

Sudoku Nonominó en Haskell

Alexander A. González Fertel C-412

a.fertel@estudiantes.matcom.uh.cu

Nonominó

Los nonominos se representan como un tipo *Record* con dos propiedades: **nid**, que se utiliza a la hora de discernir un nonominó de otro y **points**, que es una lista de tuplas donde el primer elemento es un entero representando qué dígito hay en la casilla representada por el segundo elemento que es una tupla x, y . La idea seguida es que dicha lista de 'puntos' representa desplazamientos con respecto a una casilla del tablero, dichos desplazamientos comienzan por la tupla 'raíz', es decir, el inicio de la lista de puntos debe ser el punto con desplazamiento $(0, 0)$, que además debe corresponder a la casilla más arriba a la izquierda, se implementó una función que dado un nonominó y un punto x, y , devuelve un nuevo nonominó con el desplazamiento dado a partir de este punto.

Sudoku

El sudoku esta representado por un tipo *Board*, que es simplemente una lista de *Square*. El tipo *Square* está representado por una fila, una columna, un identificador de nonomino (**nid**) y un *Either [Int] Int*, respectivamente:

- `row :: Int`
- `col :: Int`
- `nono :: MaybeInt`
- `status :: Either[Int]Int`

La propiedad **nono** es de tipo *Maybe Int* ya que inicialmente, las casillas del tablero no tienen asociadas un nonominó; el entero es el identificador del nonominó. La propiedad **status** define en que 'estado' está dicha casilla: si la casilla tiene un valor definido, entonces es *Right Int*, si no, entonces es una lista de posibles dígitos **Left [Int]**.

Principales Ideas

Ensamblado

Para ensamblar los nonominós se toman todas las permutaciones de la lista que contiene los naturales del uno al ocho y se intentan colocar los nonominós, que tienen un orden predefinido, como dicta dicha permutación. Para colocar el nonominó correspondiente, se busca la casilla más arriba a la izquierda que no está ocupada todavía por un nonominó y se intenta colocar dicho nonominó ahí.

Resolver

Luego de ensamblados los nonominós, queda como resultado un tablero de sudoku donde a diferencia del sudoku clásico, en vez de tener cuadraditos de 3×3 , contamos con nonominós, por tanto, se pudiera proceder de manera análoga a como se procedería en el clásico, solo que ahora, no pueden ocurrir dos dígitos dentro de un mismo nonominó. La solución es ordenar de forma no decreciente por longitud de las posibilidades a las casillas del tablero, de forma tal que siempre se traten de escribir primero los dígitos

de las casillas en las que solo hay una posibilidad, reduciendo notablemente la complejidad temporal del algoritmo. Por lo demás, es un backtrack bastante simple.

Función de inicio

La función de inicio es *main :: IO()*, que se encuentra en el archivo *Main.hs* y se puede encontrar implementada con 1 ejemplo de uso, para la prueba de diferentes nonominós, simplemente se cambia la lista en el archivo *Test.hs* y se vuelve a ejecutar la función. Esta llama a la función que ensambla la lista de nonominós y luego llama a la función que resuelve el sudoku resultante. En la salida se muestran los nonominós ensamblados, el tablero sin resolver y el tablero resuelto.

Notas

Se hace uso de las mónadas `[]` (lista) y **IO**. Se provee de comentarios en todas las partes importantes del código. Se intenta mostrar en consola con colores, pero por alguna extraña razón muestra caracteres desconocidos.