

Taller 1

Modelamiento e Implementación de CSPs

Programación por restricciones

Escuela de Ingeniería de Sistemas y Computación



Profesor Juan Francisco Díaz Frias

Marzo de 2024

1. Introducción

A continuación Usted encontrará una serie de ejercicios que debe resolver individualmente, modelando el problema enunciado como un CSP, implementando el modelo en MiniZinc, haciendo pruebas y análisis de las pruebas enfocándose en la corrección del modelo, y en la eficiencia de la(s) estrategia(s) de búsqueda implementadas.

	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. Los problemas

2.1. Sudoku

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.

2.2. 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.

	9	34	4		
9					
13					
13					
			11	3	
	7				
	19				

			10	15			
		9					
				13			
7	24				23		
7	20		12			3	
12				10			
	14	14	9	8			
		22	17				
			11				

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`.

2.3. 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.

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 considerablemente 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`.

2.4. 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`.

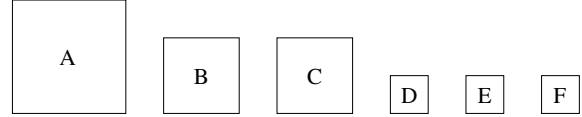
2.5. Ubicación de personas en una reunión

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. $next(A, B)$: La persona A debe estar junto a la persona B en la fotografía.

2. $separate(A, B)$: La persona A debe estar separada de la persona B (y viceversa).

3. $distance(A, B, M)$: 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:

$personas = ["p1" \ "p2" \ \dots \ "pn"]$

$next = [[p_{i_1}, p_{j_1} | \dots | p_{i_k}, p_{j_k}]]$ donde $p_{i_1}, \dots, p_{i_k}, p_{j_1}, \dots, p_{j_k} \in \{1, 2, \dots, n\}$

$separate = [[p_{x_1}, p_{y_1} | \dots | p_{x_s}, p_{y_s}]]$ donde $p_{x_1}, \dots, p_{x_s}, p_{y_1}, \dots, p_{y_s} \in \{1, 2, \dots, n\}$

$distance = [[p_{a_1}, p_{b_1}, c_1 | \dots | p_{a_w}, p_{b_w}, c_w]]$ donde $p_{a_1}, \dots, p_{a_w}, p_{b_1}, \dots, p_{b_w} \in \{1, 2, \dots, n\}$ y $c_1, \dots, c_w \in \{1, \dots, n-2\}$

Un ejemplo de entrada sería:

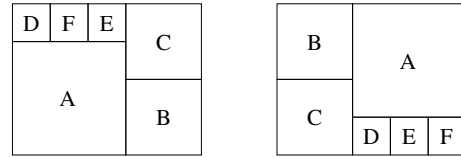
$personas = ["jose" \ "julian" \ "diana" \ "juan" \ "david" \ "claudia" \ "edgar" \ "mario"]$

$next = [[5, 8]]$

$separate = [[6, 7]]$

$distance = [[6, 5, 2]]$

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



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`.

3. Informe, entrega y evaluación

3.1. Sobre el Informe...

El grupo deberá entregar un informe del proyecto, en formato pdf, que contenga, al menos, los siguientes aspectos, por cada uno de los ejercicios desarrollados:

- El modelo: una descripción del modelo (parámetros, variables, restricciones, anotaciones de búsqueda) y una justificación de su adecuación al problema planteado.
- Detalles importantes de implementación: lo más relevante de la implementación, sin incluir código (por ejemplo restricciones redundantes y rompimiento de simetrías).
- El análisis de los árboles de búsqueda generados por su modelo.
- Pruebas: descripción de las pruebas realizadas a su implementación.
- Análisis: del modelo, de las pruebas, de las estrategias de búsqueda.
- Conclusiones: Esta es una de las partes más interesantes del trabajo (pero no por ello la

2.6. 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 diseñar una estrategia de distribución.

que más vale). En ella se espera que usted analice los resultados obtenidos y **justifique** claramente sus afirmaciones.

3.2. Grupos de trabajo

El taller puede ser desarrollado por grupos de máximo 2 personas.

3.3. Entrega

La entrega se debe realizar vía el campus virtual en las fechas previstas para ello, por uno sólo de los integrantes del grupo. **La fecha de entrega límite es el 16 de abril de 2024 a las 17:00.** Se debe subir al campus virtual en el enlace correspondiente a este taller un archivo comprimido **.zip** que siga la convención *CódigoEstudiante1-CódigoEstudiante2-Taller1-PPR-2023-I.zip*. El comprimido deberá contener:

1. Archivo **Readme.txt** que describa todos los archivos entregados y las instrucciones para ejecutarlos.
2. Archivo **InformeTaller1.pdf** acorde a la Sección 3.1.
3. Archivos ***.mzn** y ***.dzn** con la implementación del modelo y los datos de prueba respectivos.

3.4. Evaluación

La evaluación de cada taller se hará de acuerdo al número de ejercicios desarrollados (idealmente todos), a la claridad y calidad de los informes y al funcionamiento de las implementaciones.

En específico se calificará según los siguientes criterios:

- Definición de variables de los modelos (10)

Se espera que las variables de los modelos estén precisamente definidas (su significado), y sus dominios estén precisamente definidos. Nunca se confunden variables con parámetros. Además de estar precisamente definidos, los dominios de las variables son apropiados para los problemas que modelan y no contienen valores que desde el inicio se sabe que no pueden tomar.

- Definición de las restricciones de los modelos (15)

Se espera que al definir las restricciones de los problemas, se tengan en cuenta todas las restricciones mencionadas en el problema, modelando todos los aspectos relevantes. Las restricciones se definen independientemente de la implementación, en un lenguaje formal preciso y claro, y se justifica su corrección o adecuación, cuando esta no es evidente. Las restricciones, en general, no están condicionadas por los valores que puedan tomar las variables; este tipo de restricciones sólo se dejan en caso de no haber otra opción.

- Claridad para imponer restricciones que rompen simetrías (10)

Se espera que cuando el problema lo permita se definan restricciones que rompan simetrías, se expliquen claramente qué simetrías rompen y cómo recuperar las soluciones que se pierden al imponer esas restricciones. Si no hay lugar a restricciones que rompan simetrías, se espera al menos un párrafo argumentando por qué no, o que diga que no lograron definir ninguna.

- Claridad para imponer restricciones redundantes (10)

Se espera que cuando el problema lo permita se definan restricciones redundantes, se explique claramente por qué son redundantes y se justifique que con ellas el espacio de búsqueda se poda. Si no hay lugar a restricciones redundantes, se espera al menos un párrafo argumentando por qué no o que diga que no lograron definir ninguna.

- Claridad en la escogencia de las mejores estrategias de búsqueda (15)

Se espera que no sólo se hayan explorado los modelos con la estrategia de búsqueda por defecto, sino que se haya reflexionado alrededor del tema de la exploración, y producto de esa reflexión se hayan definido diversas anotaciones (al menos dos, pero pueden ser más) que den lugar a diversas estrategias de búsqueda, y que se analicen y comparen dichos comportamientos aprovechando la estadística que

provee MiniZinc sobre las búsquedas (nodos explorados, nodos fallidos, tiempo de respuesta, ...).

- Calidad de las pruebas (5)

Se espera que cada modelo se haya probado con un suficiente número de pruebas para poder analizar a profundidad el modelo. Por ello este criterio lo que evalúa es si esa batería de pruebas es suficiente para verificar la corrección del modelo y para explorar las diferentes estrategias de búsqueda y compararlas. También se espera que los resultados de las pruebas se sistematicen de alguna manera (tablas, gráficas, ...) y se presenten en el informe.

- Coherencia entre los modelos, las implementaciones y los análisis presentados en el informe (25)

Se espera que todos los modelos descritos hayan sido implementados coherentemente en MiniZinc y que los análisis correspondan con las pruebas de esos modelos en MiniZinc.

- Redacción del informe (10)

Se espera que el informe sea completo, claro, preciso, fácil de leer, bien presentado, contenga los análisis más importantes y las conclusiones.

Éxitos!!!