

ACTIVIDAD 4

Resolución de Sudokus mediante PSR

El Sudoku es un rompecabezas matemático cuyo objetivo es rellenar una cuadrícula de $N \times N$ celdas dividida en subcuadrículas de $N \times N$ con las cifras del 1 al N partiendo de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ninguna cifra en una misma fila, columna o subcuadrícula. Sea el siguiente ejemplo para el caso de $N=4$.

	4	3	2
3	2	4	
4			3
2	3		4

1	4	3	2
3	2	4	1
4	1	2	3
2	3	1	4

<http://www.sudoku-online.org>

Se podría realizar una formalización de la siguiente forma (OPCIONAL):

```
function psr=crea_Sudoku_psr
%%% psr es una estructura con los siguientes elementos
%%% asignacion: vector fila cada columna corresponde con el valor asignado
%%% a la variable(i). Inicialmente debe estar a un valor imposible
%%% dominios: matriz donde cada fila representa los valores discretos de
%%% cada variable

N=4; %% Otro tamaño posible de 9 x 9
variables=N*N; %% 1 variable por cada celda
dominios=ones(N*N,1:N); % 1 a N

X=[0 4 3 2
    3 2 4 0
    4 0 0 3
    2 3 0 4];

psr=struct('variables',variables, ...
          'asignacion',asignacion,...
          'dominios', dominios);
```

1. **RESTRICCIONES.** Sea la siguiente formulación de un Sudoku:

En este problema las restricciones deben satisfacer que no se debe repetir ninguna cifra en una misma fila, columna o subcuadrícula. **Completa la función para que contemple todas las posibilidades:**

```
function c=consistenteSudoku(psr,vari,valor)
%% de vari obtener fila y columna
[fil,col]=ind2sub(size(psr.asignacion),vari);
%% Ejemplo de comprobación: si en la misma fila y col no está ya el valor
c= (all(psr.asignacion(fil,~:)==val)) && ...
    ((all(psr.asignacion(:,col)==valor)) && ...
```

2. **BACTRACKING:** Resuelve el problema aplicando el algoritmo de backtracking y comprueba que todo funciona correctamente.
3. **ARCO-CONSISTENCIA:** Añade una función para ir haciendo los dominios de las variables Arco-Consistentes con sus restricciones al inicio de la resolución y en cada paso de ejecución.
4. **PRUEBA FINAL:** Prueba final del código con el siguiente Sudoku 9x9:

0	0	2	0	0	0	0	0	0	0	9	1	2	5	7	3	8	6	4
3	7	6	1	0	0	0	2	0	0	3	7	6	1	8	4	5	2	9
8	0	0	0	6	9	0	3	0	0	8	5	4	2	6	9	7	3	1
0	0	8	0	0	2	0	4	5	0	7	6	8	3	1	2	9	4	5
0	4	3	0	5	0	2	0	6	1	4	3	9	5	8	2	7	6	0
0	0	0	0	0	0	0	0	0	5	2	9	6	4	7	3	1	8	0
4	0	0	7	3	0	6	0	0	0	4	8	5	7	3	1	6	9	2
0	0	0	8	0	0	0	5	0	2	3	1	8	9	6	4	5	7	0
6	0	7	0	2	5	0	8	3	6	9	7	4	2	5	1	8	3	0