Labos LI: Práctica 5. Prolog avanzado Primavera 2019

Problema(s) A:

A.1. "Hacer Aguas": disponemos de un grifo de agua, un cubo de 5 litros y otro de 8 litros. Se puede verter el contenido de un cubo en otro, llenar un cubo, o vaciar un cubo del todo, y queremos saber la secuencia mínima de operaciones para obtener exactamente 4 litros de agua en el cubo de 8 litros.

A.2. "Misioneros": buscamos la manera más rápida para tres misioneros y tres caníbales de cruzar un río, disponiendo de una canoa que puede ser utilizada por 1 o 2 personas (misioneros o caníbales), pero siempre evitando que haya misioneros en minoría en cualquier orilla (por razones obvias).

Usa (es obligatorio) el siguiente esquema Prolog para resolver los dos problemas:

```
camino( E,E, C,C ).
camino( EstadoActual, EstadoFinal, CaminoHastaAhora, CaminoTotal ):-
  unPaso( EstadoActual, EstSiguiente ),
  \+member(EstSiguiente,CaminoHastaAhora),
  camino( EstSiguiente, EstadoFinal, [EstSiguiente|CaminoHastaAhora], CaminoTotal ).

solucionOptima:-
  nat(N),                          % Buscamos solución de "coste" 0; si no, de 1, etc.
  camino([0,0],[0,4],[[0,0]],C), % En "hacer aguas": -un estado es [cubo5,cubo8], y
  length(C,N),                     % -el coste es la longitud de C.
  write(C).
```

Problema B: Un programa escrito en el lenguaje *sumbol* tiene la siguiente sintaxis:

```
<programa>   -->    begin  <instrucciones>  end
<instrucciones>   -->    <instruccion>
<instrucciones>   -->    <instruccion> ; <instrucciones>
<instruccion>   -->    <variable> = <variable> + <variable>
<instruccion>   -->    if <variable> = <variable> then <instrucciones> else <instrucciones>  endif
<variable>-->    x
<variable>-->    y
<variable>-->    z
```

Tres ejemplos de programas *sumbol*:

```
  begin x=x+z end
  begin x=x+y; z=z+z; x=y+x end
  begin x=y+z; if z=z then x=x+z; y=y+z else z=z+y endif; x=x+z end
```

Escribe en Prolog un sencillo analizador sintáctico para el lenguaje *sumbol*, es decir, una cláusula programa(P) que se satisface si la lista de átomos P contiene un programa *sumbol* sintácticamente correcto, y que falla en caso contrario. Para ello (es obligatorio), haz corresponder una cláusula a cada regla de la gramática. Ejemplos:

```
?- programa( [begin, z, =, x, +, y, end] ).
yes
?- programa( [begin, z, =, x, +, y, ;, x, =, z, z, end] ).  % aqui falta un "+"
no
```

Problema C: File `tsp.pl` contains a branch-and-bound solution to a Traveling-Salesman-like problem. Adapt `helicopter.pl` using the same idea to solve the following problem. We have a helicopter that can carry its pilot plus at most two passengers at the same time. We are given a list of passengers, each with his/her origin and destination, given as coordinates in the plane (a pair of integers in 0..1000). The aim is to find the shortest route for the helicopter to start from its base, handle all passengers, and return to its base.