Labos LI: Pràctica 5. Prolog avançat Tardor 2024

Problema A: Adapta el problema de "xifres" de la pàgina 15 del tema **p6** a la *Col·lecció d'apunts bàsics de lògica* perquè:

- també faci servir la divisió (sense donar mai divisió per zero),
- trobi les solucions més curtes primer (les que usen menys elements de la llista L),
- i a més, comenci intentant trobar la solució exacta, i a continuació les que es trobin a distàncies més grans.

Per exemple:

```
?- main([2,3,4,6],31).
% probem valor :31: a distancia :0       %% <<<< no hi ha cap expressió amb valor exacte
% probem valor :31: a distancia :1       %% <<<< hi ha 88 solucions a distància +/-1
(2+3)*6 = 31-1 = 30                      %%      primer amb 3 números
...
2+3*(4+6) = 31+1 = 32                    %%      després amb 4 numeros
...
% probem valor :31: a distancia :2       %% <<<< hi ha 32 solucions a distància +/-2
2+(3+4*6) = 31-2 = 29                    %%      amb 4 numeros
...
% probem valor :31: a distancia :3       %% <<<< hi ha 48 solucions a distància +/-3
(2+3)*6+4 = 31+3 = 34                    %%      amb 4 numeros
...
```

Problema B: Adapta (és obligatori) l'esquema Prolog de baix per a resoldre els tres problemes `B1`, `B2`, `B3`. L'única cosa que cal fer en cadascun d'ells és; a) definir què és un estat, i b) quins passos hi ha per a passar d'un estat al següent.

B.1. Busquem la manera més ràpida per a tres missioners i tres caníbals de travessar un riu, disposant d'una canoa que pot ser utilitzada per 1 o 2 persones (missioners o caníbals), però sempre evitant que els missioners quedin en minoria en qualsevol riba (per raons òbvies).

B.2. Donat un natural $n > 0$, una posició inicial $(Fila_I, Columna_I)$, una posició final $(Fila_F, Columna_F)$, i un nombre de passos P, trobar un camí de $(Fila_I, Columna_I)$ a $(Fila_F, Columna_F)$, en un tauler d'escacs de $n \times n$ en exactament P passos de cavall, i passant per caselles diferents. El programa ha de fallar si per a la n en qüestió no existeix tal camí.

B.3. Tracta d'esbrinar la manera més ràpida que tenen quatre persones P_1, P_2, P_5 i P_8 per a creuar de nit un pont que només aguanta el pes de dos, on tenen una única i imprescindible llanterna i cada P_i triga i minuts a creuar. Dos junts triguen com el més lent dels dos.

```
main :- EstatInicial = ...,    EstatFinal = ...,
        between(0, 1000, CostMax),          % Busquem solució de cost 0; si no, de 1, etc.
        cami(CostMax, EstatInicial, EstatFinal, [EstatInicial], Cami),
        reverse(Cami, Cami1), write(Cami1), write(' amb cost '), write(CostMax), nl, halt.

cami(0, E, E, C, C).                        % Cas base: quan l'estat actual és l'estat final.
cami(CostMax, EstatActual, EstatFinal, CamiFinsAra, CamiTotal) :-
        CostMax > 0,
        unPas(CostPas, EstatActual, EstatSeguent),  % En B.1 i B.2, CostPas és 1.
        \+ member(EstatSeguent, CamiFinsAra),
        CostMax1 is CostMax-CostPas,
        cami(CostMax1, EstatSeguent, EstatFinal, [EstatSeguent|CamiFinsAra], CamiTotal).

unPas(...) :- ...
...
```

Problema C: El fitxer `tsp.pl` conté una solució amb *branch-and-bound* a un problema semblant al *Traveling-Salesman problem*. Intenta millorar-lo tal com s'indica en el fitxer.