Motion Planning — Exercise 4

Wolfgang Hönig and Andreas Orthey, TU Berlin SS 2024

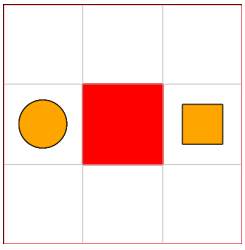
Non-Programming

1. Consider A* with a heuristic that is not admissible.
   (a) Show that the resulting algorithm is not optimal.
   (b) Is the algorithm still complete? Explain why or why not.
2. The lecture introduced A* using an OPEN priority queue and a closed set. Other resources, e.g., `https://en.wikipedia.org/wiki/A*_search_algorithm#Pseudocode`, only use the OPEN priority queue.
   (a) Provide an example of an admissible heuristic that is not consistent.
   (b) Are those two variants (wikipedia vs. lecture) equivalent?
   (c) What are respective advantages and disadvantages?
3. Consider a problem where no solution exists.
   (a) Explain the structure of such a graph. What kind of property does it have?
   (b) What will A* do, and how will it terminate?

Programming

4. Implement A* for a simple agent moving in a 4-connected grid.

   Consider the following example, where a circular robot has to plan a motion towards the square goal, while avoiding the red obstacle:



   The planning problem is described in a yaml-file, including: bounded workspace, obstacles, start, and goal configuration:

```yaml
map:
  dimensions: [3,3]
  obstacles:
    - [1, 1]
robot:
  start: [0, 1]
  goal: [2, 1]
```

The goal is to generate the shortest path from the start to the goal and output it as another yaml file, e.g.:

```yaml
- [0, 1]
- [0, 2]
- [1, 2]
- [2, 2]
- [2, 1]
```

We already provide a visualization script that can animate your generated solution.

(a) Implement A*, such that you can read a problem specification file and output a solution file. **Hints:** For Python, you can use the `yaml` package for file IO and `heapq` for a priority queue[1].

(b) Test your approach with three different heuristics: the 0-heuristic, Manhattan distance, and a non-admissible heuristic of your choice. What difference do you observe with respect to solution quality and computational efficiency?

[1] https://docs.python.org/3/library/heapq.html#priority-queue-implementation-notes