Journal-Spring 2010

Son Nguyen
John Lindser
Dhinali Peiris

June 10, 2010

At the beginning of the quarter the goal assigned was to write code to auto generate code to compute the germinal configuration given the inputs of the functions. And, also in the process to find the most efficient way to compute them effectively. Unfortunately though the first couple of weeks we tried to get somewhere, we were not successful.

Hence in the middle of the quarter we started to write code to generate all possible cases of geminal configurations given the constraints

- The number of nodes in a geminal configuration to be between 2 and 6
- The number of edges in a geminal configuration to be between 1 and 3
- The number of isolated nodes to be not more than one in a geminal configuration
- The total number of edges and isolated nodes in a geminal configuration to be between 1 and 3
- Eliminate the case where there are more than one edge on two nodes in consideration
- Eliminate duplicates due to symmetry (we coded a couple of routines on shifting and deleting)

Once the duplicates were eliminated by all the combinations generated, we found that all could be grouped in to 11 categories. Namely if each node was numbered starting from zero to a maximum of five.

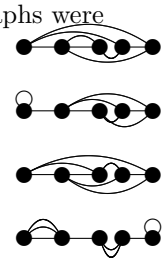
The eleven categories that were found were

- ((0,1))
- ((0,1),(2))
- ((0,1),(1,2))
- ((0,1),(0,2),(1,2))
- ((0,1),(0,2),(0,3))
- ((0,1),(1,2),(3))
- ((0,1),(1,2),(2,3))
- ((0,1),(2,3))
- ((0,1),(2,3),(4))
- ((0,1),(1,2),(3,4))
- ((0,1),(2,3),(4,5))

which were the same as we found when computed by hand earlier.

Further the symmetry group was found for each of the above listed case. Hence, the each case and the corresponding Symmetry group will let us determine all graphical representations of all geminal configurations by Son's code which he did last quarter.

By generating all geminal configurations by the program it was found that the case of (0,1,2)(3,4) generated four graphs extra than what was computed by hand. The missing four graphs were



Also during the quarter the two posters on finding all cases of a given graphical representation by hand and LaTeXthem, and writing an algorithm to find all cases of graphical representation and the code done in Python was combined in order to present for the 2010 Ohio University Student Research and Creative Activity Expo.