

Taller Modelamiento e Implementación CSPs
Programación por restricciones
Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle

Robinson Duque, Juan Francisco Díaz

A continuación **Usted encontrará 8 ejercicios de los cuales debe presentar un total de 7**. La solución a todos y cada uno de los puntos del taller se debe almacenar en un archivo .mzn tal como se describe en cada punto. Usted debe entregar como solución al taller, el conjunto de esos archivos empaquetado y comprimido en un solo archivo cuyo nombre debe ser análogo a taller1PRJFDiaz.zip o taller1PRJFDiaz.tgz o taller1PRJFDiaz.tar.gz o taller1PRJFDiaz.rar. Adicionalmente debe entregar un archivo practica.pdf, con el desarrollo del taller que no corresponde a código MiniZinc. Debe subir su solución al campus a más tardar el día viernes 6 de Diciembre (2019).

PARTE 1 (CSPs)

1. Sudoku (Calentamiento)

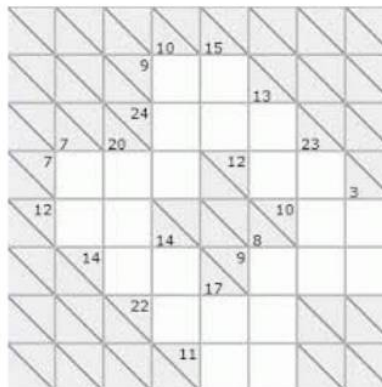
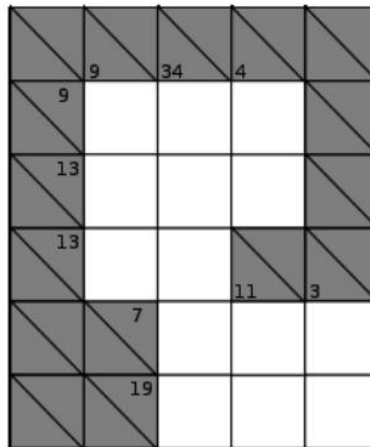
Sudoku es un rompecabezas matemático de colocación que se popularizó en Japón en 1986 y se dio a conocer en el ámbito internacional en 2005. El objetivo es rellenar una cuadrícula de 9×9 celdas (81 casillas) dividida en subcuadrículas de 3×3 (también llamadas “cajas” o “regiones”) con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ninguna cifra en una misma fila, columna o subcuadrícula. Un sudoku está bien planteado si la solución es única. La resolución del problema requiere paciencia y ciertas dotes lógicas.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 9 | | | | | 1 | |
| 8 | | 4 | | 2 | | 3 | 7 |
| | 6 | | 9 | | 7 | 2 | |
| | | 5 | | 3 | | 1 | |
| | 7 | | 5 | | 1 | | 3 |
| | | 3 | | 9 | | 8 | |
| | 2 | | 8 | | 5 | | 6 |
| 1 | | 7 | | 6 | | 4 | 9 |
| | 3 | | | | | | 8 |

Modele el problema del Sudoku como un CSP, especifique las variables, dominios y restricciones que requiera. Implemente su solución en MiniZinc . Utilice diferentes implementaciones de las restricciones que modeló y diferentes estrategias de distribución. Muestre claramente cuáles son las ventajas y desventajas de las diferentes implementaciones que usted realice. Guarde su solución en el archivo **sudoku.mzn**.

2. Kakuro

El Kakuro es un juego tipo crucigrama derivado del Sudoku, aunque menos conocido. Este tipo de crucigramas no tiene claves verbales sino numéricas y como en ellos, cada respuesta sirve de ayuda para resolver las otras que se cruzan. Por este motivo, el nombre original del juego era rompecabezas de sumas cruzadas. El objetivo del Kakuro es rellenar las casillas con números del 1 al 9 de forma que en ninguna serie de números, en horizontal o vertical, se repitan y que su suma dé como resultado el número que se indica.



Modele el problema del Kakuro como un CSP, especifique las variables, dominios y restricciones que requiera. Implemente su solución en MiniZinc . Utilice diferentes implementaciones de las restricciones que modeló y diferentes estrategias de distribución. Muestre claramente cuáles son las ventajas y desventajas de las diferentes implementaciones que usted realice. Guarde su solución en el archivo **kakuro.mzn**.

3. Secuencia Mágica

Una secuencia mágica de longitud n es una secuencia x_0, \dots, x_{n-1} de enteros tales que para cada x_i ($i = 0, \dots, n-1$):

- x_i en un entero entre 0 y $n-1$.
- El número i ocurre exactamente x_i veces en la secuencia

Ejemplo: 6,2,1,0,0,0,1,0,0,0 es una secuencia mágica de longitud 10.

1. Usted debe escribir un script parametrizado en un número n que pueda enumerar todas las secuencias mágicas de longitud n .

2. El tamaño del árbol de búsqueda puede ser conderablemente reducido imponiendo propagadores para las restricciones redundantes:

- $x_0 + \dots + x_{n-1} = n$
- y
- $(-1)x_0 + \dots + (n-2) * x_{n-1} = 0$

Modifique su solución para incluir estas restricciones y explique por qué estas restricciones son redundantes. Guarde su solución en el archivo **secuencia.mzn**

Ayuda: Dentro del listado de restricciones globales de MiniZinc (<https://www.minizinc.org/doc-2.3.2/en/lib-globals.html>) existe un conjunto de restricciones que permiten realizar conteos (4.2.6.5. Counting constraints), dentro de estas restricciones existe una que crea un propagador para imponer la restricción que hayan exactamente n variables dentro de un arreglo x que tomen el valor v .

4. Acertijo Lógico

A tres amigos les gusta diferentes tipos de música. Dadas las pistas a continuación, identifique a cada uno de ellos, diga qué edad tiene, cuál es el tipo de música que le agrada y su nombre (nombre y apellido).

1. Juan es más viejo que González, a quien le gusta la música clásica.
2. El fan de la música pop, que no es García, no tiene 24.
3. Oscar, quien no es López, tiene 25.
4. La música favorita de Darío no es el jazz.
5. Las edades de los 3 amigos están entre 24 y 26 años.

Modele este problema como un CSP, especifique las variables, dominios y restricciones además de la estrategia de distribución. Implemente su solución en MiniZinc. Guarde su solución en el archivo **acertijo.mzn**.

5. Ubicación de personas en una reunión

Un grupo de N personas desea tomarse una foto, para lo cual las personas se organizan en una fila. Las personas pueden definir una o varias de las siguientes preferencias sobre su posición en la fila para la fotografía:

1. $\text{next}(A\ B)$: La persona A debe estar junto a la persona B en la fotografía.
2. $\text{separate}(A\ B)$: La persona A debe estar separada de la persona B (y viceversa).
3. $\text{distancia}(A\ B\ M)$: Las personas A y B deben estar separadas como máximo por M personas.

Escriba un modelo que permita calcular el orden de la fila dada una entrada de la forma:

personas=["p1", "p2", ..., "pn"]

next=[$[p_{i1}, p_{j1}] \dots [p_{ik}, p_{jk}]$], donde $p_{i1}, \dots, p_{ik}, p_{j1}, \dots, p_{jk} \in \{1, \dots, n\}$

separate=[$[p_{x1}, p_{y1}] \dots [p_{xs}, p_{ys}]$], donde $p_{x1}, \dots, p_{xs}, p_{y1}, \dots, p_{ys} \in \{1, \dots, n\}$

distancia=[$[p_{a1}, p_{b1}, c_1] \dots [p_{ad}, p_{bd}, c_d]$], donde $p_{a1}, \dots, p_{ad}, p_{b1}, \dots, p_{bd} \in \{1, \dots, n\}$ y $c_1, \dots, c_d \in \{1, \dots, n-2\}$

Un ejemplo de entrada sería:

personas=["jose", "julian", "diana", "juan", "david", "claudia", "edgar", "mario"]

next=[5,8]

separate=[3, 7]

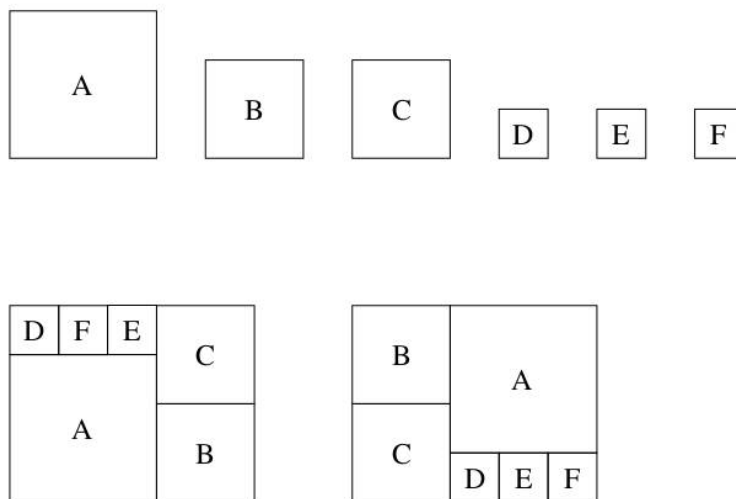
distancia=[6,5,2]

Guarde su solución en el archivo **reunion.mzn**.

6. Construcción de un rectángulo

Considere el problema de la construcción de un rectángulo a partir de un conjunto de cuadrados: n cuadrados de tamaños dados (algunos del mismo tamaño otros de diferente tamaño) deben colocarse unos al lado de otros y unos encima de otros de tal manera que se forme un rectángulo de anchura y altura dadas. Dos cuadrados cualesquiera no se pueden solapar. Todo cuadrado debe ser utilizado y, obviamente todo cuadrado debe ser colocado con sus lados paralelos al eje X y al eje Y.

Por ejemplo, dados seis cuadrados de lados 3,2,2,1,1,1 respectivamente, construir un rectángulo de anchura 5 y altura 4 (ver figura). En este punto Ud. debe modelar el problema, establecer las restricciones y explorar distintas estrategias de distribución.



Modele este problema como un CSP, especifique las variables, dominios y restricciones además de la estrategia de distribución. Implemente su solución en MiniZinc. Guarde su solución en el archivo **rectangulo.mzn**.

PARTE 2 (COPs)

7. Construcción de Transformadores

Walter construye dos tipos de transformadores y tiene disponibles 6 toneladas de material ferromagnético y 28 horas de tiempo de trabajo. El transformador 1 requiere 2 toneladas de material

ferromagnético y 7 horas de trabajo, y el transformador 2 requiere 1 tonelada de material ferromagnético y 8 horas de trabajo. Los precios de venta de los transformadores 1 y 2 son, son 120 y 80 mil dolares respectivamente. ¿Cuántos transformadores de cada tipo debe fabricar Walter para maximizar sus beneficios?

a Dibuje una tabla que represente los datos del problema.

b Proponga un modelo COP para el problema utilizando los valores presentados en la tabla.

c Presente la implementación del modelo anterior utilizando código MiniZinc.

d Proponga un modelo COP genérico para n transformadores asumiendo que los valores en la tabla pueden cambiar.

8. Producción de Gasolina

Una compañía petrolera mezcla gasolina de tres ingredientes: butano, catalítico reformado y nafta pesado. A continuación se detallan las características de los ingredientes y los requisitos mínimos para la gasolina regular:

| | Butano | Catalítico Reformado | Nafta Pesado | Gasolina |
|------------------|--------|----------------------|--------------|-----------|
| Octano | 120 | 100 | 74 | ≥ 89 |
| Vapor de Presión | 60 | 2.5 | 4.0 | ≤ 11 |
| Volatilidad | 105 | 3 | 12 | ≥ 17 |

El costo (por galón) de butano es de \$ 0.58, de \$ 1.55 para catalítico reformado y \$ 0.85 para nafta pesado. ¿Cuántos galones de los tres ingredientes deben mezclarse para producir 12,000 galones de gasolina a un costo mínimo?

Aclaración: note que cada galón de gasolina debe cumplir con ciertos valores de Octano, Valor de Presión y Volatilidad. Los valores de la tabla representan la cantidad de octano, valor de presión y volatilidad contenidos en cada galón de butano, catalítico reformado y nafta pesado.

a Proponga un modelo COP para el problema utilizando los valores presentados en la tabla.

b Presente la implementación del modelo anterior utilizando código MiniZinc.

c Proponga un modelo genérico asumiendo que los valores en la tabla pueden cambiar dependiendo del proveedor de cada uno de los ingredientes.

d Presente la implementación del modelo anterior utilizando código MiniZinc.