INF5390 - Mandatory Assignment 1

Magnus Andersen
University of Oslo, Department of Informatics

March 4, 2013

Contents

Abstract

This report contains my suggestions for the three tasks given in the assignment[1], and deals with *intelligent agents*, *problem-solving through search* and *logical agents*, respectively.

1 Intelligent Agents

I have defined the concepts as follows:

a) An *agent* is something/someone who usually operates alone (autonomous), doing some fairly specialized task (at least when I think of movies/tv shows about (special) agents). In the world of AI, an agent would be something that is capable of receiving input from the environment while also being capable of affecting said environment ('producing output to the environment'), in an autonomous manner.

b) An *agent function* is one (of possibly) several functions, i.e. purposes, the agent should be able to do/fulfill.

c) An *agent program* is a program (software) that implements the purpose, i.e. the function(s) of the agent, while also considering other important factors such as how the agent's performance should be measured etc (the whole package).

d) *Rationality* is related to reason, that is, acting/thinking in a consistently reasonable/«profitable» way given the circumstances. For example, if one were given the choice to make a bet in favor of some random event happening with a probability of less than 0.5, it would not be rational to go through

[1] http://www.uio.no/studier/emner/matnat/ifi/INF5390/v13/undervisningsmateriale/ovinger/inf5390-2013-oving-1.pdf

with the bet, since in >50% of the cases one would lose the stake. In other words, one has not acted profitably/rationally.

e) *Autonomy* is the quality of being self-governed, free from external control/micromanagement.

2 Solving Problems by Searching

Three cannibals, three missionaires and a boat are all situated at the same side of some riverbank. They all want to get to the opposite side of the river, but they are subjected to the following constraints:

- The boat has a minimum carrying capacity of one person (it is not autonomous) and a maximum capacity of two persons.
- At any one place (the boat or either side of the riverbank), if at least one missionary is present, there cannot simultaneously be more cannibals than missionaries (otherwise, the cannibals are thought to overpower and consume their religious counterparts).

Our task is to look at this as a search problem, whereby a solution (some number of successive river crossings satisfying above contraints) can (hopefully) be found through expansion of the search graph (it is a graph, not a tree, because of multiple paths to the same node, see Figure 1 below).

2.1 Problem Specification and State Space Diagram

States

There are many possible ways in which one could define a state. The current state of the system needs to consider three things:

1. Where are the cannibals, and how many?
2. Where are the missionaries, and how many?
3. Where is the boat?

1. and 2. can be expressed as a two-dimensional vector describing the number of missionaries/cannibals at the wrong and right side, but it would be *even simpler* to use a single variable describing the number of cannibals/missionaries at e.g. the right side, since one can easily infer one from the other. Lastly, 3. can be expressed by a single variable describing the boat's location. A state S can thus be described by a three-dimensional vector[2]

$$S = \begin{bmatrix} c & m & b \end{bmatrix}$$

where $c \in \{0, 1, 2, 3\}$ indicates the number of cannibals at the right side, $m \in \{0, 1, 2, 3\}$ indicates the number of missionaries at the right side, and $b \in \{0, 1\}$indicates where the boat is (0 for wrong side, 1 for desired side).

As per the above the initial state can be expressed as

$$S_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

[2]MATLAB-style-ish

Actions

There are two possible moves here:

1. Move 1 or 2 cannibals/missionaries (or 1 cannibal and 1 missionary) from the *wrong* side to the *right* side
2. Move 1 or 2 cannibals/missionaries (or 1 cannibal and 1 missionary) from the *right* side to the *wrong* side

That is, an action of either one of the above three can be either moving C cannibals and/or M missionaries from the wrong side to the right side,

$$\begin{bmatrix} c & m & 0 \end{bmatrix} \rightarrow \begin{bmatrix} (c+C) & (m+M) & 1 \end{bmatrix} \tag{1}$$

or moving C cannibals and/or M missionaries from the right side to the wrong side,

$$\begin{bmatrix} c & m & 1 \end{bmatrix} \rightarrow \begin{bmatrix} (c-C) & (m-M) & 0 \end{bmatrix} \tag{2}$$

where (1) and (2) satisfies $(C, M \in \{0, 1, 2\}) \wedge (1 \leq C + M \leq 2) \wedge ((c \pm C) \leq (m \pm M))$.

Goal test

Check if the current state is $S = \begin{bmatrix} 3 & 3 & 1 \end{bmatrix}$. If it is, all three cannibals/missionaries have arrived safely (i.e. satisfying the constraints) on the other side of the river.

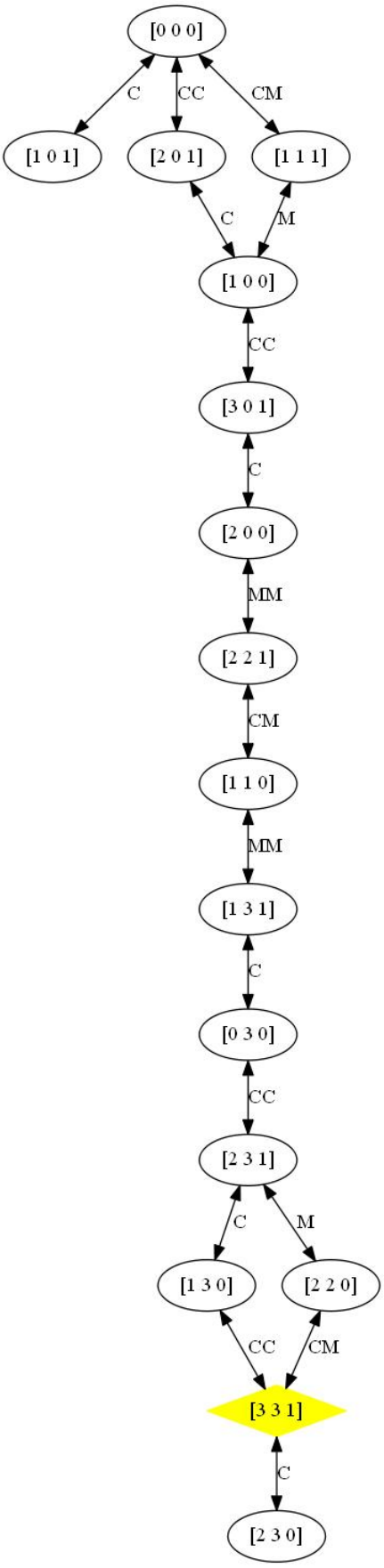
Solution and path cost

A *solution* is any arbitrary number of successive river crossings satisfying the constraints leading to a state that passes the goal test. The *cost* of the solution (i.e. path cost) is the total number of river crossings needed to reach the solution.

State space diagram

The state space diagram is shown in Figure 1 on the next page. $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$is the initial state. The yellow one, $\begin{bmatrix} 3 & 3 & 1 \end{bmatrix}$, is the goal state. A C or M along the edges indicates that one cannibal or one missionary has crossed the river. Similarly, a *CC, MM* or *CM* indicates that two cannibals, two missionaries or one cannibal and one missionary has crossed the river, respectively.

Figure 1: State Space Diagram



2.2 Choice of Uninformed Search Algorithm

Depth-limited search is by its very definition not complete, so that leaves it out of the question. Depth-first search *could* be used, but only if we keep track of already visited nodes (the state tree is bidirectional/looping). In really large trees one wishes to not have to keep track of already visted nodes due to memory constraints, but this tree is fairly small (few nodes), so depth-first search would be feasible if we kept track of previously visited nodes.

By looking at Figure 1 on the previous page we see that the branching factor, i.e. the average number of successors to each node, is $b \approx 1.1$ (most nodes have only one successor). We also see that the depth of the solution is $d = 11$. Breadth-first search would have a time complexity of $O(b^d) \approx O(3)$, and similar space complexity. Iterative deepening search would have the same time complexity as breadt-first search, but a space complexity of $O(bd) \approx O(12.1)$. Bidirectional search would have a time complexity of $O(b^{d/2}) \approx O(1.7)$ and similar space complexity. Since all step costs are equal, uniform-cost search would have a time complexity and a space complexity of $O(b^{d+1}) \approx O(3.1)$. And finally, depth-first search would have a time complexity of $O(b^m) \approx 3.1$ and a space complexity of $O(bm) = 13.2$, where $m = 12$ is the maximum depth of the tree (if we think of it as a tree).

Each one of the above cases requires that we keep track off previously visited nodes, since edge is bidirectional and nodes thus could be expanded indefinitely. Bidirectional search has both the lowest time complexity and space complexity, so one could argue its case over the other alternatives, although the differences aren't that big and for all practical purposes it would not matter what we chose in this particular scenario.

3 Logical Agents

We shall state which of the following statements are correct:

a) $False \models True$. True.

b) $True \models False$. False.

c) $(A \wedge B) \models (A \Leftrightarrow B)$. From the truth tables for $(A \wedge B)$ and $(A \Leftrightarrow B)$ one can observe whenever $(A \wedge B)$ is True, $(A \Leftrightarrow B)$ is also True (but not the other way around). Statement is True.

d) $(A \Leftrightarrow B) \models (A \vee B)$. From the truth tables we see that there is one case in which $(A \Leftrightarrow B)$ is True but not $(A \vee B)$. Statement is False.

e) $(A \Leftrightarrow B) \models (\neg A \vee B)$. From the truth tables we see that whenever $(A \Leftrightarrow B)$ is True, $(\neg A \vee B)$ is also True. Statement is True.