SIA- TP1-

GRUPO7: Juan Pablo Rey, Santiago Camisay, Fernando Bejarano.

Contents

Problema asignado: CalcuDoku

El juego Calcudoku es una versión modificada del juego Sudoku. En el mismo se cuenta con un tablero de N por N casilleros. Dentro de este juego se tiene grupos, los cuales son un conjuntos de casilleros junto con una operación matemática que, aplicada a dichos casilleros, debe dar un resultado definido por el mismo.

Además, también incluye las reglas propias del Sudoku original:

- los números con los que se puede completar el tablero son del 1 al N inclusive.
- no se puede repetir el mismo número por fila ni columna.

Heuristicas (función h):

Admisibles

h1

H2

H3

No admisibles

Estructuras de datos implementados en el problema

Tablero (Board.java)

Contiene los grupos, los valores de los casilleros y el n. Los valores están implementados en un BitSet para poder manipular los datos lo más rápido posible.

Estructura de un estado (CalcudokuState)

- la matriz de bits que representa el tablero.
- lista que contiene la representación en bits(almacenada en bitset) de cada numero que se puede emplear.
- lista de grupos. Las misma es final y no se copia cuando se copian estados (se comparte, no tiene sentido copiarlos por deep copy): ### Estructura de un grupo
- result
- operador (MINUS,PLUS,MULTIPLY,DIVIDE,IDENTITY); “Identity” es simplemente el valor resultado.Este valor es un enum
- Lista de posiciones (i,j) de casilleros que forman el grupo.

Llenado del tablero

Se llena todo el tablero de la siguiente manera: por cada fila se toman números del 1 al N inclusive y luego de los reordena en forma aleatoria. De esta manera se logra cumplir la restricción de que cada fila no tenga repetidos; el objetivo de esto es simplificar el problema y disminuir la cantidad de estados a analizar por los distintos algoritmos.

ACEPTABLE POR LA CATEDRA. Luego swapeos por fila solo.

Reglas

Se swapea cualquiera con cualquiera.s Las reglas se crean una sola vez al comienzo.Se descartan reglas simetricas.

Parser

Para generar el tablero inicial, parsea un string y devuelve el estado inicial. El estado inicial tiene el bitset en cero todo.

Algoritmos de búsqueda

desinformadas (no usan la función heuristica).

*bfs
*dfs

informadas (usan la funcion heuristica).

* A*.[listo.]
* iterativo (IDDFS): en gpsengine.getComparator hacer lo mismo que en dfs. Luego en GPSSolut

Conclusiones

Pruebas de rendimiento

Para realizar las pruebas de rendimiento que se encuentran a continuación se tomaron 3 ejemplos de tableros para cada N, y se realizaron 5 corridas por cada combinación de estrategia de búsqueda con cada una de las heurísticas (para el caso de IDDFS, BFS, y DFS no se emplea heurística según se vio en clase).

Bibliografía.

- http://www.conceptispuzzles.com/index.aspx?uri=puzzle/calcudoku/rules
- https://es.wikipedia.org/wiki/Kenken