

# Lógica para Ciencias de la Computación

Laboratorio Prolog: Clase 2

# Términos

- Utilizan un espectro de valores no vacío llamado Universo del Discurso.
- Representan propiedades y relaciones entre los objetos del dominio.
- Pueden ser constantes o variables.

# Constantes

- Las constantes dan nombre a objetos o individuos conocidos de nuestro universo. Sirven para representar propiedades y relaciones entre los objetos del dominio. Existen de dos tipos:
  - **Átomos**: Cadenas de letras, dígitos y subrayado (  ) que empiezan por una letra **minúscula**.
    - Toda cadena de caracteres encerrada entre comillas simples ('').
    - Toda combinación especial de signos: “:-”, “?-”, etc.
  - **Números**: son utilizados con el fin de poder realizar operaciones aritméticas.
    - Enteros: utilizables en el espectro de valores  $[-2^{23}, 2^{23}-1]$ .
    - Reales: números decimales, suele utilizarse notación científica.

# Valores válidos vs valores no válidos

Átomos válidos	Átomos no válidos		Números Válidos	Números no Válidos
f	2sillas		-135	135-
vacio	Vacio		1.67	.2
juan_parra	juan-parra		1.4E4	2.
'Juan Parra'	_Juan		1.4E+4	1.4e4
a356	356a		1.4E-4	1.4+4

# Variables

- Son utilizadas para representar cualquier objeto del Universo que se encuentre en la base de conocimiento.
- A diferencia de los átomos, las variables empiezan siempre con letra mayúscula o con el símbolo de subrayado (\_).
- Cualquier identificador que empiece con mayúscula será considerado como una variable.
- Cuando se trabaje con objetos desconocidos cuya identidad no nos interesa se puede utilizar la variables anónima (\_).
- Cuando una variable es instanciada, su contenido no puede cambiar.

# Ejemplo

- ***gusta(juan, X).*** -----> equivale a  $\forall x$  ***gusta(juan, x)***
- Desde la base de conocimiento significa que a Juan le gusta cualquier cosa.
- Desde una consulta a la base de conocimiento significa “¿Qué cosas le gustan a Juan?”, por lo que prolog indicará todas las cosas que a Juan le gustan.

# Reglas

- Agrega conocimiento complejo a la base de Prolog.
- Se utiliza el símbolo “:-” para separar la regla
  - De lado izquierdo la cabeza de la regla (cuantificación universal)
  - De lado derecho el cuerpo de la regla (cuantificación existencial).
- Por ejemplo enseñemosle a Prolog cuando una persona es abuelo de otra.
  - ¿Cuando una persona es abuelo de otra?
    - R: cuando es padre de su padre.

```
esAbuelo(X,Y) :- esPadre(X,Z) , esPadre(Z,Y) .
```

- Esto se lee: “*X es abuelo de Y, si X es padre de Z y Z es padre de Y*”.

# Conectivos lógicos

- Útiles para trabajar con sentencias más complejas que necesitan de la combinación de fórmulas atómicas combinadas mediante conectivos.
- La lógica de primer orden permite utilizar los conectores:
  - Conjunción
  - Disyunción
  - Negación
  - Implicancia



# Conjunción

- Conocida también como “Y” y “AND”. Es representado por una coma “,” en prolog.
- Sirve para conectar diferentes sentencias. Para satisfacer la secuencia se deberán satisfacer ***todas*** las sentencias.
- Ejemplo:
  - “X es abuelo de Y, si X es padre de Z y Z es padre de Y”

```
esAbuelo(X,Y) :- esPadre(X,Z) , esPadre(Z,Y) .
```

# Disyunción

- Conocida también como “O” y “OR”. Es representado por un punto y coma “;” en prolog.
- Sirve para conectar diferentes sentencias. Para satisfacer la secuencia se deberá satisfacer **al menos una** de las sentencias.
- Ejemplo:
  - “H es hijo de P, si P es padre de H o P es madre de H”.

```
esHijo(H, P) :- esPadre(P, H) ; esMadre(P, H) .
```

# Negación

- Aunque Prolog no permite explícitamente representar la negación lógica, se puede hacer de manera implícita.
- Al utilizar la sentencia de negación, se tendrá éxito si el objetivo fracasa.
- No es una verdadera negación lógica, sino que una negación por fallo.
  - Se puede obtener la negación de X de las siguientes maneras:
    - `not(X)`
    - `\+ X`

# Implicación o condicional

- Indica que un hecho depende de un grupo de otros hechos
- Lo hemos estado utilizando desde la primera regla que creamos, utiliza el símbolo “:-” para indicar que a continuación viene un conjunto de reglas unidas por conectores AND y OR.

```
cabeza_de_la_regla :- cuerpo_de_la_regla.
```

# Formas de leer las reglas

La regla  $C :- O_1, O_2, \dots, O_n$ .

- La cabeza describe el hecho a definir, mientras que el cuerpo los objetivos que deben satisfacerse para que la cabeza sea cierta.
- Declarativo: “La demostración de la cláusula  $C$  se sigue de la demostración de los objetivos  $O_1, O_2, \dots, O_n$ ”.
- Procedimental: “Para ejecutar el procedimiento de  $C$ , se debe llamar para su ejecución los objetivos  $O_1, O_2, \dots, O_n$ ”.
- Cómo proposición:
  - cuerpo\_de\_las\_regla  $\rightarrow$  cabeza\_de\_la\_regla

# Ejemplos

```
esNieto(N,A):-esPadre(X,N),esPadre(A,X) ; esPadre(X,N),esMadre(A,X)  
; esMadre(X,N),esMadre(A,X) ; esMadre(X,N),esPadre(A,X) .
```

```
sonPrimos(H1,H2):- esPadre(X,H1),esPadre(Y,H2),sonHermanos(X,Y) .
```

```
not(son_amigos(philip, francisco)) .
```

# Simbología y resumen

- (“,”): aplica conjunción entre dos o más condiciones (**AND**).
- (“;”): aplica disyunción entre dos o más condiciones (**OR**).
- **A=B**: intenta unificar A y B, devuelve true si es así.
- **A\=B**: es falso si A y B se unifican.
- **A is B**: se evalúa o calcula B y se unifica con A.
- **A:=B**: Evalúa y compara A y B. Verdadero si son iguales.
- **A\=B**: Evalúa y compara A y B. Falso si son iguales.
- **=<, >=, < y >** se comportan normalmente.

# Actividad 1

- Realice la base de conocimiento para los operadores lógicos:
  - Negación (NOT)
  - Disyunción (OR)
  - Conjunción (AND)
  - Implicancia
  - Equivalencia



## Actividad 2

- A partir de lo visto anteriormente, generar una base de conocimiento en prolog de su familia.
- Modo fácil: **esPadre()** ← involucra al padre o a la madre.
- Modo normal: se debe crear **esPadre()** y **esMadre()** por separado.
- Realizar las consultas que validen la información almacenada en la base de conocimiento.

# Actividad 3

- Analice el siguiente sistema de diagnóstico de enfermedades y responda:
  - ¿Qué se le debe recetar a Manuel?
  - ¿Qué se le debe recetar a Alicia?

```
enfermo_de(manuel,gripe).
tiene_sintoma(alicia,cansancio).
sintoma_de(fiebre,gripe).
sintoma_de(tos,gripe).
sintoma_de(cansancio,anemia).
elimina(vitaminas,cansancio).
elimina(aspirinas,fiebre).
elimina(jarabe,tos).
recetar_a(X,Y):-enfermo_de(Y,A),alivia(X,A).
alivia(X,Y):-elimina(X,A),sintoma_de(A,Y).

enfermo_de(X,Y):-tiene_sintoma(X,Z),sintoma_de(Z,Y).
```

# Actividad 4

- Calcule la potencia de un número (ej:  $N1$  elevado a  $N2$ ).