CS 1571 Homework 3 Writeup

Christopher Grant

For my implementation of the project, I chose to use python as it has better string handling than java. For my implementation of forward chaining, I chose to implement the algorithm in the book in figure 9.3. Although I am sure that there are more efficient forward chaining algorithms, I chose to use this one because for the relatively small knowledge bases that are being constructed in this project, a faster or more efficient implementation of forward chaining wouldn't make much of a difference.

The incremental forward chaining algorithm that was implemented in FOL_IFC.py is a more efficient algorithm than the standard forward chaining algorithm implemented in FOL_FC.py because of the time savings from not having to call the unification algorithm when we know that it will not find any matches. Because unify is run a number of times for each iteration of of the for loop and run for each clause in the Knowledge base after q' is derived, being able to short circuit out of the for loop when we know that there will not be any other new clauses added to the knowledge base can be a great time saver. Additionally, running the "time" unix program to time the execution of each program, incremental forward chaining is slightly faster than the regular forward chaining and if this was expanded for larger knowledge bases with larger FOL statements, this additional time benefit would be more evident due to skipping the calls to unify. To see "time" output, look in the outputs folder, and look at time_IFC.txt and time_FC.txt