in \{\to, \leftrightarrow, \wedge, \vee\})$
3. $H_{\neg} : \mathtt{FORM}_A \times B \to B$
4. $H_{\forall}, H_{\exists} : \mathrm{Var}_A \times \mathtt{FORM}_A \times B \to B$

T) Entonces existe una única función $F : \mathtt{FORM}_A \to B$ tal que:

- $F(\alpha) = H_{at}(\alpha)$ si $\alpha \in \mathrm{AT}_A$
- $F((\alpha \square \beta)) = H_{\square}(\alpha, F(\alpha), \beta, F(\beta))$
- $F((\neg\alpha)) = H_{\neg}(\alpha, F(\alpha))$
- $F(((\forall x)\alpha)) = H_{\forall}(x, \alpha, F(\alpha))$
- $F(((\exists x)\alpha)) = H_{\exists}(x, \alpha, F(\alpha))$

Variables libres y ligadas

Alcance de cuantificadores

Def [alcance o radio de acción]

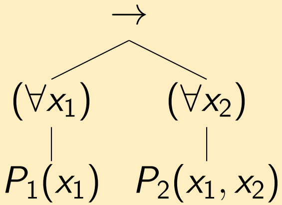
- El alcance del cuantificador $\forall x$ en la fórmula $((\forall x)\alpha)$ es la fórmula α.
- El alcance del cuantificador $\exists x$ en la fórmula $((\exists x)\alpha)$ es la fórmula α.

$$(\forall x_1)\ \boxed{P_1(x_1)} \rightarrow (\forall x_2)\ \boxed{P_2(x_1, x_2)}$$

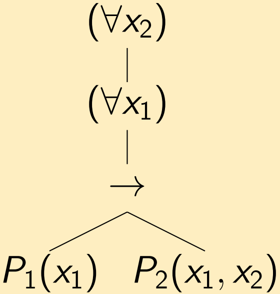
$$(\forall x_2)\ \boxed{(\forall x_1)\ \boxed{(P_1(x_1) \rightarrow P_2(x_1, x_2))}}$$

Alcance de cuantificadores

$(\forall x_1)\ \boxed{P_1(x_1)} \rightarrow (\forall x_2)\ \boxed{P_2(x_1, x_2)}$



$(\forall x_2)\ \boxed{(\forall x_1)\ \boxed{(P_1(x_1) \rightarrow P_2(x_1, x_2))}}$



Def. Ocurrencias libres y ligadas

- Una *ocurrencia* de una variable x en α *está ligada* si se encuentra bajo alcance de un cuantificador $(\forall x)$ o $(\exists x)$, o si es la variable de un cuantificador $(\forall x)$ o $(\exists x)$.
- Si una ocurrencia de una variable x no está ligada en α, se dice que es una *ocurrencia libre*.

Ejemplo

- $(\forall {\color{blue}x_1})P_1({\color{blue}x_1}) \to (\forall x_2)P_2({\color{red}x_1}, x_2)$
- $(\forall {\color{blue}x_1})P_1(c_1)$

Las ocurrencias azules de x_1 están ligadas, mientras que la roja no.

Def. Variables libres y ligadas

- Una *variable* x *está ligada* en α si x tiene alguna ocurrencia ligada en α.
- Una *variable* x *está libre* en α si x tiene alguna ocurrencia libre en α.

Ejemplo

Sea $\alpha = (\forall x_1)P_1(x_1) \to (\forall x_2)P_2(x_1, x_2)$

- x_1 tiene 2 ocurrencias ligadas en α
  - entonces x_1 es ligada en α
- x_1 tiene 1 ocurrencia libre en α
  - entonces x_1 es libre en α

Observación

- Una *ocurrencia* de variable en una fórmula está o bien libre o bien ligada (¡no ambas!).
- Una *variable* puede estar libre y ligada en una misma fórmula.

Def 2.3.6. Conjunto de variables libres de un término

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

Definimos $\mathrm{FV} : \mathtt{TERM}_A \to \wp(\mathsf{Var})$ recursivamente en $\mathtt{TERM}_A$:

- $\mathrm{FV}(x) = \{x\}$ si $x \in \mathsf{Var}$
- $\mathrm{FV}(c_i) = \emptyset$
- $\mathrm{FV}(f_i(t_1, \dots, t_{a_i})) = \mathrm{FV}(t_1) \cup \dots \cup \mathrm{FV}(t_{a_i})$

Def 2.3.7. Conjunto de variables libres de una fórmula

Definimos $\mathrm{FV} : \texttt{FORM}_A \to \wp(\mathsf{Var})$ recursivamente en $\texttt{FORM}_A$:

- $\mathrm{FV}(\bot) = \emptyset$
- $\mathrm{FV}(P_i(t_1, \ldots, t_{r_i})) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_{r_i})$
- $\mathrm{FV}(t_1 =' t_2) = \mathrm{FV}(t_1) \cup \mathrm{FV}(t_2)$
- $\mathrm{FV}((\alpha \Box \beta)) = \mathrm{FV}(\alpha) \cup \mathrm{FV}(\beta)$
- $\mathrm{FV}((\neg \alpha)) = \mathrm{FV}(\alpha)$
- $\mathrm{FV}(((\forall x)\alpha)) = \mathrm{FV}(\alpha) - \{x\}$
- $\mathrm{FV}(((\exists x)\alpha)) = \mathrm{FV}(\alpha) - \{x\}$

Variables ligadas de una fórmula

Ejercicio

Definir recursivamente en $\mathtt{FORM}_A$ la función $\mathrm{BV} : \mathtt{FORM}_A \to \wp(\mathsf{Var})$ que calcula el conjunto de variables ligadas de una fórmula.

Términos y fórmulas cerrados

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

Def 2.3.8 [términos y fórmulas cerradas]

- Un término t es *cerrado* si $FV(t) = \emptyset$.
- Una fórmula α es *cerrada* si $FV(\alpha) = \emptyset$. También se dice en este caso que α es una *sentencia*.
- Una fórmula α es *abierta* si no tiene cuantificadores.

Notación:

$$\mathtt{TERM}_{C\,A} = \{t \in \mathtt{TERM}_A \mid t \text{ es cerrado}\}$$
$$\mathtt{SENT}_A = \{\alpha \in \mathtt{FORM}_A \mid \alpha \text{ es cerrada}\}$$

Sustituciones en TERM y en FORM

Def 2.3.9 Sustitución de términos en términos

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

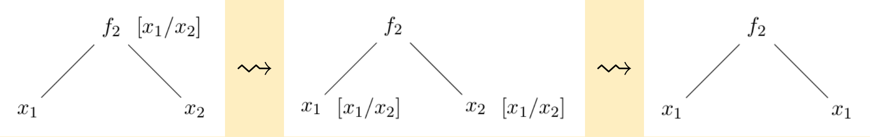
Sean $s, t \in \mathtt{TERM}_A$ y $x_j \in$ Var. Definimos $s[t/x_j]$ del siguiente modo:

1. $x_i[t/x_j] = \begin{cases} t & \text{si } i = j \\ x_i & \text{si } i \neq j \end{cases}$
2. $c_i[t/x_j] = c_i$
3. $f_i(t_1, \ldots, t_{a_i})[t/x_j] = f(t_1[t/x_j], \ldots, t_{a_i}[t/x_j])$

Ejemplo (Sustitución)

Sea $\mathcal{L}$ un lenguaje de tipo $\langle 1, 2; 1, 2; 2 \rangle$.

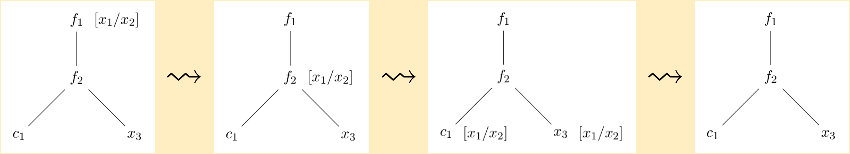
- $f_2(x_1, x_2)[x_1/x_2] = f_2(x_1, x_1)$
- $f_1(f_2(c_1, x_3))[c_2/x_3] = f_1(f_2(c_1, c_2))$
- $f_1(f_2(c_1, x_3))[c_2/x_1] = f_1(f_2(c_1, x_3))$



Ejemplo (Sustitución)

Sea $\mathcal{L}$ un lenguaje de tipo $\langle 1, 2; 1, 2; 2 \rangle$.

- $f_2(x_1, x_2)[x_1/x_2] = f_2(x_1, x_1)$
- $f_1(f_2(c_1, x_3))[c_2/x_3] = f_1(f_2(c_1, c_2))$
- $f_1(f_2(c_1, x_3))[c_2/x_1] = f_1(f_2(c_1, x_3))$



Def 2.3.10 Sustitución de variables por términos en fórmulas (1 de 3)

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

Sean $t \in \mathtt{TERM}_A, x_j \in \mathsf{Var}, \alpha \in \mathtt{FORM}_A$. Definimos $\alpha[t/x_j]$ del siguiente modo:

- $\bot[t/x_j] = \bot$
- $P_j(t_1, \ldots, t_{r_j})[t/x_j] = P_j(t_1[t/x_j], \ldots, t_{r_j}[t/x_j])$
- $t_1 =' t_2[t/x_j] = t_1[t/x_j] =' t_2[t/x_j]$

Def 2.3.10 Sustitución de variables por términos en fórmulas (2 de 3)

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

Sean $t \in \mathtt{TERM}_A, x_j \in \mathrm{Var}, \alpha \in \mathtt{FORM}_A$. Definimos $\alpha[t/x_j]$ del siguiente modo:

- $(\alpha \Box \beta)[t/x_j] = \alpha[t/x_j] \Box \beta[t/x_j]$
- $(\neg \alpha)[t/x_j] = (\neg \alpha[t/x_j])$

Def 2.3.10 Sustitución de variables por términos en fórmulas (3 de 3)

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

Sean $t \in \mathtt{TERM}_A, x_j \in \mathrm{Var}, \alpha \in \mathtt{FORM}_A$. Definimos $\alpha[t/x_j]$ del siguiente modo:

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

Sustitución y pasaje de parámetros

```
z := 45;
print (sumar2(z));
```

```
FUNCTION sumar2(x: integer) : integer;
VAR y : integer;
BEGIN
  y := 2;
  sumar2 := x + y
END;
```

Ejemplo (Sustitución)

Sea $\mathcal{L}$ un lenguaje de tipo $\langle 1, 2; 2; 2\rangle$.

- $P_1(f_1(x_1, x_2))[x_1/x_2] = P_1(f_1(x_1, x_1))$
- $(P_1(x_1) \to P_2(c_1, x_3))[c_2/x_1] = (P_1(c_2) \to P_2(c_1, x_3))$
- $((\exists x_1)P_2(x_1, x_3))[c_3/x_3] = ((\exists x_1)P_2(x_1, c_3))$
- $((\exists x_1)P_2(x_1, x_3))[c_1/x_1] = ((\exists x_1)P_2(x_1, x_3))$
- $((\exists x_1)P_2(x_1, x_3))[x_1/x_3] = ((\exists x_1)P_2(x_1, x_1))$

Ejemplo (Sustitución)

Sea $\mathcal{L}$ un lenguaje de tipo $\langle 1, 2; 2; 2 \rangle$.

- $P_1(f_1(x_1, x_2))[x_1/x_2] = P_1(f_1(x_1, x_1))$
- $(P_1(x_1) \to P_2(c_1, x_3))[c_2/x_1] = (P_1(c_2) \to P_2(c_1, x_3))$
- $((\exists x_1)P_2(x_1, x_3))[c_3/x_3] = ((\exists x_1)P_2(x_1, c_3))$
- $((\exists x_1)P_2(x_1, x_3))[c_1/x_1] = ((\exists x_1)P_2(x_1, x_3))$
- $((\exists x_1)P_2(x_1, x_3))[x_1/x_3] = ((\exists x_1)P_2(x_1, x_1))$
    - ¡Apareció una ligadura nueva! ¡No queremos estas situaciones!

¿Sustitución y pasaje de parámetros?

```
y := 45;
print (sumar2(y));
```

```
FUNCTION sumar2(x: integer) : integer;
VAR y : integer;
BEGIN
  y := 2;
  sumar2 := x + y
END;
```

¿Qué falló en el último caso?

- $((\exists x_1)P_2(x_1, x_3))[x_1/x_3] = ((\exists x_1)P_2(x_1, x_1))$
  - la variable x_3 estaba libre en $((\exists x_1)P_2(x_1, x_3))$
  - al sustituir x_3 por x_1 (que es la variable cuantificada por el $\exists$), queda ligada $((\exists x_1)P_2(x_1, x_1))$
- Lo mismo hubiera pasado si en vez de $[x_1/x_3]$ se pone $[t/x_3]$ con $x_1 \in \mathrm{FV}(t)$.
- El problema ocurre cuando hacemos $((\exists x_i)\alpha)[t/x]$ y x_i está en t : $x_i \in \mathrm{FV}(t)$
- Tenemos que exigir que $x_i \notin \mathrm{FV}(t)$
- Hay algunos casos en que no hace falta pedir esa condición. Entre ellos está cuando la sustitución "no se realiza" porque $x \notin \mathrm{FV}((\exists x_i)\alpha)$

Def 2.3.11. Término libre para una variable en una fórmula

Sean $t \in \mathtt{TERM}, \varphi \in \mathtt{FORM}$. *$t$ está libre para x en φ* si:

1. φ es atómica
2. $\varphi = (\varphi_1 \square \varphi_2)$ y t está libre para x en φ_1 y en φ_2
3. $\varphi = (\neg\varphi_1)$ y t está libre para x en φ_1
4. $\varphi = ((\forall y)\varphi_1)$ (o $\varphi = ((\exists y)\varphi_1)$) y se cumple alguna de las siguientes:
    1. $x \notin \mathrm{FV}(((\forall y)\varphi_1))$ (resp. $x \notin \mathrm{FV}(((\exists y)\varphi_1))$)
    2. $y \notin \mathrm{FV}(t)$ y t está libre para x en φ_1

Restricción en sustituciones

De ahora en adelante sólo aplicaremos la sustitución $\varphi[t/x]$ cuando t esté libre para x en φ. Si esto no sucede: ¡no permitimos aplicar la sustitución!

$((\exists x_1)P_2(x_1, x_3))[c_3/x_3]$ Podemos aplicar la función de sustitución, la aplicamos, y el resultado es $((\exists x_1)P_2(x_1, c_3))$.

$((\exists x_1)P_2(x_1, x_3))[c_1/x_1]$ Podemos aplicar la función de sustitución, la aplicamos, y el resultado es $((\exists x_1)P_2(x_1, x_3))$, igual que el argumento.

$((\exists x_1)P_2(x_1, x_3))[x_1/x_3]$ No podemos aplicar la función de sustitución.

Ejemplo

- x_2 está libre para x_1 en $(\exists x_1)P_1(x_1, x_3)$ pues $x_1 \notin \mathrm{FV}((\exists x_1)P_1(x_1, x_3))$.
- Cualquier t está libre para x_2 en $(\exists x_1)(\forall x_2)P_1(x_1, x_2)$ pues $x_2 \notin \mathrm{FV}((\exists x_1)(\forall x_2)P_1(x_1, x_2))$
- $f(x_3, x_1)$ no está libre para x_2 en $(\forall x_3)P_2(x_2)$ pues $x_2 \in \mathrm{FV}((\forall x_3)P_2(x_2))$, y $x_3 \in \mathrm{FV}(f(x_3, x_1))$ (aunque $f(x_3, x_1)$ está libre para x_2 en $P_2(x_2)$)
- $f(x_3, x_1)$ no está libre para x_2 en $(\forall x_4)(\exists x_3)(x_3 =' x_2)$ pues $x_2 \in \mathrm{FV}((\forall x_4)(\exists x_3)(x_3 =' x_2))$ y $f(x_3, x_1)$ no está libre para x_2 en $(\exists x_3)(x_3 =' x_2)$ (aunque $x_4 \notin \mathrm{FV}(f(x_3, x_1))$)

Sustitución simultánea en términos y fórmulas

- $t[t_1, \ldots, t_n/x_1, \ldots, x_n]$ es el resultado de sustituir las ocurrencias de cada x_i por t_i en t *simultáneamente* $(i = 1, \ldots, n, x_i \neq x_j$ si $i \neq j)$
- $\alpha[t_1, \ldots, t_n/x_1, \ldots, x_n]$ se define análogamente

Sustitución simultánea $\neq$ sustitución secuencial

No es lo mismo la sustitución simultánea que la composición de sustituciones:

- $x_1[x_2, c_2/x_1, x_2] = x_2$
- $x_1[x_2/x_1][c_2/x_2] = x_2[c_2/x_2] = c_2$

Notación: $[\alpha(x), \alpha(t)]$

- Para simplificar la notación $\alpha[t/x]$, en matemática se utilizan (meta)expresiones de la forma $\alpha(x)$ o $\alpha(x, y, z)$
- Esta notación no significa que las variables listadas ocurran libres en la fórmula ni que la fórmula no tenga otras variables libres que no sean las listadas...
- Sólo se utiliza para escribir informalmente la sustitución:
  Por ejemplo, $\alpha(t)$ notará el resultado de sustituir t por x en $\alpha(x)$; $\alpha(s, t, u)$ notará el resultado de sustituir s por x, t por y y u por z en $\alpha(x, y, z)$.

Símbolo de predicado $

- Las sustituciones definidas hasta el momento, permiten poner un *término* dado en el lugar de una variable.
- Eso es diferente de la sustitución de $\mathtt{PROP}$, que permitía poner una *fórmula* en el lugar de una *fórmula atómica*.
- Para hacer esto, se tiene que hacer una modificación en el lenguaje.

Agregamos a la definición de $\mathtt{FORM}$ una cláusula más:

- $\$ \in \mathtt{FORM}$ ($ es una variable de fórmula, la usamos como comodín para sustituir una fórmula en otra).

Def 2.3.13. Fórmula libre para \$

Sean $\alpha, \varphi \in \texttt{FORM}$. φ está libre para \$ en α si se cumple alguna de las siguientes condiciones:

1. α es atómica
2. $\alpha = (\alpha_1 \square \alpha_2)$ y φ está libre para \$ en α_1 y en α_2
3. $\alpha = (\neg \alpha_1)$ y φ está libre para \$ en α_1
4. $\alpha = ((\forall x)\alpha_1)$ (o $\alpha = ((\exists x)\alpha_1)$) y se cumple alguno de los siguientes:
    1. \$ no ocurre en α_1
    2. $x \notin \mathrm{FV}(\varphi)$ y φ está libre para \$ en α_1

Def 2.3.14. Sustitución de fórmulas en fórmulas

Sean $\alpha, \varphi \in \texttt{FORM}$ tal que φ está libre para \$ en α.
Definimos $\alpha[\varphi/\$]$ recursivamente en α:

1. si α es atómica, $\alpha[\varphi/\$] = \begin{cases} \alpha & \text{si } \alpha \neq \$ \\ \varphi & \text{si } \alpha = \$ \end{cases}$
2. $(\alpha_1 \Box \alpha_2)[\varphi/\$] = (\alpha_1[\varphi/\$] \Box \alpha_2[\varphi/\$])$
3. $(\neg \alpha_1)[\varphi/\$] = (\neg \alpha_1[\varphi/\$])$
4. $((\forall x)\alpha_1)[\varphi/\$] = ((\forall x)(\alpha_1[\varphi/\$]))$
5. $((\exists x)\alpha_1)[\varphi/\$] = ((\exists x)(\alpha_1[\varphi/\$]))$

Resumen

Resumen de Sintaxis de la Lógica de Predicados de Primer Orden I

- Se definieron dos familias de lenguajes:
  - uno para representar elementos del universo
  - otro para representar afirmaciones sobre esos elementos
- Se definieron los principios de Inducción y Recursión primitivas para esos lenguajes con el objetivo de:
  - hacer demostraciones
  - definir funciones recursivamente sobre esos lenguajes
- Se estudiaron los aspectos sintácticos que introducen las variables y se definieron algunos conceptos relativos a eso:
  - ocurrencia libre y ligada de una variable
  - término libre para una variable en una fórmula dada

Resumen de Sintaxis de la Lógica de Predicados de Primer Orden II

- Se definieron funciones de sustitución:
  - de una variable por un término en fórmulas y términos
  - de fórmulas en fórmulas

Bibliografía I

Dirk Van Dalen.
Logic and Structure.
Secciones 3.1, 3.2 y 3.3, 5ta ed.