Universität Zürich^UZH

Introduction to Artificial Intelligence
Exercise Sheet 2

Laurin van den Bergh, 16-744-401
Yufeng Xiao, 19-763-663
Nora Beringer, 19-734-227

Universität Zürich
Institut für Informatik

Due: March 9, 2022

Exercise 2.1

farmer = f, wolf = w, goat = g, cabbage = c.
We abstract away all the positions of the boat as the boat is only relevant in combination with the farmer. Also we abstract away the space between the shores, as it is only relevant which "items" are on the left or right shore.

- States: S = $\langle f, w, g, c\rangle$, where $f, c, g, w \in \{L, R\}$; We can ignore the boat as we can assume the boat is always where the farmer is. Note: Farmer is the first position, wolf 2nd, goat 3rd and cabbage 4th
- Initial state: $\langle L, L, L, L\rangle$
- Goal state: $\langle R, R, R, R\rangle$
- Actions: $\{move\_c, move\_g, move\_w, move\_f\}$
- Transition model: T: S x A $\rightarrow$ S $\cup\{\perp\}$; Each action works as follows: A move is characterized by switching the letter of the item that is moved, where we assume that moving each item (w, g, c) also moves the farmer f and $move\_f$ only moves the farmer. If the letter initially is a L then it gets turned into a R, and vice-versa. E.g. $move\_c(\langle L, L, L, L\rangle) = \langle R, L, L, R\rangle$. we find that four states are sink states and therefore actions leading to one of these end the "game". These are: $\langle L, R, \cdot, R\rangle$, $\langle L, \cdot, R, R\rangle$, $\langle R, \cdot, L, L\rangle$, $\langle R, L, \cdot, L\rangle$, where $\cdot \in \{L, R\}$. All actions that get applied on an item i and state s where $f \neq i$ are considered invalid and lead to $\perp$.

Below we show two solution paths:

$$\begin{tikzcd}
& \langle L, L, L, L\rangle \arrow[d] & \\
& \langle R, L, R, L\rangle \arrow[d] & \\
& \langle L, L, R, L\rangle \arrow[dl] \arrow[dr] & \\
\langle R, R, R, L\rangle \arrow[d] & & \langle R, L, R, R\rangle \arrow[d] \\
\langle L, R, L, L\rangle \arrow[dr] & & \langle L, L, L, R\rangle \arrow[dl] \\
& \langle R, R, L, R\rangle \arrow[d] & \\
& \langle L, R, L, R\rangle \arrow[d] & \\
& \langle R, R, R, R\rangle &
\end{tikzcd}$$

Exercise 2.2

a) Incorrect. If the algorithm doesn't take into account of states already visited, then it could visit each state an infinite number of times.

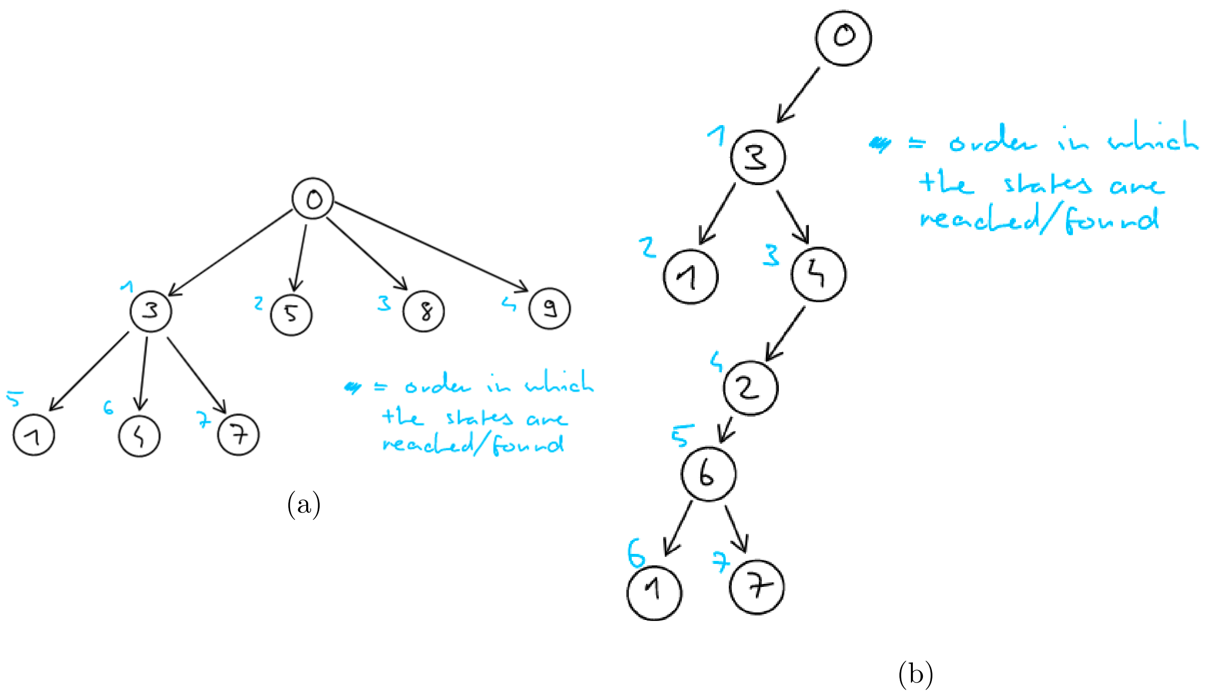
b) False, it isn't complete but it is useful as it uses less memory, implements an acyclic state space and works faster then BFS.

c) True, because BFS with uniform cost is complete and optimal.

d) True, in uniform-cost search, goal test on the node is performed when it is selcted for expansion. Otherwise the first node generated could be a sub-optimal path. Hence with early goal testing the search is cost optimal.

e) Correct as it combines the benefits of BFS and DFS and one can limit the depth.

Exercise 2.3



= order in which the states are reached/found
(a)
= order in which the states are reached/found
(b)

Figure 1: (a) Shows the search tree of the breadth first search algorithm, (b) shows the search tree of the depth first search algorithm. The numbers in blue denote the order in which the states are explored.