

Programación Declarativa
Clase Práctica 6
Curso 2018-2019

Tema: Metaprogramación y bases de datos lógicas relacionales.

1. Se tiene un conjunto de seis personas (Pepe, Juan, Rafael, Ana, Concha, Eva) a las cuales debemos repartir los siguientes regalos (coche, casa, viaje, barco). Se conoce que existen personas incompatibles que no aceptan tener el mismo regalo. Las incompatibilidades son las siguientes:
 - Pepe con el resto de las personas.
 - Juan con Rafael, Ana y Concha
 - Rafael con Concha y Eva
 - Ana con Concha
 - Concha con Eva
 - a. Definir un programa Prolog que permita obtener asignaciones de regalos posibles.
 - b. Formule un objetivo que determine cuantas soluciones factibles existen regalándole un viaje a Ana.
 - c. Formule un objetivo que permita hallar una repartición y a partir de ella determinar las personas que recibirán un coche como regalo.
 - d. Formule un objetivo que permita indicarle al usuario cuantos regalos de cada tipo debe comprar.
2. Se cuenta con una teoría en Prolog que se refiere a una base de datos de una biblioteca circulante. En dicha teoría se definieron los siguientes predicados:
 - libro/2, hecho que establece cuantos ejemplares de un libro tiene la biblioteca:
libro(libro_100, 8).
libro(libro_25, 2).
libro(libro_59, 0).
 - libro_prestado/2, hecho que establecen a qué persona se le ha prestado qué libro:
libro_prestado('Ana', libro_100).
libro_prestado('Pedro', libro_25).
 - libro_en_consulta/2, hecho que establecen a qué persona se le ha dado un libro para que lo consulte en la biblioteca:
libro_en_consulta('Miguel', libro_100).

libro_en_consulta('Miguel', libro_25).

- fotocopiando/1, hecho que establecen que la fotocopidora está siendo utilizada para fotocopiar un libro dado.

fotocopiando(libro_59).

A los usuarios de esta biblioteca circulante se les prestan facilidades que se dividen en básicas y adicionales. Son facilidades básicas:

- la referencia (revisar el catálogo, solicitar información de documentos sobre un tema)
- la consulta (consultar documentos, libros, etc. en la propia biblioteca)

Son facilidades adicionales:

- el préstamo (de libros y documentos)
- las fotocopias (solicitarlas).

Defina los siguientes predicados para el programa:

- info_prestamos(LP): que da en LP la lista de todos los libros que están prestados.
- info_consulta(LC): que da en LC la lista de todos los libros que están siendo consultados.
- info_libros(CantidadMinima, [Libro:Ejemplares|Resto]): que da una lista cuyos elementos tienen el formato Libro:Ejemplares para dar a conocer la información de todos los libros que tiene la biblioteca cuya cantidad sea mayor que CantidadMinima.
- facilidad(U, F): determina la facilidad de un usuario U dando en la variable F el tipo de facilidad (básica o adicional) seguido de dos puntos más el nombre específico de la facilidad que tenga (referencia, consulta, préstamo, fotocopias). Vea que si se piden más respuestas a este predicado (;) se obtienen poco a poco todas las facilidades de U. Tenga en cuenta los siguientes axiomas:
 - Un usuario siempre puede solicitar la facilidad básica de la referencia.
 - Un usuario puede consultar si a lo sumo está consultando un libro, es decir, si está consultando dos libros, no puede pedir consultar un tercero.
 - Un usuario tiene la facilidad de préstamo si no tiene un libro prestado.
 - Un usuario puede fotocopiar si no hay nadie utilizando la fotocopidora.

Por ejemplo, una posible respuesta para los fragmentos del programa dados es:

:-facilidad('Ana', X).

X = básica:referencia;

X = básica:consulta;

false.

:-facilidad('Miguel', básica:consulta).

false.

- e. prestar(U, L): presta al usuario U el libro L y actualiza la información de la base de conocimientos de la biblioteca. Tenga en cuenta que a un usuario se le puede prestar un libro si tiene la facilidad de préstamo (no tiene otro libro en préstamo), si el libro existe en la biblioteca y si la cantidad de este libro es mayor que cero.
- f. consultar(U, L): permite al usuario U consultar el libro L y actualiza la información de la base de conocimientos de la biblioteca. Tenga en cuenta que U debe tener la facilidad de consultar y que el libro que quiere consultar está disponible.
- g. fotocopiar(U, L): permite al usuario U solicitar la fotocopia del libro L y actualiza la información de la base de conocimientos de la biblioteca, si en esos momentos se puede utilizar la fotocopidora y el libro está en la biblioteca.
- h. Declare además las operaciones inversas: devolver(U, L), terminar_consulta(U, L), fotocopia_terminada(L).

3. Se supone definida una base de datos de familias, con una sola relación que tiene la siguiente estructura:

familia(persona(Nombre, Apellido1, Apellido2, fecha(Dia, Mes, Ano), trabaja(Compania, Salario)), persona(Nombre, Apellido1, Apellido2, fecha(Dia, Mes, Ano), trabaja(Compania, Salario)), [persona(Nombre, Apellido1, Apellido2, fecha(Dia, Mes, Ano), desempleada) | Lista_otros_hijos]).

Donde la primera persona corresponde al marido, la segunda a la mujer y la lista de personas del tercer argumento de "familia" corresponde a los hijos de la pareja. *Ejemplo:*

familia (persona(gustavo, gimenez, marques, fecha(25,9,66), trabaja(drt, 3),
 persona(maria, martinez, gil, fecha(15,3,65), trabaja(tte, 3),
 [persona(ramon, gimenez, martinez, fecha(3,5,93), desempleada)]).

Se pide escribir objetivos en PROLOG que permitan obtener:

- a) Todos los datos del marido de la familia.
- b) Todos los datos de la mujer de la familia.
- c) La lista con todos los hijos de la familia.
- e) Todos los datos del hijo n-ésimo de la familia.
- f) Todos los datos de todas las personas de la base de datos.
- g) La fecha de nacimiento de las personas.
- h) El salario de las personas.

- i) Nombre y apellidos de las mujeres que trabajan.
 - j) Nombre y apellidos de las personas desempleadas que nacieron antes de 1978.
 - k) Nombre y apellidos de las personas que nacieron antes de 1961 que cobren menos de 3
 - l) Primer apellido del padre y primer apellido de la madre de las familias que tienen al menos 3 hijos.
 - m) Calcular el total de ingresos por familia.
 - n) Calcular los ingresos por miembro de familia en cada familia.
 - o) Calcular la media de ingresos de todas las familias.
 - p) Nombre y apellidos de las personas de familias sin hijos.
 - q) Nombre y apellidos de las personas de familias cuyos hijos no tengan empleo.
 - r) Nombre y apellidos de las personas de familias cuyo marido está desempleado y la mujer trabaje.
 - s) Nombre y apellidos de los hijos cuyos padres se diferencian en edad más de 15 años.
 - t) Definir la relación gemelos(Hijo1, Hijo2).
4. Defina el predicado `cumplen(L,C,Cumplidores)` que dada una lista `L` y un predicado unario `C` obtenga la lista de todos los elementos de `L` que cumplen la condición `C`.
 5. Programe el predicado `pares(L,P,Pares)` que dada una lista de elementos `L` y el nombre de un predicado binario `P`, obtenga la listas de pares de la forma `(E,R)` donde `E` es un elemento de `L` y `R` es el valor que une con el segundo argumento del predicado `P` al evaluar `P(E,R)`.
 6. Defina el metapredicado `=../2` a partir de `functor/3` y `arg/3`. Luego de `_na functor/3` y `arg/3` a partir de `=../2`.
 7. Defina un predicado que dado un objetivo `O` devuelva la lista de todas las soluciones que este posee, respondiendo al siguiente formato estructural: `soluciones(O,[sol(O)|Resto])`.
 8. Declare un predicado `soluciones_distintas(O,N)` que dado un objetivo `O` determine en `N` la cantidad de soluciones distintas que posee.
 9. Defina un predicado `listaNPrimeros(L,N,LN)` que dada una lista `L` y un número natural `N` permita obtener otra lista que contenga los `N` primeros elementos de `L`.
 10. Defina un predicado `nDesdeP(L,N,P,R)` que dada una lista `L` y dos números naturales `N` y `P`, brinde la lista que contenga los `N` primeros elementos de `L` contados a partir de la posición `P`.
 11. Programe `findall/3` utilizando los metapredicados `assert/1` y `retract/1`.