Ian Pascoe

Jean-Yves Hervé

CSC412

8 December 2018

Prog06 Report

For this assignment, we were to implement the replacement rule in an equation of a curve that was to be drawn to a GLUT window. The implementation that we started with used doubly linked lists to hold the data for the equation. What I ended up adding to the application was the multithreading ability with the use of mutex locks for access to the curve equation. The reason why this must be done is so the GLUT that we use does not access the equation while the program manipulates it. I started by creating a single thread version of the program that just replaced 1 instance of the left side of the replacement rule. Once I got that working, which involved adding 3 new equations and a struct to hold the rules, I implemented threads and mutexes. There was to be a thread created for each of the rules in the rules file. The thread would take care of replacing every instance of that rule in the original curve equation. Unfortunately, when implementing my solutions, I ran into several problems with the pointers. I have tried just about everything to fix the errors I saw but for some reason, GLUT was refusing to read the last segment of the curve equation after replacements. I print the resultant equation after a replacement, and it shows all of the nodes and the correct resultant equation but GLUT misses the last node. I also print when a thread or GLUT is acquiring the mutex, and it does in the correct order so that theory was refuted. I have always sort of struggled with pointer arithmetic but I believe what I was doing was right, I must just be missing something. Version 2 is not fully implemented due to time constraints. I was too busy trying to completely finish and debug version 1. I am very disappointed in the outcome of this assignment after the amount of work I put in. Hopefully with the next implementation of locks and semaphores, I will be able to fully implement it successfully.