



PROLOG y la programación lógica

- Paradigmas de programación: imperativa, orientación a objetos, funcional y declarativa
- Programación lógica
 - Existe un motor de inferencias



1. Elementos de Prolog

1. Hechos
2. Consultas
3. Variables
4. Conjunciones
5. Reglas



1.1. Hechos

- Un hecho describe una propiedad de un objeto
 - Ejemplo: el diamante es valioso
 - Hecho:
valioso(diamante).
 - Predicado monádico: Px
- El programador debe definir la interpretación de los objetos y las relaciones entre ellos:
 - Objetos: juan, libro
 - Relación:
tiene(juan, libro).
 - Predicados poliádicos, en este caso, diádico: Pxy
- El orden de la relación es importante: $Pxy \neq Pyx$



1.1. Hechos

- Ejemplo:
 - Objetos: Juan, Maria, Cocido, Vino.
 - Relaciones: A <Objeto> le gusta <Objeto>
 - Hechos:
 - A Juan le gusta el cocido
 - A Juan le gusta el vino
 - A Maria le gusta el vino
 - PROLOG:
 - a_le_gusta(juan, cocido).
 - a_le_gusta(juan, vino).
 - a_le_gusta(maria, vino).
 - Con esto, se ha hecho la definición del “mundo”



1.2. Consultas

- Comienzan por:
?-
- Equivale a preguntar: “la consulta se deduce de la base de conocimiento (hechos + reglas).”
- Ante una consulta, PROLOG intenta hacer un búsqueda (*matching*) sobre la base de conocimiento:
 - Busca el mismo predicado
 - Verifica que hay el mismo número de argumentos
 - Igualdad entre argumentos: términos / variables (instanciadas)

1.2. Consultas

- Las respuestas a una consulta pueden ser:
 - Yes → *Se deduce de la BC*
 - No → *No se deduce; no es lo mismo que falso*
- ¿Qué responder a una consulta?
 - [ENTER] → Termina
 - ; → ¿Hay más respuestas?

Base de conocimiento	Consultas	Respuestas
a_le_gusta(jose, maria).	?- le_gusta_a(maria,jose).	No
a_le_gusta(maria, libro).	?- a_le_gusta (maria, libro).	Yes
a_le_gusta(juan, coche).	?- a_le_gusta (juan, pescado).	No
a_le_gusta(jose, pescado).	?- a_le_gusta (jose, pescado).	Yes



1.3. Variables

- Una variable puede estar:
 - libre
 - ligada (sujeta a cuantificador universal o existencial)
- En PROLOG todas las variables empiezan por Mayúsculas o “_” (más adelante)



1.3. Variables

?- a_le_gusta(maria, X).

?- a_le_gusta(juan, X), a_le_gusta(maria, X).

- Aunque en PROLOG se suelen utilizar X, Y,... las variables pueden tener nombres nemotécnicos:

?- a_le_gusta(maria, Persona).

- Ante una consulta se busca la primera instanciación (ligadura) de la variable que satisfaga el predicado (si hay alguna).
- En ese momento se añade una marca y si pedimos más respuestas ([:]) se consulta a partir de esa marca.



1.4. Conjunciones

- Permiten expresar múltiples objetivos que deben satisfacerse simultáneamente:

- ¿Se gustan Juan y María?

?- a_le_gusta(juan, maria), a_le_gusta(maria,juan).

?- a_le_gusta(jose, maria), a_le_gusta(maria, jose).

Viene a ser como el conjunto de fórmulas satisfacibles.

- Permitimos la CONJUNCIÓN LÓGICA entre (n) objetivos
- PROLOG intentará satisfacer TODOS los objetivos, en el orden dado (de **izquierda a derecha**).
- Tras satisfacer cada objetivo, marca el punto en la BC y las instanciaciones que ha hecho para llegar ahí.



1.4. Conjunciones

- Ejemplo: ¿Hay algo que le guste a Juan y a María?

?- a_le_gusta(juan, Algo), a_le_gusta(maria, Algo).
¿?

?- a_le_gusta(juan, X), a_le_gusta(jose, Y).
¿?

- Las conjunciones implican instanciar y desinstanciar, para satisfacer o resatisfacer objetivos → reevaluación (con backtracking)



1.5. Reglas

- Establece dependencias entre hechos, de forma compacta: *afirmación general sobre objetos y sus relaciones*.
- Esto está íntimamente ligado con el concepto de CONSECUENCIA LÓGICA
- Una regla tendrá:
<cabeza> :- <cuerpo>
<cuerpo> → hechos con conjunciones y variables

Ejemplo:

Juan es amigo de aquellos a quienes les gustan los coches.

En PROLOG:

es_amigo(Juan,Y) :- a_le_gusta(Y, coche).



4. Ejercicios propuestos

- Queremos una BC sobre las personas de una familia, que represente un árbol genealógico de tres niveles (abuelos, padres e hijos):

Definimos las personas (hombres y mujeres) que lo componen implícitamente a través de su género.

Definimos las relaciones de parentesco más sencillas:

`es_padre_de(X,Y)`

`matrimonio_con(U,V)` V: familiar político

Define reglas que permitan saber:

- Quién es hijo de quién.
- Quién es abuelo/a de quién.
- Quién es padre/madre de quién.
- Quién es hermana/hermano de quién.
- Quién es tío/a de quién.
- Quién es primo/a de quién.
- Quién es sobrino/a de quien.