



Técnicas de los Sistemas Inteligentes

Enunciado de la Práctica 2

En esta práctica se pide al estudiante resolver la siguiente relación de problemas usando MiniZinc. Cada ejercicio se resolverá en un **fichero MZN por separado**, debiendo entregarse el código debidamente comentado. La solución entregada debe ser **correcta y completa**, es decir, todas las soluciones válidas son admitidas por la codificación entregada y ésta no admite ninguna solución no válida. NO se entregará ninguna memoria, pero **el código debe estar perfectamente comentado, explicando y justificando la solución presentada, variables y restricciones usadas para codificar cada problema, y cualquier otro aspecto relativo a las decisiones tomadas durante el proceso de codificación. Un código cuyos comentarios sean, a criterio del profesor, extremadamente deficientes podrá invalidar total o parcialmente aquellos problemas a los que afecte.**

Nota: Los problemas con **errores sintácticos** en MiniZinc automáticamente califican con una puntuación de 0 (cero) puntos.

La entrega consistirá en un **fichero ZIP** que contenga los 10 ficheros MZN de cada problema.

Fecha de entrega: 16/05/2021 23:59

Modo de entrega: A través de la plataforma **Prado**, en el enlace correspondiente.

1. Puzzle Cripto-aritmético. Se plantea un problema criptoaritmético, de forma que cada letra codifica un único dígito (es decir, un número entero en $[0,9]$) y cada dígito está asignado a una única letra. Se pide encontrar una asignación de dígitos a letras que satisfaga la siguiente suma:

```
SIX
+ SEVEN
+ SEVEN
=====
TWENTY
```

La salida de MiniZinc debe formatearse como una suma perfectamente alineada, por ejemplo:

```
  6 5 0
+ 6 8 7 8 2
+ 6 8 7 8 2
-----
1 3 8 2 1 4
```

2. Se desea calcular la cantidad mínima de monedas para alcanzar un importe concreto. Usando las monedas de 1, 2, 5, 10, 20 y 50 céntimos, así como las de 1 y 2 euros, ¿cuál es la cantidad mínima de monedas para alcanzar un importe de 7.99€? La solución debe indicar el número de monedas así como la cantidad de monedas de cada tipo, y debe estar formateada de la siguiente manera (por ejemplo, si el importe buscado fuese 3.79€):

```
monedas: [0, 2, 1, 0, 1, 1, 1, 1]
total monedas: 7
```



Nota: Este problema tiene una parte de satisfacción (respetar la restricción de la suma de los importes) y un parte de optimización (minimizar la cantidad de monedas). Para resolverlo se recomienda comenzar por la parte de satisfacción y guardar en una variable la “cantidad” de monedas de la solución, y tras esto resolver la parte de optimización usando la sentencia:

`solve minimize <variable>`

3. Se desea colorear el mapa de América del Sur con 5 colores, de forma que dos países limítrofes sean coloreados con distinto color. La siguiente tabla muestra los países limítrofes a cada país:

PAÍS	VECINOS
Argentina	Bolivia, Brasil, Chile, Paraguay, Uruguay
Bolivia	Argentina, Brasil, Chile, Paraguay, Perú
Brasil	Argentina, Bolivia, Colombia, Guayana Francesa, Guyana, Paraguay, Perú, Surinam, Uruguay, Venezuela
Chile	Argentina, Bolivia, Perú
Colombia	Brasil, Ecuador, Perú, Venezuela
Ecuador	Colombia, Perú
Guayana Francesa	Brasil, Surinam
Guyana	Brasil, Surinam, Venezuela
Paraguay	Argentina, Bolivia, Brasil, Uruguay
Perú	Bolivia, Brasil, Chile, Colombia, Ecuador
Surinam	Brasil, Guayana Francesa, Guyana
Uruguay	Argentina, Brasil, Paraguay
Venezuela	Brasil, Colombia, Guyana

Adicionalmente, el uso de un color para colorear un país tiene un coste diferente (si, por ejemplo, se colorean dos países con el mismo color, el coste total sería el doble del coste de dicho color), de acuerdo a la siguiente tabla:

Verde	Naranja	Rojo	Azul	Amarillo
100€	200€	300€	400€	500€

Se pide encontrar la coloración de dicho mapa de forma que el coste total sea mínimo. La solución debe indicar el color de cada uno de los países por orden alfabético, así como el coste total de dicha coloración siguiendo el siguiente formato:

total: #coste_total

```
"Argentina      ": #color_asignado  #coste_de_dicho_color
"Bolivia        ": #color_asignado  #coste_de_dicho_color
"Brasil         ": #color_asignado  #coste_de_dicho_color
"Chile          ": #color_asignado  #coste_de_dicho_color
"Colombia       ": #color_asignado  #coste_de_dicho_color
"Ecuador        ": #color_asignado  #coste_de_dicho_color
"Guayana Francesa": #color_asignado  #coste_de_dicho_color
"Guyana         ": #color_asignado  #coste_de_dicho_color
"Paraguay       ": #color_asignado  #coste_de_dicho_color
"Peru           ": #color_asignado  #coste_de_dicho_color
"Surinam        ": #color_asignado  #coste_de_dicho_color
```



"Uruguay ": #color_asignado #coste_de_dicho_color
"Venezuela ": #color_asignado #coste_de_dicho_color

4. Un grupo de siete amigos se reúne y no se pone de acuerdo sobre el día de la semana en el que se encuentran. Cada uno de ellos afirma:
- Juan: "Hoy es martes".
 - María: "Eso no es correcto, mañana es martes".
 - Carmen: "Ambos estáis equivocados. Ayer fue martes".
 - Francisco: "No Carmen, ayer fue sábado".
 - Marta: "Hoy es o jueves o viernes".
 - Olga: "No es así, hoy es domingo".
 - Pepe: "No, hoy no es domingo".

Teniendo en cuenta que únicamente una de las afirmaciones anteriores es correcta, ¿en qué día de la semana estamos? La salida debe estar formateada de modo que se indique el día que es hoy en modo texto y quién ha emitido la afirmación correcta (indicando el nombre, en modo texto), es decir:

hoy: <día>
emisor: <persona>

5. Encontrar una asignación de horarios, para un día de clase, que satisfaga las siguientes condiciones:
- a. Existen tres aulas, disponibles entre las 9:00 y las 13:00.
 - b. Cada aula solo puede estar ocupada por un único profesor/a al mismo tiempo.
 - c. Existen 3 asignaturas (IA, TSI y FBD), que se imparten en periodos de 1h semanal.
 - d. Los estudiantes se dividen en 4 grupos (G1, G2, G3 y G4).
 - e. Cada grupo recibe docencia de una única asignatura en cada momento. Es decir, por ejemplo, G1 no puede tener clase a la misma hora de dos asignaturas diferentes.
 - f. Cada profesor imparte docencia de un único grupo/asignatura en cada momento.
 - g. El profesor 2, debido a sus problemas de movilidad, solo puede dar clase en el Aula 2 (que está especialmente habilitada para ello). Lo mismo ocurre con el profesor 4 y el aula 3.
 - h. La asignatura de TSI solo se puede dar en las Aulas 2 o 3, porque son las que tienen el software y hardware adecuado para ello.
 - i. Los grupos 1 y 3 de IA deben tener clase de 09:00 a 10:00 y de 11:00 a 12:00, respectivamente. Además, este último grupo tiene que dar clase en el Aula 1 obligatoriamente.
 - j. Cada profesor da docencia a los grupos definidos en la siguiente tabla. Además, cada profesor tiene las restricciones de horarios definidas en la siguiente tabla, que muestra las horas en las que, un cierto profesor, no puede dar clase.

Profesor	Asignaturas	Horas disponibles para el profesor
Prof1	FBD-G1, FBD-G2, FBD-G3, IA-G1	Todas
Prof2	IA-G2, FBD-G4	09:00-10:00 10:00-11:00
Prof3	IA-G3, TSI-G1, TSI-G2, TSI-G4	Todas
Prof4	IA-G4, TSI-G3	Todas



El formato en que se debe presentar la salida es el siguiente (se muestra un ejemplo cualquiera; no necesariamente la salida deseada a este problema concreto):

	Aula1	Aula2	Aula3
" 9" - "10"	->"FBD-G1 (P2)"	"TSI-G2 (P1)"	"FBD-G3 (P3)"
"10" - "11"	->"TSI-G3 (P3)"	" IA-G4 (P4)"	" IA-G2 (P1)"
"11" - "12"	->"TSI-G4 (P3)"	"TSI-G1 (P1)"	"FBD-G4 (P3)"
"12" - "13"	->" IA-G1 (P1)"	" IA-G3 (P4)"	"FBD-G2 (P2)"

Nota: el número de soluciones de este ejercicio es 5.

6. Encontrar una asignación de horarios que satisfaga las siguientes condiciones:

- Existe un aula disponible en seis franjas consecutivas de 1h (por ejemplo, de 8:00 a 14:00) de lunes a viernes.
- Existen nueve asignaturas (A1..A9). El número de horas semanales de cada asignatura se detalla en la siguiente tabla:

A1	A2	A3	A4	A5	A6	A7	A8	A9
4 hrs	2 hrs	4hrs	4hrs	4hrs	2hrs	2hrs	2hrs	1hr

- Las asignaturas {A1,A3,A4,A5,A8} deben impartirse en bloques de 2h consecutivas, mientras que el resto, es decir {A2,A6,A7,A9}, se imparten en bloques de 1h.
- En cada día de la semana solo se puede impartir, como máximo, un bloque de cada asignatura.
- El profesor/a de cada asignatura es el siguiente: Prof1={A1,A3}; Prof2={A4,A5}; Prof3={A6,A9}; Prof4={A2,A7,A8}.
- Cada profesor solo puede impartir un bloque de alguna de sus asignaturas cada día, excepto Prof4 (que puede impartir más de una).
- La cuarta franja horaria debe reservarse para el recreo ("NA"); es decir, no asignar ninguna asignatura.
- El orden de cada bloque es importante, en el sentido de que el primer bloque de cada asignatura se posiciona en un día anterior que el segundo bloque.
- Varios profesores tienen ciertas restricciones horarias:
 - El Profesor 1 solo puede dar clase en las dos últimas horas de la mañana.
 - El Profesor 2 solo puede dar clase en las dos primeras horas de la mañana.
 - El Profesor 3 solo puede dar clase en la hora justo antes del recreo.
- Varias asignaturas también deben ser impartidas ciertos días concretos de la semana:
 - Asignatura 1: lunes o martes
 - Asignatura 3: miércoles o jueves
 - Asignatura 4: lunes o martes
 - Asignatura 5: jueves o viernes
 - Asignatura 6: miércoles o viernes
 - Asignatura 7: martes o viernes
 - Asignatura 8: miércoles
 - Asignatura 9: lunes



La salida de Minizinc debe estar formateada del siguiente modo:

```
"      Lunes      "      Martes      "      Miercoles      "      Jueves      "      Viernes      "
"08:00-09:00" -> "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF"
"09:00-10:00" -> "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF"
"10:00-11:00" -> "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF"
"11:00-12:00" -> "      NA      "      NA      "      NA      "      NA      "      NA      "
"12:00-13:00" -> "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF"
"13:00-14:00" -> "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF" "#ASIG-BLOQ-PROF"
```

Nota: el número de soluciones de este ejercicio es 2.

7. En la tabla adjunta aparece la información necesaria para llevar a cabo la construcción de un avión. En la primera columna aparecen los identificadores de las tareas necesarias para construirlo. En la segunda columna aparece la descripción de cada una de estas. En la tercera columna se muestra la duración (en días) de cada una de las tareas. Y la cuarta columna muestra la relación de precedencia entre tareas: por ejemplo, si el contenido de la celda correspondiente a la tarea “G” es “C,D” se debe interpretar como que “las tareas C y D deben finalizarse antes que comience la tarea “G”. Se pide encontrar una asignación de tiempos de inicio a estas tareas de forma que se pueda construir el avión en el menor tiempo posible, asumiendo que cada tarea la realiza un único trabajador y que se disponen de tantos trabajadores como se necesiten.

Nota: La primera tarea debe comenzar en tiempo “1”.

Tarea	Descripción	Duración	Predecesoras
A	Diseño general de la aeronave	60	Ninguna
B	Transporte de secciones grandes pre-ensambladas	10	A
C	Unión de fuselaje delantero y trasero	5	B
D	Conexión de cables y tuberías	5	C
E	Revestimientos térmicos y acústicos	5	D
F	Piso del avión	2	D
G	Revisión de sistemas del avión (hidráulico, eléctrico, combustible,...) y superficies de control	7	D,E
H	Montaje del tren de aterrizaje	3	C
I	Montaje de timones, alerones, flaps, y cono de cola	7	E,F
J	Colocación de los baños, la cocina, las sillas	2	E,F
K	Ensamblaje final e instalación del motor	5	I,J
L	Proceso de certificación	1	K

Formatear la salida de MiniZinc de modo que muestre por pantalla lo siguiente:

```
TIEMPO MINIMO = ##
Tarea "A" empieza en tiempo 1
Tarea "B" empieza en tiempo ##
Tarea "C" empieza en tiempo ##
Tarea "D" empieza en tiempo ##
Tarea "E" empieza en tiempo ##
Tarea "F" empieza en tiempo ##
Tarea "G" empieza en tiempo ##
Tarea "H" empieza en tiempo ##
Tarea "I" empieza en tiempo ##
Tarea "J" empieza en tiempo ##
Tarea "K" empieza en tiempo ##
Tarea "L" empieza en tiempo ##
```



8. Imaginemos ahora un problema similar al anterior, pero en donde lo que se busca construir es una casa (véase la siguiente tabla de tareas y duraciones).

Tarea	Descripción	Duración	Predecesoras
A	Levantar muros	7	Ninguna
B	Carpintería de tejado	3	A
C	Poner tejado	1	B
D	Instalación eléctrica	8	A
E	Pintado fachada	2	C,D
F	Ventanas	1	C,D
G	Jardín	1	C,D
H	Techado	3	A
I	Pintado interior	2	F,H

Se busca resolverlo suponiendo que disponemos de tres trabajadores (que en cada momento solo pueden estar dedicados a una única tarea) y que las tareas requieren el siguiente número de trabajadores:

A	B	C	D	E	F	G	H	I
2	3	2	2	1	2	1	1	2

La salida de MiniZinc debe estar formateada como sigue:

```
TIEMPO MINIMO = ##
*****
```

```
Tarea "A" empieza en tiempo 1
Tarea "B" empieza en tiempo ##
Tarea "C" empieza en tiempo ##
Tarea "D" empieza en tiempo ##
Tarea "E" empieza en tiempo ##
Tarea "F" empieza en tiempo ##
Tarea "G" empieza en tiempo ##
Tarea "H" empieza en tiempo ##
Tarea "I" empieza en tiempo ##
*****
```

Asignacion:

Matriz en la que cada fila es un trabajador (por tanto, con 3 filas), cada columna es un día, y cada valor se corresponde con la tarea desarrollada. Incluir un 0 si el trabajador no realiza ninguna tarea en un cierto momento.

9. Acaba de llegar un paciente al hospital y todavía no ha sido diagnosticado. No obstante, se sabe que presenta tos seca, dolor de cabeza, y dolor de garganta. Se muestra a continuación la base de datos de pacientes anteriores y su diagnóstico.

	Fatiga	Tos Seca	Dolor de Cabeza	Dolor de Garganta	Diagnóstico
Paciente 1	0	0	1	1	COVID
Paciente 2	1	1	1	0	COVID
Paciente 3	1	1	0	1	COVID
Paciente 4	1	0	0	1	COVID
Paciente 5	1	1	0	1	COVID
Paciente 6	0	1	1	1	COVID



ugr

Universidad de Granada

Departamento de Ciencias de la Computación
e Inteligencia Artificial



Paciente 7	1	0	0	0	COVID
Paciente 8	1	1	0	0	COVID
Paciente 9	0	1	0	0	COVID
Paciente 10	0	0	1	1	COVID
Paciente 11	1	0	0	1	Gripe
Paciente 12	1	0	0	1	Gripe
Paciente 13	0	1	1	0	Gripe
Paciente 14	1	1	0	0	Gripe
Paciente 15	1	0	1	0	Gripe
Paciente 16	1	0	0	0	Gripe
Paciente 17	1	0	0	1	Alergias
Paciente 18	0	1	1	0	Alergias
Paciente 19	0	0	1	0	Alergias
Paciente 20	1	0	1	0	Alergias
Nuevo paciente	0	1	1	1	¿?

Se pide aplicar una aproximación similar al principio del vecino más cercano para realizar el diagnóstico del nuevo paciente. Es decir, se debe calcular la distancia de este nuevo paciente a todos los demás, y seleccionar la distancia mínima, que determinará cuál es el diagnóstico más probable. Para ello, se proporcionan los siguientes predicados, para que sean utilizados por el alumnado:

```
% Calcular la distancia entre dos pacientes.
%
predicate distancia(array[int] of int: a, array[int] of var int: v, var int:
d) =

    d = sum(i in index_set(a)) (
        (a[i]-v[i])*(a[i]-v[i])
    )
    /\ d >= 0
;

% Recuperar el índice del menor valor de un array (i.e. argmin)
%
predicate indice_minimo(var int: mi, array[int] of var int: x) =
    exists(i in index_set(x)) (
        x[i] = min(x)
        /\
        mi = i
    )
;
```

La salida de MiniZinc debe mostrar las distancias calculadas, el paciente de la base de datos más parecido y el probable diagnóstico.

10. Un explorador necesita partir rápidamente para un viaje a la selva. Para ello, debe escoger varios objetos, que no excedan, en total, los 5kgs, dado que el peso no debe reducir su movilidad y capacidad para desplazarse son rapidez. Cada objeto tiene distinto grado de preferencia para el explorador (la preferencia se mide en una escala de 0 a 100, donde los números más altos representan los objetos más útiles o necesarios). En la tabla siguiente se muestran los objetos, su peso y su preferencia. Encontrar el conjunto de objetos cuya suma de preferencias sea máxima sin exceder el peso máximo determinado.

*ugr***Universidad de Granada**Departamento de Ciencias de la Computación
e Inteligencia Artificial

Objeto	Peso (en gramos)	Preferencia
Mapa	50	100
Cuchillo	500	90
Prismáticos	1000	75
Agua	3000	95
Linterna	1500	85
Cerveza	3000	10
Novela	500	5
Manual de Supervivencia	1000	70
Plátanos	2000	90
Caramelos	500	20
Cerillas	250	50
Espejo	500	40

La salida de MiniZinc se debe formatear del siguiente modo:

```
Objetos = #lista_de_objetos | Preferencias = #suma_preferencias |  
PesoFinal = #peso_final kgs
```