

Programación con restricciones – Práctica 1 – Fdi - UCM

Importante:

- No simplificar restricciones ni usar “lógica humana” para deducir nada; solo intentar modelar el problema de forma lo más precisa posible.
- Las soluciones se copiarán a un fichero solución.txt Por favor no cambiar el formato del fichero.

- 1) [1 punto] Imagínate que estás en una isla en la que conviven caballeros, que siempre dicen la verdad, y duendes que siempre mienten. Te encuentras con 2 seres, a los que llamaremos A y B que dicen:

A: B es un duende

B: A es un duende

¿qué son cada uno de ellos? Escribe un modelo que encuentre todas las posibles soluciones

- 2) [2 puntos] Bertoldo siempre dicen la verdad lunes, miércoles y sábados, pero miente el resto de los días. Esta mañana me ha dicho: “Mañana diré la verdad”. ¿Qué día es hoy? Nota: solo hay una solución.
- 3) [2 puntos] El profesor de gimnasia, profesor Cacaseno, ha encontrado en el almacén pelotas de tenis, fútbol, baloncesto y rugby. De un tipo había 2, de otro 3, de otro 4 y de otro 5, pero no recuerda qué número corresponde a qué tipo de pelotas. Lo que sí recuerda es que
- Hay menos pelotas de tenis que de fútbol
 - Hay 3 pelotas más de rugby que de baloncesto

¿Cuántas pelotas de cada tipo hay?

Nota: El programa debe empezar de la forma:

```
% num pelotas de cada tipo
array [1..4] of var 2..5:pelotas;
% para acceder al array
int:tenis=1; int:futbol=2; int:baloncesto=3; int:rugby=4;
```

- 4) [2 puntos] Entre los amigos de Herminia, Bertoldo y Casimira se rumorea que:
- Casimira es mayor que Herminia
 - Herminia es mayor que Bertoldo
 - Casimira es mayor que Bertoldo

No se sabe si estos rumores son ciertos, lo que sí podemos decir es que si las dos primeras frases son ciertas, entonces la tercera es...¿verdadera o falsa? Escribe un modelo MiniZinc para resolver esta peliaguda pregunta.

Nota: Habrá pista alrededor de las 15h.

Nota2: La respuesta es obvia, pero se trata de hacer un modelo MiniZinc que determine, él solito, tal obviedad.

5) [2 puntos] Partimos de la siguiente declaraciones

```
int:N=4;
array [1..N] of var 1..N:a;
array [1..N] of var 1..N:b;
array [1..N] of var 1..2*N*N:c;

constraint C1;

constraint C2;

constraint C3;

solve satisfy;
```

Escribir las restricciones C1, C2 y C3 de forma que:

C1: Asegura que el array a está ordenado de menor a mayor, es decir si $i < j$, $a[i] < a[j]$

C2: Asegura que el array b está ordenado de mayor a menor, es decir si $i > j$, $b[i] > b[j]$ y que además b no coincide con a en ninguna posición $i=1..N$.

C3: Para todo $i=1..N$, $c[i] = a[i]*b[i]$

6) [1 punto] Queremos escribir un modelo MiniZinc para obtener números perfectos entre 1 y N. Los números perfectos son aquellos x cuya suma de divisores (excluyendo al propio x), es x.

Por ejemplo 6 es un número perfecto, porque sus divisores son 1,2,3, y $1+2+3 = 6$.

Encontrar la restricción C1 que permite encontrar estos números en este programa:

```
int:N=1000;
var 1..N:a;
constraint C1;
solve satisfy;
```

Para probar: deben salir 3 soluciones, 6, 28, 496

Nota: Difícil y sin pistas, para probar la excelencia del alumnado y la crueldad del profesorado, respectivamente.