50.021 Artificial Intelligence

Theory Homework Week 2

Lu Jiankun 1002959

1. Problem Formulation

State space:
Triple (x,y,z) where $0 \leqslant x, y \leqslant 3$, z = 0 or 1.
x, y and z represent the number of missionaries, cannibals and boats currently on the original bank.

Initial state:
(3,3,1), which means there are 3 missionaries, 3 cannibals and 1 boat on the original bank.

Goal test:
Reach the goal state (0,0,0), which means all missionaries, cannibals and the boat reach the other bank of the river.

Actions:
From each state, either bring one missionary, one cannibal, two missionaries, two cannibals, or one of each type to the other bank.

Path cost:
One unit per crossing

Note: not all the states are achievable. For example, (0,0,1) is not attainable. Some of the states where cannibals outnumber missionaries on one side of the river are illegal.

2. Tree Search V.S. Graph Search

a) The main difference is that Graph Search avoids re-visiting visited nodes while tree search does not. In Graph search we hold a list of explored nodes while in Tree Search, we do not. Graph Search checks whether a node is in Frontier or Explored before inserting it into Frontier, while tree search inserts nodes without checking. This is because tree does not contain cycles and whereas graph can.

b) A state is a representation of the physical configuration. A node is a data structure constituting part of the search tree or graph. Nodes comprise state, parent-node, child-node(s), action, path-cost, depth. On the other hand, states do not have parents, children, depth or path cost.

c) Explored set keeps track of states. A node is a data structure constituting part of the search tree or graph. Different nodes may contain the same state. Keeping track of states can avoid the algorithm to create redundant nodes as no matter how the search strategy reaches a state, it does not affect whether there exists a solution from this state. So, if a state is reached before, there is no need to explore it again. (In the case of UCS, a new node with smaller path cost will replace old node of the same state but with bigger path cost)

3. Breadth-First Search (BFS) and Depth-First Search (DFS)

a) BFS as a graph search

	Frontier/ Queue	Explored Set
Step 1	A	A
Step 2	AB, AC	A, B, C
Step 3	AC, ABD	A, B, C, D
Step 4	ABD	A, B, C, D
Step 5	ABDX	A, B, C,D,X

Solution: ABDX

b) DFS as a graph search

	Frontier/ Queue	Explored Set
Step 1	A	A
Step 2	AB, AC	A, B, C
Step 3	AB, ACD	A, B, C, D
Step 4	AB, ACDX	A, B, C, D, X

Solution: ACDX

c) BFS as tree search, following nodes will be inserted:
ABC, ACB, ACD.

d) DFS as tree search, following nodes will be inserted:
ACB, ACDB, ACDBC

4. More BFS/DFS

a) BFS as graph search.

	Frontier/ Queue	Explored Set
Step 1	A	A
Step 2	AB, AC	A, B, C
Step 3	AC, ABD	A, B, C, D
Step 4	ABD, ACE, ACF	A, B, C, D, E, F
Step 5	ACE, ACF, ABDH, ABDX	A, B, C, D, E, F, H, X

Solution: ABDX

b) DFS as graph search.

	Frontier/ Queue	Explored Set
Step 1	A	A
Step 2	AB, AC	A, B, C
Step 3	AB, ACE, ACF	A, B, C, E, F
Step 4	AB, ACE, ACFI	A, B, C, E, F, I
Step 5	AB, ACE, ACFIH	A, B, C, E, F, H, I
Step 6	AB, ACE, ACFIHD, ACFIHX	A, B, C, D, E, F, H, I, X

Solution: ACFIHX

5. Uniform Cost Search (USC)

	Frontier/ Queue	Explored Set
Step 1	A *(path cost: 0)*	A
Step 2	AC *(path cost: 3)*, AB *(path cost: 5)*	A, B, C
Step 3	AB *(path cost: 5)*, ACF*(path cost: 8)*, ACE *(path cost: 153)*	A, B, C, E, F
Step 4	ACF*(path cost: 8)*, ABD*(path cost:55)*, ACE *(path cost: 153)*	A, B, C, D, E, F
Step 5	ACFI *(path cost:11)*, ACFE *(path cost: 13)*, ABD*(path cost: 55)*	A, B, C, D, E, F, I
Step 6	ACFE *(path cost: 13)*, ACFIH *(path cost: 14)*, ABD*(path cost: 55)*	A, B, C, D, E, F, H, I
Step 7	ACFIH *(path cost: 14)*, ACFED *(path cost: 15)*	A, B, C, D, E, F, H, I
Step 8	ACFED *(path cost: 15)*, ACFIHX *(path cost: 24)*	A, B, C, D, E, F, H, I, X
Step 9	ACFEDX *(path cost: 16)*, ACFIHX *(path cost: 24)*	A, B, C, D, E, F, H, I, X

Solution: ACFEDX (path cost: 16)