Recursión, relaciones

Objetivos

- Introducir definiciones recursivas en Prolog
- Prolog es relacional
- Es Prolog completamente declarativo? Algunos problemas

Definiciones recursivas

- Los predicados en Prolog pueden ser definidos recursivamente.

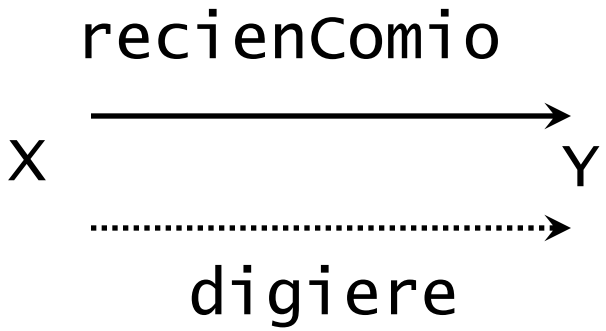
- Una definición es recursiva si el objeto definido contiene referencias a si mismo.

Ejemplo 1

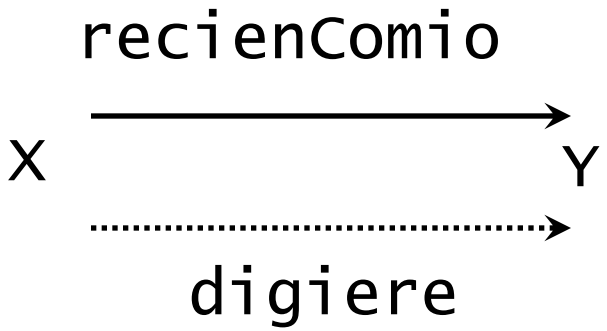
```
digiere(X,Y):- recienComio(X,Y).
digiere(X,Y):- recienComio(X,Z), digiere(Z,Y).

recienComio(mosquito,sangre(juan)).
recienComio(rana,mosquito).
recienComio(gaviota,rana).
```

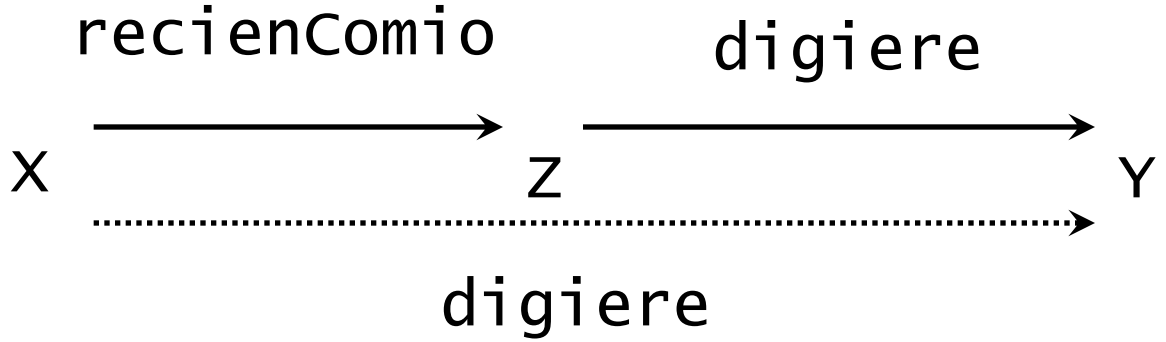
```
?-
```

Situación



Situación





Ejemplo 1

```
digiere(X,Y):- recienComio(X,Y).
digiere(X,Y):- recienComio(X,Z), digiere(Z,Y).

recienComio(mosquito,sangre(juan)).
recienComio(rana,mosquito).
recienComio(gaviota,rana).
```

```
?- digiere(gaviota,mosquito).
```

Otra definición recursiva

```
p:- p.
```

```
?-
```

Otra definición recursiva


```
p:- p.
```

```
?- p.
```

Otra definición recursiva


```
p:- p.
```

```
?- p.
ERROR: out of memory
```

Ejemplo 2: Descendiente

```
hijo(juan,pedro).
hijo(pedro,ana).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,Y).
```

Ejemplo 2: Descendiente

```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,Y).
```

Ejemplo 2: Descendiente


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,Y).
```

```
?- descendiente(juan,laura).
no
?-
```

Ejemplo 2: Descendiente


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,U), hijo(U,Y).
```

```
?-
```

Ejemplo 2: Descendiente


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,Y).
descendiente(X,Y):- hijo(X,Z), hijo(Z,U), hijo(U,Y).
```

```
?-
```

Ejemplo 2: Descendiente


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), descendiente(Z,Y).
```

```
?- descendiente(juan,jose).
```

Arbol de búsqueda

Arbol de búsqueda para

?- descendiente(juan,laura).

Ejemplo 3: naturales en unario

Definición

1. **0** es un natural.
2. Si **X** es un natural, s(**X**) es un natural.

Ejemplo 3: naturales en unario

Definición

1. **0** es un natural.
2. Si **X** es un natural, s(**X**) es un natural.
3. Estos son todos los naturales.

(3, clausura, en general se sobreentiende en las definiciones inductivas)

Ejemplo 3: naturales en unario

```
natural(0).
natural(s(X)):- natural(X).
```

Ejemplo 3: naturales en unario

```
numeral(0).
numeral(succ(X)):- numeral(X).
```

```
?- natural(s(s(s(0)))).
yes
?-
```

Ejemplo 3: naturales en unario

```
natural(0).
natural(s(X)):- natural(X).
```

```
?- natural(X).
```

Ejemplo 3: naturales en unario

```
natural(0).
natural(s(X)):- natural(X).
```

```
?- natural(X).
X=0;
X=s(0);
X=s(s(0));
X=s(s(s(0)));
X=s(s(s(s(0))))
```

Ejemplo 4: suma

?- suma(s(s(0)),s(s(0)),Res). (Res=2+2)

Res=(s(s(s(s(0)))))

yes

Ejemplo 4: suma

?- suma(s(s(0)),s(s(0)),Res). (Res=2+2)

Res=(s(s(s(s(0)))))

yes

QUEREMOS UNA SUMA DE 3 ARGUMENTOS !!!

Ejemplo 4: suma

```
suma(0,X,X).                    % cláusula base
```

```
?- suma(s(s(0)),s(s(0)),Res).     (Res=2+2)
Res=(s(s(s(s(0)))))
yes
```

Ejemplo 4: suma

```
suma(0,X,X).                          % cláusula base

suma(s(X),Y,s(Z)):-                   % cláusula recursiva
    suma(X,Y,Z).
```

```
?- suma(s(s(0)),s(s(0)),Res).
Res=(s(s(s(s(0)))))
yes
```

Ventajas de una definición relacional

Definir la resta, usando suma.

Definir el producto.

¿División entera?

Prolog y Lógica

Prolog fue el primer intento razonable de lenguaje de programación lógico

- **Especificación del problema** usando el lenguaje de la **lógica** :
  - programa Prolog = conjunto finito de cláusulas
- El programador **no indica cómo** resolver el problema
- Para obtener información, se hacen **consultas** a la base de conocimientos definida (el conjunto finito de cláusulas, o sea, el programa).

Prolog y Lógica

- Prolog realiza avances en el sentido de ser un lenguaje de especificación, pero no es perfecto:

- Prolog tiene un modo específico de:
  - Seleccionar una cláusula en la BC
    - Secuencial, desde el comienzo
  - Seleccionar un predicado en la consulta
    - De izquierda a derecha
  - Realizar backtracking cuando llega a un punto de falla
    - Recorrida en profundidad del árbol de búsqueda

descendiente1.pl


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente(X,Y):- hijo(X,Z), descendiente(Z,Y).
```

```
?- descendiente(X,Y).
X=juan
Y=pedro
```

descendiente2.pl


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente (X,Y):- hijo(X,Z), descendiente(Z,Y).
descendiente(X,Y):- hijo(X,Y).
```

```
?- descendiente(X,Y).
X=juan
Y=jose
```

descendiente3.pl


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente (X,Y):- descendiente(Z,Y), hijo(X,Z).
descendiente(X,Y):- hijo(X,Y).
```

```
?- descendiente(X,Y).
ERROR: OUT OF LOCAL STACK
```

descendiente4.pl


```
hijo(juan,pedro).
hijo(pedro,ana).
hijo(ana,laura).
hijo(laura,jose).

descendiente(X,Y):- hijo(X,Y).
descendiente (X,Y):- descendiente(Z,Y), hijo(X,Z).
```

```
?- descendiente(X,Y).
```

Resumen

- Predicados recursivos (natural, suma, descendiente, etc.)
- Algunas discrepancias entre significado declarativo y procedural de Prolog:
  - Importa el orden de las cláusulas en un programa
  - Importa el orden de los predicados en una consulta o en el cuerpo de una regla
  - La unificación de Prolog es *defectuosa*

Próxima

Introducir **listas** en Prolog

- Estructura de datos recursiva muy importante
- Predicados sobre listas