Christian Zinck

DMP Planarity Decision

For my project I chose to implement the DMP planarity decision algorithm by path addition. I based my implementation off of the pseudocode provided at http://www.mathcs.emory.edu/~rg/book/chap6.pdf, pages 14 - 17. The first step was to eliminate some easy cases, as the source suggests. Accordingly, all self and multiple edges are removed and only the largest biconnected component is considered. Additionally, all graphs with less than 5 vertices are planar, all graphs with $e > 3v - 6$ are nonplanar, and all acylic graphs are planar.

The core of the algorithm is as follows: starting with a single cycle, and therefore a graph with 2 faces, iteratively add paths to the graph, each time increasing the number of chordless cycles and faces by 1, until the graph is completely embedded, planar, or a path is found with no viable embedding, nonplanar. The difficulty in the implementation comes from managing the regions of the graph, path selection, and deciding how many possible embeddings a path can have.

To manage the regions, I chose to maintain a list of cycles known as a basis. A basis of a graph is a complete collection of chordless cycles in the graph, which for a biconnected graph includes all vertices and edges. Every time a path is added to the graph, it will split a region in 2, creating 2 new regions, both boundaries of which contain the path edge list, and the original region is removed from the basis, as it is no longer chordless.

To select a path for embedding, I chose to brute force test combinations of endpoints within the graph until a path was found. The alternative would've been to search for path possibilities within the basis, but since the basis size scales linearly with problem size while the maximum number of nodes stays constant, the larger the problem becomes, the less efficient it is to search for paths within the same region.

To determine the number of possible embeddings for a path, the fact that a path can be embedded in as many ways as its endpoints share regions was used, meaning if 2 nodes share a path that has not already been embedded, that path can be embedded in any one of their common regions. If a path has 0 possible embeddings, the graph is obviously nonplanar. If a path has 1 possible embedding, it must be embedded in that region. If a path has more than 1 possible embedding, it cannot be embedded until it is certain that no path with 1 possible embedding exists.

I have tested the implementation on graphs of known planarity and it decides correctly. It has a worst case time complexity of $O(fv^2)$ when all nodes have a path with more than 1 possible embedding and a best case time complexity of $O(f)$ when all paths have exactly 1 possible embedding. It has a slower average time complexity than other competing algorithm categories, such as edge and vertex addition, but has the fastest best case because the number of faces is always less than the number of edges or vertices.