

Jerry Huang

Period 2

APCS

Kuszmaul

Logical Expression Readme

I: Introduction:

As a part of our final in the aikido master Mr. Kuszmaul's class, we were required to create a program that will evaluate a logical expression inputted by the user. In this final project, we were asked to create an Interface LogicalExpression that requires a constructor that takes a String as an input that represents a logical expression. My version of this program contains the methods valid(), satisfiable(), contingent(), equivalent(), and entails().

II: How This Fulfills the Specification:

This project fulfills the specifications for the Logical Expression program. It contains all the necessary methods that return boolean values or either True or False. It also produces a truth table to show the different possible combinations of truths.

III: Explanation of Current Errors:

Currently, the program only has one tiny error. The entails method does not work properly for every logical expression. I tried fixing this, but couldn't find out how. However, everything else worked after debugging and after working with Sean Chapman, Shivum Agarwal, Blake Novak, and Gregory Jerian.

IV: Overview of Code

The code in my project is split into different methods that are encapsulated very well in order for clearer understanding. It begins by creating 4 instances of my LogicalExpressionClass and tests each by creating an array of the elements of the infix expression. Each element in the array is then analyzed to determine whether there are variables or operators. Once the variables have been identified, the convertToRPN() method is run in order to convert the infix form of the logical expression into reverse polish notation. This is necessary in order to remove the parentheses. The program then runs the truthTable() method to display the truth table. Lastly, the methods valid(), satisfiable(), contingent(), equivalent(), and entails() are run.

V: Major Challenges

Some of the major challenges included writing the equivalent and entails methods.

VI: Acknowledgements

Special thanks to Sean Chapman, Gregory Jerian, Blake Novak, and Shivum Agarwal for helping me with last minute errors and helping me figure out how to fix some parts of my program in order to get it to run properly. Also, thanks to the videos I watched for providing me with a clear understanding of the Shunting-Yard formula.