Joshua Kostura
COS 570

Resolution Theorem Prover

A resolution theorem prover, sounds easy in theory but in practice was a lot harder. I was able to prove some simple things with my RTP, but it does not handle variables correctly, or really at all. If can prove things form a slightly more complex axiom list, as long as it contains literals. In order to process variables I would have to write my own comparison function for clauses. It would only need to compare the "function" of the clause, for example dead, human , Pompeian, if these are equivalent then it can bind the variables with constant and use the unify function to make sure that the binding is valid. Unfortunately I haven't gotten to this point completely, so the code that I have turned in will prove rather simple things as describe above. The knowledge basis are included in the file axioms.lisp, this is loaded at the top my main lisp file. The main control strategy I used was to maintain a list of things we need to prove, then attempt to resolve them one at a time until we are let will nil, which would indicate that are process was successful. How when rtp cannot prove something it most of the time crashes, I was more focused on getting it to prove something that I didn't make it handle failure gracefully. All of the function within my code file are comment, hopefully well enough for you to understand. I have included a text file with the output of 3 sample runs of my rtp, it is fairly straight forward to understand the output, it prints what you are trying to prove, what it is trying to resolve, and then what it successfully resolves with.