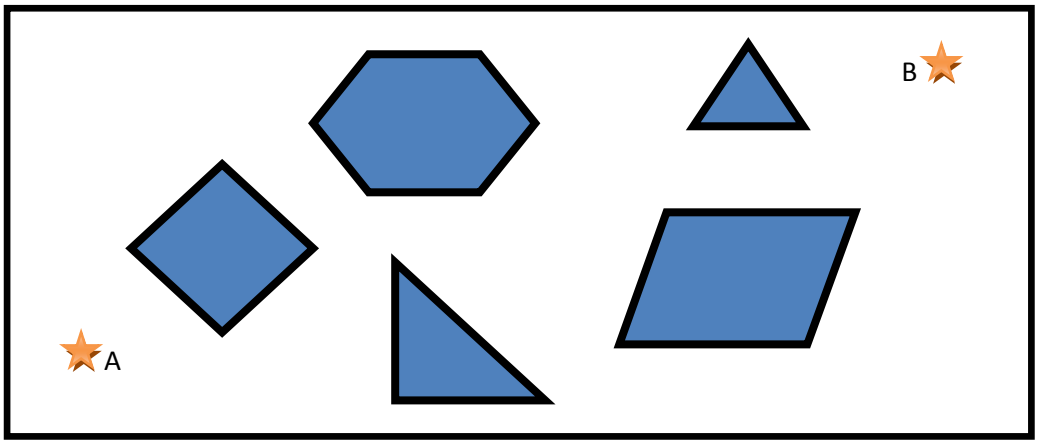
Intelligent Systems

Exercise sheet 5
Search Methods

This exercise sheet is about search methods. The exercise consists of two parts, first you will have to model the environment in a way that is recognizable by a computer agent. Second, you will have to implement a search algorithm in a generic way and thereafter apply it to the problem[1].



Assume a plane with convex obstacles similar as shown in the picture above. For any two points (A) and (B) find the shortest path between them; this is an idealization of the problem that a robot has to navigate through an environment.

Exercise 1 (10 points)

Suppose the state space of the agents consists of all positions (x,y) in the plane. How many states are there? How many paths are there to the goal?

Explain briefly why the shortest path from one polygon vertex to any other in the scene must consist of straight-line segments joining some of the vertices of the polygons. Define a good space state now. How large is the state space?

Define the necessary functions to implement the search problem, including a successor function that takes a vertex as input and returns a set of vertices that can be reached in a straight line from the given vertex.[2] Use the straight line distance for the heuristic function.

Now that you abstracted the problem to a successor and heuristic function, implement it in your favorite programming language. Implement an example plane that is of similar complexity as the one shown in the figure above. Note that your successor function can be implemented statically, i.e. your

[1] Exercise adapted from Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Prentice Hall, 1995

[2] Do not forget the neighbors on the same polygon

implementation does not need to compute it for some environment, but can look it up in a table that you filled in advance.

Exercise 2 (10 points)

Implement the A* algorithm as defined on the lecture slides, keep the implementation independent[3] of the problem given in Exercise 1. If done so, apply it to solve the problem of Exercise 1.

[3] i.e. you should be able to reuse it for any other admissible problem