Instituto Tecnológico de Costa Rica

Sede San Carlos

II Proyecto programado: Estrategias para la solución de un Sudoku

Análisis de Algoritmos

Alumnos:

Xavier Blanco Arias

Brandon Chaves Villegas

Profesora: Ana Lorena Valerio

Descripción del Problema:

Implementar distintas estrategias de algoritmos para la resolución de el popular puzzle Sudoku de tamaño 9x9 y analizar de manera empírica y analítica los resultados obtenidos por cada de estas estrategias y así determinar cuál es la más eficiente en términos de tiempo y uso de memoria.

Solución del Problema:

La primera estrategia utilizada para la resolución del sudoku fue la del Backtracking, en este caso recorremos la matriz de 9x9 avanzando celda por celda, cada vez que nos posicionamos en una celda se le hace una llamada a una función que verifica si el numero 1 puede ser colocado en ese lugar cumpliendo con las reglas básicas del sudoku, de no ser posible se seguirá intentando colocar otro numero aumentando hasta llegar al número máximo, el numero 9, si en algún momento en determinada celda no se puede colocar ningún número del 1 al 9 es donde entra el juego el backtracking, pues nos devolveremos a la celda anterior e intentaremos colocar en ella un número más alto que el que ya tenía (siempre prefiriendo el numero más bajo posible) y a partir de este punto se repetirá la secuencia reemplazando números colocados anteriormente hasta haber llenado exitosamente todas las celdas del sudoku.

Lógica del algoritmo mediante seudocódigo:

funcionBacktracking(fila, columna,matriz,numero){

for i=0; i<9; i++ {

for x=0;x<9;x++ {

while numero<10:

if numero cumpleLeyes en matriz[fila,columna]:

matriz[fila,columna]=numero;

numero=1;

break;

else:

numero+1;

if ningun numero colocado:

funcionBacktracking(fila,columna-1,matriz,matriz[fila,columna]+1)

}

}

}

Otra de las estrategias utilizadas para la resolucion del problema, fue utilizar un algoritmo probabilistico, el cual no utiliza nigun tipo de razonamiento logico humano ni Backtracking, sino que funciona realizando suposiciones bien fundamentadas, funciona de manera que el puzzle del sudoku es representado por 729 probabilidades distintas, una para cada posible digito del 1 al 9 en cada celda posible de la matriz de 9x9 (81 celdas en total).

El algoritmo inicializa las probabilidades basandose en la cantidad y valor de los numeros iniciales proporcionados e inicia un ciclo en el que elije el valor con la mejor probabilidad para determinada celda, una vez colocado un valor este es definitivo y no se cambiara mas.

Lógica del algoritmo mediante seudocódigo:

funcionSudokuProb(int[,] SudokuInicial, numeroCeldasAResolver){

inicializarProbabilidades(SudokuInicial);

numerodeCeldasResueltas=0;

while numerodeCeldasResueltas<NumeroCeldasAResolver:

fijarNumeroenCelda();

}

La última estrategia utilizar para resolver el sudoku fue la estrategia de modo genético, estrategia que trata de hacer posibles combinaciones entre las distintas filas para ir generando nuevas resultados posibles para la solución del sudoku, luego de realizar las combinaciones los espacios que quedaron vacíos deben ser rellenados con los números correctos utilizando la mutación, la mutación se hace de forma arbitraria, o sea, se deben probar números del 1 al 9 en cada casilla y el número que cumpla las normas del juego será el número elegido para quedarse en dicha posición.

Lógica del algoritmo utilizando pseudocódigo:

funcionSudokuGenetico(int[,] sudokuInicial){

for(int i=0;i<9;i++){

seleccionarMejorCombinacion(fila i, fila i+1)

fila i= mejorCombinacion();

}

seleccionarMejorCombinacion(fila 7, fila 8)

fila 8= mejorCombinacion();

mutarEspaciosenCero();

Análisis de Resultados:

El proyecto fue concluido de manera satisfactoria, completando exitosamente todas las distintas estrategias utilizadas y utilizando las distintas cantidades de números iniciales predeterminados, a excepción del algoritmo genético el cual no logra resolver en su totalidad algunos sudokus con cierta cantidad de números iniciales, sin embargo este algoritmo resuelve el sudoku hasta cierto punto y deja solo algunos espacios vacios en el puzzle.

Analizando las distintas mediciones realizadas a los algoritmos (adjuntadas en documento aparte, podemos concluir que la estrategia de Backtracking es más eficiente en términos de asignaciones y comparaciones realizadas, por lo tanto mejor manejo de recursos, mientras que la estrategia probabilística puede llegar a tener mejor tiempo de ejecución con ciertas cantidades de números iniciales, además la cantidad y posición de estos números es generada de manera aleatoria, por ende esto puede variar dado la complejidad del puzzle.

Conclusiones y Recomendaciones:

El proyecto fue concluido de manera satisfactoria, obteniendo nuevos conocimientos sobre la utilización de útiles algoritmos de programación como el backtracking y otros poco convencionales como lo son los algoritmos genéticos y probabilísticos.

El proyecto en general deja una agradable sensación pues fue concluido de manera exitosa y podemos asegurar que hemos mejorado nuestros conocimientos y habilidades en el arte de la programación.

# Bibliografía

Bello, L. (s.f.). *lucianobello.com*. Recuperado el 21 de Noviembre de 2013, de http://www.lucianobello.com.ar/sudoku/sudoku.pdf

*DotNetPerls*. (s.f.). Recuperado el 21 de Noviembre de 2013, de http://www.dotnetperls.com/sort-dictionary

Gottlieb, M. A. (s.f.). *Feynman Lectures*. Recuperado el 18 de Noviembre de 2013, de http://www.feynmanlectures.info/sudoku/pss.html

InteractivePython. (s.f.). *InteractivePython*. Recuperado el 18 de Noviembre de 2012, de http://interactivepython.org/courselib/static/pythonds/AlgorithmAnalysis/analysis.html

NeoSeeker. (s.f.). *NeoSeeker*. Recuperado el 19 de Noviembre de 2013, de http://www.neoseeker.com/forums/86/t1755394-in-how-you-output-contents-of-dictionary-class/

StackOverflow. (s.f.). *StackOverflow*. Recuperado el 20 de Noviembre de 2013, de http://stackoverflow.com/questions/5096414/recursive-backtracking

Youtube. (s.f.). *Youtube*. Recuperado el 10 de Noviembre de 2013, de Youtube: http://www.youtube.com/watch?v=-hZAnscUViE