Trying to find a way to actually run the input for a specified grammar. Looking into chapter 4. Making a simple calculator.

Ran into a problem. BaseVisitor file was not made so I need to figure out how to use the command line for this.

replace

export  **with** set

/usr/local/lib **with** C:\Users\Arinze\Desktop\UROP2015Copy

antlr-4.0-complete **with** antlr-4.2-complete

removed parenthesis around the java –jar… complete.jar in alias(doskey)

replace alias **with** doskey

had to add “$\*” to the end of the doskey command

cd C:\Users\Arinze\Desktop\UROP2015Copy

set CLASSPATH=".:C:\Users\Arinze\Desktop\UROP2015Copy/antlr-4.2-complete.jar:$CLASSPATH

java -jar C:\Users\Arinze\Desktop\UROP2015Copy/antlr-4.2-complete.jar

doskey antlr4=java -jar C:\Users\Arinze\Desktop\UROP2015Copy/antlr-4.2-complete.jar $\*

After running into a lot of problems with, I switched to a new task of finding a minimal element circuit for a three input system. Programming in java. Will return to ANTLR after Bryan has looked into it and decided whether or not it would be a good use of time.

Made a class for the circuits. Need to implement the truthTable of the circuit when made from two. Must first confirm that I can iterate through a list of elements and make the different levels.

Worked on the file for a couple more hours at night. Made a function that gets all the circuits to a certain level. Implemented the new truthTable solver. Checked the circuits up to level 3, checked the truth tables up to level 2.

Must learn how to make tests for a function. Must also finish up the rest of the program by making a dictionary (or dictionary equivalent) and using it to find the minimum circuit for each of the truth values. Then will generalize to n inputs and will possibly also generalize for different gates.

\*for some odd reason y.concat(“1”); didn’t work. I had to use y+=”1”