Sintaxis y Propiedades

Estructuras

Def 2.2.1 [estructura]
Una estructura es una secuencia ordenada

$$\mathcal{M} = <A, R_1,\ldots R_n, F_1,\ldots F_m, \{c_i \mid i \in I\}>$$

tal que:

- A es un conjunto no vacío, (Notacion: $A = |\mathcal{M}|$)
- $R_1,\ldots R_n$ son relaciones sobre A ($n \geq 0$)
- $F_1,\ldots F_m$ son funciones en A ($m \geq 0$)
- c_i ($i \in I$) son elementos distinguidos de A

Ejemplos:

- $< N, Par, \leq, +, *, 0, 1>$ naturales
- $<N, <>$ CPO de los naturales
- $< Z, +, -, 0 >$ grupo de los enteros

Tipo de Similaridad

Def 2.2.2 [tipo de similaridad de una estructura]

El tipo de similaridad de $<A, R_1,\ldots R_n, F_1,\ldots F_m, \{c_i \mid i \in I\}>$ es una secuencia $<r_1,\ldots,r_n;\ a_1,\ldots,a_m;\ k>$ tal que:

- $R_i \subseteq A^{ri}$ $(1 \le i \le n$ y $r_i \ge 0)$
- $F_j : A^{aj} \to A$ $(1 \le j \le n$ y $a_j \ge 0)$
- $k = |\{c_i \mid i \in I\}|$ (el cardinal del conjunto $\{c_i \mid i \in I\}$)

Ejemplos:

$< N,Par,\le,+,*,0,1>$ tiene tipo $<1,2 ; 2,2 ; 2>$

$<N,<>$ tiene tipo $< 2 ; - ; 0 >$

$< Z, +, -, 0 >$ tiene tipo $<- ; 2,1 ; 1 >$

Alfabeto de Primer Orden

Def [alfabeto de un lenguaje de primer orden]

Un alfabeto para un lenguaje de primer orden de tipo $<r_1,\ldots,r_n;\ a_1,\ldots,a_m;\ k>$ consiste de los siguientes símbolos:

- Símbolos de relación: $P_1, P_2, \ldots, P_n, =$‘
- Símbolos de función: $f_1, f_2, \ldots, f_m$
- Símbolos de constantes: c_i tal que $i \in I$ y $|I| = k$
- Variables: $x_1, x_2, x_3, ..$
- Conectivos : $\to, \leftrightarrow, \neg, \wedge, \vee, \bot$
- Cuantificadores: $\forall, \exists$
- Auxiliares : ( ) ,

Términos

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $\langle r_1 \ldots r_n; a_1 \ldots a_m; k \rangle$

Def 2.3.1 [términos de un lenguaje de primer orden]

- El conjunto $TERM_A$ de los términos del lenguaje de primer orden con alfabeto A se define inductivamente por:

  i) $x_i \in TERM_A \ (i \in N)$

  ii) $c_i \in TERM_A \ (i \in I)$

  iii) si $t_1 \ldots t_{ai} \in TERM_A$, entonces $f_i (t_1, .. t_{ai}) \in TERM_A$

Fórmulas

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $\langle r_1 \ldots r_n; a_1 \ldots a_m; k \rangle$

Def 2.3.2 [FORM]

- El conjunto $FORM_A$ de las fórmulas del lenguaje de primer orden con alfabeto A se define inductivamente por:

  i) $\perp \in FORM_A$

  ii) Si $t_1 \ldots t_{rj} \in TERM_A$, entonces $P_j (t_1 \ldots t_{rj}) \in FORM_A$

  iii) Si $t_1, t_2 \in TERM_A$, entonces $t_1 = ' t_2 \in FORM_A$

  iv) Si $\alpha, \beta \in FORM_A$ entonces $(\alpha \in \beta) \in FORM_A$ para $\in \in \{\rightarrow, \leftrightarrow, \wedge, \vee\}$

  v) Si $\alpha \in FORM_A$ entonces $(\neg \alpha) \in FORM_A$

  vi) Si $\alpha \in FORM_A$ entonces $((\forall x_i)\alpha)$ y $((\exists x_i)\alpha) FORM_A$

Ejemplos de Términos y Fórmulas

Sea A un alfabeto P_1,P_2,f_1,f_2,c_1,c_2 de tipo <1,2; 1,2; 2 >

- $f_2(c_1,x_4) \in TERM_A$?
- $f_1(c_1,x_4) \in TERM_A$?
- $((\forall x_1)\ P_2(f_1(x_1),c_1))\rightarrow((\exists x_2)\ P_1(x_2)) \in FORM_A$?
- $((\exists x_2)\ f_2(x_1, c_2)) \in FORM_A$?
- $((\forall x_1)P_1(x_1, c_1)) \in FORM_A$?
- $((\exists x_1)\ ((\exists x_2)\ ((\exists x_3)\ P_3(x_1, x_2, x_3)))) \in FORM_A$?

OJO!!!

No confundir símbolo de predicado y símbolo de función!

$f_2(x_1, c_2) \in TERM$ y $P(x_1) \in FORM$

Reglas de parentización

- Para simplificar la escritura de las fórmulas, omitimos ciertos paréntesis:
  - Las reglas de precedencia de conectivos son las mismas que para 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: ejemplos

Atención: No confundir las siguientes fórmulas

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

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

Ejemplo: Parentizar la siguiente expresión:

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

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

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

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

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

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

Var, Const, AT

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Def [Var]

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

Def [$Const_A$]

- $Const_A$ es el conjunto de los símbolos de constante de A $(\{c_i | i \in I\})$.

Def [fórmulas atómicas, AT_A]

- AT_A es el conjunto de fórmulas de $FORM_A$ que se obtienen con las cláusulas base ($\bot$, $P_j(t_1,\ldots,t_{rj})$, $t_1 = t_2$).

Principio de Inducción Primitiva para TERM

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Lema 2.3.3 [principio de inducción para $TERM_A$]

Sea $\mathcal{P}$ una propiedad sobre $TERM_A$.

- Si se cumple:

  i) $\mathcal{P}(x)$ para todo $x \in Var$.

  ii) $\mathcal{P}(\underline{c})$ para todo $\underline{c} \in Const_A$.

  iii) si $\mathcal{P}(t_1) \ldots \mathcal{P}(t_{ai})$ entonces $\mathcal{P}(f_i\ (t_1, .. t_{ai}))$ para todo $1 \leq i \leq m$

Entonces para todo $t \in TERM_A$ se cumple $\mathcal{P}(t)$

Principio de Inducción Primitiva para FORM

Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Lema 2.3.4 [principio de inducción para $FORM_A$]

Sea $\mathcal{P}$ una propiedad sobre $FORM_A$.

- Si se cumple:

  i) $\mathcal{P}(\alpha)$ para todo α atómico.

  ii) si $\mathcal{P}(\alpha)$ y $\mathcal{P}(\beta)$ entonces $\mathcal{P}(\alpha \square \beta)$ $(\square \in \{\rightarrow, \leftrightarrow, \wedge, \vee\})$

  iii) si $\mathcal{P}(\alpha)$ entonces $\mathcal{P}(\neg\alpha)$

  iv) si $\mathcal{P}(\alpha)$ entonces $\mathcal{P}((\forall x)\alpha)$ y $\mathcal{P}((\exists x)\alpha)$ para todo $x \in Var$.

Entonces para todo $\alpha \in FORM_A$ se cumple $\mathcal{P}(\alpha)$

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.

$(\forall x_1)P_1(x_1) \rightarrow (\forall x_2)P_2(x_1,x_2)$

$(\forall x_2)(\forall x_1)(P_1(x_1) \rightarrow P_2(x_1, x_2))$

Variables Libres y Ligadas

Def [ocurrencias libres y ligadas]

- Una ocurrencia de una variable x en α es 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 es ligada en α en se dice que es una ocurrencia libre.

Def [variables libres y ligadas]

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

Ocurrencias Libres y Ligadas Ejemplos

$((\forall x_1)\ P_1(x_1)) \rightarrow (\forall x_2)\ P_2(x_1,x_2)$

Ocurrencia ligada

Ocurrencia libre

$(\forall x_1)\ P_1(c_1)$

Ocurrencia ligada

Variables Libres y Ligadas Ejemplo

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

- 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

Obs:

- una ocurrencia de variable en una fórmula es o bien libre o bien ligada (no ambas!)
- una variable puede ser libre y ligada en una misma fórmula

Esquema de Recursión Primitiva para Term

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Lema [esquema de recursión primitiva para $TERM_A$]

- Sean las siguientes funciones:
  - H_b: $Var \cup Const_A \rightarrow B$
  - H_i : $(TERM_A \times B)^{ai} \rightarrow B$, con $i \in \{1,\ldots,m\}$

Entonces existe una única función $F:TERM_A \rightarrow B$ tal que:

- $F(t) = H_b(t)$ si $t \in Var \cup Const_A$
- $F(f_i(t_1,\ldots,t_{ai})) = H_i(t_1, F(t_1), \ldots, t_{ai}, F(t_{ai}))$

Esquema de Recursión Primitiva para FORM

Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Lema [esquema de recursión primitiva para $FORM_A$]

Sean las siguientes funciones:

H_{at}: $AT_A \rightarrow B$

$H_{\square}$: $(FORM_A \times B)^2 \rightarrow B$ $(\in \in \{\rightarrow, \leftrightarrow, \wedge, \vee\})$

$H_{\neg}$: $FORM_A \times B \rightarrow B$

$H_{\forall}, H_{\exists}$: $Var_A \times FORM_A \times B \rightarrow B$

Entonces, existe una única función $F:FORM_A \rightarrow B$ tal que

$F(\alpha) = H_{at}(\alpha)$ si $\alpha \in AT_A$

$F(\alpha \square \beta) = H_{\square}(\alpha, F(\alpha), \beta, F(\beta))$ $(\in \in \{\rightarrow, \leftrightarrow, \wedge, \vee\})$

$F(\neg \alpha) = H_{\neg}(\alpha, F(\alpha))$

$F((\forall x)\alpha) = H_{\forall}(x, \alpha, F(\alpha))$

$F((\exists x)\alpha) = H_{\exists}(x, \alpha, F(\alpha))$

Variables de un Término

Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Def 2.3.6 [conj. de variables (libres) de un término]

Definimos $FV : TERM_A \rightarrow \mathcal{P}(Var)$ recursivamente en $TERM_A$:

$FV(x) = \{ x \}$ si $x \in Var$

$FV(\underline{c}_i) = \varnothing$

$FV(f_i (t_1, \ldots, t_{ai})) = FV(t_1) \cup \ldots \cup FV(t_{ai})$.

Variables de una Fórmula

Def 2.3.7 [conj. de variables libres de una fórmula]

- Definimos $FV : FORM_A \rightarrow P(Var)$ recursivamente en $FORM_A$:
  - $FV(\bot) = \varnothing$
  - $FV(P_j (t_1, \ldots, t_{rj})) = FV(t_1) \cup \ldots, \cup FV(t_{rj})$
  - $FV(t_1 =' t_2) = FV(t_1) \cup FV(t_2)$
  - $FV(\alpha \square \beta) = FV(\alpha) \cup FV(\beta)$ $(\square \in \{\rightarrow, \leftrightarrow, \wedge, \vee\})$
  - $FV(\neg \alpha) = FV(\alpha)$
  - $FV((\forall x)\alpha) = FV((\exists x)\alpha) = FV(\alpha) - \{ x \}$

Ejercicio: Definir recursivamente en $FORM_A$ la función $BV : FORM_A \rightarrow P(Var)$ que calcula el conjunto de variables ligadas de una fórmula

Terminos y Fórmulas Cerrados

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Def 2.3.8 [términos y fórmulas cerradas]

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

Notación:

$TERMc_A = \{t \in TERM_A \mid t \text{ es cerrado}\}$

$SENT_A = \{\alpha \in FORM_A \mid \alpha \text{ es cerrada}\}$

Sustituciones en TERM y en FORM

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i$ $(i \in I)$ de tipo $<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Def 2.3.9 [sustitución de términos en términos]

- Sean $s,t \in TERM_A$ y $x_j \in VAR$. Definimos $s[t/x_j]$ del siguiente modo:

  i. $x_i\,[t/x_j] = \begin{cases} t & \text{si } i=j \\ x_i & \text{si } i \neq j \end{cases}$

  ii. $\underline{c}_k\,[t/x_j] = \underline{c}_k$

  iii. $f_i(t_1, \ldots, t_{ai})\,[t/x_j] = f(t_1[t/x_j], \ldots, t_{ai}[t/x_j])$

Sustitución de Variables Libres por Términos en Fórmulas (1)

- Sea A un alfabeto $P_1 \ldots P_n, f_1 \ldots f_m, c_i (i \in I)$ de tipo$<r_1 \ldots r_n; a_1 \ldots a_m; k>$

Def 2.3.10 [sustitución de términos en fórmulas]

- Sea $t \in TERM_A$, $x_j \in VAR$ y $\alpha \in FORM_A$.

Definimos $\alpha[t/x]$ del siguiente modo:

i. $\bot [t/x_j] = \bot$

ii. $P_j(t_1,..., t_{rj}) [t/x_j] = P_j(t_1[t/x_j],..., t_{rj}[t/x_j])$

iii. $(t_1 =' t_2) [t/x_j] = (t_1[t/x_j] =' t_2[t/x_j])$

iv. $(\alpha \square \beta) [t/x_j] = (\alpha [t/x_j] \square \beta [t/x_j])$ $(\in \in \{\rightarrow, \leftrightarrow, \wedge, \vee\})$

v. $(\neg\alpha)[t/x_j] = \neg(\alpha[t/x_j])$

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

$((\exists x_i)a\ )[t/x_j]$ --- análogo a $((\forall x_i)\ \alpha)[t/x_j]$ ---

Ejemplos de Sustitución

Sea L un lenguaje de tipo <1,2;2,2;2>

- $P_1(f_2(x_1,x_2))[x_1/x_2] = P_1(f_2(x_1,x_1))$
- $(P_1(x_1) \rightarrow P_2(c_1,x_3))[c_2/x_1] = (P_1(c_2) \rightarrow P_2(c_1,x_3))$
- $((\exists x_1)\ P_2(x_1,x_3))[c_3/x_3] = ((\exists x_1)\ P_2(x_1,c_3))$
- $((\exists x_1)\ P_2(x_1,x_3))[c_1/x_1] = ((\exists x_1)\ P_2(x_1,x_3))$
- $((\exists x_1)\ P_2(x_1,x_3))[x_1/x_3] = ((\exists x_1)\ P_2(x_1,x_1))$ ← cambia totalmente el sentido!

↓

No queremos estas situaciones!

Ejemplo trucho

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

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

Si acá pongo y

…y acá pongo y

Identificaré las dos variables, y devolveré 4

Ejemplo trucho (cont.)

- El ejemplo anterior no es real. Los lenguajes tipo Pascal implementan pasajes de parámetro por copia o por referencia, y no realizan una sustitución textual del código, como nosotros.
- Sin embargo, igualmente ilustra la dificultad de no considerar el contexto en que se realiza una sustitución.

Que 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 $((\exists \mathbf{x_1})\ P_2(\mathbf{x_1},\mathbf{x_3}))$
- ➔ al sustituir **x_3** por x_1 (que es la variable que cuantifica el ∃), queda **ligada**: $((\exists \mathbf{x_1})\ P_2(\mathbf{x_1},\mathbf{x_1}))$
- y lo mismo hubiera pasado si en vez de $[x_1/x_3]$ se pone $[t/x_3]$ con $x_1 \in FV(t)$.
- El problema ocurre cuando hacemos $((\exists x_i)\alpha)[t/x]$ y x_i está en t : **$x_i \in FV(t)$**

➔ ***luego, pedimos que*** **$x_i \notin FV(t)$**

- Hay algunos casos en que no hace falta pedir esa condición. Entre ellos está cuando la sustitución "no se realiza" porque **$x \notin FV((\exists x_i)\ \alpha)$**

Término Libre para una Variable

Def 2.3.11 [término libre para una variable]

- Sean $t \in TERM, \alpha \in FORM$: **t está libre para x en α** si se cumple alguna de las siguientes condiciones:
  - i. α es atómica
  - ii. $\alpha = (\alpha_1 \square \alpha_2)$ y t está libre para x en α_1 y en α_2
  - iii. $\alpha = (\neg\alpha_1)$ y t está libre para x en α_1
  - iv. $\alpha = (\forall x_i)\ \alpha_1$ y ($x \notin FV((\forall x_i)\alpha_1)$ o $x_i \notin FV(t)$ y t está libre para x en α_1)
  - v. $\alpha = (\exists xi)\ \alpha_1$ --- análogo que $(\forall xi)\ \alpha_1$ ---
- De ahora en adelante sólo aplicaremos la sustitución $\alpha[t/x]$ cuando t esté libre para x en α. Si esto no sucede: no se puede aplicar la sustitución!

Ejemplos

- x_2 está libre para x_1 en $(\exists x_1)P_1(x_1,x_3)$
  - pues $x_1 \notin FV((\exists x_1)P_1(x_1,x_3)$
- cualquier t está libre para x_2 en $(\exists x_1)(\forall x_2)P_1(x_1,x_2)$
  - pues $x_2 \notin FV((\exists x_1)(\forall x_2)P_1(x_1,x_2))$
- $f(x_3,x_1)$ no está libre para x_2 en $(\forall x_3)P_2(x_2)$
  - pues $x_2 \in FV((\forall x_3)P_2(x_2))$
  - y $x_3 \in V(f(x_3,x_1))$ (aunque $f(x_3,x_1)$ está libre para x_2 en $P_2(x_2)$)

Ejemplos

- $f(x_3,x_1)$ no está libre para x_2 en $(\forall x_4)(\exists x_3)(x_3=`x_2)$
  - pues $x_2 \in FV((\forall x_4)(\exists x_3)(x_3=`x_2))$
  - y $f(x_3,x_1)$ no está libre para x_2 en $(\exists x_3)(x_3=`x_2)$
    - pues $x_2 \in FV((\exists x_3)(x_3=`x_2))$
    - y $x_3 \in V(f(x_3,x_1))$ (aunque $x_4 \notin V(f(x_3,x_1)))$

Sustitución Simultánea

Def [sustitución simultánea]

- **t [t_1,..., t_n/ x_1,..., x_n]** es el resultado de sustituir las ocurrencias de cada x_i por t_i en t *simultáneamente* (i=1,...,n)
- α [t_1,..., t_n/ x_1,..., x_n] se define análogamente
- CUIDADO! No es lo mismo sustitución simultánea que 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: [α(x), α(t)]

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

$ y Fórmulas Libres para $

Agregamos a la definición de FORM una cláusula más:

- $ $\in$ FORM ($ es una variable de fórmula, la usamos como comodín para sustituír una fórmula en otra)

Def 2.3.13 [fórmula libre para $]

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

i. α es atómica

ii. $\alpha = (\alpha_1 \square \alpha_2)$ y φ está libre para $ en α_1 y en α_2

iii. $\alpha = (\neg\alpha_1)$ y φ está libre para $ en α_1

iv. $\alpha = (\forall x_i)\alpha_1$ o $\alpha = (\exists x_i)\alpha_1$ y ($ no ocurre en α_1 o $x_i \notin FV(\varphi)$ y φ está libre para $ en α_1)

Sustitución de Fórmulas en Fórmulas

Def 2.3.14 [sustitución de fórmulas en fórmulas]

Sean $\alpha,\varphi \in$ FORM tal que φ está libre para $ en α.

Definimos $\alpha[\varphi/\$]$ del siguiente modo:

i. si α es atómica, entonces $\alpha[\varphi/\$] = \begin{cases} \alpha & \text{si } \alpha \neq \$ \\ \varphi & \text{si } \alpha = \$ \end{cases}$

ii. $(\alpha \square \beta)[\varphi/\$] = (\alpha[\varphi/\$] \square \beta[\varphi/\$])$

iii. $(\neg\alpha)[\varphi/\$] = \neg(\alpha[\varphi/\$])$

iv. $((\forall x_i)\alpha)[\varphi/\$] = (\forall x_i)(\alpha[\varphi/\$])$

v. $((\exists x_i)\alpha)[\varphi/\$] = (\exists x_i)(\alpha[\varphi/\$])$

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

- Se definieron dos familias de lenguajes:
  - Uno para representar elementos del universo.
  - Otro para representar afirmaciones sobre esos elems.
- Se definieron los principios de Inducción y Recursión primitivas para esos lenguajes con el objetivo de:
  - Hacer demostraciones
  - Definir funciones recursivamente sobre esos lenguajes.

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

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