Programming Assignment (optional)
due: May 17th, 2010

We encourage you to complete this assignment. It will account for 20% of your final grade, provided that it increases the grade. Please submit this assignment directly to Efi Fogel. Further details will be given in the course website.

Exercise: Develop a function template called `triangulate()` that triangulates arrangements using the regularization and triangulation stages introduced by Preparata and Shamos [3]; see also [1, Section 25.2] for a summary. This algorithm is very similar to the one presented in class [2] for a single polygon.

The function accepts an arrangement $A$ of segments as input. It inserts pairwise interior-disjoint segments into the arrangement that triangulate all the bounded faces of $A$. It consists of two steps. First, it inserts the appropriate segments to make all bounded faces $x$-monotone. Then, it inserts the remaining segments, which complete the triangulation.

Use the following declaration:

```
template <typename Arrangement> void triangulate(Arrangement& arr);
```

When the function template is instantiated the `Arrangement` template parameter must be substituted with an instance of the `CGAL::Arrangement_2<Traits,Dcel>` class-template.

Develop a simple `main()` function that reads an arrangement stored in an input ASCII file, issues a call to an instance of the `triangulate()` function-template, and writes the resulting arrangement to the standard output stream.

The implementation must handle all degeneracies. Examples of input files will be distributed later on through the course website.

References

