**DOCUMENTATION:**

**Prerequisites:**

Before running source code, install the following:

* Python module anytree, via running command “pip install anytree” on command prompt if on windows.
* Graphviz from <http://www.graphviz.org/download/> . Making sure to then add the “/bin” folder to your system PATH.

To run python source code (designed for python 3.7.7) run it from the command line with the input text file as its only argument: e.g. “python CD.py input.txt” .

**Produced files:**

Once run, the source code will produce a log file, “log.txt”, containing either a message saying success, telling you both your input file and supplied formula were valid, or, if not, an error message telling you what was wrong.

The code will also produce a grammar and store it inside a file called “grammar.txt”, containing the set of terminals (\*Vt\*), the set of non-terminals (\*Vt\*) and a set of all the production rules (\*P\*) in the grammar .

If both the input file and the formula were valid, the program will also produce a visual depiction of the corresponding parse tree, contained in “tree.png”.

If the input file format was valid, but the formula was not, the appropriate grammar will still be displayed within “grammar.txt”, in order to assist you in troubleshooting your formula.

Note: if you define predicates but no variables, the log will give you a warning message indicating this before showing any error or success message. The predicates will be included in the terminal symbols of the grammar, but will not appear in the production rules, as upon arriving at a predicate, there would never be a way to turn <V> into a terminal, as no such variables exist.

**Input:**

Your input file must be in a strict format (as per the specification), if this is wrong, the log will tell you what the issue is.

As an example of what a file should look like observe the following (also contained in example.txt):

“

variables: w x y z

constants: C D

predicates: P[2] Q[1]

equality: =

connectives: \land \lor \implies \iff \neg

quantifiers: \exists \forall

formula: \forall x ( \exists y ( P(x,y) \implies \neg Q(x) )

\lor \exists z ( ( (C = z) \land Q(z) ) \land P(x,z) ) )

“

The output of the source code on this example has also been provided (“log.txt”, “grammar.txt” and “tree.png”).

Note: there must be exactly one space between the colon and the first symbol of each category, and one space between each subsequent symbol, finishing the line with a newline (\n) and no space before it, if not an error for extra whitespace will be outputted.

Furthermore, even if you do not want to define any constants or variables, you must still provide an empty section as such: “constants:\n” (with \n being a newline character)

However, the formula may contain extra whitespace including tabs and line breaks between formula symbols.

Note: all indexing of symbol numbers in log file error messages are 0 indexed.