Lógica de Predicados: Motivación

$\underbrace{\text{Todo natural es entero}}_{p}$ y $\underbrace{\text{2 es un natural.}}_{q}$ Luego $\underbrace{\text{2 es entero.}}_{r}$

$p, q \models r$ es claramente un razonamiento válido pero

no es posible demostrarlo desde la Lógica Proposicional

Lógica proposicional NO es suficientemente expresiva para captar esta relación

$$\frac{\begin{array}{c}\forall x\,(x \in \mathrm{N} \rightarrow x \in \mathrm{Z}) \\ 2 \in \mathrm{N}\end{array}}{2 \in \mathrm{Z}}$$

La validez del razonamiento depende de la estructura interna de las proposiciones

debe expresarse usando Lógica de Predicados

Lógica de Predicados de Primer Orden

LENGUAJE DE PRIMER ORDEN

- Símbolos para denotar individuos
  - constantes (ej. 2, Juan, bicicleta)
  - variables (ej. x, y, z)
  - funciones (ej. sucesor, +, *, para definir nuevos individuos como sucesor(2), (1+1), (2*1))
- Símbolos de relaciones (entero(x), hermano(x, y))
- Conectivos
- Cuantificadores (existencial, universal)

Lenguaje de Primer Orden

Alfabeto básico consta de los siguientes símbolos:

- Variables: Var = $\{x, y, z, x_1, x_2, \ldots\}$
- Conectivos $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- Cuantificadores: $\forall$ (universal) $\exists$ (existencial)
- Símbolos auxiliares (,)

Símbolos comunes a todo lenguaje

- Conjunto de símbolos F, símbolos de funciones n-arias, $n \geq 1$ f, g, h
- Conjunto de símbolos C, símbolos de constantes a, b, c
- Conjunto de símbolos R, símbolos de relaciones o predicados n-arios, $n \geq 1$ P, Q, R

Símbolos propios de cada lenguaje

Lenguaje de Primer Orden

✓ Caracterizado por sus símbolos propios

$L = \langle R, F, C \rangle$ puede ocurrir $R = \emptyset$, $F = \emptyset$, o $C = \emptyset$

Ejemplos

Sea $L_1 = \langle R, F, C \rangle$

$R = \{P, Q\}$ P unario, Q binario

$F = \{ f \}$ f unaria

$C = \emptyset$

Fórmulas bien definidas en L_1

$Q(x, x) \wedge \exists x P(x)$

$\forall x(P(x) \rightarrow Q(f(x), x))$

Ejemplos

Sea $L_2 = \langle R, F, C \rangle$

$R = \{P\}$ P binario

$F = \{ f, g \}$ f unaria g binaria

$C = \{a, b\}$

Fórmulas bien definidas en L_2

$P(f(a), b) \rightarrow \forall x P(x, b)$

$\exists x(P(x, a) \wedge P(g(x, b), x))$

Lenguaje de Primer Orden

Dado L = <R, F, C>

✓ Términos de L Ter(L) (representan elementos del dominio)

a) C $\subseteq$ Ter(L)

b)Var $\subseteq$ Ter(L)

c) Si $t_1, t_2, ..., t_n \in$ Ter(L) y f $\in$ F es un símbolo de función n-aria, entonces $f(t_1, t_2, ..., t_n) \in$ Ter(L)

Ejemplos

Sea L_2 = <R, F, C>

R = {P} P binario

F = { f, g } f unaria g binaria

C = {a, b}

Ter(L_2)

a) a, b (las constantes son términos)

b) x, y, z, ... (las variables son términos)

c) f(a), g(a, b), f(g(a, b)), g(f(a), x), ...

(funciones aplicadas a términos son términos)

Lenguaje de Primer Orden

Dado L = <R, F, C>

✓ Fórmulas Atómicas de L At(L)

Si $t_1, t_2, ..., t_n \in$ Ter(L) y P $\in$ R es un símbolo de relación n-aria, entonces $P(t_1, t_2, ..., t_n) \in$ At(L)

Ter(L_2)

a) a, b

b) x, y, z, ...

c) f(a), g(a, b), f(g(a, b)), g(f(a), x), ...

At(L_2)

P(a, b), P(a, a), P(x, y), P(x, b),

P(f(a), g(a, b)), P(z, f(g(a, b))), ...

Lenguaje de Primer Orden

Dado L = <R, F, C>

✓ Fórmulas de L $F_m(L)$

a) $At(L) \subseteq F_m(L)$

b) Si A, B $\in F_m(L)$ entonces $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B) \in F_m(L)$

c) Si A $\in F_m(L)$ entonces $(\forall xA)$, $(\exists xA)$ $\in F_m(L)$ $\quad$ $(x \in Var)$

$At(L_2)$

P(a, b), P(a, a), P(x, y), P(x, b),

P(f(a), g(a, b)), P(z, f(g(a, b))),

...

$F_m(L_2)$

$(P(a, b) \wedge P(a, a))$

$(\exists x(P(x, y) \vee (\neg P(x, b))))$

$(P(f(a), g(a, b)) \wedge P(z, f(g(a, b))))$

...

Lenguaje de Primer Orden

Para simplificar la escritura de las fórmulas, podemos eliminar ciertos paréntesis, siguiendo las reglas:

- $\neg$, $\forall x$, $\exists x$ tienen mayor precedencia que los conectivos binarios
- los conectivos binarios tienen la misma precedencia que en la Lógica Proposicional: $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ (de mayor a menor)

$F_m(L_2)$

	Eliminando paréntesis
$(P(a, b) \wedge P(a, a))$	$P(a, b) \wedge P(a, a)$
$(\exists x(P(x, y) \vee (\neg P(x, b))))$	$\exists x(P(x, y) \vee \neg P(x, b))$
$(P(f(a), g(a, b)) \wedge P(z, f(g(a, b))))$	$P(f(a), g(a, b)) \wedge P(z, f(g(a, b)))$

Lenguaje de Primer Orden

Alcance de un cuantificador

Es la fórmula afectada por el cuantificador. Si $\forall x A$ o $\exists x A$ es una fórmula, el alcance del cuantificador $\forall x$ o $\exists x$ es la fórmula A

Ejemplos

$\forall x\, P(x) \rightarrow \forall y\, R(x, y)$

$\forall x (P(x)) \rightarrow \forall x (R(x, y))$

$\forall x\, \forall y ((P(x) \rightarrow R(x, y)))$

Lenguaje de Primer Orden

Variables libres y ligadas

Una ocurrencia de una variable x en una fórmula A se dice ligada si está dentro del alcance de un cuantificador. En caso contrario se dice libre.

Ejemplos

Ocurrencia libre de x

$\forall x\, P(x) \rightarrow \forall y\, R(x, y)$

Ocurrencia ligada de x

Ocurrencia ligada de y

En una fórmula A

- cada ocurrencia de una variable es o libre o ligada en A.

- una misma variable puede tener ocurrencias libres y ligadas en A

Lenguaje de Primer Orden

Una **variable** x **ocurre libre** en una fórmula A si:

- Si A es atómica, x ocurre libre en A sí y sólo sí x es variable de A

- Si A = ¬B, x ocurre libre en A sí y sólo sí x ocurre libre en B

- Si A = B * C, x ocurre libre en A sí y sólo sí x ocurre libre en B o en C

(siendo * alguno de los conectivos binarios)

- Si A = ∀yB o A = ∃yB, x ocurre libre en A sí y sólo sí x ≠ y y x ocurre libre en B.

$A(x_1, x_2, ..., x_n)$ indica que las variables libres de A están en el conjunto $\{x_1, x_2, ..., x_n\}$

Lenguaje de Primer Orden

Una **fórmula** A se dice **cerrada** (o **sentencia**) cuando no tiene variables libres (cada variable está dentro del alcance de un cuantificador).

$\forall x\ \forall y\ (P(x) \rightarrow R(x, y))$ FORMULA CERRADA

$\forall x\ P(x) \rightarrow \forall x\ R(x, y)$ FORMULA NO CERRADA (y es libre)

La **clausura universal de una fórmula** $A(x_1, x_2, ..., x_n)$ es la sentencia

$\forall x_1\ \forall x_2 \dots \forall x_n\ A(x_1, x_2, ..., x_n)$

La **clausura existencial de una fórmula** $A(x_1, x_2, ..., x_n)$ es la sentencia

$\exists x_1\ \exists x_2 \dots \exists x_n\ A(x_1, x_2, ..., x_n)$

Ciencias de la Computación II - Filminas de Clase – Facultad Cs. Exactas – UNCPBA - 2019

Lenguaje de Primer Orden

Sustitución

Si A es una fórmula, x una variable libre de A y t un término, la sustitución de x por t en A, A(x/t), es la fórmula que se obtiene al reemplazar en A cada ocurrencia libre de x por el término t.

Ejemplo

$A = \forall x R(x, y) \wedge B(y)$ $A(y/c) = \forall x R(x, c) \wedge B(c)$ c constante

$A = \forall x P(x) \rightarrow Q(x)$ $A(x/c) = = \forall x P(x) \rightarrow Q(c)$ c constante

Sustitución simultánea

Si A es una fórmula $A(x_1, x_2, ..., x_n)$, y $t_1, t_2, ..., t_n$ son términos entonces la sustitución simultánea $A(x_1/t_1, x_2/t_2, ..., x_n/t_n)$ es la fórmula que se obtiene al reemplazar en A cada ocurrencia libre de x_i por el término t_i.

Lenguaje de Primer Orden

Un **término** t se dice **libre para una variable** x en una **fórmula** A si ninguna ocurrencia libre de x está dentro del alcance de un cuantificador $\forall y$ o $\exists y$ donde y es una variable de t.

Si t es libre para x en A entonces t se puede sustituir en todas las ocurrencias libres de x sin que alguna variable y de t quede dentro del alcance de un cuantificador.

Resumiendo:

- Sólo sustituiremos ocurrencias libres de las variables

- Las ocurrencias de variables que aporte cada término sustituyente deben resultar libres en la fórmula final.

Estas restricciones garantizan que la fórmula resultante de la sustitución será (in)satisfacible si la original lo era.

Lenguaje de Primer Orden

Ejemplos

$A_1 = \forall x(B(x) \rightarrow C(y))$

- El término t = f(x), f símbolo de función unaria, no es libre para y en A_1

- El término t = f(z) es libre para y en A_1

$A_1(y/f(z)) = \forall x(B(x) \rightarrow C(f(z)))$

$A_2 = \forall xB(x, y) \rightarrow \forall zB(z, x)$

- El término t = g(x, w), g símbolo de función binaria, no es libre para y en A_2

- El término t = g(y, z) es libre para y en A_2 y pero no es libre para x en A_2

$A_2(y/g(y, z)) = \forall xB(x, g(y, z)) \rightarrow \forall zB(z, x)$

Lenguaje de Primer Orden

Composición de sustituciones: Dadas e_1 y e_2 sustituciones

$e_1 = \{ x_1/t_1, x_2/t_2, ..., x_n/t_n\}$ $\qquad$ $e_2 = \{ y_1/s_1, y_2/s_2, ..., y_k/s_k\}$

$e_1.e_2 = \{ x_i/t_ie_2 : x_i \neq t_ie_2, i = 1, ..., n\} \cup \{ y_j/s_j : y_j \neq x_i , i = 1, ..., n, j = 1, ..., k\}$

Ejemplo:

$e_1 = \{ x/g(y, a), y/b, z/f(w), u/w\}$ $\qquad$ $e_2 = \{ y/f(b), w/u, t/g(a, f(b))\}$

a, b ctes.

$e_1.e_2 = \{ x/g(f(b), a), y/b, z/f(u), w/u, t/g(a, f(b))\}$

Ciencias de la Computación II - Filminas de Clase – Facultad Cs. Exactas – UNCPBA - 2019

Semántica de Primer Orden

Modelos o interpretaciones:

Sea L = <R, F, C> un lenguaje de primer orden. Un modelo M en L es una estructura M = <D, R^D, F^D, C^D> donde:

- ✓ **D dominio** o universo de interpretación (conjunto no vacío del cual las variables toman valores)
- ✓ **R^D** conjunto de relaciones n-arias sobre D tal que para cada símbolo $P \in R$ existe una relación $P^D \subseteq D^n$ asignada a P
- ✓ **F^D** conjunto de funciones n-arias sobre D tal que para cada símbolo $f \in F$ existe una función $f^D: D^n \rightarrow D$ asignada a f
- ✓ **C^D** conjunto de elementos distinguidos de D tal que para cada constante $c \in C$ existe un elemento $c^D \in D$ asignado a c

Semántica de Primer Orden

Ejemplo:

Sea L = <R, F, C> un lenguaje de primer orden con una relación binaria Q, una función unaria f y una constante c

Dadas $F_1 = \forall x Q(x, c)$ y $F_2 = \exists x Q(f(x), c) \rightarrow \exists z Q(z, z)$ sobre L

Para el modelo M = <D, {Q^D}, {f^D}, {c^D}>

D = {1, 2, 3} $c^D = 2$

$Q^D(x, y) = \{(1, 2), (2, 1), (3, 3)\}$ $f^D(1) = f^D(2) = 3$ $f^D(3) = 1$

F_1 es falsa en M y F_2 es verdadera en M

Formalización de Lenguaje Natural

Formalizar frase en lenguaje natural → encontrar expresión en lenguaje formal que la represente fielmente

🡣 No hay procedimientos generales para la formalización

🡡 Existen algunas estrategias o heurísticas

- Si la estructura sintáctica de la frase es compleja, se puede **reescribir** con una estructura más sencilla que mantenga el mismo significado
- Definir el **dominio** al cual pertenecen los elementos a utilizar
- Determinar:
  - **Constantes**: elementos concretos del dominio
  - **Variables**: elementos genéricos
  - **Funciones**: representan cómo un elemento queda determinado por otros
  - **Predicados unarios**: representan propiedades de un elem.
  - **Predicados de aridad > 1**: representan relaciones entre elem
- Identificar conectivas linguísticas y cuantificadores y sustituir por **conectivos** y **cuantificadores** de la lógica de primer orden

Formalización de Lenguaje Natural

<u>Patrones más habituales</u>:

- **Universal afirmativo** $\forall x(A(x) \rightarrow B(x))$

  Todo A es B - Sólo los B son A – No hay ningún A que no sea B

- **Universal negativo** $\forall x(A(x) \rightarrow \neg B(x))$

  Ningún A es B

- **Existencial afirmativo** $\exists x(A(x) \wedge B(x))$

  Algún A es B – Alguien es a la vez A y B

- **Existencial negativo** $\exists x(A(x) \wedge \neg B(x))$

  Algún A no es B – No todos los A son B

Formalización de Lenguaje Natural

Relación entre cuantificadores:

- **Universal/Existencial**

  $\neg \forall x A(x) \equiv \exists x \neg A(x)$

"No todos son A" equivale a decir "Algunos no son A"

- **Existencial/Universal**

  $\neg \exists x A(x) \equiv \forall x \neg A(x)$

"No hay A" equivale a decir "Todos son no A"