

Informe de Programación Declarativa Sudoku Nonominó

Roberto Zahuis Benítez
C-411

May 27, 2019

1 Estructuras

Para la realización del algoritmo fue necesario crear 2 estructuras auxiliares:

1. Point: Es la que almacena cada punto de los nonominos, los cuales pueden tener, o no, valores. En caso negativo, este tendrá valor 0.
2. Nonomino: La cual tiene la representación del nonominó. Está conformada por 9 puntos y un color, el cual es necesario para diferenciar las piezas.

Para definir el tablero de sudoku se utiliza una lista de listas, para poder llevar a cabo mejor el algoritmo de resolución de este.

2 Algoritmos

Para resolver el problema, primeramente se organizan los Nonominos de forma tal que generen un tablero válido. Después para resolver el sudoku, se computa por cada celda, los posibles números que esta puede contener, los cuales se mantienen consistentes en todo momento de la ejecución del algoritmo. Una vez posicionados los nonominos, computado los posibles números en cada casilla, se utiliza la recursividad para en cada casilla, ir seleccionando los números posibles y seguir resolviendo el sudoku con el tablero nuevo y la lista de posibles números por casilla.

2.1 Organizar nonominos

Para organizar los nonominos se calculan las permutaciones de estos y por cada una se prueban cuales son las válidas. Para esto, por cada nonomino, se busca en el sudoku la primera casilla libre más arriba a la izquierda, se chequea si se puede poner el nonomino (mirando posición a posición si no está ocupada en el sudoku), en caso afirmativo, se coloca y se calcula con los nonominos restantes en el sudoku resultante. Una vez ubicada todos los nonominos, se chequea que el sudoku sea válido, o sea, que no halla repetición de números tanto en filas como en columnas.

2.2 Cálculo de números posibles por casilla

Para realizar este precomputo, en cada celda, se computan los posibles números por fila, columna y sección de nonomino y se intersectan.

Por fila, al ser una lista de listas, se filtran los elementos de la fila que sean mayores que 0 y se les halla la diferencia de la lista de los números del 1 al 9 con el resultado de este filtro.

Por columna, se halla la traspuesta de la matriz y se calcula por fila.

Por nonomino, se busca el nonomino al cual pertenece la casilla ubicada en la posición (x,y), y se le hace el mismo tratamiento que para calcular por fila.

2.3 Consistencia de números posibles

Para mantener la consistencia, cada vez que se va a posicionar un número para entrar en el caso recursivo, se elimina este de la lista de los posibles de su propia fila, columna y de las casillas de su sección.