Gregory Jerian

12/10/15

Period 4

Readme

INTRO AND HOW THIS MEETS THE SPECIFICATIONS:

This class uses a shunting-yard algorithm to convert “infix” notation, or the notation used by normal people, into “reverse-polish notation”, a notation that has operators after the operands. This is done using a queue and a stack, thanks to “What’s a Creel”’s video on YouTube. After this, I generated a truth table (thanks to Blake for his code), with n columns (n is number of variables) and 2n rows. Then, I evaluated this notation using another stack, thanks to “Hurray Banana”’s video. After this, I evaluated the various logic methods detailed in the requirements.

PROBLEMS WITH CODE:

The code is kind of buggy because there was a time crunch towards the end. The biggest problem is that entails and equivalent do not work all the time. The other three methods do though (valid, satisfiable, contingent)! Also, I could have implemented it better by using less data structures and not switching between them as much which would have cut back on the O(n) time. Finally, the interface is empty. I couldn’t figure out what to put in it so I just left it.

CREDITS:

<https://www.youtube.com/watch?v=QzVVjboyb0s> This video for explaining reverse-polish notation and shunting-yard.

<https://www.youtube.com/watch?v=LQ-iW8jm6Mk> This video for explaining evaluating stacks.

Sean Chapman, Shivum Agarwal, Blake Novak, Jerry Huang. We all communicated with each other on this project because we were all very confused as of what to do. We helped each other a lot and were able to create something that none of us could have created on our own.