Task 4 – Putting it Together

Author: Josh Harkness

```
%%%%%%%%%%%%%%%  Code brought in and changed from crypto/v4/crypto.pro
%%%%%%%%%%%%%%%%%%%%%%%%%%%

:-consult('~/Documents/CSC366/Assignments/crypto/v1/crypto.pro').

:-consult('~/Documents/CSC366/Assignments/crypto/v3/crypto.pro').


establishCryptoProblem(numbers(N1,N2,N3,N4,N5),goal(G)) :-

        addCryptoProblemToKnowledgeBase(N1,N2,N3,N4,N5,G).


addCryptoSolutionToKB(Expression) :-

        retract(solution(_)),

        assert(solution(Expression)).

addCryptoSolutionToKB(Expression) :-

        assert(solution(Expression)).


solve(numbers(N1,N2,N3,N4,N5),goal(G)) :-

        retract(solution(_)),

        establishCryptoProblem(numbers(N1,N2,N3,N4,N5),goal(G)),

        displayProblem,

        solveProblemHeuristically,

        displaySolution.

solve(numbers(N1,N2,N3,N4,N5),goal(G)) :-

        establishCryptoProblem(numbers(N1,N2,N3,N4,N5),goal(G)),

        displayProblem,

        solveProblemHeuristically,

        displaySolution.


 solve :-
```

```
   retract(solution(_)),
   generateRandomCryptoProblem,
   displayProblem,
   solveProblemHeuristically,
   displaySolution.
 solve :-
   generateRandomCryptoProblem,
   displayProblem,
   solveProblemHeuristically,
   displaySolution.

demo(0).
demo(N) :-
        solve,
        K is N - 1,
        demo(K).

displaySolution :-
        solution(S),
        displayResult(S),
        nl.
displaySolution.

displayResult(ex(A,O,B)) :-
        number(A),number(B),
        write('( '),write(A),write(' '),write(O),write(' '),write(B),write(' )').
displayResult(ex(A,O,B)) :-
        number(A),B = ex(A1,O1,B1),
        write('( '),write(A),write(' '),write(O),write(' '),
            displayResult(ex(A1,O1,B1)),write(' )').
displayResult(ex(A,O,B)) :-
            number(B),A = ex(A1,O1,B1),
            write('( '),displayResult(ex(A1,O1,B1)),write(' '),write(O),write(' '),
            write(B),write(' )').
displayResult(ex(A,O,B)) :-
            A = ex(A1,O1,B1),B = ex(A2,O2,B2),
            write('( '),displayResult(ex(A1,O1,B1)),write(' '),write(O),write(' '),
            displayResult(ex(A2,O2,B2)),write(' )').

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rule(1,situation1,action1).
rule(2,situation2,action2).
rule(3,situation3,action3).
rule(4,situation4,action4).
rule(5,situation5,action5).
rule(6,situation6,action6).
rule(7,situation7,action7).
rule(8,situation8,action8).

solveProblemHeuristically :-
            rule(Number,Situation,Action),
            write('considering rule '),write(Number),write('...'),nl,
            Situation,
            write('application of rule '),write(Number),write(' produces '),
            Action.
solveProblemHeuristically.
```

```
%%%%%%%%%% rules created for the heuristics %%%%%%%%%%%%%%%%

delete_one(_, [], []).

delete_one(Term, [Term|Tail], Tail).

delete_one(Term, [Head|Tail], [Head|Result]) :-

  delete_one(Term, Tail, Result).


other_numbers(special(G),others(A,B,C,D)) :-

        problem(numbers(N1,N2,N3,N4,N5),goal(G)),

        delete_one(G,[N1,N2,N3,N4,N5],[A,B,C,D]).


doubleton(doubleton(A,B),rest(C,D,E)) :-

        problem(numbers(N1,N2,N3,N4,N5),_),

        combos(set(N1,N2,N3,N4,N5),combo(A,B),extras(C,D,E)),

        A = B.


doubleton :-

        problem(numbers(N1,N2,N3,N4,N5),_),

        combos(set(N1,N2,N3,N4,N5),combo(A,B),extras(_,_,_)),

        A = B.


member(X,[X|R],R).

member(X,[Y|R],Result) :-

        member(X,R,Subresult),

        Result = [Y|Subresult].


oneLess(A,G) :- A is G + 1.

oneLess(G,[N|R],N,R) :-

        oneLess(N,G).
oneLess(G,[X|R],OneMoreThanGoal,Rest) :-
        oneLess(G,R,OneMoreThanGoal,More),
        Rest = [X|More].

makeGoalFromThree(goal(G),numsUsed(C,D,E)) :-
        crypto(C,D,E,G,_).
makeGoalFromThree(goal(G),numsUsed(C,D,E),Expression) :-
        crypto(C,D,E,G,Expression).

makeGoalFromFour(G,numsUsed(A,B,C,D)) :-
        crypto(A,B,C,D,G,_).
makeGoalFromFour(G,numsUsed(A,B,C,D),Expression) :-
        crypto(A,B,C,D,G,Expression).

isHalf(G,Numbers,[A,B,C,D]) :-
        Half is G / 2,
        delete_one(Half,Numbers,[A,B,C,D]).
isHalf(G,Numbers,[A,B,C,D],Half) :-
        Half is G / 2,
        delete_one(Half,Numbers,[A,B,C,D]).

isDouble(G,Numbers,[A,B,C,D]) :-
        Double is G * 2,
        delete_one(Double,Numbers,[A,B,C,D]).
isDouble(G,Numbers,[A,B,C,D],Double) :-
        Double is G * 2,
        delete_one(Double,Numbers,[A,B,C,D]).
```

```
%Heuristic one-------------------------------------------------------------------------------------------------------------
---
situation1 :-
        problem(Numbers,Goal),
        Goal = goal(0),
        Numbers = numbers(N1,N2,N3,N4,N5),
        member(0,[N1,N2,N3,N4,N5]).

action1 :-
        problem(Numbers,_),
        Numbers = numbers(N1,N2,N3,N4,N5),
        addCryptoSolutionToKB(ex(N1,*,ex(N2,*,ex(N3,*,ex(N4,*,N5))))).

%Heuristic two-------------------------------------------------------------------------------------------------------------
----
situation2 :-
        problem(numbers(N1,N2,N3,N4,N5),goal(G)),
        member(G,[N1,N2,N3,N4,N5]),
        member(0,[N1,N2,N3,N4,N5]),
        not(G=0).

action2 :-
        problem(_,goal(G)),
        other_numbers(special(G),others(A,B,C,D)),
        addCryptoSolutionToKB(ex(G,+,ex(A,*,ex(B,*,ex(C,*,D))))).

%Heuristic three-----------------------------------------------------------------------------------------------------------
------
situation3 :-
        problem(_,goal(0)),
        doubleton.

action3 :-
        doubleton(doubleton(A,B),rest(C,D,E)),
        addCryptoSolutionToKB(ex(ex(A,-,B),*,ex(C,*,ex(D,*,E)))).

%Heuristic four-------------------------------------------------------------------------------------------------------------------
----
situation4 :-
        problem(numbers(N1,N2,N3,N4,N5),goal(G)),
        not(G=0),
        doubleton,
        member(G,[N1,N2,N3,N4,N5]).

action4 :-
        problem(_,goal(G)),
        doubleton(doubleton(A,B),rest(C,D,E)),
        delete_one(G,[C,D,E],[X,Y]),
        addCryptoSolutionToKB(ex(G,+,ex(ex(A,-,B),*,ex(X,*,Y)))).

%Heuristic five-------------------------------------------------------------------------------------------------------------------
---
situation5 :-
        problem(_,goal(G)),
        not(G=0),
        doubleton(doubleton(_,_),rest(C,D,E)),
        makeGoalFromThree(goal(G),numsUsed(C,D,E)).

action5 :-
```

```
        problem(_,goal(G)),

        doubleton(doubleton(A,B),rest(C,D,E)),

        makeGoalFromThree(goal(G),numsUsed(C,D,E),Expression),

        addCryptoSolutionToKB(ex(ex(A,/,B),*,Expression)).


%Heuristic six------------------------------------------------------------------------------------------------------------------
--

situation6 :-

        problem(numbers(N1,N2,N3,N4,N5),goal(G)),

        G > 1,

        isHalf(G,[N1,N2,N3,N4,N5],[A,B,C,D]),

        makeGoalFromFour(2,numsUsed(A,B,C,D)).


action6 :-

        problem(numbers(N1,N2,N3,N4,N5),goal(G)),

        isHalf(G,[N1,N2,N3,N4,N5],[A,B,C,D],Half),

        makeGoalFromFour(2,numsUsed(A,B,C,D),Expression),

        addCryptoSolutionToKB(ex(Half,*,Expression)).


%Heuristic seven------------------------------------------------------------------------------------------------------------------
------

situation7 :-

        problem(numbers(N1,N2,N3,N4,N5),goal(G)),

        G < 5,

        G > 0,

        isDouble(G,[N1,N2,N3,N4,N5],[A,B,C,D]),

        makeGoalFromFour(2,numsUsed(A,B,C,D)).


action7 :-
```

```
          problem(numbers(N1,N2,N3,N4,N5),goal(G)),

          isDouble(G,[N1,N2,N3,N4,N5],[A,B,C,D],Double),

          makeGoalFromFour(2,numsUsed(A,B,C,D),Expression),

          addCryptoSolutionToKB(ex(Double,/,Expression)).


%Heuristic eight-----------------------------------------------------------------------------------------------------------------
-----

situation8 :-

          problem(numbers(N1,N2,N3,N4,N5),goal(G)),

          oneLess(G,[N1,N2,N3,N4,N5],_,[A,B,C,D]),

          makeGoalFromFour(1,numsUsed(A,B,C,D)).


action8 :-

          problem(numbers(N1,N2,N3,N4,N5),goal(G)),

          oneLess(G,[N1,N2,N3,N4,N5],OneMoreThanGoal,[A,B,C,D]),

          makeGoalFromFour(1,numsUsed(A,B,C,D),Expression),

          addCryptoSolutionToKB(ex(OneMoreThanGoal,-,Expression)).
```