

PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES

Grado en Ingeniería Informática / Grado en Matemáticas e Informática

17 de Junio de 2015

Examen Final Ordinario

Nombre:

Matrícula:

INSTRUCCIONES: El examen consta de 3 ejercicios. Todos los ejercicios deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse a los ejercicios tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

DURACIÓN DEL EXAMEN: 90 minutos

EJERCICIO 1 (3,5 puntos – 30 minutos)

Se pide al alumno que escriba un **programa lógico puro** para realizar el recorrido en anchura de un árbol binario, codificando los siguientes predicados, así como cualquier otro que sea necesario.

El predicado principal *anchura(A,L)* es cierto si y solo si *L* es la lista de los nodos del árbol *A* en el orden en que se visitan en un recorrido en anchura (de izquierda a derecha). El predicado auxiliar *hijos(As,Hs,Ns,L)* es cierto si y solo si *Hs* es la lista de los hijos de los árboles de la lista *As* y el par *Ns-L* es la lista diferencia de los nodos de dichos árboles (los de *As*). Se debe utilizar la representación de listas habitual en Prolog y explicar la que se utilice para los árboles.

SOLUCION:

```
anchura(A,L):- niveles([A],L).
```

```
niveles([],[]).
```

```
niveles([A|As],L):-  
    hijos([A|As],Hs,L,LT),  
    niveles(Hs,LT).
```

```
hijos([],[],L,L).
```

```
hijos([A|As],Hs,L,T):-  
    no_vacio(A,Hs,HsT,L,LT),  
    hijos(As,HsT,LT,T).
```

```
no_vacio(void,Hs,Hs,L,L).
```

```
no_vacio(tree(N,I,D),[I,D|Hs],Hs,[N|L],L).
```

La representación de árboles es la habitual: void para el árbol vacío y tree(N,I,D) para un árbol no vacío con nodo N, hijo izquierdo I e hijo derecho D.

PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES
Grado en Ingeniería Informática / Grado en Matemáticas e Informática

17 de Junio de 2015

Examen Final Ordinario

Nombre:

Matrícula:

INSTRUCCIONES: El examen consta de 3 ejercicios. Todos los ejercicios deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse a los ejercicios tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

DURACIÓN DEL EXAMEN: 90 minutos

EJERCICIO 2 (3,5 puntos – 30 minutos)

Las ciudades en un determinado país están unidas por carreteras de un único sentido. Dichas carreteras se representan con un functor carretera/4, cuyos argumentos se refieren respectivamente a ciudad origen, ciudad destino, tipo de carretera (autopista, autovía, carretera nacional, carretera comarcal, carretera local), y kilómetros (entre el origen y el destino).

Se pide al alumno que escriba un **programa lógico en Prolog** que incluya al menos los siguientes predicados:

- recorrido/3 (recorrido(O,D,C) que se verifique si C es un camino de la ciudad origen (O) a la ciudad destino (D). Dicho camino debe representarse con un functor camino/2 (camino(R,Km), donde R es la lista de ciudades del recorrido y Km el número de kilómetros totales del recorrido).
- vias/2 (vias(I,Ts)) que se verifique si Ts es la lista (sin redundancias) de los tipos de vía del itinerario I. Los itinerarios se representan con un functor del que no se conoce el nombre cuyos argumentos son respectivamente una lista con las ciudades del itinerario, una lista con todos los tipos de vía del itinerario, y un tercer argumento para representar los kilómetros totales del itinerario.
- filtrarCarreteras/3 (filtrarCarreteras(T,K,Cs)) que se verifique si Cs es el listado de carreteras (de toda la red del país) del tipo T y cuyos kilómetros (entre el origen y el destino) son menores que K.

SOLUCION:

%%% Base de Hechos (ejemplo) %%%

```
carretera(a,c,autopista,10).
carretera(b,a,autovia,5).
carretera(b,c,cNacional,35).
carretera(d,b,cComarcal,55).
carretera(d,e,cLocal,45).
carretera(c,e,autovia,12).
carretera(c,d,autovia,24).
carretera(e,b,autovia,15).
```

```
redC([carretera(a,c,autopista,10), carretera(b,a,autovia,5), carretera(b,c,cNacional,35),
      carretera(d,b,cComarcal,55), carretera(d,e,cLocal,45), carretera(c,e,autovia,12),
      carretera(c,d,autovia,24), carretera(e,b,autovia,15)]).
```

%% recorrido/3 (recorrido(O,D,C) %%

```
recorrido(O,D,camino(R,Km)):-
    recorridoAux(O,[D],R,0,Km).
```

```
recorridoAux(O,[O|R],[O|R],K,K).
recorridoAux(O,[Y|R1],R,K1,Km):-
    carretera(C,Y,T,K),
    Kms is K1+K,
    \+(member(C,[Y|R1])),
    recorridoAux(O,[C,Y|R1],R,Kms,Km).
```

%% vias/2 (vias(I,Ts) %%

```
vias(I,T):-
    I=..[_,_Ns,Ts,_Kms],
    setof(X,member(X,Ts),T).
```

%% filtrarCarreteras/3 (filtrarCarreteras(T,K,Cs) %%

```
filtrarCarreteras(Tipo,Kms,Cs):-
    redC(RedC),
    findall(C,(member(C,RedC),arg(3,C,T),Tipo==T,arg(4,C,D),Kms>D),Cs).
```

PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES
Grado en Ingeniería Informática / Grado en Matemáticas e Informática

17 de Junio de 2015

Examen Final Ordinario

Nombre:

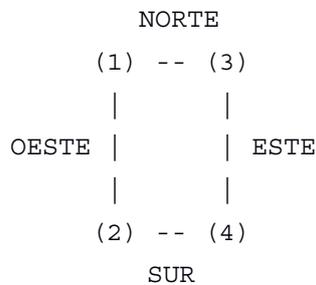
Matrícula:

INSTRUCCIONES: El examen consta de 3 ejercicios. Todos los ejercicios deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse a los ejercicios tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

DURACIÓN DEL EXAMEN: 90 minutos

EJERCICIO 3 (3 puntos – 30 minutos)

Un pequeño archipiélago del Atlántico Sur está compuesto por 4 islas (denominadas Heru, Tamarón, Techaide y Bayou) conectadas por puentes como puede verse en el croquis que sigue a continuación.



Como puede verse, en el croquis no aparece reflejada la localización de las islas, sino un número que hace referencia cada una de ellas (1, 2, 3 y 4). Se sabe que cada una de las islas exporta al exterior un producto diferente (piñas, cocos, plátanos y frutas de la pasión), y por otro lado, ofrece a los turistas un único tipo de paisaje (volcánico, selva, sabana y desierto).

Se sabe además lo siguiente sobre la ubicación de las diferentes islas:

- La isla desértica está al sur de Heru.
- La isla que exporta piñas está al oeste de Tamarón.
- La isla volcánica está al este de la isla que exporta frutas de la pasión.
- Bayou y la isla con paisaje sabánico están conectadas por un puente vertical (norte-sur).
- Techaide y la isla que exporta cocos están conectadas por un puente horizontal (este-oeste).
- Las islas con paisajes selvático y sabánico no tienen conexión directa mediante un puente.

Se pide al alumno que escriba un **programa lógico con restricciones en dominios finitos** utilizando la sintaxis de **Ciao Prolog** que permita asociar cada isla anónima del boceto (representadas por los números 1, 2, 3 y 4 en dicho boceto) con su nombre real, su paisaje autóctono y el producto que exporta al exterior. **Explicar claramente la semántica de los predicados definidos para resolver el problema e identificar claramente las variables de entrada y salida y su significado.**

PISTA: Cada isla tiene tres características únicas (nombre, producto que exporta al exterior y tipo de paisaje). Se recomienda al alumno que represente mediante variables diferentes cada valor posible para cada una de estas características, y formule restricciones que asignen a todas estas variables valores enteros en el dominio [1,4] que representan a cada una de las islas. De esta manera, los tríos de variables (características) que poseen idénticos valores hacen referencia a la misma isla.

SOLUCIÓN:

Siguiendo la estrategia recomendada en el enunciado, se definen 12 variables (una por cada valor distinto de cada característica). Las variables se separan entonces en 3 grupos de 4 variables (un grupo por cada característica). Las variables pertenecientes a cada grupo se restringen de tal manera que pueden tomar un valor único —es decir, ninguna otra variable del mismo grupo puede tomar el mismo valor— del dominio [1,2,3,4]. De esta manera, tres variables pertenecientes a grupos distintos que toman el mismo valor en el dominio [1,2,3,4] representan características de la misma isla.

```
use_package(fd).
```

```
solve_islands([Heru, Tamaron, Techai de, Bayou], [Ananas, Coconuts, Bananas, Passi onFrui ts], [Vol cani c, Sel vati c, Savannah, Deserti c]) :-
```

```
    [Heru, Tamaron, Techai de, Bayou] in 1..4,
    all_different([Heru, Tamaron, Techai de, Bayou]),
    [Ananas, Coconuts, Bananas, Passi onFrui ts] in 1..4,
    all_different([Ananas, Coconuts, Bananas, Passi onFrui ts]),
    [Vol cani c, Sel vati c, Savannah, Deserti c] in 1..4,
    all_different([Vol cani c, Sel vati c, Savannah, Deserti c]),
    Deserti c .=. Heru + 1,
    Ananas .=. Tamaron - 2,
    Vol cani c .=. Passi onFrui ts + 2,
    (
        (Bayou .=. Savannah - 1);
        (Bayou - 1 .=. Savannah)
    ),
    (
        (Techai de .=. Coconuts + 2);
        (Techai de + 2 .=. Coconuts)
    ),
    Sel vati c + Savannah .=. 5,
    labeling([Heru, Tamaron, Techai de, Bayou]),
    labeling([Ananas, Coconuts, Bananas, Passi onFrui ts]),
    labeling([Vol cani c, Sel vati c, Savannah, Deserti c]).
```

Si se ejecuta este programa, se obtiene una única solución al problema planteado:

```
?- solve_islands(A, B, C).
```

```
A = [1, 4, 2, 3],
B = [2, 4, 3, 1],
C = [3, 1, 4, 2] ? ;
```

no

De acuerdo a la cabecera del predicado `solve_islands/3`, el resultado anterior se traduce en lo siguiente:

- La isla 1 es Heru, una isla selvática que exporta frutas de la pasión.
- La isla 2 es Techaide, una isla desértica que exporta piñas.
- La isla 3 es Bayou, una isla volcánica que exporta plátanos.
- La isla 4 es Tamaron, una isla con paisaje sabánico que exporta cocos.