



Formalisation de l'algorithme de Bentley-Ottmann

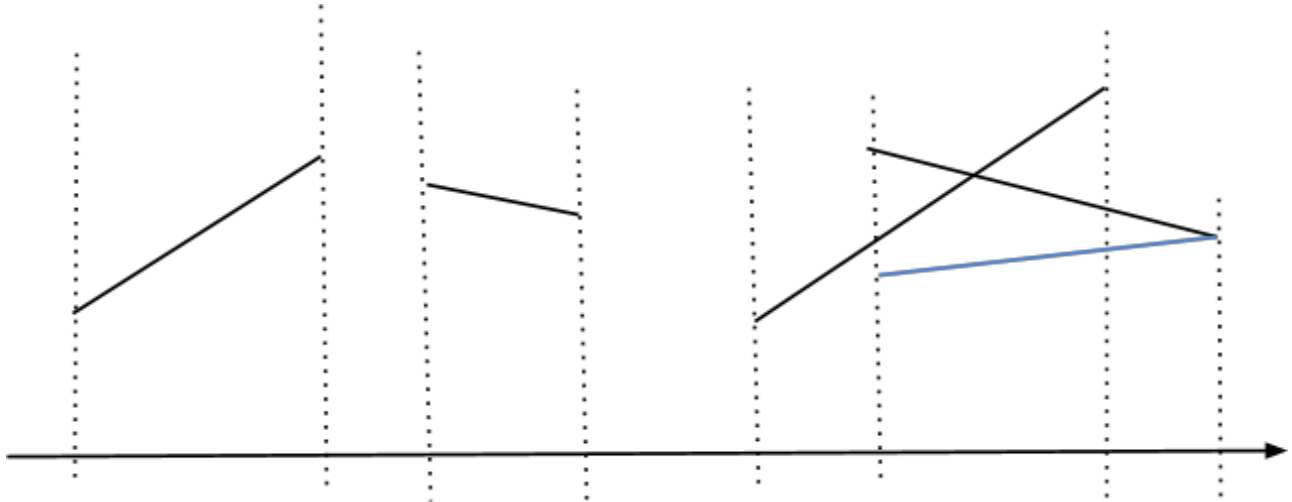
Encadrants: Pascal Schreck (schreck@unistra.fr) et Julien Narboux (narboux@unistra.fr)

Lieu du stage: Université de Strasbourg, Laboratoire ICube

L'objectif de ce stage de Master 2 est la formalisation à l'aide de l'assistant de preuve Coq d'un algorithme de géométrie algorithmique. Il s'agira de programmer et prouver la correction de l'algorithme de Bentley-Ottmann [BO79]. Cet algorithme permet, étant donnés des segments du plan, de calculer les coordonnées des points d'intersections de ces segments.

Un algorithme naïf pour résoudre ce problème consiste à examiner toutes les paires de segments, ce qui conduit à $n(n+1)/2$ tests d'intersection. Cependant, on peut se dire que en général, tous les segments ne se coupent pas 2 à 2, et que même, en général, le fait que 2 segments se coupent est plutôt rare. En fait, l'algorithme de Bentley-Ottmann a une complexité qui dépend des données en entrée (comme dans la définition) mais aussi du résultat, en l'occurrence, du nombre de points d'intersection.

Pour diminuer significativement le nombre de tests d'intersections entre segments, l'idée est de placer les segments dans des bandes de plan verticales qui sont en quelque sorte des boîtes englobantes suivant la direction Ox. Deux segments ne peuvent avoir une intersection que s'ils partagent bande de plan verticale commune.



Le travail se déroulera de la manière suivante:

1. Implantation et preuve d'un algorithme de calcul de l'intersection entre deux segments.
2. Implantation en Coq de l'algorithme naïf.
3. Extraction et test de l'algorithme naïf.
4. Preuve de la correction de l'algorithme naïf.
5. Implantation en Coq de l'algorithme de Bentley-Ottmann (en se basant sur les structures de données existantes en Coq pour la gestion d'un ensemble trié).
6. Extraction et test de l'algorithme de Bentley-Ottmann.
7. Rédaction d'une preuve détaillée de l'algorithme sur papier sous les hypothèses simplificatrices (que les segments sont tous distincts et ne s'intersectent pas en leurs extrémités, *etc.*).
8. Formalisation de cette preuve en Coq.

Ce travail pourrait se prolonger par une étude des problèmes de stabilité numérique [BP00].

Profil: Ce sujet nécessite des compétences en programmation fonctionnelle, ainsi qu'une expérience d'un assistant de preuve comme Coq par exemple.

Bibliographie:

- [BO79] Jon Bentley & Thomas Ottmann, "Algorithms for Reporting and Counting Geometric Intersections", IEEE Trans. Computers C-28, 643-647 (1979)
- [BP00] Boissonat, J.-D.; [Preparata, F. P.](#) (2000), "[Robust plane sweep for intersecting segments](#)", SIAM Journal on Computing 29 (5): 1401–1421.