Formal proofs on paths avoiding collisions

Yves Bertot
Yves.Bertot@inria.fr

The general context of this work focusses on planning the movements of an object in a two-dimensional space where obstacles are described by a collection of straight line segments.

We assume a vertical cell decomposition algorithm has already produced a collection of cells that are guaranteed to be free of obstacles, together with a description of "open boundaries" between these cells, which make it possible to move from one cell to the next. The object is now to provide an algorithm that constructs paths between two points in the space, together with a proof that this algorithm will never produce paths that collide with the obstacles.

The algorithm and the proof will be described using the Coq proof system. Mathematical questions should be restricted to simple algebraic reasoning about intersections between straight line segments, naive set theory, recursive reasoning about sequences of segments, and graph traversals.

This internship requires a part of programming with symbolic representations of mathematical objects (line segments) and together with the corresponding step of formal proof. We expect the formal proof work to be performed using the Coq proof system [1], more precisely with the Mathematical Components library [4]. This work will give the opportunity of further work in the direction of NURBS (Non-Uniform Rational Basis Splines).

We also wish to derive outreach material from this experiment, where animations of the algorithm's working could be produced for the public to see.

This internship will be supervised by Yves Bertot.

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/`