

PROLOG

LENGUAJE DECLARATIVO

Prolog es un lenguaje de programación declarativo, lo que significa que es un lenguaje basado en la declaración de condiciones, proposiciones, afirmaciones...

En el caso de prolog se hace la declaración de **hechos** y **reglas** como se verán mas adelante.

BASE DE CONOCIMIENTO

- Para responder a las preguntas o consultas formuladas por el programador, Prolog consulta una **base de conocimiento**. Esta base conocimiento representa el programa como tal, programa que se compone únicamente de clausulas, que con el uso de la lógica, me expresan el conocimiento deseado por el programa.
- La base de conocimiento o el programa se guarda en un archivo con la extensión '.pl', archivo que puede ser abierto y a partir de esto poderle hacer consultas a mi programa.

TÉRMINOS

Los términos en prolog son los componentes que conforman el lenguaje, y en este caso éstos van a ser los únicos elementos que componen un programa.

Constantes

Átomo: Son nombres de objetos, propiedades, o relaciones. Estos deben empezar en minúscula.

`atomo(luis).`

`atomo(color).`

`atomo(padre).`

`atomo('pedro').`

Número: Valores que solo pueden ser entero o reales, pueden llevar el signo.

Ejemplos de las diferentes formas de expresar un número:

numero(2).

numero(216565).

numero(1.54521).

numero(-5).

numero(-5.0).

numero(2e10).

Variables

Se representan mediante cadenas representadas por letras, o por el símbolo '_', para que Prolog las tome como variables, éstas deben empezar en mayúscula o con '_'.

`variable(X).`

`variable(Variable).`

`variable(_).`

`variable(_var).`

Una variable anónima se representa por el nombre '_' con la cual en cada instancia de ésta variable se refiere a una variable distinta..

Estructuras

Estos son términos compuestos por otros términos, donde la sintaxis que se tiene es la siguiente:

nombre_atomo(termino1, termino2, ..., terminoN).

Donde esos términos del 1 al N, se les llama **argumentos** . Además, al nombre del átomo también se le llama **predicado**.

padre(luis). % Estructura que toma un solo argumento.

edad(luis, 30). % Estructura que ya se compone por más de un argumento.

color(X). % Estructura con átomo llamado color y con un argumento que es una variable.

EXPRESIONES

Los operadores nos permiten manipular diferentes tipos de datos.

OPERADORES ARITMÉTICOS

Con estos podemos llevar a cabo operaciones aritméticas entre números de tipo entero o real, sin embargo se tuvieron en cuenta sólo los básicos, pero existen para las funciones trigonométricas, valor absoluto, mínimo, máximo, entre otros muchas más.

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/ y //	División real y entera
^ y **	Potencia
+	Positivo
-	Negativo

OPERADORES RELACIONALES CON EVALUACIÓN

Las operaciones relacionales nos permiten establecer relaciones de orden.

Este tipo de operadores recibe valores numéricos y/o expresiones antes de realizar unificación o comparaciones evalúa el valor de la expresión.

Operador	Significado	Ejemplo
is	Unificación	$X \text{ is } 10 + 2.$
==	Igualdad	$10 + 2 == 5 + 7.$
!=	Desigualdad	$10 + 2 != 5 + 8.$
>	Mayor que	$11 * 3 > 3 ^ 2.$
<	Menor que	$2 ** 10 < 5 * 2.$
>=	Mayor o igual que	$99.0 >= 0.$
=<	Igual o menor que	$-15 =< 15.$

OPERADORES RELACIONALES SIN EVALUACIÓN.

Operador	Significado	Ejemplo
=	Unificación	$X = 10 + 2.$
==	Igualdad	$10 + 2 == 10 + 2.$
\==	Desigualdad	$10 + 2 \neq 5 + 7.$

CLÁUSULAS

Las cláusulas en Prolog están basadas en cláusulas de Horn.

$$p_1 \wedge p_2 \wedge \dots \wedge p_m \Rightarrow p$$

Lo cual sería equivalente a tener en Prolog:

p :- p1, p2, ..., pm.

Donde p es la Cabeza y todos los pi son el Cuerpo, y cada uno de ellos son átomos.

come(A,B) :- carnivoroso(A), animal(B), masDebil(B, A);

herbivoroso(A), plantaComestible(B).

Reescritas a lenguaje natural:

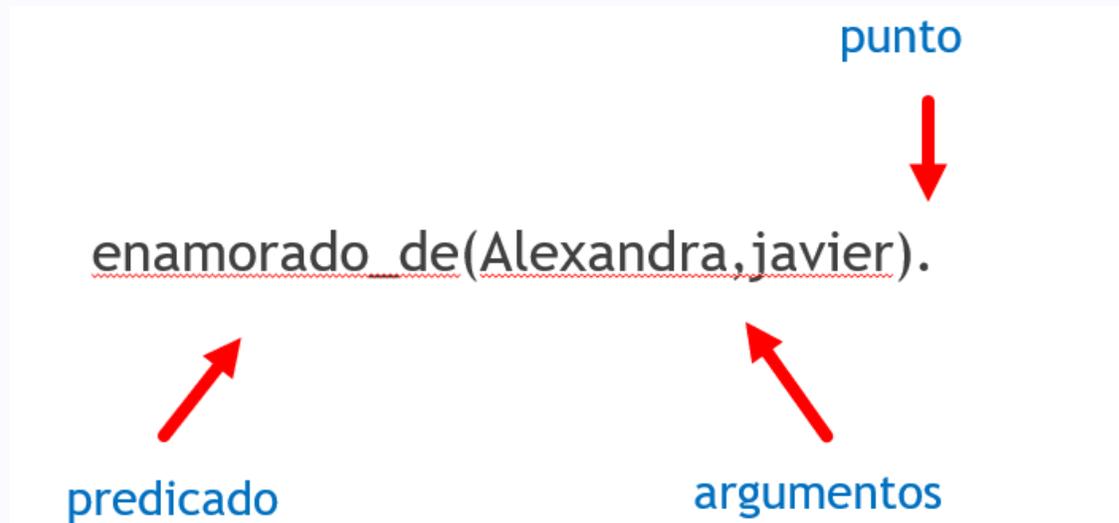
"A come a B si, A es carnívoro y B es animal y B es más débil que A, o si A es herbívoro y B es una planta comestible."

TIPOS DE CLÁUSULAS

- Una cláusula con cabeza y cuerpo es llamada **Regla**.
- Sin cuerpo es un **Hecho** o **Afirmación**.
- Sin cabeza es una **Pregunta** o **Consulta**.

HECHOS

- Un hecho es un mecanismo para representar propiedades o relaciones de los objetos que se están representando. Los hechos declaran los valores que van a ser verdaderos o afirmativos para un predicado en todo el programa.
- Los hechos siguen la siguiente sintaxis: nombre_predicado(argumentos).



- Los hechos se dividen en 2 tipos: propiedades y relaciones.

PROPIEDADES

Las propiedades se caracterizan por llevar un solo argumento y de esta manera expresan una propiedad de los objetos. Por ejemplo:

`color(azul).`% azul es color - Denota la propiedad del azul de ser un color

`color(verde).` % verde es color

`padre(juan).`%Juan es padre - Denota la propiedad que tiene juan y es la de ser padre.

`padre(pablo).` % Pablo es padre

RELACIONES

Las relaciones se caracterizan por llevar más de un argumento y de esta manera expresan la relación entre varios objetos. Por ejemplo:

`padrede('juan', 'maria').` % Juan es padre de maría - Este hecho expresa una relación de padre-hijo

`padrede('pablo', 'juan').` % Pablo es padre de juan

`edad(juan, 30).` % Juan tiene la edad de 30 años - Este hecho está relacionando a juan con un edad de 30 años, expresando una verdad o afirmación

`edad('pablo', 50).`

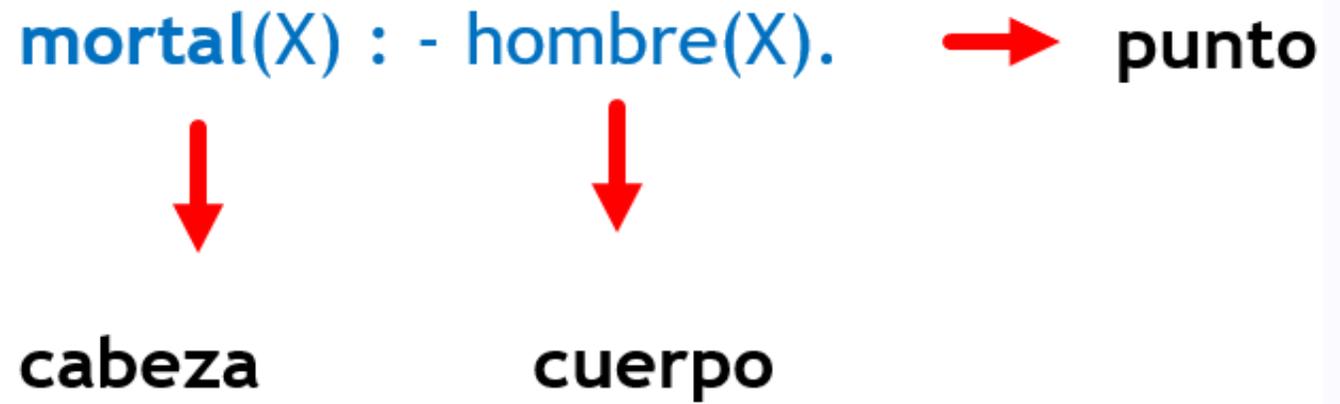
REGLAS

Cuando la verdad de un hecho depende de la verdad de otro hecho o de un grupo de hechos se usa una regla.

Permiten establecer relaciones más elaboradas entre objetos donde se declaran las condiciones para que un predicado sea cierto, combinando hechos para dar el valor de verdad del predicado.

La sintaxis base para una regla es la siguiente:

CABEZA :- CUERPO



La forma como se debe leer esta sintaxis es de la siguiente manera: “La cabeza es verdad si el cuerpo es verdad”.

De esta manera se obtendrá el valor de verdad de la cabeza con el valor que se obtenga en el cuerpo, si el cuerpo resulta falso, la cabeza será falsa.

Por ejemplo:

%A es hijo de B si B es padre de A

hijode(A,B) :- **padrede**(B,A).

% A es abuelo de B si A es padre de C y C es padre B

abuelode(A,B) :- **padrede**(A,C), **padrede**(C,B).

CONJUNCIONES

Se usa una coma para separar los hechos del cuerpo de la regla. Este 'separador' se traduce como un AND lógico, concatenado cada hecho con un AND.

Por ejemplo:

% X es hermano de Y si existe algún padre Z que sea padre de X y Y

hermano(X, Y) :- padre(Z), **padrede**(Z, X), **padrede**(Z, Y).

DISYUNCIONES

Se usa un punto y coma para separar los hechos del cuerpo de la regla. Este 'separador' se traduce como un OR lógico, concatenado cada hecho con un OR.

Por ejemplo:

% A y B son familiares si A es padre de B o A es hijo de B o A es hermano de B

familiarde(A,B) :- padrede(A,B); hijode(A,B); hermanode(A,B).

CONSULTAS

Es el mecanismo para extraer conocimiento del programa, donde una consulta está constituida por una o más metas que Prolog debe resolver.

- **Hecho:**

1 amigos(pedro, antonio).

- **Consulta:**

?- amigos(pedro, antonio).