29 de octubre de 2019

Pedro Tamargo allué - 758267

Memoria práctica 1

inteligencia artificial 2019-2020

Contenido

[Problema de los Caníbales y los Misioneros 2](#_Toc22662933)

[Comparación de ejecución del problema 8-puzzle con distintos algoritmos de búsqueda 3](#_Toc22662934)

[Tabla comparativa de tiempos entre distintos algoritmos 3](#_Toc22662935)

[Explicación de cada algoritmo 4](#_Toc22662936)

[Definición del problema del 15-puzzle 5](#_Toc22662937)

# Problema de los Caníbales y los Misioneros

El problema de los caníbales y los misioneros consiste en trasladar 3 caníbales y 3 misioneros de una orilla del río a otra, contando con la restricción de que no pueden producirse estados donde el número de caníbales de una orilla es mayor que el de los misioneros de esa orilla.

Para la resolución del problema se plantea el siguiente estado:

Dónde, es el número de caníbales en la orilla del río , es el número de misioneros en la orilla del río y es la posición del bote en el estado, si , en caso contrario, el bote se encuentra en la orilla derecha.

Por lo tanto, el estado inicial sería:

Para definir las transiciones posibles se plantea el siguiente predicado auxiliar:

Siendo el estado resultante de realizar el movimiento, y el predicado que indica si un estado es peligroso o no, en base a la restricción planteada por el problema.

Los movimientos posibles para este problema serían:

Por lo tanto, se definen las condiciones para poder realizar un movimiento de la siguiente manera:

Siendo cierto si , es decir, si el bote está a en la orilla izquierda, el estado que se alcanzaría después de mover un caníbal de la orilla izquierda a la derecha y el estado que se alcanzaría después de mover un caníbal de la orilla derecha a la izquierda.

Sería análogo para el resto de movimientos posibles .

El estado final para este problema sería que todos los caníbales, los misioneros y el bote estuvieran en la orilla derecha .

# Comparación de ejecución del problema 8-puzzle con distintos algoritmos de búsqueda

### Tabla comparativa de tiempos entre distintos algoritmos

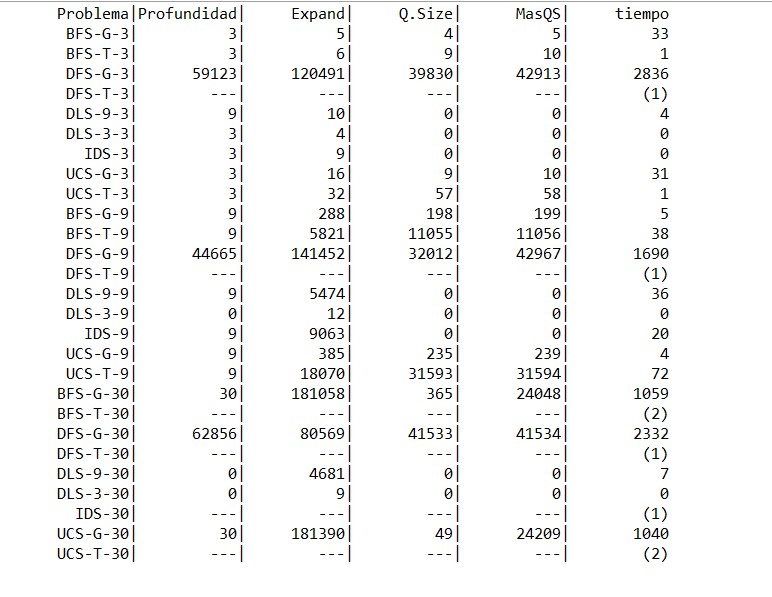


Tabla comparativa entre Profundidad, Nodos Expandidos, Tamaño de la frontera, Máximo tamaño de la frontera y tiempo entre los distintos algoritmos evaluados.

1. Este algoritmo no termina porque el algoritmo ha encontrado un bucle, esto puede porque se trataba de una búsqueda en árbol, ya que no guardan lista de nodos explorados.
2. Este algoritmo no termina porque la máquina virtual de Java se queda sin memoria dinámica para representar los nodos.

### Explicación de cada algoritmo

* **BFS-G-3**
* **BFS-T-3**
* **DFS-G-3**
* **DFS-T-3**
* **DLS-9-3**
* **DLS-3-3**
* **IDS-3**
* **UCS-G-3**
* **UCS-T-3**
* **BFS-G-9**
* **BFS-T-9**
* **DFS-G-9**
* **DFS-T-9**
* **DLS-9-9**
* **DLS-3-9**
* **IDS-9**
* **UCS-G-9**
* **UCS-T-9**
* **BFS-G-30**
* **BFS-T-30**
* **DFS-G-30**
* **DFS-T-30**
* **DLS-9-30**
* **DLS-3-30**
* **IDS-30**
* **UCS-G-30**
* **UCS-T-30**

# Definición del problema del 15-puzzle

El problema del 15-puzzle se define de la misma manera que el del 8-puzzle, con la salvedad de que en este caso serían 14 fichas y 1 hueco en blanco, que se desplazaría de la misma manera que en el 8-puzzle.

El estado inicial es análogo al del 8-puzzle, un conjunto de los 15 primeros números naturales (empezando por el 0), en una lista, de tal manera que cada número positivo es la casilla con dicho número, y el 0 es el hueco, que puede moverse.

Las funciones usadas en el 8-puzzle son similares a las utilizadas en el 15-puzzle, con la diferencia que debemos comparar con 16 valores en lugar que 8.