

---

## Taller No.2: Cálculo de predicados

---

William A. Gómez Roa

*wa.gomez@javeriana.edu.co*

*Bioingeniería y Ciencia de*

*Datos*

1. Si se define un nuevo cuantificador  $\epsilon$  de la forma:

$$\epsilon x P(x) = P(x_1) \oplus P(x_2) \oplus \dots \oplus P(x_n)$$

- (a) Cuando es verdadero. Cuando es falso.

El cuantificador  $\epsilon x P(x) = P(x_1) \oplus P(x_2) \oplus \dots \oplus P(x_n)$  es verdadero en uno de estos dos casos: 1) cuando solo un término es verdadero y todos los demás son falsos, y 2) cuando solo un término es falso y todos los demás son verdaderos. En ambos casos, el cuantificador es verdadero. Por otro lado, el cuantificador es falso en los demás casos. Al igual que la operación XOR.

- (b)Cuál es la relación de este cuantificador con el cuantificador universal o con el cuantificador existencial.

Mientras que el cuantificador universal se enfoca en la verdad de todas las proposiciones, el cuantificador existencial en la verdad de al menos una proposición, el nuevo cuantificador se enfoca en la verdad de exactamente una proposición o exactamente todas menos una proposiciones.

2. Determine si las siguientes formulas son equivalentes

$$\forall x \exists y P(x, y); \exists y \forall x (x, y)$$

2.

$$\forall x \exists y P(x, y) ; \exists y \forall x P(x, y)$$

1.  $\forall x \exists y P(x, y)$

2.  $\forall x (P(x, y_1) \vee P(x, y_2) \vee \dots \vee P(x, y_n))$

3.  $\exists y \forall x P(x, y)$

4.  $\exists y (P(x_1, y) \wedge P(x_2, y) \wedge \dots \wedge P(x_n, y))$

5. ② no es equivalente a ④

3. Determine la validez del siguiente argumento.

Alguien en esta clase disfruta de la programación en Python. Cada persona que disfruta de la programación en Python se interesa por ser un desarrollador en Python. Por lo tanto, hay una persona en esta clase que se interesa por ser desarrollador en Python.

→ Alguien en esta clase disfruta de la programación en Python

$$\exists x \text{ Disfruta}(x)$$

→ Cada persona que disfruta de la programación en Python se interesa por ser un desarrollador en Python!

$$\forall x (\text{Disfruta}(x) \rightarrow \text{Interesa}(x))$$

•  $\text{Disfruta}(x)$ : "x disfruta de la programación en Python"

•  $\text{Interesa}(x)$ : "x se interesa por ser un desarrollador en Python"

Queremos demostrar: "Hay una persona en esta clase que se interesa por ser desarrollador de Python".

$$\exists x \text{ Interesa}(x)$$

⇒ Demostrar:

$$\exists x \text{ Disfruta}(x), \forall x (\text{Disfruta}(x) \rightarrow \text{Interesa}(x))$$

∴

$$\exists x \text{ Interesa}(x)$$

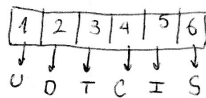
1.  $\text{Disfruta}(x)$  Eliminación Cuant. Existencial
2.  $\text{Disfruta}(x) \rightarrow \text{Interesa}(x)$  Eliminación Cuant. Universal
3.  $\text{Interesa}(x)$  Modus Ponens
4.  $\exists x \text{ Interesa}(x)$  Introducción Cuant. Existencial

4. Una sala de profesores consta de seis puestos de trabajo para los profesores Ismael García, Gerardo Tole, Eduardo E Kassir, Eddy Herrera, Liliana Barreto y Marisol Correa, los puestos están ubicados y numerados de la siguiente manera:

1	2	3	4	5	6
---	---	---	---	---	---

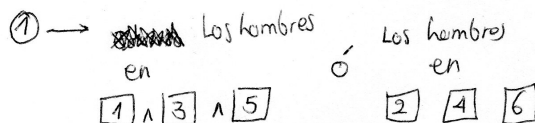
Cómo debe asignar el director del departamento a cada profesor un puesto de trabajo, cumpliendo con los siguientes requerimientos:

"Ningun hombre puede quedar al lado de otro hombre. Eduardo debe quedar al lado de Eddy; Liliana debe quedar en un extremo; si Gerardo queda en el puesto 4 Marisol no puede quedar a su lado; si algun hombre queda en el puesto 2, en el puesto 5 debe estar Eddy. "



- Ningun hombre puede quedar en lado de otro hombre. → ①
- Eduardo debe quedar al lado de Eddy → ②
- Liliana debe quedar en un extremo → ③
- Si Gerardo queda en el puesto 4 Marisol ~~no~~ puede quedar a su lado → ④
- Si algún hombre queda en el puesto 2, en el puesto 5 debe estar Eddy. → ⑤

Eduardo	- Eddy
Ismael	- Liliana
Gerardo	- Marisol



$$\begin{aligned} ② \rightarrow & (U(\text{Eduardo}) \wedge D(\text{Eddy})) \vee (D(\text{Eduardo}) \wedge (U(\text{Eddy}) \vee T(\text{Eddy}))) \vee (T(\text{Eduardo}) \wedge (D(\text{Eddy}) \vee C(\text{Eddy}))) \\ & \vee (C(\text{Eduardo}) \wedge (I(\text{Eddy}) \vee T(\text{Eddy}))) \vee (I(\text{Eduardo}) \wedge (S(\text{Eddy}) \vee C(\text{Eddy}))) \\ & \vee (S(\text{Eduardo}) \wedge I(\text{Eddy})) \end{aligned}$$

$$③ \rightarrow U(\text{Liliana}) \vee S(\text{Liliana})$$

$$④ \rightarrow C(\text{Gerardo}) \wedge (\neg T(\text{Marisol}) \vee \neg I(\text{Marisol}))$$

$$⑤ \rightarrow (D(\text{Gerardo}) \vee D(\text{Eduardo}) \vee D(\text{Ismael})) \rightarrow I(\text{Eddy})$$

1	2	3	4	5	6
Liliana	Gerardo	Marisol	Eduardo	Eddy	Ismael

5. Utilice un CAS para evaluar predicados. Sugerencia : Como referencia corra el archivo Predicates.cdf.


**SWISH**

File Edit Examples Help


96 users online

Program

```

1 % Definición de Las relaciones de parentesco
2 padres(william, nelly).
3 padres(nelly, laura).
4 padres(mateo, nelly).
5 padres(candelaria, nelly).
6 padres(misael, william).
7 padres(cecilia, william).
8 padres(william, nislo).
9 padres(nislo, juan).
10 padres(mateo, marina).
11 padres(mateo, nataly).
12 padres(candelaria, marina).
13 padres(candelaria, nataly).
14
15 % Regla para determinar si alguien es abuelo o abuela
16 abuelo(Abuelo, Nieto) :-
17     padres(Abuelo, PadreMadre),
18     padres(PadreMadre, Nieto).

```


abuelo(mateo, william).

false


abuelo(mateo, laura).

true

Next 10 100 1,000 Stop


abuelo(mateo, andres).

false

? - abuelo(mateo, andres).