Universidad del Cauca

ISO 9001:2015 SC- CER 450832

IQNet: CO- SC-CER450832

Una Acreditación con Rostro Humano

Hacia una Universidad comprometida con la paz territorial

Universidad del Cauca

ESTRUCTURAS DE LENGUAJES

INTRODUCCIÓN
Prolog

Prof: Pablo H. Ruiz
Unicauca

Introducción

- PROLOG (“PROgramation et LOGique”), fue creado por Alain Colmerauer y sus colaboradores alrededor de 1970 en la Universidad de Aix-Marseille, Francia para hacer deducciones a partir de texto.
- PROLOG juega un importante papel dentro de la Inteligencia Artificial (Sistemas expertos)
- Se basa en el cálculo de predicados.
- Programar en Prolog consiste en dar al ordenador un Universo finito en forma de hechos y reglas. -> Base de conocimiento
- A continuación, si se hacen las preguntas adecuadas, Prolog buscará las respuestas en dicho Universo y las presentará en la pantalla.
- La programación en Prolog consiste en:
  - Declarar algunos **HECHOS** sobre los objetos y sus relaciones,
  - Definir algunas **REGLAS** sobre los objetos y sus relaciones, y
  - Hacer **PREGUNTAS** sobre los objetos y sus relaciones.

Predicados

Universidad del Cauca

- Cláusula de Horn sin cabeza
- En Prolog se llaman **hechos**.
  - Ej. figura(cuadrado).
  - Ej. hermano(david, jorge).
- Los nombres de todos los objetos y relaciones deben comenzar con una letra minúscula.
- Primero se escribe la relación o propiedad: *predicado*
- Los objetos se escriben separándolos mediante comas y encerrados entre paréntesis: *argumentos*.
- Al final del predicado debe ir un punto (".").

Términos

Universidad del Cauca

- Los términos pueden ser *constantes* o *variables*.
- Las constantes representan objetos concretos y conocidos en el dominio. Son valores fijos que no cambian durante la ejecución. Existen dos tipos de constantes: átomos y números.
  - **Átomos**: Un átomo puede ser una cadena de letras, dígitos o caracteres especiales, que usualmente empiezan con una letra minúscula.
  - También pueden ser cadenas encerradas en comillas simples.
    - Átomos sin comillas: Ejemplo: carlos, perro, ventana_1.
    - Átomos con comillas: Ejemplo: 'Madrid', 'Hola Mundo'.
    - Combinaciones especiales: Algunas combinaciones de signos como ?-, :- son usadas en Prolog para representar consultas o reglas.
      - Ejemplo: :- es usado para definir reglas.

Términos

Átomos con comillas vs átomos sin comillas

Característica	Átomos sin comillas	Átomos con comillas
Formato	Solo letras minúsculas, dígitos, guiones bajos. Debe comenzar con minúscula.	Cualquier cadena de caracteres, incluyendo espacios, mayúsculas, y símbolos.
Usos típicos	Nombres simples, etiquetas.	Cadenas más complejas, nombres con espacios o mayúsculas.
Ejemplo	carlos, perro, ventana_1	'Madrid', 'Hola Mundo', 'Juan Pérez'
Cuando usarlos	Para identificadores simples, nombres de objetos.	Cuando se necesitan caracteres especiales, espacios o formato mixto.

Términos

Universidad del Cauca

- En Prolog, los números son considerados átomos cuando representan valores que se utilizan para operaciones aritméticas o para representar información numérica.
- Existen dos tipos de números: enteros y reales.

%Enteros

edad(carlos, 30).
enteroaltura(edificio, 100).
enteropeso(perro, 15).

%Reales

pi(3.1416).
altura(montaña, 8848.86).
temperatura(hoy, 21.5).

Términos

Universidad del Cauca

- Ejemplos de constantes:

Átomos válidos	Átomos no válidos	Números válidos	Números no válidos
carlos	2peras	12	123-
juan_perez	Nombre	-1.23	.2
‘Juan Perez’	juan-perez	1.2E+3	2.
a123	_sandra	1.2E-3	1.2+3

Variables

Universidad del Cauca

- Las **variables** se utilizan para representar cualquier objeto del Universo. Objetos desconocidos en ese momento, es decir, son las incógnitas del problema.
- Las variables son usadas para hacer **inferencias y relacionar** diferentes términos.
- Representan valores que pueden ser asignados o "**unificados**" en la ejecución del programa.
- Se diferencian de los átomos en que empiezan siempre con **una letra mayúscula o con el signo de subrayado** (_).
- Se puede representar una variable anónima usando sólo el símbolo (_).
- Ejemplos:
  - X
  - Sumando
  - Primer_factor
  - _indice
  - _

Variables

Universidad del Cauca

- Las variables anónimas se representan usando solo el símbolo de guion bajo (_) y son útiles cuando no interesa el valor que tome una variable en una consulta o una regla.
- A diferencia de las variables normales, las variables anónimas no guardan el valor que unifican ni son reutilizables dentro de la misma consulta o regla.

Ejemplos:

1. Ignorar un argumento en una consulta

```
es_persona(carlos).
es_persona(maria).
 es_persona(juan).

?- es_persona(_).
```

si hay alguna persona en la base de datos, pero **no importa quién es**, se puede usar una variable anónima

2. Ignorar un argumento en una relación

```
vive_en(carlos, madrid).
 vive_en(maria, barcelona).
vive_en(juan, madrid).

?- vive_en(_, madrid).
```

Si se quiere saber si alguien vive en Madrid, **pero no importa el nombre de la persona**, se puede usar una variable anónima para ignorar el primer argumento

Átomos vs variables

Universidad del Cauca

Característica	Átomo	Variable
Formato	Empieza con letra minúscula o está entre comillas simples.	Empieza con letra mayúscula o guion bajo.
Rol	Representa un valor fijo o constante.	Representa un valor indeterminado que puede cambiar.
Valor	Inmutable, siempre es el mismo.	Se puede unificar con diferentes valores durante la ejecución.
Ejemplos	carlos, 'Madrid', perro.	X, Persona, _Variable.
Uso	Para nombrar objetos concretos.	Para inferencias, consultas y reglas generales.

Conectivas lógicas

Universidad del Cauca

- Las conectivas que se utilizan en la Lógica de Primer Orden son:
  - ***conjunción, disyunción, negación* e *implicación***.

- La **conjunción (**“y”), se representa poniendo una coma entre los objetivos “**,**”.

- La **disyunción (** “o”) , tendrá éxito si se cumple alguno de los objetivos que la componen. Se utiliza un punto y coma “**;**” entre cada objetivo. Se recomienda representar la disyunción usando objetivos separados.

- Ej. gusta(X, fiesta) :- joven(X);colombiano(X).
  - Se puede representar:
    - gusta(X, fiesta) :- joven(X).
    - gusta(X, fiesta) :- colombiano(X).

Conectivas lógicas

Universidad del Cauca

- La **negación** lógica no puede ser representada explícitamente en Prolog, sino que se representa implícitamente por la falta de aserción : “no”, tendrá éxito si el objetivo *X* fracasa.
  - No es una verdadera negación, en el sentido de la Lógica, sino una negación “por fallo”. La representamos con el predicado predefinido ***not***

```
comida(pescado).
comida(pollo).
caducado(pescado).
comestible(A):-comida(A), not(caducado(A)).
```

No se niega el hecho en sí. Se niega su capacidad de ser probado.
Es una **negación por ignorancia**: si no lo sé, lo doy por falso.

```
persona(ana).
persona(carlos).
trabaja(ana).
no_trabaja(X) :- persona(X), not(trabaja(X)).
```

Reglas

Universidad del Cauca

- Cláusula de Horn con cabeza
- La **implicación** o **condicional**, sirve para representar que un hecho depende de un grupo de otros hechos.
  - Similar a "si ... entonces ...".
- En Prolog se usa el símbolo "**:-**". Se conoce como una **regla**.
- cabeza_de_la_regla :- cuerpo_de_la_regla.
  - La cabeza describe el hecho que se intenta definir; el cuerpo describe los objetivos que deben satisfacerse para que la cabeza sea cierta.
  - La cabeza es verdad si el cuerpo es verdad.
  - Por deducción la cabeza es falsa si el cuerpo es falso

```
puede_votar(X) :- mayor_de_edad(X), ciudadano(X).
```

(mayor_de_edad(X) y ciudadano(X))⇒puede_votar (X)

Si X es mayor de edad **y** X es ciudadano, **entonces** X puede votar.

Es un **condicional lógico** porque Prolog **no afirma directamente que puede_votar(X) sea cierto**, sino que **lo será solo si se cumplen las condiciones del cuerpo de la regla**.

Reglas

Universidad del Cauca

- Ej. humano(X) :- hombre(X);mujer(X).
- (hombre(X) ∨ mujer(X))⇒humano(X)
- En Prolog es mejor estilo separar disyunciones en reglas distintas para mantener claridad y trazabilidad:
  - humano(X) :- hombre(X).
  - humano(X) :- mujer(X).
- Un mismo nombre de variable representa el mismo objeto siempre que aparece en la regla. Así, cuando *X* se instancia a algún objeto, todas las *X* de dicha regla también se instancian a ese objeto.

Preguntas

- Al hacer una pregunta a un programa lógico queremos determinar si esa pregunta es ***consecuencia lógica*** del programa.
- Prolog considera que todo lo que hay en la Base de conocimientos es verdad, y lo que no, es falso.
- Se hace una búsqueda intentando encontrar hechos que ***coincidan*** con la pregunta. Dos hechos "coinciden" (se pueden unificar) si sus predicados son el mismo (se escriben de igual forma) y si cada uno de los respectivos argumentos son iguales entre sí.
- ***Si Prolog responde "true" es que ha podido demostrarlo, y si responde "false" es que no lo ha podido demostrar (no debe interpretarse como "falso" si no que con lo que Prolog conoce no puede demostrar su veracidad).***
- Ej.

```
1 ?- humano(clara).
true.

2 ?- humano(pluto).
false.

3 ?- perro(maria).
false.

4 ?- humano(X).
```

Un programa en Prolog

Universidad del Cauca

- Hechos y reglas:

```
mujer(clara).
mujer(maria).
hombre(cesar).
hombre(josé).
perro(pluto).
gato(garfield).
humano(X) :- hombre(X);mujer(X).
animal(X) :- perro(X);gato(X).
```

Preguntas:

```
1 ?- humano(clara).
true.

2 ?- humano(pluto).
false.

3 ?- perro(maria).
false.

4 ?- humano(X).
```

Comentarios

Universidad del Cauca

```
/*
 Programa de ejemplo en Prolog
*/

mujer(clara). %Clara es una mujer
mujer(maria).
hombre(cesar).
hombre(josé).
perro(pluto).
gato(garfield).
humano(X) :- hombre(X);mujer(X). % Esto es una regla
animal(X) :- perro(X);gato(X).
```

Operadores aritméticos y relacionales

Universidad del Cauca

is en Prolog

- is es un operador evaluador aritmético.
- Se utiliza para evaluar expresiones matemáticas y unificar el resultado con una variable.
- Variable is expresion.
- La expresión puede contener operaciones como +, -, *, /, mod, etc.
- Ejemplo:
- ?- Y is 3 + 4.
  Y = 7.
- Uso en reglas:
  - doble(X, Y) :- Y is X * 2.

  ?- doble(5, Y).
  Y = 10.

Operadores aritméticos y relacionales

Universidad del Cauca

Operador	Símbolo
Potencia	^ ó **
Producto	*
División	/
División entera	//
Resto división entera	mod
Suma	+
Signo positivo	+
Resta	-
Signo negativo	-
Igualdad	=
Distinto	\=
Menor que	<
Menor o igual que	=<
Mayor que	>
Mayor o igual que	>=
Evaluación aritmética	is

```
cuadrado(X,Y) :- Y is X*X.
```

```
1 ?- X is random(5).
X = 1.

2 ?- X is sqrt(9).
X = 3.0.
```

Operadores aritméticos y relacionales

Operador / Función	Descripción	Ejemplo de Uso	Resultado
+	Suma	X is 5 + 3.	X = 8
-	Resta	X is 5 - 3.	X = 2
*	Multiplicación	X is 5 * 3.	X = 15
/	División real	X is 5 / 2.	X = 2.5
//	División entera	X is 5 // 2.	X = 2
mod	Módulo (resto de división)	X is 5 mod 2.	X = 1
**	Exponenciación	X is 2 ** 3.	X = 8
^	Potencia (similar a **)	X is 2 ^ 3.	X = 8
abs/1	Valor absoluto	X is abs(-5).	X = 5
max/2	Máximo entre dos valores	X is max(5, 3).	X = 5
min/2	Mínimo entre dos valores	X is min(5, 3).	X = 3
sqrt/1	Raíz cuadrada	X is sqrt(9).	X = 3.0
round/1	Redondeo al entero más cercano	X is round(2.6).	X = 3
floor/1	Redondeo hacia abajo	X is floor(2.9).	X = 2
ceiling/1	Redondeo hacia arriba	X is ceiling(2.1).	X = 3
truncate/1	Truncamiento	X is truncate(2.9).	X = 2
sin/1	Seno	X is sin(0).	X = 0.0
cos/1	Coseno	X is cos(0).	X = 1.0
tan/1	Tangente	X is tan(0).	X = 0.0
exp/1	Exponencial (e^x)	X is exp(1).	X ≈ 2.718
log/1	Logaritmo natural (base e)	X is log(1).	X = 0.0

Universidad del Cauca

Ejemplo:

```
cuadrado(X,Y) :- Y is X*X.
```

```
1 ?- X is random(5).
X = 1.

2 ?- X is sqrt(9).
X = 3.0.
```

Ejemplos:

Universidad del Cauca

Escriba una regla en Prolog que calcule el área de un círculo dado su radio. La fórmula para calcular el área de un círculo es A = π * R^2, donde R es el radio.

```
area_circulo(Radio, Area) :-
        Area is 3.1416 * Radio * Radio.
```

```
?- area_circulo(5, Area).
```

Ejemplos:

Universidad del Cauca

Escriba una regla en Prolog que calcule el área de un triángulo dado su base y su altura

```
area_triangulo(Base, Altura, Area) :-
Area is 0.5 * Base * Altura.

area_triangulo(2,3,Area).
```

Ejercicio

Universidad del Cauca

Escriba una regla en Prolog que calcule el volumen de un cilindro dado su radio y su altura. La fórmula para calcular el volumen de un cilindro es:

$V = \pi \times R \times R \times H$

donde R es el radio y H es la altura.

```
volumen_cilindro(Radio, Altura, Volumen) :- Volumen is 3.14159 * Radio^2 * Altura.

volumen_cilindro(3, 7, Volumen).
```

Ejercicio

Escriba una regla en Prolog que calcule la hipotenusa de un triángulo rectángulo dados los valores de los catetos a y b.

```
hipotenusa(A, B, C) :- C is sqrt(A^2 + B^2).

hipotenusa(3, 4, C).
```

Ejercicios

Universidad del Cauca

- Tenga en cuenta la siguiente información sobre un restaurante:
  - Un restaurante tiene en su menú los siguientes platos:
    - Entradas: ensalada, queso, patacón
    - Carnes: milanesa, lomo viche
    - Pescado: tilapia frita, trucha
    - Postre: helado, fruta
  - Un plato principal sólo puede ser carne o pescado
  - Un Menú se puede componer de:
    - Una entrada, un plato principal y un postre
    - Una entrada y un plato principal
    - Un plato principal y un postre
- Implemente un programa en Prolog que permita contestar las siguientes preguntas:
  - ¿El queso es una entrada?
  - ¿El helado es una entrada?
  - ¿La tilapia frita es un pescado?
  - ¿La trucha es un plato principal?
  - ¿Se puede pedir un menú con ensalada como entrada, trucha como plato principal y fruta como postre?
  - ¿Se puede pedir un menú con ensalada como entrada y helado como plato principal?
  - ¿Cuáles son las entradas?
  - ¿Cuáles son los postres?
  - ¿Cuáles son los platos principales?

Ejercicios

```
entrada(ensalada).
entrada(queso).
entrada(patacon).
carne(milanesa).
carne(lomo_viche).
pescado(tilapia_frita).
pescado(trucha).
postre(helado).
postre(fruta).
%reglas
plato_principal(X) :- carne(X).
plato_principal(X) :- pescado(X).
menu(Entrada, Principal, Postre) :-
    entrada(Entrada),
    plato_principal(Principal),
    postre(Postre).
menu(Entrada, Principal) :-
    entrada(Entrada),
    plato_principal(Principal).
menu(Principal, Postre) :-
    plato_principal(Principal),
    postre(Postre).
```

entrada(queso).
entrada(helado).
pescado(tilapia_frita).
plato_principal(trucha).
menu(ensalada, trucha, fruta).
menu(ensalada, helado).
entrada(X).
postre(X).
plato_principal(X).

Ejercicios

Universidad del Cauca

- Escribir un programa Prolog que responda consultas acerca de cuáles son los rivales de una determinada selección en un campeonato mundial.
  - Una selección tiene como rivales todos los otros equipos de su mismo grupo (¡nunca contra sí misma!).
  - Incluir en el programa la siguiente información:
  - El grupo A está formado por Colombia, Camerún, Jamaica e Italia.
  - El grupo B está formado por Argentina, Nigeria, Japón y Escocia.
- El programa debe ser capaz de responder:
  - ¿Cuáles son los rivales de Argentina?
  - ¿Cuáles son los rivales de una selección X?
  - ¿Cuáles son los equipos del grupo A?
  - ¿Cuáles son los equipos de un grupo X?

Ejercicios

Universidad del Cauca

```
grupo(colombia, a).
grupo(camerun, a).
grupo(jamaica, a).
grupo(italia, a).
grupo(argentina, b).
grupo(nigeria, b).
grupo(japon, b).
grupo(escocia, b).
```

```
rival(X, Y) :-grupo(X, G),grupo(Y, G), X \= Y.
equipo_del_grupo(X, G) :- grupo(X, G).
```

rival(argentina, Rival).

rival(X, Rival).

equipo_en_grupo(a, Equipo).

equipo_en_grupo(X, Equipo).

Ejercicios

Universidad del Cauca

- La siguiente es la nómina de personal de una empresa:
  - Departamento de ventas: empleada María, aprendices Juan y Roque
  - Departamento de compras: empleada Nora, aprendiz Pedro
  - Departamento de administración: empleados Felipe y Hugo, aprendiz Ana.

Escribir un programa que responda las siguientes consultas:

- ¿Quiénes trabajan en un departamento X?
- Dadas dos personas A y B, ¿puede A darle órdenes a B?
  - Decimos que A puede darle órdenes a B si y sólo si trabajan en el mismo departamento y A tiene un cargo superior a B. Se considera que "empleado" es un cargo superior a "aprendiz".