

## **Exhaustive Crypto Solver Essay**

The exhaustive crypto solver relies on the idea of splitting up a large problem into multiple smaller subproblems that are easier to solve. The exhaustive solver was built incrementally, starting with the global variable ADT, random problem generation, order 2,3,4 problem solver, order 5 solver, and the final exhaustive solver which ties everything together and provides easy ways to generate fully parenthesized solutions to problems (either random or specific).

The problems are generated by making use of the global variable ADT and the built in prolog random function. The domain for the crypto numbers is defined by two global variables (low and high) at the beginning of the program. In this case, the domain is [0,15]. Crypto problems are assembled by generating 6 random numbers (5 for the number set and 1 for the goal). The problem is then inserted into the ADT for future use. In our case, one problem can be defined at a time in the ADT.

Crypto problems are solved by making use of combinatorial sets, which allows for the generation of every permutation / combination of a set of tokens from size 2-5. By having the ability of generating every possible permutation of numbers, we can brute force all of the possible solutions to the problem. The program makes use of the fact that computers can perform calculations much faster than a human. Obviously this is not a "smart" algorithm in sense that it does not make use of any heuristics. It simply exhaustively attempts to solve each possible permutation of numbers for the problem.