Tyler Egan
Benjamin Hetz

December 3rd, 2017
CS457

Theorem Prover

We had a really hard time getting this project started. The data structures were pretty deeply nested and convoluted to access in a meaningful way. Being shown some code interacting with it in class was really helpful and was definitely the only reason we were able to finish in time.

For heuristics for determining which pair of sentences to work on next, we iterated over the predicates in each pair of sentences. If a predicate could be removed (ie the predicates matched and had opposite values for negation), then the value would be decremented. In all other cases, the value was incremented. The higher the value for any given pair, the lower priority it was considered in our queue. In the case of ties, the number of predicates in the first in the pair of sentences was used as a secondary priority. Again, the lower the value the higher the relative priority was.

During testing, we very quickly realized that none of the examples could actually be resolved, so we made a small modification so that we could actually check. The

Printing out the actual proof gave us a fair bit of trouble. In the end, we printed out the final sentence (the empty one) and printed its parents all the way up to the top.