Individual DEMO:

Heuristic 1: – H1/Zeros

CODE

```
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),
        assert(solution(ex(N1,*,ex(N2,*,ex(N3,*,ex(N4,*,N5)))))).
```

DEMO

```
?- solve(numbers(0,4,7,5,3), goal(0)).

Problem: Numbers = {0, 4, 7, 5, 3} Goal = 0
considering rule 1 ...
application of rule 1 produces ( 0 * ( 4 * ( 7 * ( 5 * 3 ) ) ) )
true .

?- solve(numbers(1,4,7,0,3), goal(0)).

Problem: Numbers = {1, 4, 7, 0, 3} Goal = 0
considering rule 1 ...
application of rule 1 produces ( 1 * ( 4 * ( 7 * ( 0 * 3 ) ) ) )
true .

?- solve(numbers(1,0,7,2,3), goal(0)).

Problem: Numbers = {1, 0, 7, 2, 3} Goal = 0
considering rule 1 ...
application of rule 1 produces ( 1 * ( 0 * ( 7 * ( 2 * 3 ) ) ) )
true .
```

Heuristic 2: – Zero and Goal

CODE

```
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)),
        assert(solution(ex(G,+,ex(A,*,ex(B,*,ex(C,*,D)))))).
```

DEMO

```
?- solve(numbers(1,4,7,0,3), goal(3)).

Problem: Numbers = {1, 4, 7, 0, 3} Goal = 3
considering rule 1 ...
considering rule 2 ...
application of rule 2 produces ( 3 + ( 1 * ( 4 * ( 7 * 0 ) ) ) )
true .

?- solve(numbers(1,4,8,0,3), goal(8)).

Problem: Numbers = {1, 4, 8, 0, 3} Goal = 8
considering rule 1 ...
considering rule 2 ...
application of rule 2 produces ( 8 + ( 1 * ( 4 * ( 0 * 3 ) ) ) )
true .

?- solve(numbers(1,4,5,0,3), goal(5)).

Problem: Numbers = {1, 4, 5, 0, 3} Goal = 5
considering rule 1 ...
considering rule 2 ...
application of rule 2 produces ( 5 + ( 1 * ( 4 * ( 0 * 3 ) ) ) )
true .
```

Heuristic 3: – Zero Goal and Pair

CODE

```
situation3 :-
        problem(_,goal(0)),
        doubleton.
action3 :-
        doubleton(doubleton(A,B),rest(C,D,E)),
        assert(solution(ex(ex(A,-,B),*,ex(C,*,ex(D,*,E))))).
```

DEMO

```
?- solve(numbers(1,4,7,3,3), goal(0)).

Problem: Numbers = {1, 4, 7, 3, 3} Goal = 0
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
application of rule 3 produces ( ( 3 - 3 ) * ( 1 * ( 4 * 7 ) ) )
true .

?- solve(numbers(1,6,4,4,3), goal(0)).

Problem: Numbers = {1, 6, 4, 4, 3} Goal = 0
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
application of rule 3 produces ( ( 4 - 4 ) * ( 1 * ( 6 * 3 ) ) )
true .

?- solve(numbers(1,4,7,1,3), goal(0)).

Problem: Numbers = {1, 4, 7, 1, 3} Goal = 0
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
application of rule 3 produces ( ( 1 - 1 ) * ( 4 * ( 7 * 3 ) ) )
true .
```

Heuristic 4: – Same and Goal Same
CODE

```
situation4 :-
        problem(numbers(N1,N2,N3,N4,N5),goal(G)),
        N1=N2,N2=N3,N3=N4,
        N5=G.
action4 :-
        problem(numbers(N1,N2,N3,N4,N5),_),
         assert(solution(ex(N5,+,ex(ex(N1,-,N2),+,ex(N3,-,N4))))).
```

DEMO

```
?- solve(numbers(4,4,4,4,4), goal(4)).

Problem: Numbers = {4, 4, 4, 4, 4} Goal = 4
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
application of rule 4 produces ( 4 + ( ( 4 - 4 ) + ( 4 - 4 ) ) )
true .

?- solve(numbers(7,7,7,7,7), goal(7)).
```

```
Problem: Numbers = {7, 7, 7, 7, 7} Goal = 7
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
application of rule 4 produces ( 7 + ( ( 7 - 7 ) + ( 7 - 7 ) ) )
true .

?- solve(numbers(3,3,3,3,3), goal(3)).

Problem: Numbers = {3, 3, 3, 3, 3} Goal = 3
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
application of rule 4 produces ( 3 + ( ( 3 - 3 ) + ( 3 - 3 ) ) )
true .
```

Heuristic 5: – Same and Goal One

CODE

```
situation5 :-
        problem(numbers(N1,N2,N3,N4,N5),goal(G)),
        N1=N2,N2=N3,N3=N4,N4=N5,
        1=G.
action5 :-
        problem(numbers(N1,N2,N3,N4,N5),_),
         assert(solution(ex(ex(N1,/,N2),+,ex(ex(N3,-,N4),*,N5)))).
```

DEMO

```
?- solve(numbers(3,3,3,3,3), goal(1)).

Problem: Numbers = {3, 3, 3, 3, 3} Goal = 1
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
application of rule 5 produces ( ( 3 / 3 ) + ( ( 3 - 3 ) * 3 ) )
true .

?- solve(numbers(2,2,2,2,2), goal(1)).

Problem: Numbers = {2, 2, 2, 2, 2} Goal = 1
```

```
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
application of rule 5 produces ( ( 2 / 2 ) + ( ( 2 - 2 ) * 2 ) )
true .

?- solve(numbers(9,9,9,9,9), goal(1)).

Problem: Numbers = {9, 9, 9, 9, 9} Goal = 1
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
application of rule 5 produces ( ( 9 / 9 ) + ( ( 9 - 9 ) * 9 ) )
true .
```

Heuristic 6: – Pair, Three make zero and Goal One

CODE

```
situation6 :-
        problem(_,goal(G)),
       G=1,
        doubleton(_,rest(C,D,E)),
        crypto(C,D,E,0,_).
action6 :-
        doubleton(doubleton(A,B),rest(C,D,E)),
        crypto(C,D,E,0,ThreeMakeZero),
        assert(solution(ex(ex(A,/,B),+,ThreeMakeZero))).
```

DEMO

```
?- solve(numbers(9,9,4,2,6), goal(1)).

Problem: Numbers = {9, 9, 4, 2, 6} Goal = 1
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
application of rule 6 produces ( ( 9 / 9 ) + ( ( 4 + 2 ) - 6 ) )
true .

?- solve(numbers(6,4,3,7,6), goal(1)).
```

```
Problem: Numbers = {6, 4, 3, 7, 6} Goal = 1
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
application of rule 6 produces ( ( 6 / 6 ) + ( ( 4 + 3 ) - 7 ) )
true .

?- solve(numbers(4,4,5,1,4), goal(1)).

Problem: Numbers = {4, 4, 5, 1, 4} Goal = 1
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
application of rule 6 produces ( ( 4 / 4 ) + ( ( 5 - 1 ) - 4 ) )
true .
```

Heuristic 7: – Pair and Rest Goal

CODE

```
situation7 :-
        problem(_,goal(G)),
        doubleton(doubleton(A,B),rest(C,D,E)),

can_make_goal_with_three_numbers(the_three(C,D,E),the_rest(A,B),the_goal(
G)).
action7 :-
        problem(_,goal(G)),
        doubleton(doubleton(A,B),rest(C,D,E)),
        crypto(C,D,E,G,ThreeMakeGoal),
        assert(solution(ex(ex(A,/,B),*,ThreeMakeGoal))).
```

DEMO

```
?- solve(numbers(4,4,5,1,4), goal(8)).

Problem: Numbers = {4, 4, 5, 1, 4} Goal = 8
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
application of rule 7 produces ( ( 4 / 4 ) * ( ( 5 - 1 ) + 4 ) )
true .

?- solve(numbers(3,1,2,3,4), goal(7)).

Problem: Numbers = {3, 1, 2, 3, 4} Goal = 7
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
application of rule 7 produces ( ( 3 / 3 ) * ( 4 + ( 1 + 2 ) ) )
true .

?- solve(numbers(8,1,8,3,2), goal(6)).

Problem: Numbers = {8, 1, 8, 3, 2} Goal = 6
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
application of rule 7 produces ( ( 8 / 8 ) * ( 2 + ( 1 + 3 ) ) )
true .
```

Heuristic 8: – Pair and Goal
CODE

```
situation8 :-
        problem(numbers(A,B,C,D,E),goal(G)),
        doubleton(doubleton(A,B),rest(C,D,E)),
        member(G,[C,D,E]).
action8 :-
        problem(_,goal(G)),
        doubleton(doubleton(A,B),rest(C,D,E)),
        make(C,D,E,G,X,Y),
        assert(solution(ex(G,+,ex(ex(A,-,B),*,ex(X,+,Y))))).
```

DEMO

```
?- solve(numbers(3,3,5,4,7), goal(5)).

Problem: Numbers = {3, 3, 5, 4, 7} Goal = 5
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
considering rule 8 ...
application of rule 8 produces ( 5 + ( ( 3 - 3 ) * ( 4 + 7 ) ) )
true .

?- solve(numbers(3,3,5,4,7), goal(4)).

Problem: Numbers = {3, 3, 5, 4, 7} Goal = 4
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
considering rule 8 ...
application of rule 8 produces ( 4 + ( ( 3 - 3 ) * ( 5 + 7 ) ) )
true .

?- solve(numbers(2,2,5,1,3), goal(5)).

Problem: Numbers = {2, 2, 5, 1, 3} Goal = 5
considering rule 1 ...
considering rule 2 ...
considering rule 3 ...
considering rule 4 ...
considering rule 5 ...
considering rule 6 ...
considering rule 7 ...
considering rule 8 ...
application of rule 8 produces ( 5 + ( ( 2 - 2 ) * ( 1 + 3 ) ) )
true .
```