Assignment 1: Search Algorithms

Xavier F.C. Sánchez-Díaz

20th August 2024

Deadline: 13.09.2024, 23:59 hrs

Overview

In this assignment, you will experiment with the search algorithms we covered in the lecture, in a well-known context. This is an exercise set where you will manually generate the list of expanded nodes and get a feel of the differences between various informed and uninformed search strategies.

The assignment consists of **two parts**. The **first part** is uninformed search, while the **second part** will cover informed search. You need to make a **passable effort** on **both** parts in order to get this **assignment approved**. Each section specifies a set of required tasks you must do, either on your own or teaming up with at most one teammate.

1 The problem: The wolf, the goat, and the cabbage

This is a very well-known problem in computer science. It is often defined as a "logical riddle", and sometimes appears in programming interviews. The problem is as follows:

> *A farmer with a **wolf**, a **goat** and a **cabbage** must cross a river by boat. The boat can carry only the farmer and a single item. If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage. How can they cross the river, without anything being eaten?*

Modelling this problem is an interesting way to practice our skills in data structures and mathematics. Luckily for you, we have conveniently provided a visualisation of our model in Figure 1.

Every vertex represents a state, and every edge represents an action and has a direction. Redundant actions are marked as 'undo actions', which will simply take you back to

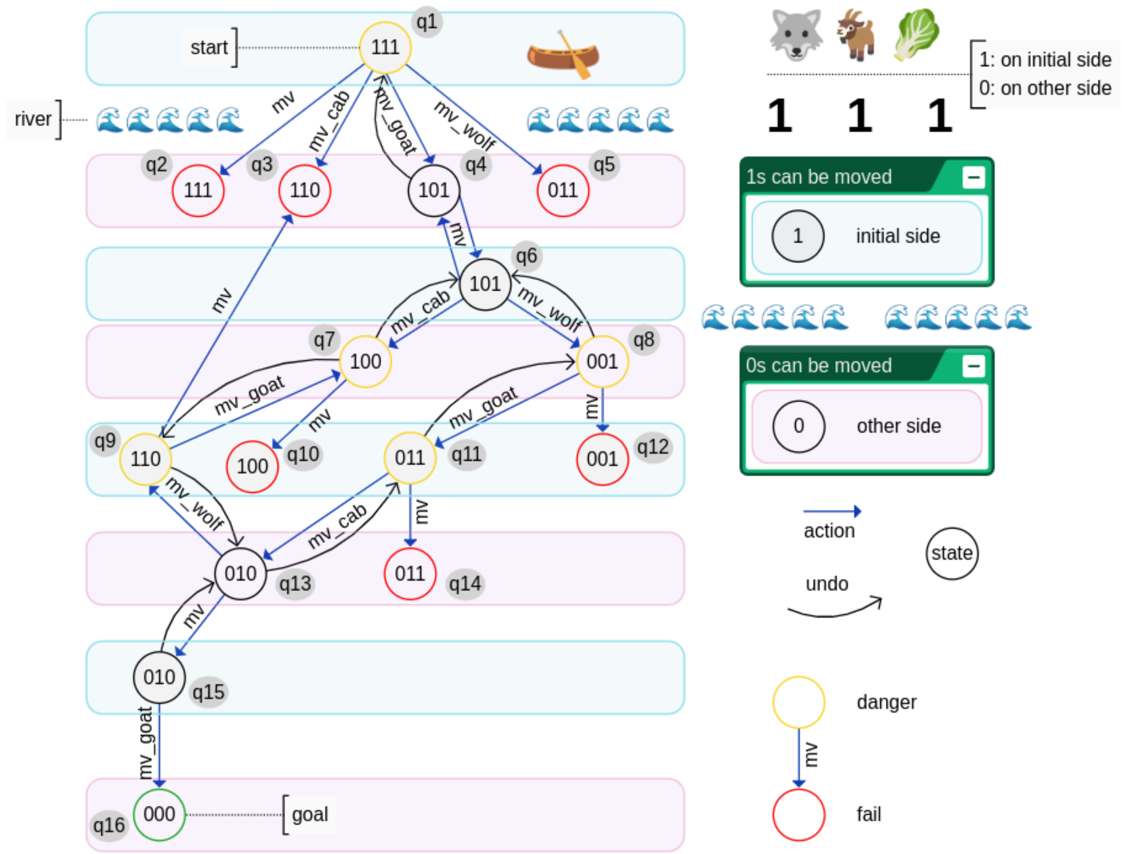

Figure 1: The Wolf, goat and cabbage problem visualised. For simplicity, the states have been labelled from q_1 to q_{16} so you can refer to them in your solutions.

the previous state. Each state is represented as a 3-bit string that indicates the presence or absence of an item on the initial side of the river. For example:

- The **starting node**, `111` (or q_1), indicates the *presence* of the **wolf**, **goat** and **cabbage** items on the *initial* side of the **river**.
- The **goal node**, `000` (or q_{16}) indicates the *absence* of the **wolf**, **goat** and **cabbage** items on the *initial* side of the **river**, i.e., they are on the *other* side.

As you can see, there is **no cost** associated with performing an action, so that means that all actions cost the same. Your goal is, then, to find the solution by simulating the different search algorithms.

1.1 Part One: Uninformed Search

1. Execute **tree-like** search on the graph shown in Figure 1 for the given algorithms below. Tie breaking: Left-to-right order of successors.

   a) Breadth-first search

   b) Depth-first search

   c) Uniform-cost search

Write down the order of expansion, the path from start to goal (or write *None* if no path to goal is found), and write the *cost* (i.e., count the number of steps) of the path found. For example:

> Example of output
>
> **BFS**:
>
> - Order of expansion: S, A, B, (G)
> - Found path: SBG
> - Path cost: 4

You can include your search trees as diagrams as well, but it is not necessary.

1.2 Part Two: Informed Search

As we discussed in the last section, there are no costs associated with actions in the graph from Figure 1. For an informed search strategy (like A^*) to work on a graph like this, it would need a **heuristic function**. What is a good way to quantify *how close* we are to the goal?

1. Design a heuristic function for the graph presented in Figure 1 so that the A^* and Greedy Best First strategies are able to solve the problem:

    a) Assign an h value to each state (or better yet, write it in a function form!)

    b) Is this heuristic **admissible**?

    c) Is this heuristic **consistent**?

2. Execute **tree-like** search on the graph shown in Figure 1 for the given algorithms below, using your proposed heuristic and the left-to-right order of successors tie-breaking strategy.

    a) Greedy Best-First

    b) A^*

As with the previous part, write down the order of expansion, the path from start to goal (or write *None* if no path to goal is found), and write the *cost* of the path found.

2 Deliverables

You must upload a **single** PDF containing **typeset** text, equations, and/or diagrams of your solutions.

- Include **natively digital** equations and diagrams. You can as well use hand writing on an tablet as long as it is readable. Do **not** upload scans or photos of hand-written solutions as these will be ignored by the TAs

- You can typeset your equations in Microsoft Word/Google Docs. If you're feeling adventurous, try LaTeX. Overleaf is a great place to start