ÍNDICE

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

OBJETIVO

El objetivo del presente Trabajo Práctico Especial es crear un Sistema de Producción que será usado para resolver el problema asignado, en este caso el juego Deep Trip.
Para esto, fue provisto un motor de inferencia reducido programado en Java, al cual se le hicieron las modificaciones necesarias para completar el trabajo.

1. DESCRIPCIÓN del TRABAJO REALIZADO

Funcionamiento del Juego

El juego DeepTrip consiste en un tablero de c columnas y f filas, lleno de fichas de distintos colores. La finalidad es dejar el tablero vacío. Cada vez que se encuentran tres o más fichas del mismo color contiguas, éstas se consumen (desaparecen) y las fichas que se encuentran más arriba caen hasta ocupar los espacios vacíos. Solamente se puede rotar cualquiera de las filas n lugares (donde $1 \leq n \leq c$)

Implementación del Juego

➔ *Lógica*

Consiste en dos actividades:

- ***Drop****:* Baja las fichas en caso de que encuentre un espacio vacío.
- ***Consume:*** Busca y consume si encuentra 3 o más fichas del mismo color juntas.

Básicamente hace un loop entre Drop y Consume.

➔ *Reglas*

Dado que las reglas se basan en todos los movimientos posibles de un tablero, en total son $f \cdot c$ reglas definidas de la misma manera: rotar una determinada fila n veces. Sin embargo, dado que no todas las reglas son válidas en cada jugada (pues, por ejemplo, no tiene sentido rotar una fila vacía), cuando una regla no puede aplicarse, se lanza una excepción.

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

➔ *Función de costo*

Dado que ningún movimiento es más costoso que otro, la función de costo de cada regla es constante: $g(n) = r$, siendo n un nodo cualquiera y r la cantidad de movimientos hecha para llegar al nodo n desde el nodo inicial.

➔ *Heurísticas*

◆ *Heurística Uno:*

Sean n un nodo cualquiera y t la cantidad total de fichas que hay en el tablero de n, se define $h1$ de la siguiente manera:

$h1(n) = \infty$, si existe un color con menos de 3 fichas en n,

$h1(n) = 0$, si n es el tablero objetivo, o sino

$h1(n) = t/8$.

Esta función heurística (codificada en la clase HeuristicOne), se basa en la cantidad mínima de "consumos" que pueden ocurrir (es decir, de fichas contiguas de un mismo color que desaparecen de una sola vez). De un "consumo", el cluster de fichas puede tener como máximo 8 fichas (ver **Figura 1**). Sin embargo, esta situación ocurre en el juego con probabilidades muy bajas, además de depender de la cantidad de fichas que haya de cada color.

Asimismo, el hecho de que retorne como valor "infinito" cuando el tablero tiene un color con menos de 3 fichas se refiere a que puede asegurarse que ese tablero -a partir de ese estado-, no puede llegar a ninguna solución. Y cuando el nodo es igual al tablero objetivo, quiere decir que se alcanzó la solución (y por definición de heurística, debe devolver 0).

De todo esto se deriva que el costo aproximado faltante para llegar a la solución va a ser menor para $h1$ que para $h*$, es decir que:

$h1(n) < h*(n) \ \forall \ n$

Ergo, $h1$ es admisible.

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

- ***Heurística Dos:***

  Sean n un nodo cualquiera y c1,c2,...,cf la cantidad de fichas de cada color (habiendo f colores en el tablero) se define $h2$ tal como se expresa a continuación:

  $h2(n) = \infty$, si existe un color con menos de 3 fichas en n,

  $h2(n) = 0$, si n es el tablero objetivo, o

  $$h2(n) = \sum_{i=1}^{f} s(ci) \ ;$$

  donde $s(c)$ es la resolución del problema de la moneda para el conjunto de números 3,4 y 5.

  Esta función heurística (codificada en la clase HeuristicTwo) se basa en identificar para cada color la cantidad mínima de clusters de 3, 4 y 5 fichas que se pueden formar, y luego la suma de todas esas cantidades da valor a la heurística.

  Los primeros dos casos de la función tienen la misma explicación que en la heurística anterior.

  Se concluye que, existen casos en los que esta heurística indica una cantidad mayor de movimientos a la real, por lo que no es admisible (aunque no deja de ser útil).

- ***Heurística Tres:***

  Sean n un nodo cualquiera y c1,c2,...,cf la cantidad de fichas de cada color (habiendo f colores en el tablero) se define $h3$, como se detalla seguidamente:

  $h3(n) = \infty$, si existe un color con menos de 3 fichas en n,

  $h3(n) = 0$, si n es el tablero objetivo, o

  $$h3(n) = \sum_{i=1}^{f} \frac{1}{ci} \sum_{k=1}^{ci} |xk - mi)| \ ;$$

  donde mi es el "centro de masa" de los puntos con un determinado color.

  Esta función heurística (codificada en la clase HeuristicThree) , se basa en el hecho de que cuanto más cerca estén las piezas unas de otras, menos movimientos se van a necesitar para agruparlas.

Para los primeros dos casos de la función, resulta válido lo dado a conocer para la heurística uno.

2. ANÁLISIS de los RESULTADOS OBTENIDOS. CONCLUSIONES

Para realizar las mediciones, se utilizaron 2 tamaños diferentes de tablero:

- ➔ ***Chico*** **(de 3x3)**, y
- ➔ ***Medio*** **(de 4x4).**

A su vez, para cada uno de ellos se evaluaron tres tableros.

Los resultados obtenidos de dichas evaluaciones, se encuentran expresados en el **ANEXO** de esta presentación.

Aún cuando los resultados dependen mucho de los tableros usados (los cuales, por cuestiones de tiempo son bastante chicos), se puede arribar a ciertas conclusiones:

1) El algoritmo DFS es aquel que tarda más en encontrar una respuesta, independientemente de que el tablero tenga o no solución. Esto tiene sentido, puesto que para hallar una respuesta recorre el árbol de nodos en profundidad (incluyendo aquellos nodos que no tienen solución).

2) El algoritmo BFS es mucho más rápido que el anterior (e incluso, a veces, más aún que los algoritmos informados). Eso sucede debido a que recorre el árbol por niveles, y cuando ve una solución, la retorna de inmediato. En los tableros que se usaron, las soluciones no requieren muchos pasos (4, a lo sumo 5 movimientos), y por eso pueden llegar a ser más eficientes que los algoritmos informados (que requieren más procesamiento). No obstante, este algoritmo requiere mucha cantidad de memoria (que con tableros muy grandes puede producir excepciones por falta de memoria).

3) El algoritmo IDDFS, al combinar ventajas de DFS y BFS, es rápido como BFS si el tablero tiene solución, pero es más lento que DFS si no tiene solución (debido a que, como hace muchos árboles DFS de distintas alturas, va a tener que llegar hasta la profundidad máxima antes de

decir que no hay solución al tablero).

4) Comparando los algoritmos A* y Greedy entre sí, puede decirse que ambos son más o igual de eficientes que los desinformados (DFS, BFS y IDDFS). Sin embargo, A* tiende a ser más eficiente y rápido sin importar el tipo de tablero (hay que considerar que se usaron tableros bastante sencillos de prueba, y por eso Greedy a algunos de ellos los resolvía casi instantáneamente). Sin embargo, en ciertos casos, los resultados hechos con Greedy tendían a valores mucho más altos que A*. Eso se debe a su naturaleza recursiva y de backtracking, heredada de DFS.

5) Con respecto a las heurísticas, cualquiera de las tres es útil (en el sentido de que ayudan a encontrar una respuesta al problema), aunque las heurísticas 2 y 3 tienden a ser más precisas que la heurística 1 (lo que hace que, en ciertos tablero, con aquellas se recorran menos nodos que con la última). Sin embargo, hay que tener en cuenta que las heurísticas 1 y 2 son más fáciles de calcular que la 3, y eso hace que el tiempo de procesamiento por nodo sea menor.

ANEXO

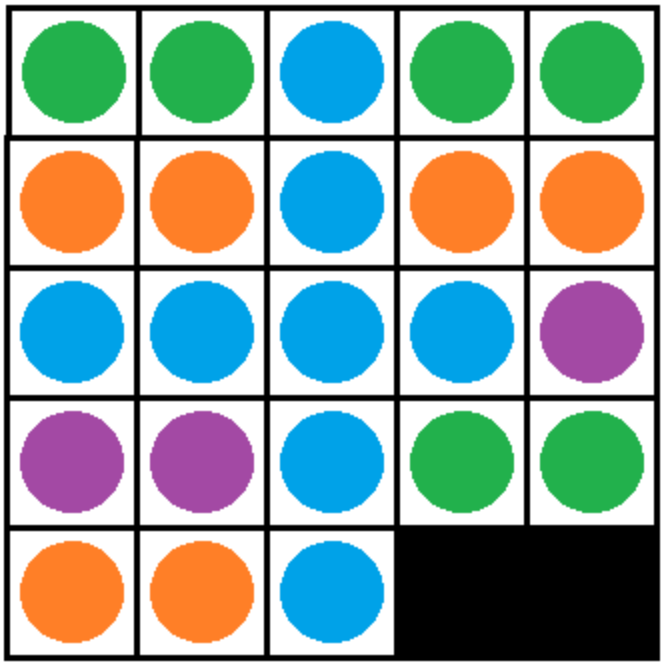

Figura 1

TABLEROS CHICOS (3x3)

Tablero 1

1	2	3
3	4	1
2	3	1

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	2251	614		2330	706	126	1	2330	1
Nodos en frontera	0	0		0	0	0	0	0	0
Nodos generados	2251	614		2330	706	126	1	2330	1
Profundidad	-	-	-	-	-	-	-	-	-
Tiempo de procesamiento	1783 ms	706 ms	∞	2288	1019 ms	187 ms	6 ms	2077 ms	7 ms

Tablero 2

1	**2**	**3**
3	**2**	**1**
2	**3**	**1**

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	39	11	9	94	3	3	10	65	65
Nodos en frontera	13	33	7	8	15	15	15	15	15
Nodos generados	52	44	16	102	18	18	25	80	80
Profundidad	4	2	2	2	2	2	8	12	12
Tiempo de procesamiento	90 ms	33 ms	56 ms	159 ms	18 ms	18 ms	48 ms	179 ms	207 ms

Tablero 3

1	**2**	**1**
3	**2**	**3**
2	**3**	**1**

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	5	12	18	57	4	4	3	5	5
Nodos en frontera	18	27	4	4	15	15	7	7	5
Nodos generados	23	39	22	61	19	19	10	12	10
Profundidad	5	2	2	2	2	2	3	3	2
Tiempo de procesamiento	13 ms	31 ms	66 ms	122 ms	23 ms	25 ms	13 ms	14 ms	14 ms

TABLEROS MEDIOS (4x4)

Tablero 1

1	**2**	**3**	**4**
3	**4**	**1**	**2**
2	**3**	**1**	**4**
1	**1**	**2**	**3**

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	21416	1994		2074	2074	2525	16056	16056	9792
Nodos en frontera	0	0		0	0	0	0	0	0
Nodos generados	21416	1994		2074	2074	2525	16056	16056	9792
Profundidad	-	-	-	-	-	-	-	-	-
Tiempo de procesamiento	163989 ms	1403 ms	∞	1852 ms	1594 ms	3133 ms	205784 ms	204112 ms	67217 ms

Tablero 2

1	**2**	**3**	**4**
3	**4**	**1**	**2**
1	**2**	**3**	**4**
1	**2**	**3**	**4**

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	34997	5	8	1	14	1	1	33339	1
Nodos en frontera	4	43	4	11	39	11	11	7	11
Nodos generados	35001	48	12	12	53	12	12	33346	12
Profundidad	1	1	1	1	1	1	1	1	1
Tiempo de procesamiento	911152 ms	31 ms	73 ms	13 ms	70 ms	20 ms	17 ms	893256 ms	25 ms

Tablero 3

1	**2**	**3**	**4**
3	**4**	**1**	**2**
1	**6**	**3**	**4**
1	**2**	**3**	**4**

Grupo 1:
DOMINGUES, Matias (Leg. 50278)
FONTANELLA DE SANTIS, Teresa (Leg. 52455)
MARTINEZ CORREA, Facundo (Leg. 49139)

	DFS	BFS	IDDFS	A*	A*	A*	GS	GS	GS
Heurística	-	-	-	h1	h2	h3	h1	h2	h3
Nodos expandidos	43620	2350		2350	2350	2350	1	1	1
Nodos en frontera	0	0		0	0	0	0	0	0
Nodos generados	43620	2350		2350	2350	2350	1	1	1
Profundidad	-	-	-	-	-	-	-	-	-
Tiempo de procesamiento	131334 0 ms	1817 ms	∞	2561 ms	2576 ms	5168 ms	13 ms	9 ms	9 ms