

Futoshiki (不等式)

Futoshiki, es un puzzle lógico originario de Japón, su nombre significa desigualdad.

Se juega en una matriz cuadrada, en este caso de 9X9. Su objetivo es completar la matriz con números de 1 al 9 de tal manera que cada fila y columna contenga solo uno de cada dígito.

Para cada acertijo, inicialmente se proporcionan algunos dígitos posicionados aleatoriamente en la matriz. Y en el caso de las desigualdades, estas ya se encuentran especificadas al inicio entre algunas celdas de forma que una celda debe ser mayor o menor a su celda vecina.

En este caso también se implementa la restricción que no se deben repetir los dígitos del 1 al 9 en los cuadrados de 3X3 de la matriz.

Búsqueda de solución.

Se encuentra una solución al acertijo, fácilmente utilizando restricciones de la librería CLP(FD).

Primeramente, se define y se verifica que cada fila sea de longitud 9 y también que tengan elementos distintos.

Se realiza un **transpose** para verificar que los elementos también sean distintos viéndolos de manera vertical.

Para aplicar la restricción de no repetición en bloques, se divide la lista de filas y se coloca cada una en una variable distinta (*As, Bs, Cs, etc*) y se llama al predicado bloques donde inicialmente se tienen listas vacías, luego en el caso de **bloques([N1,N2,N3|Ns1], [N4,N5,N6|Ns2], [N7,N8,N9|Ns3])**, se verifica que N1, N2, N3, N4, N5, N6, etc, sean todos distintos, entonces, se llama recursivamente usando como parámetro las listas restantes (sin los primeros 3 elementos).

Entonces, los bloques se llamarán primero con 3 filas de 9 elementos, verificará que los primeros 3 de cada fila sean distintos y se llamará a si mismo con 3 listas de 6 elementos, se verifica nuevamente y se vuelve a llamar con 3 listas de 3 elementos para luego finalmente llamar con 3 listas vacías.

Predicados importantes.

Entre los predicados más importantes, se encuentran las restricciones, las cuales se mencionaron en la búsqueda de soluciones, estas restricciones se encuentran en el predicado **futoshiki**.

Además de las restricciones se tienen los predicados donde se cargan las pistas, los cuales son **cargarPist**, donde se llaman a los predicados de carga de pistas, **cargarPistas**, para las pistas numéricas, **cargarPistas2** para las pistas horizontales y **cargarPistas3** para las pistas verticales.

En **cargarPistas** se utiliza un findall, donde se le da un formato P a las pistas para enviarlas en una lista y luego cargarlas de manera recursiva al tablero. Esta carga se realiza llamando al predicado **callunequals**, donde llama al predicado **unequal** que a su vez con el predicado **celda** se especifica un patrón de restricción para que la carga sea exitosa en la matriz la cual es una lista de listas, este mismo procedimiento se aplica a **cargarPistas2**, la cual se utiliza para la carga de pistas horizontales.

Finalmente, en **cargarPistas3**, primeramente, como este predicado se llama en **cargarPist**, se realiza la transpuesta debido a que en las pistas verticales las columnas son las que coinciden y de esa forma al pasar las columnas como filas, se puede volver a seguir el patrón que sigue cargarPistas2.

Ejemplo utilizado.

El siguiente ejemplo, es el utilizado para la prueba, el tablero tiene las pistas iniciales como también las desigualdades, las cuales también son pistas iniciales.

Estas pistas se ven reflejadas en el archivo **pistas.pl**, el cual está incluido en el **futoshiki.pl**.

Lo único necesario para la ejecución del mismo, es escribir en la consola **resolver(X)**.

Las imágenes a continuación.



Fig 1: Tablero inicial.



Fig 2: Solución.

Referencias y Bibliografía.

Archivo base para futoshiki.

<https://www.swi-prolog.org/pldoc/man?section=clpfd-sudoku>

Ayuda para generación de ejemplo y base para imagen de tablero.

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

<https://es.goobix.com/juegos-en-linea/futoshiki/>

<https://en.wikipedia.org/wiki/Futoshiki>