Programación con Restricciones
Constraint Programming [MII-771]
Capítulo 3: Modelado

Dr. Ricardo Soto
[ricardo.soto@ucv.cl]
[http://www.inf.ucv.cl/~rsoto]

Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

1. Introducción

Solving = Modeling + Search

1. Introducción

Ejemplo 1 - SEND+MORE=MONEY

Resolver la siguiente ecuación, reemplazando las letras por dígitos distintos.

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

$$\begin{array}{cccccc} & 9 & 5 & 6 & 7 \\ + & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & 2 \end{array}$$

1. Introducción

Modelo

- Variables

$$\text{S,E,N,D,M,O,R,Y} \in [0, 9]$$

- Restricciones

$$\begin{array}{rrrrrrrrrrrrr} & & & 1000 \cdot S & + & 100 \cdot E & + & 10 \cdot N & + & D \\ & & + & 1000 \cdot M & + & 100 \cdot O & + & 10 \cdot R & + & E \\ = & 10000 \cdot M & + & 1000 \cdot O & + & 100 \cdot N & + & 10 \cdot E & + & Y \end{array}$$

$$S \neq E, \quad S \neq N, \quad S \neq D \quad \ldots \quad R \neq Y$$

1. Introducción

Usando Global Constraints

- Variables

$$S,E,N,D,M,O,R,Y \in [0,9]$$

- Restricciones

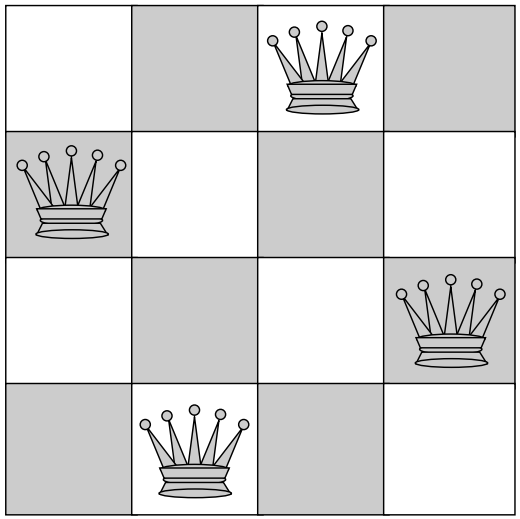
$$\begin{array}{rrrrrrrrrrrrr} & & & 1000 \cdot S & + & 100 \cdot E & + & 10 \cdot N & + & D \\ & & + & 1000 \cdot M & + & 100 \cdot O & + & 10 \cdot R & + & E \\ = & 10000 \cdot M & + & 1000 \cdot O & + & 100 \cdot N & + & 10 \cdot E & + & Y \end{array}$$

$$alldifferent(S, E, N, D, M, O, R, Y)$$

1. Introducción

Ejemplo 2 - N-Queens

Ubicar **n** reinas en un tablero de ajedrez de **n** $\times$ **n**, de manera tal que no se puedan atacar.



1. Introducción

Modelo

- Variables

$Q_1, Q_2, Q_3, Q_4 \in [1, 4]$

- Restricciones (para $i \in [1, 3]$ y $j \in [i + 1, 4]$)

$Q_i \neq Q_j$ (filas)

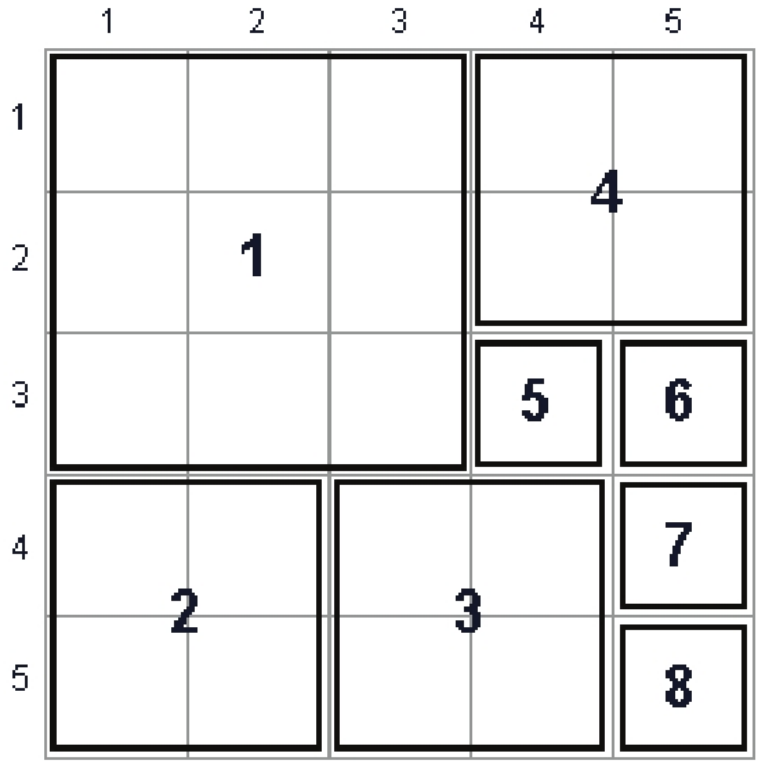
$Q_i + i \neq Q_j + j$ (diagonal 1)

$Q_i - i \neq Q_j - j$ (diagonal 2)

2. Ejercicios

Ejercicio 1 - Packing Squares

Ubicar un conjunto de cuadrados dentro una base cuadrada de tal manera que ningún cuadrado se translape con otro.



2. Ejercicios

- Variables

$x_1, x_2, ..., x_{squares} \in [1, sideSize]$
$y_1, y_2, ..., y_{squares} \in [1, sideSize]$

- Constantes

$sideSize$
$squares$
$size_1, size_2, ..., size_{squares}$

- Restricciones (para $i \in [1, squares]$) //inside

$x_i \leq sideSize - size_i + 1$
$y_i \leq sideSize - size_i + 1$

2. Ejercicios

- Restricciones (para $i \in [1, squares]$ y $j \in [i+1, squares]$) //noOverlap

$x_i + size_i \leq x_j$ *OR*
$x_j + size_j \leq x_i$ *OR*
$y_i + size_i \leq y_j$ *OR*
$y_j + size_j \leq y_i$

2. Ejercicios

Ejercicio 2 - Sudoku

Completar una matriz de 9x9 de manera tal que cada fila, cada columna y cada una de las sub-matrices de 3x3 tengan dígitos distintos del 1 al 9.

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

2. Ejercicios

- Variables

$$x_{1,1}, x_{1,2}, \ldots, x_{9,9} \in [1, 9]$$

- Restricciones (para $k \in [1, n], i \in [1, n], j \in [i + 1, n]$) //differentInRowsAndColumns

$$x_{k,i} \neq x_{k,j}$$
$$x_{i,k} \neq x_{j,k}$$

- Restricciones (para $k1, j1, k2, j2, k3, j3 \in [1, 3]$ y si $(k2 \neq k3 \ \ AND \ \ j2 \neq j3)$) //differentInSubSquares

$$x_{(k1-1)*3+k2,(j1-1)*3+j2} \neq x_{(k1-1)*3+k3,(j1-1)*3+j3}$$

2. Ejercicios

Usando Global Constraints

- Variables

$$x_{1,1}, x_{1,2}, \ldots, x_{9,9} \in [1, 9]$$

- Restricciones (para $i \in [1, n]$) //differentInRowsAndColumns

$$alldifferent(getColumn(x, i))$$
$$alldifferent(getRow(x, i))$$

- Restricciones (para $i, j \in [1, 3]$) //differentInSubSquares

$$alldifferent(getSubMatrix(x, (i-1)*3+1, i*3, (j-1)*3+1, j*3));$$

2. Ejercicios

Ejercicio 3 - Stable Marriage

Considere un grupo de n hombres y n mujeres que quieren casarse. Cada mujer tiene un ranking de preferencia para su posible marido y así como también cada hombre para su posible esposa. El objetivo es formar los matrimonios de manera tal de que no haya 2 personas de sexo opuesto que se gusten más que sus respectivas parejas.

2. Ejercicios

- Variables

$husband_1, husband_2, ..., husband_n \in [1, n]$
$wife_1, wife_2, ..., wife_n \in [1, n]$

- Constantes

$man\_rank_{1,1}, man\_rank_{1,2}, ..., man\_rank_{n,n}$
$woman\_rank_{1,1}, woman\_rank_{1,2}, ..., woman\_rank_{n,n}$

2. Ejercicios

- Restricciones (para $m \in [1, n]$) //matchHusbandWife

$$husband_{wife_m} = m$$

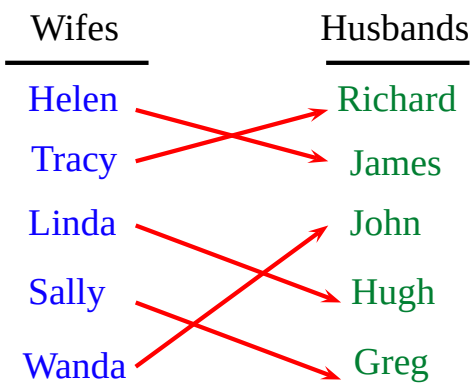
- Restricciones (para $w \in [1, n]$)

$$wife_{husband_w} = w$$

- Restricciones (para $m \in [1, n], w \in [1, n]$)

$$man\_rank_{m,w} < man\_rank_{m,wife_m} \Rightarrow woman\_rank_{w,husband_w} < woman\_rank_{w,m}$$
$$woman\_rank_{w,m} < woman\_rank_{w,husband_w} \Rightarrow man\_rank_{m,wife_m} < man\_rank_{m,w}$$

2. Ejercicios



Men Rank	Helen	Tracy	Linda	Sally	Wanda
Richard	5	1	2	4	3
James	4	1	3	2	5
John	5	3	2	4	1
Hugh	1	5	4	3	2
Greg	4	3	2	1	5

Women Rank	Richard	James	John	Hugh	Greg
Helen	1	2	4	3	5
Tracy	3	5	1	2	4
Linda	5	4	2	1	3
Sally	1	3	5	4	2
Wanda	4	2	3	5	1

Ej: Tracy está casada con Richard, Tracy prefiere a Richard en 3er lugar y Richard la prefiere en 1er lugar. Si Tracy quisiese cambiar su marido no podría ya que John (su primera preferencia) prefiere a su señora (Wanda) más que a Tracy igual que Hugh (casado con Linda) que prefiere a Tracy en último lugar.

2. Ejercicios

Ejercicio 4 - Social Golfers

Considere un grupo de n golfistas que juegan una vez por semana y siempre en grupos de tamaño g. El objetivo es organizar un calendario para w semanas de manera tal que 2 golfistas no juegen juntos más de una vez.

2. Ejercicios

- Variables

$$schedule_{1,1}, schedule_{1,2}, \ldots, schedule_{weeks,groups} \in setof[1, players]$$

- Constantes

$weeks$
$groups$
$groupSize$

- Restricciones (para $w \in [1, weeks], g \in [1, groups]$) //groupSize

$$card(schedule_{w,g}) = groupSize$$

2. Ejercicios

- Restricciones (para $w \in [1, weeks], g1 \in [1, groups], g2 \in [g1 + 1, groups]$) //playOncePerWeek

$$card(schedule_{w,g1} \cap schedule_{w,g2}) = \varnothing$$

- Restricciones (para $w1 \in [1, weeks], w2 \in [w1 + 1, weeks], g1 \in [1, groups], g2 \in [1, groups]$) //differentGroups

$$card(schedule_{w1,g1} \cap schedule_{w2,g2}) \leq 1$$