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

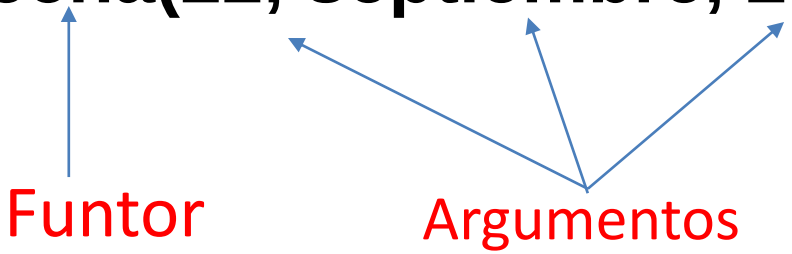
Prolog – Estructuras compuestas

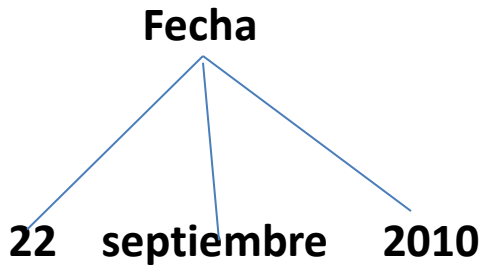
Prof: Pablo H. Ruiz
Unicauca

Funtor

Universidad del Cauca

- Un funtor es el nombre principal de una estructura compuesta en Prolog. Sirve para identificar el tipo de relación o entidad que se está representando.
- Es un término con un nombre (functor) y uno o más argumentos.
- Ejemplo: una estructura para representar la fecha. **fecha(22, septiembre, 2010)**





Sintaxis y uso básico de un Funtor

Universidad del Cauca

Forma general:

- **estructura(arg1, arg2, ..., argN).**

Ejemplo:

- **persona(carlos, 35, bogota).**
  ?- persona(Nombre, Edad, bogota).
- libro('1984', orwell).
- producto(lapiz, 1200, papeleria).
- evento(clase, lunes, 10).
- vehiculo(toyota, 2020, blanco).

-Agrupa datos relacionados.

-Permite definir reglas y consultas más organizadas.

-Ayuda a identificar el tipo de dato en estructuras más complejas.

Estructuras compuestas

Universidad del Cauca

Las estructuras compuestas permiten representar información más rica de forma organizada.

Ejemplo con hechos separados:

- nombre(ana).
- edad(ana, 20).
- ciudad(ana, cali).

Ejemplo usando una estructura:

- persona(ana, 20, cali).

Acceso a campos y definición de reglas

Universidad del Cauca

Reglas usando estructuras:

```
?-es_adulto(persona(ana, 20, cali)).
```

- es_adulto(persona(_, Edad, _)) :- Edad >= 18.
- La cabeza de la regla contiene un predicado estructurado:
  - persona(_, Edad, _).
- Esto significa que, antes de evaluar el cuerpo, Prolog debe unificar el término recibido en la consulta con la estructura de la cabeza.

Acceso a campos y definición de reglas

Universidad del Cauca

```
es_adulto(persona(_, Edad, _)) :- Edad >= 18.
vive_en(persona(_, _, Ciudad), Ciudad).
```

```
?- vive_en(persona(ana, 22, cali), X).
```

Consulta compuesta:

```
persona(ana, 20, cali).
persona(luis, 17, bogota).
persona(carlos, 25, medellin).
```

- ?- persona(N, E, C), es_adulto(persona(N, E, C)).

Comparación entre estructuras

Universidad del Cauca

En Prolog, dos estructuras se consideran iguales si:

- Tienen el mismo funtor (nombre)
- Tienen el mismo número de argumentos (aridad)
- Todos los argumentos coinciden (se unifican)

```
persona(ana, 20, cali).
persona(luis, 17, bogota).
persona(carlos, 25, medellin).
```

```
?- persona(ana, 20, cali) = persona(ana, 20, cali).
?- persona(ana, 20, cali) = persona(ana, 21, cali).

?- persona(X, 20, C) = persona(ana, 20, cali).
X = ana,
C = cali.
```

Comparación parcial entre estructuras

Universidad del Cauca

- Una comparación parcial ocurre cuando se compara una estructura que contiene variables con otra que contiene valores concretos.
- Prolog intenta unificar ambas estructuras, es decir, hacer que coincidan sus componentes asignando valores a las variables cuando sea posible.
- Unificación puntual

?- persona(N, 20, cali) = persona(ana, 20, cali).

```
persona(ana, 20, cali).
persona(luis, 17, bogota).
persona(carlos, 25, medellin).
```

Prolog asigna a N el valor ana, porque eso hace falta para que ambas estructuras sean **idénticas**.

Comparación parcial entre estructuras

Universidad del Cauca

Utilidad de la comparación parcial

- Extraer información de las estructuras, es decir, cuanto se tiene una estructura con valores conocidos y otros que no conoces (variables), en estos casos se puede usar la comparación parcial para obtener los valores faltantes.

- El uso de _ como variable anónima permite ignorar campos que no te interesan al comparar estructuras.

?- persona(N, _, cali) = persona(ana, 20, cali).

Uso de listas de estructuras en Prolog

Universidad del Cauca

- En Prolog, una lista de estructuras es una colección de términos compuestos, escritos entre corchetes [...].
- Son útiles para representar conjuntos de datos complejos, como personas, vehículos, productos, etc.

[persona(ana, 20, cali), persona(luis, 17, bogota), persona(carlos, 25, cali)]

Para que sirven:

- Para recorrer y analizar múltiples datos organizados.
- Para filtrar elementos que cumplen ciertas condiciones.
- Para representar bases de conocimiento pequeñas en una sola lista.
- Para usar con predicados como member, maplist, findall

Uso de listas de estructuras en Prolog

Universidad del Cauca

Recorrer y filtrar listas con member

Verifica si un elemento pertenece a una lista o permite recorrer los elementos uno por uno mediante retroceso (;).

```
?- member(3, [1, 2, 3, 4]).
true.
```

```
Lista = [persona(ana, 20, cali), persona(luis, 17, bogota)],
member(persona(N, _, cali), Lista).
```

Uso de listas de estructuras en Prolog

Universidad del Cauca

Aplicar una operación a todos los elementos con maplist

- Aplica un **predicado a cada elemento** de una lista. Es como un **for-each** en otros lenguajes.

```
doble(X) :- Y is X * 2, write(Y), nl.

?- maplist(doble, [1, 2, 3]).
```

Uso de listas de estructuras en Prolog

Universidad del Cauca

Aplicar una operación a todos los elementos con maplist

```
mostrar(persona(N, E, C)) :-
    write(N), write(' tiene '), write(E),
    write(' años y vive en '), write(C), nl.

?- maplist(mostrar, [persona(ana, 20, cali), persona(carlos, 25, cali)]).
```

Uso de listas de estructuras en Prolog

Universidad del Cauca

Genera una lista con todos los valores que cumplen una condición con findall

findall(ElementoQueQuiero, CondiciónQueDebeCumplir, ListaResultado).

findall(X, Condición, Lista).

findall(Y, (member(X, [1,2,3]), Y is X*X), Cuadrados).

Cuadrados = [1, 4, 9].

1. recorre la lista [1,2,3]
2. findall recolecta todos los valores de Y que resultan de esas operaciones y los guarda en una lista.

Uso de listas de estructuras en Prolog

Universidad del Cauca

Genera una lista con todos los valores que cumplen una condición con findall

findall(X, Condición, Lista).

Lista = [persona(ana, 20, cali), persona(luis, 17, bogota)],
findall(N, (member(persona(N,E,_), Lista), E >= 18), Adultos).

Uso de listas de estructuras en Prolog

Universidad del Cauca

- Investigar como se implementaría un árbol binario en prolog con sus respetivos recorridos preorden, inorden, posorden.
- Investigar como se puede crear un árbol binario desde una lista