Formal proofs for a cell decomposition algorithm

Yves Bertot
Yves.Bertot@inria.fr

Given a collection of obstacles, one wishes to construct paths from a given start location to a given finish location. The main constraint is that the paths should not collide with the obstacles.

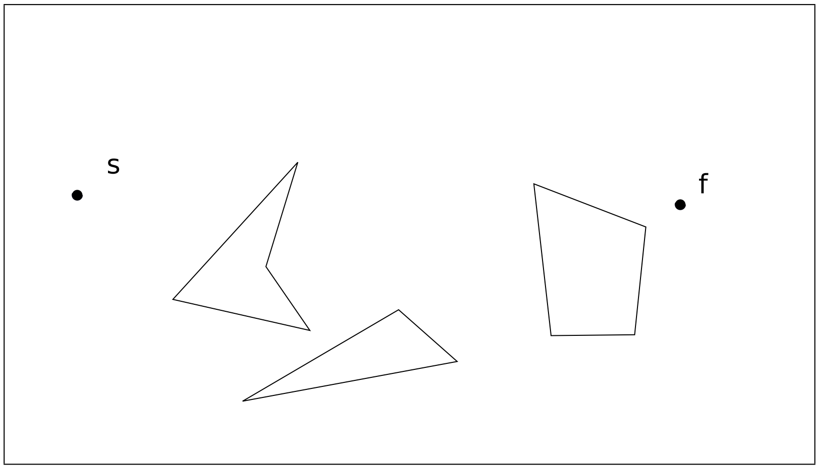
Algorithms for this problem have been known for while, and there are several variants, depending on whether one wish to have maximal clearance (moving in such a way that the distance to the closes obstacle is maximized) or the shortest length.

Another consideration of safety, is that one wishes to guarantee that the algorithm implementation does not include bugs that may provoke a collision. An approach to avoid bugs that has been advocated over the last decade is to use *interactive proof assistants* to state and guarantee the safety properties together with the algorithm implementation.

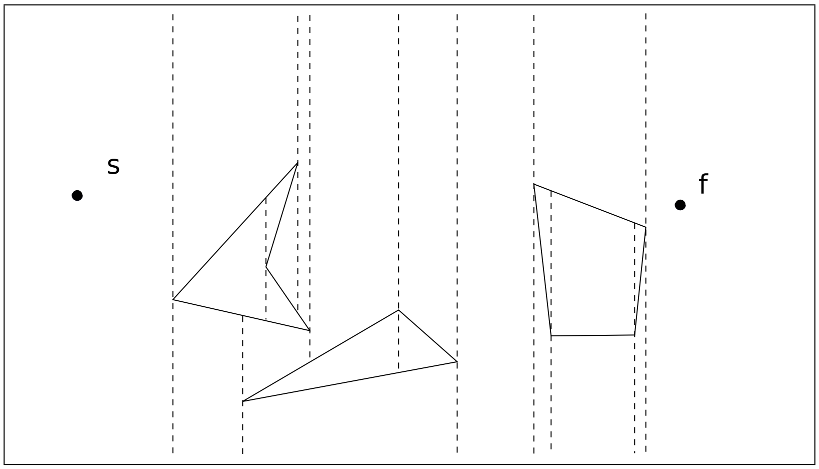
In this internship, we will study a simple motion planning algorithm and use the Coq interactive proof assistant to verify its properties.

The algorithm that is the object of this internship is a *vertical cell decomposition algorithm*. Descriptions of this algorithm can be found in the book by Latombe ([2], Ch. 5, p. 200) and in the book by LaValle ([3], Ch. 6, p., p. 251).

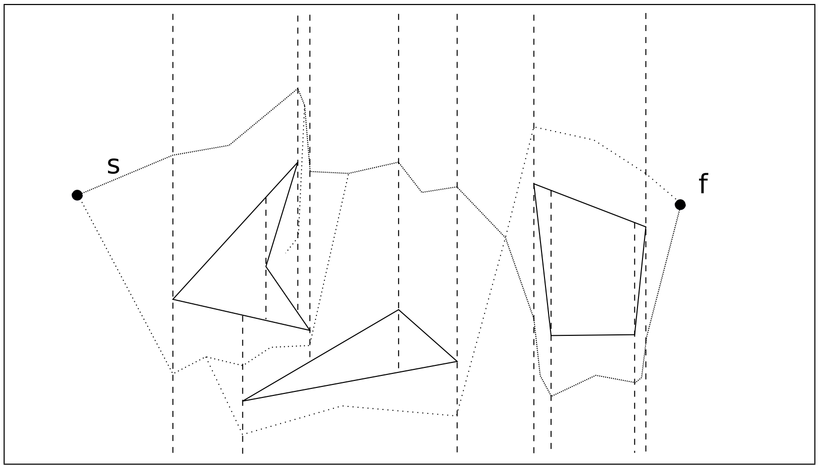
The input for the algorithm we wish to consider is a rectangle (a bounded area of the plane) and a set of straight-line segments, representing the boundaries of obstacles, as in the following figure.



When moving a vertical line from one side to the the other, one can detect the encounter of segment extremities, and draw vertical lines that reach to the next segment above and below. In the end, the full collection of added vertical lines and the input segments delimit cells with no obstacle inside. These cells have as boundaries either obstacle segments or added vertical lines, dashed lines in the following figure.



These cells are very simple, they are convex, and it easy to draw safe paths from one cell to the neigboring ones, as in the following diagram, where a sample point has been added to each of the cells.



In the end the motion planning algorithm boils down to finding a path from the cell containing the starting point to the cell containing the finishing point. This is illustrated in this figure, where one path from the starting point to the finishing point is drawn in a special style. This path can be improved in various ways, for instance by removing sample points inside cells and adding direct paths from obstacle free sides to obstacle free sides.

During the formal proof of correctness for this algorithm, the intern will have to study how to represent the various data structures to represent cells, processing queues, segments, and adacency properties and how the properties of elementary data-structures will contribute to the final safety statement of the algorithm during formal proofs performed with the Coq system [1].

Context: Proofs will be performed using the Coq system[1] and the Mathematical Components library[2]. Some experiments may require writing example implementations in Ocaml. The supervisor and his team will provide access to computers with Coq and the relevant libraries installed and training.

Prerequisites: The pre-requisite for this internship is a good knowledge of functional programming.

[1] `https://coq.inria.fr`

[2] `http://math-comp.github.io/math-comp/`

Tasks: The intern will have to write various implementations of the algorithm, either using plain inductive data structures, or using data-types for finite sets and graphs provided in the Mathematical Components library. They will also have to write specifications expressing the required safety properties for the output. They will then have to perform proofs, mostly using the Coq system and the existing theorems of the Mathematical Components library [4].

References

[1] Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions. Springer. 2004.

[2] Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers. 1991.

[3] LaValle, S. M.: Planning Algorithms. Cambridge University Press. 2006. `http://planning.cs.uiuc.edu/`

[4] Mahboubi A., Tassi E.: Mathematical Components. To appear. `https://math-comp.github.io/mcb/`