ComS 472
Homework 1

Sean Gordon

Sep 11, 2020

- 3.2 -

(a) **States:** Glass box holds key or banana
Initial State: Hold key to box 1, outside box 1
Goal Test: Holding banana
Actions: Open next box using key, move into next box, get contents of current box
Transition Model: A box is opened, a box's contents are obtained
Cost Function: Each action costs 1 unit

(b) **States:** A sequence of characters
Initial State: Sequence = ABABAECCEC
Goal Test: Sequence = E
Actions: AC = E, AB = BC, BB = E, and Ex = x for any x
Transition Model: Transform two adjacent characters using an action
Cost Function: Each transformation costs 1 unit

(c) **States:** Square is painted, unpainted, or is a bottomless pit
Initial State: All squares unpainted or bottomless, Standing on arbitrary square
Goal Test: All non-bottomless squares painted
Actions: Paint current square, move to adjacent non-bottomless unpainted square
Transition Model: A square is painted, a square is moved to
Cost Function: Each action costs 1 unit

(d) **States:** 3D grid position contains a shipping container or is empty.
Initial State: Each grid space contains a shipping container, crane at (0, 0)
Goal Test: Each grid space is empty
Actions: Move crane to (x,y) coordinate, pick up container under crane coordinate, drop container on dock
Transition Model: Crane is moved to (x,y), crane picks up container,
crane drops container on dock
Cost Function: Each action costs 1 unit

- 3.4 -

(a) **States:** Square is red or blue
Initial State: All squares blue
Goal Test: Each 3x3 sub square is of a differing color than it's neighbors
Actions: Color square, move to next square
Transition Model: Square is colored, moved to next square
Cost Function: Each action costs 1 unit

State space = $(\#states)^{\#squares} = (2)^{81} =$ 2,417,851,639,229,258,349,412,352
(Or is it infinite? The wording has me confused)

(b) **States:** Square is red or blue
Initial State: All squares blue
Goal Test: Each 3x3 sub square is of a differing color than it's neighbors
Actions: Color square and move to next square, move to next square
Transition Model: Square is colored and moved to next square,
moved to next square
Cost Function: Each action costs 1 unit

State space = $(\#states)^{\#squares} = (2)^{81} =$ 2,417,851,639,229,258,349,412,352

BFS would indeed perform faster as the new constraint would remove cycles. Iterative deepening would also perform faster for the same reason.

(c) **States:** Subsquare is red or blue
Initial State: All subsquares blue
Goal Test: Each subsquare is of a differing 7color than it's neighbors
Actions: Color entire subsquare and move to next subsquare,
move to next subsquare
Transition Model: Subsquare is colored and moved to next subsquare,
moved to next subsquare
Cost Function: Each action costs 1 unit

State space = $(\#states)^{\#squares} = (2)^{9} = 512$

(d) 2 solutions

(e) i. C $->$ B: Break each subsquare into 9 squares in a 3x3
ii. B $->$ A: Problems A and B have the same solutions

- 3.7 -

A given queen can attack at most 3 spaces in a given column.
Therefore, when placing the next queen, there are at least n-3q valid spaces,
where q is the total number of queens already placed.
Counting the total number of states: n(n-3)(n-6)(n-9)...(n-3c)
With some algebra, this can be simplified into $(n!)^{1/3}$

Exhaustive search becomes infeasible when n > 33

- 3.17 -

(a) False. The search goal may be at the far bottom left of the search tree, of which DFS will head towards immediately. A* must first go through the surrounding nodes of the starting point.

(b) True. The search heuristic is required to never overestimate the cost. This heuristic is almost guaranteed to underestimate. It will be ineffective, but it will work.

(c) False. A* is used commonly in image recognition.

(d) True. BFS will find a goal if it exists, and is only concerned with the goal with the shortest depth. As this is unaffected by step cost, BFS remains complete.

(e) False. As Manhattan distance requires only one square moved at a time, and a rook can move multiple squares at a time if given space, Manhattan distance may overestimate the cost.

- 3.22 -

If the tree to be searched is a straight line of nodes (A-B-C-D-E-...),
BFS and Iterative Deepening will use around the same amount of memory,
but BFS will not stop on its path to the goal like Iterative Deepening will.

- 3.27 -

Current	Adding	g(n)	h(n)	f(n)	Open Nodes
Start	Lugoj	0	244	244	(Lugo, 244)
Lugoj	Timisoara	111	329	440	(Timi, 440)
Lugoj	Mehadia	70	241	301	(Timi, 440), (Meha, 301)
Mehadia	Dobreta	145	242	387	(Timi, 440), (Dobr, 387)
Dobreta	Craiova	265	160	425	(Timi, 440), (Crai, 425)
Craiova	Rimnicu	411	193	604	(Timi, 440), (Rimn, 604)
Craiova	Pitesti	403	100	503	(Timi, 440), (Rimn, 604), (Pite, 503)
Timisoara	Arad	229	366	595	(Rimn, 604), (Pite, 503), (Arad, 595)
Pitesti	Bucharest	504	0	504	(Rimn, 604), (Arad, 595), (Buch, 504)
Bucharest					(Rimn, 604), (Arad, 595)

- 3.31 -

Complete: $0 < w < 2$
Optimal: $0 < w < 1$
w==0: Uninformed best-first search. **w==1:** A* search. **w==2:** Greedy best-first search