Faculty of Computer Science and Business Information Systems
5172080: Fundamentals of Mobile Robotics

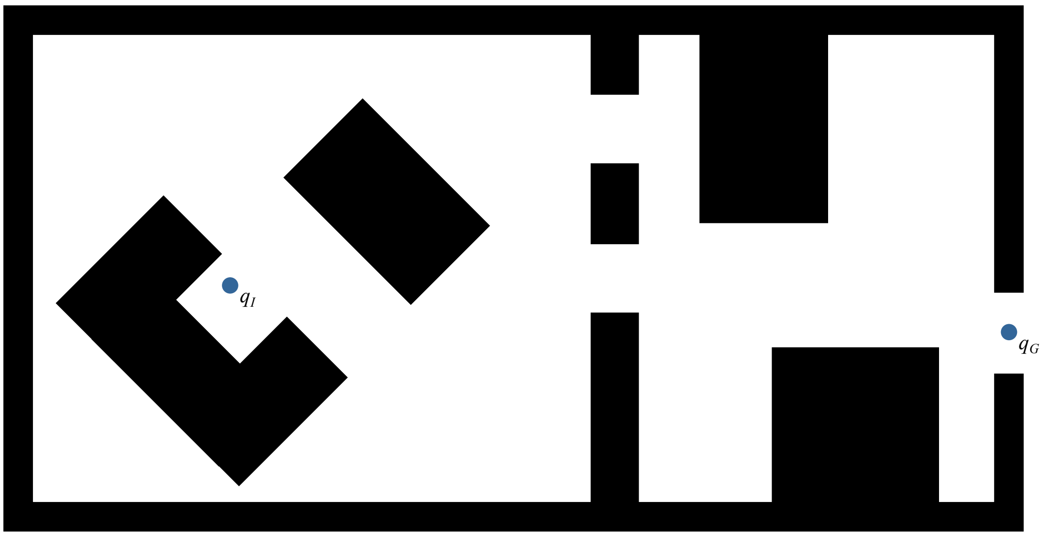
Prof. Dr. Pascal Meißner
pascal.meissner@thws.de
+49 (0)931 35118353
Franz-Horn-Str. 2 (room K.3.12)

Tutorial 7 – Solutions

Topic: Motion and Path Planning

Q 1 – Trapezoidal Cell Decomposition

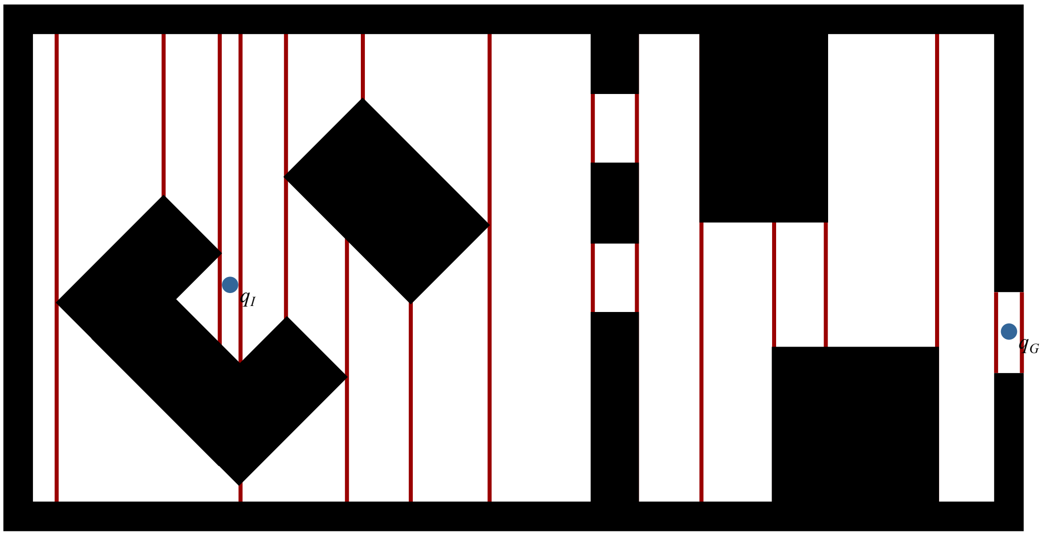
A point robot tries to plan an admissible path in the two-dimensional space below. It uses the trapezoidal decomposition as a method to discretize this polygonal configuration space.



1.a

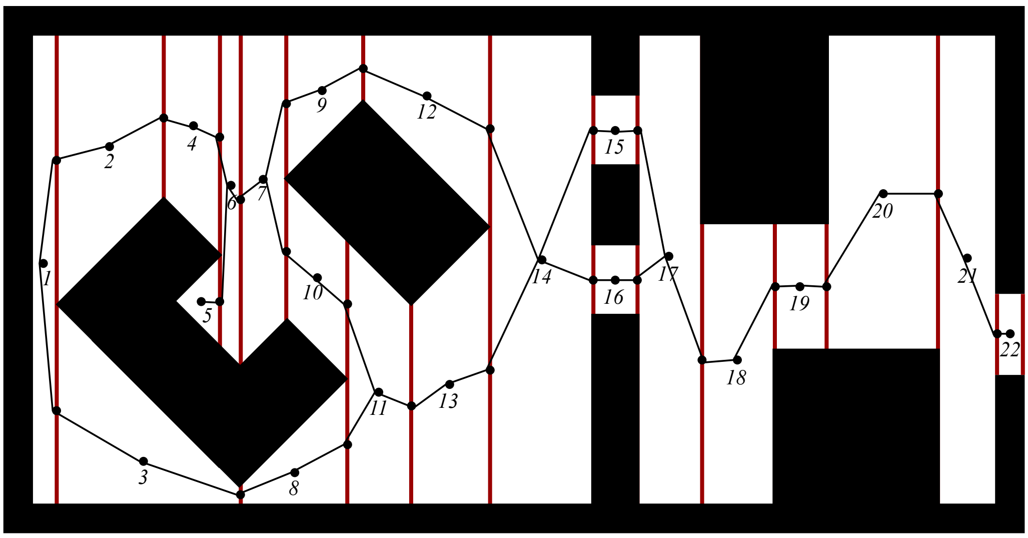
Decompose the free space (white area) into non-overlapping trapezoidal cells. Draw the cells on this sheet. Number the resulting cells from left to right. When they have the same left boundary, number the cells from top to bottom.

Vertical lines are drawn from each vertex of all polygons until an obstacle or boundary of the world is reached.



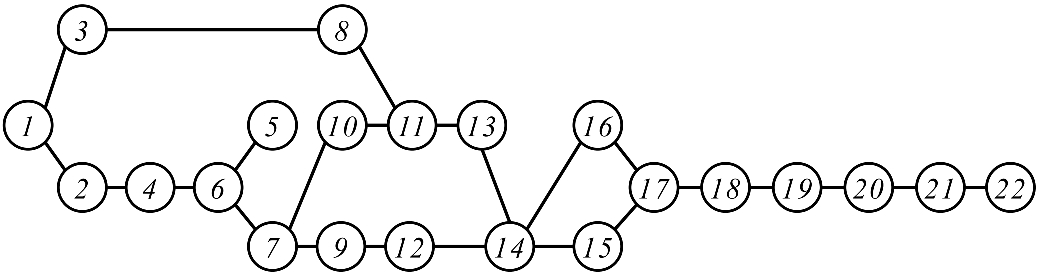
1.b

Place vertices in the decomposed free space and connect them by path segments. Construct an adjacency graph from this.



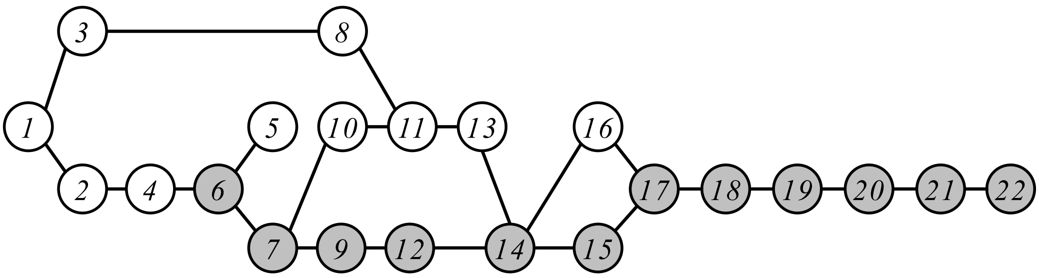
To create an adjacency graph from the cells, nodes are first placed at the centre of each cell and at the midpoints of all cell edges. Adjacent cells are then connected by straight lines using these nodes. Finally, the cells are numbered from left to right and from top to bottom.

The adjacency graph is a topological abstraction of the geometry of the configuration space: cells are simplified to nodes according to their numbers and common edges of two cells yield edges in the graph.



1.c

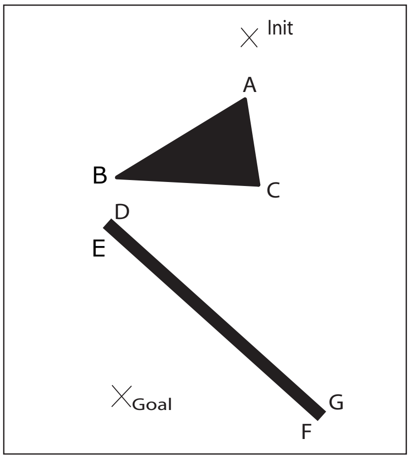
Find the shortest path from the initial configuration q_I to the goal q_G within your adjacency graph. Specify the sequence of cells traversed. Is the shortest path unique? Note: q_G and q_G are included in the above figure.



The cells 6 and 22 contain the configurations q_I and q_G. Two shortest paths between these configurations exist in the graph: (6, 7, 9, 12, 14, 15, 17, 18, 19, 20, 21, 22) and (6, 7, 9, 12, 14, 16, 17, 18, 19, 20, 21, 22). Hence, the shortest path is not unique.

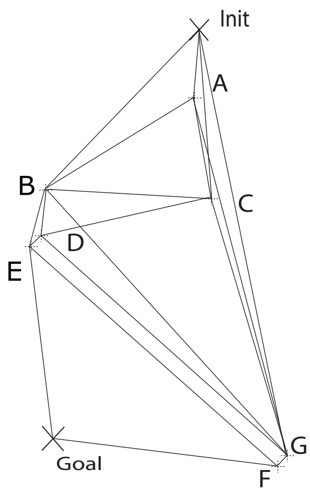
Q 2 – Visibility Graphs

Now the point robot tries to plan an admissible path in the following space instead. Given are two obstacles (grey area), the initial configuration of the robot (Init), and the goal configuration (Goal). The image frame does not represent an obstacle. The vertices in the configuration space are indicated by capital letters (A to G).



2.a

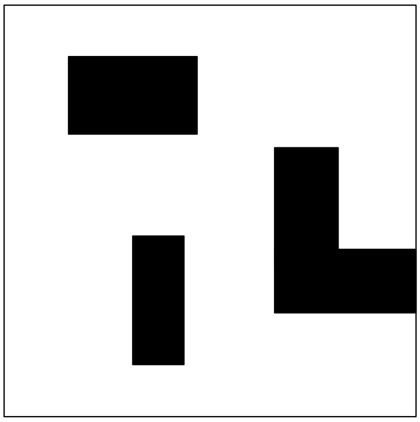
Create a collision-free roadmap between the initial and goal configurations and the obstacles by constructing a visibility graph. Draw the graph on this sheet. How many edges does the graph contain overall?



The visibility graph contains 19 edges.

Q 3 – Quadtree Decomposition

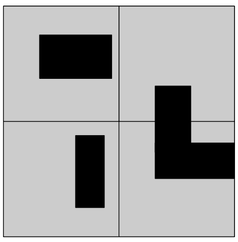
Finally the point robot decides to use the quadtree decomposition to discretize the configuration space below. This square space is split into free space (white area) and obstacle regions (black areas).

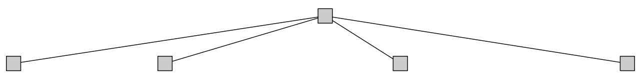


3.a

Construct the quadtree up to a depth of three and draw the resulting grid on this sheet. The root (level 0) of the tree corresponds to the square above. Process the children of a cell from left to right and bottom to top. Label the nodes of the tree as full, empty or mixed. State the numbers of full, empty, and mixed cells in the grid.

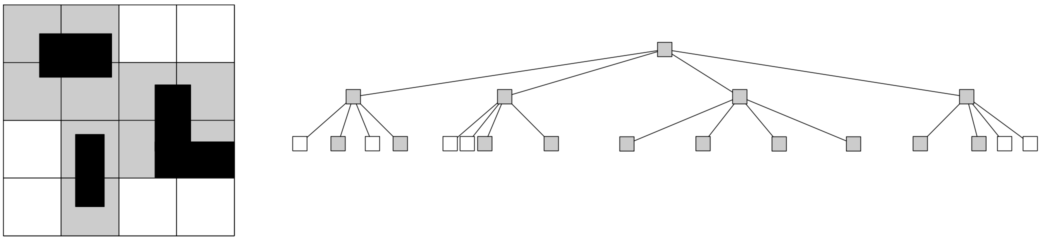
The quadtree (right) after the 1st subdivision (left):



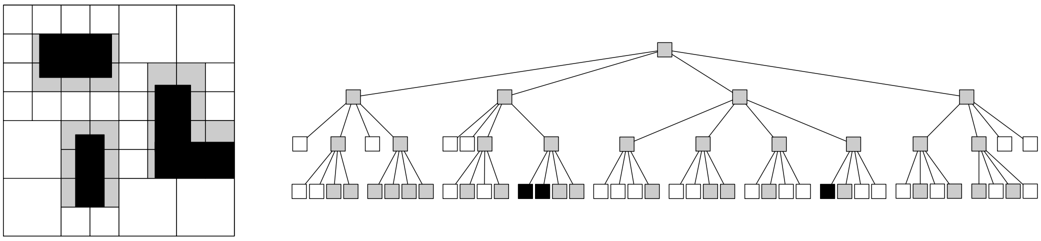


Full nodes are colored black, empty nodes are white, and mixed nodes are gray.

The quadtree after the 2nd subdivision:



The quadtree after the 3rd subdivision:



Full grid cells: 3, empty grid cells: 24, mixed grid cells: 19.

Q 4 – Search Algorithms

Prove the following propositions:

4.a

Breadth-first search is a special case of uniform-cost search.

Breadth-first search always selects the unexpanded node closest to the starting configuration, whereas uniform-cost search always selects the unexpanded node with the lowest accumulated cost from the starting configuration. If all edge costs are equal, then all nodes at row k in the search tree have an accumulated cost of k. Their distance to the starting configuration is minimal iff they have minimum accumulated path cost. In that case, the expansion orders of breadth-first and uniform-cost search are identical.

4.b

Breadth-first search, depth-first search, and uniform-cost search are special cases of greedy search.

First, let's show that depth-first and uniform-cost search are special cases of greedy search: for depth-first search, this becomes evident if we set the heuristic of greedy search to $h(n) \coloneqq -\mathrm{depth}(n)$, where $\mathrm{depth}(n)$ is the distance between the starting configuration and node n. Setting the heuristic to $h(n) \coloneqq g(n)$ instead, it becomes clear that uniform-cost search is a special case of greedy search. As one knows from 4.a that breadth-first is a special case of uniform-cost search, breadth-first is also a special case of greedy search.

4.c

Uniform-cost search is a special case of A search.*

A* search uses $f(n) = g(n) + h(n)$ as an evaluation function. If its heuristic is set to $h \equiv 0$, its evaluation function becomes $f(n) = g(n)$. In that case, the open node with lowest accumulated cost is always expanded, which corresponds to the expansion order of uniform-cost search.