Fred Morrison
CSC-366

Crypto: Heuristic Problem Solver - Task 2

The code for the Heuristic Problem Solver is relatively easy to understand. At the top of the program, there are a collection of facts named “rule” that each take 3 arguments. The first argument is the rule number, which is a simple integer. The second argument is a reference to the situation fact which determines whether or not the rule should be executed. The third and final argument is a reference to the action fact which should be executed if the situation is valid.

When you provide a problem to be solved, the Heuristic Problem Solver will traverse, in order from rule 1 to the last rule in the knowledgebase, the rules in an attempt to find one that can satisfy the problem. When the Heuristic Problem Solver pulls a rule to check from the knowledgebase, it first checks the situation fact that coincides with it. If the situation fact is true, it then performs the relevant action fact.

Essentially the program takes a “dumb” approach, for a lack of a better word, for deciding on which heuristic to use to solve the crypto problem. It has no idea of the applicability or human executability ratings for each heuristic. Therefore it simply attempts them in order from top to bottom that they were defined, until one of them works.

It is for this reason that the rules should preferably be defined from top to bottom in order of applicability, so that the most applicable heuristics are checked first. It is fast either way, but logically it makes sense to structure the rules in such a way.