Cog 366/CSC 366: Computational Modelling of Cognitive Processes Exam 2

Jacob Peck

October 30, 2011

Abstract

A ten-part exam covering heuristic generation in the context of solving order-5 Crypto problems in Prolog. Includes a brief discussion of heuristic problem solving, examples of heuristics in every-day life (if you happen to live the life of a programmer), 8 unique heuristics capable of solving order-5 Crypto problems, an examination of the applicability of these heuristics in isolation and in a pairwise fashion, a discussion on the cognitive effort of applying these heuristics, a comparison between them, code and demos for each heuristic, and a reflective evaluation essay discussing the final resulting program.

Contents

1 Heuristic Problem Solving **3**

2 Examples of Heuristics **4**

3 Crypto Heuristics **5**

4 Applicability of the Individual Heuristics **9**

5 Pairwise Applicability of the Heuristics **18**

6 Heuristic Executability **19**

7 Comparing the Heuristics **21**

8 Programming **22**

9 Heuristic Problem Solving Demo **30**

10 Reflective Evaluation Essay **31**

1 Heuristic Problem Solving

Heuristic problem solving is an approach to finding solutions to problems by the application of experience-based rules. These rules (heuristics) form the basis of a heuristic approach, and often take the form of "if (x) then [y]" where x is some verifiable situation or state the system in question may possess or be in, and y is some action to perform if x is satisfied. Most often, y is a direct solution to the problem, however in some cases, y may simply be a means to inch closer towards the goal, in effect becoming a method of achieving a subgoal.

Steve Pavlina states "Heuristics are rules intended to help you solve problems. When a problem is large or complex, and the optimal solution is unclear, applying a heuristic allows you to begin making progress towards a solution even though you can't visualize the entire path from your starting point." (`http://www.stevepavlina.com/blog/2007/05/33-rules-to-boost-your-productivity/`). This is a very good way of thinking about heuristics, as they occur everywhere in life. People apply heuristics to situations in their lives on a daily basis, and often completely unconsciously. These rules tend to be experience-based, given that if something worked relatively well in the past, there's a fairly decent chance that it will work again in the future. The same thoughts drive several approaches to problem solving in computer science. In the realm of artificial intelligence, a common undergraduate assignment is to implement a heuristic-based game playing agent. In the world of optimizations and efficiency, algorithms are often designed with heuristic thinking in mind.

Heuristics researcher Douglas Hofstadter's `Jumbo` and `Numbo` programs exhibit heuristically grounded approaches towards problem solving, despite being described by the author as something more biological than psychological (Douglas Hofstadter, *Fluid Concepts and Creative Analogies*). Hofstadter is just one among many of the researchers involved in heuristics.

2 Examples of Heuristics

1. (Recreational game: tic-tac-toe) IF (you may make three-in-a-row by playing your piece), THEN [play your piece to make three-in-a-row].

2. (Recreational game: tic-tac-toe) IF (you are to play AND your opponent can win on their next turn if you don't block them) THEN [play your piece in such a way as to block their three-in-a-row].

3. (Computer programming) IF (you are programming in Java AND you have to build a string inside a loop) THEN [use a `StringBuilder` object rather than concatenating with the + operator to increase speed].

4. (Computer programming) IF (you are programming in Lisp AND you need to use a map-like structure) THEN [use either A-lists, P-lists, or perform extractions upon a list created by mapcar].

5. (Real life) IF (you are trying to find your car in a parking lot AND you have a key-fob) THEN [constantly press the lock button while moving up and down the rows, listening for a horn].

6. (Real life) IF (you have all 8:00 AM classes AND you typically stay up far too late) THEN [set two alarms for the morning].

7. (Bonus) IF (you are programming in Io AND you want to write a book on beginning programming) THEN [write several data structures and concurrency utilities in Io to further familiarize yourself with the language].

3 Crypto Heuristics

Heuristic One

Number/Name H1 – Zeros
English If the goal is zero and zero is among the numbers, then multiply all of the numbers together.
Pseudocode
if (the goal is zero) and (zero is among the numbers)
then [multiply the numbers together]
Examples

- numbers = {5,4,0,8,9} goal = 0
  (5 * (4 * (0 * (8 * 9))))
- numbers = {0,5,3,5,8} goal = 0
  (0 * (5 * (3 * (5 * 8))))
- numbers = {7,6,7,1,0} goal = 0
  (7 * (6 * (7 * (1 * 0))))

Heuristic Two

Number/Name H2 – Zero and Goal
English If the goal is nonzero and zero and the goal are among the numbers, then add the goal to the result of multiplying all of the other numbers together.
Pseudocode
if (the goal is not zero) and (zero is among the numbers) and (the goal is among the numbers)
then [add the goal to the product of the remaining numbers]
Examples

- numbers = {7,0,9,2,6} goal = 9
  (9 + (7 * (0 * (2 * 6))))
- numbers = {5,4,3,1,0} goal = 4
  (4 + (5 * (3 * (1 * 0))))
- numbers = {0,2,3,5,3} goal = 3
  (3 + (0 * (2 * (3 * 5))))

Heuristic Three

Number/Name H3 – Zero Goal and Pair
English If the goal is zero and zero and a pair exists among the numbers, then multiply the difference between the pair of numbers by all of the remaining numbers.

Pseudocode

if (the goal is zero) and (a pair exists among the numbers)

then [multiply the difference between the pair of numbers by all of the remaining numbers]

Examples

- numbers = {4,5,6,4,9} goal = 0
  ((4 - 4) * (5 * (6 * 9)))

- numbers = {5,0,6,0,7} goal = 0
  ((0 - 0) * (5 * (6 * 7)))

- numbers = {1,0,1,2,3} goal = 0
  ((1 - 1) * (0 * (2 * 3)))

Heuristic Four

Number/Name H4 – One Goal, Zero and Neighbors

English If the goal is one and there is a set of neighbors in the numbers and there is also a zero in the numbers, then add the positive difference between the neighbors to the product of the rest of the numbers.

Pseudocode

if (the goal is one) and (a set of neighbors exists among the numbers) and (zero exists among the numbers)

then [add the positive difference between the neighbor numbers to the product of all of the remaining numbers]

Examples

- numbers = {4,5,0,4,9} goal = 1
  ((5 - 4) + (0 * (4 * 9)))

- numbers = {5,0,6,0,7} goal = 1
  ((6 - 5) + (0 * (0 * 7)))

- numbers = {1,0,1,2,3} goal = 1
  ((2 - 1) + (0 * (1 * 3)))

Heuristic Five

Number/Name H5 – Nonzero Goal, Goal, No Zero, Non-goal Pair

English If the goal is not zero and the goal is among the numbers and the numbers do not contain zero and there is a pair among the numbers which does not match the goal, then add the goal to the product found by multiplying the difference between the pair with the rest of the numbers.

Pseudocode

if (the goal is not zero) and (a pair exists among the numbers) and (the pair does not match the goal) and (zero does not exist among the numbers) and (the goal is among the numbers)

then [add the goal to the product of the difference between the pair multiplied by all of the remaining numbers]

Examples

- numbers = {1,1,2,3,5} goal = 5
  (5 + ((1 - 1) * (2 * 3)))

- numbers = {2,3,7,5,2} goal = 3
  (3 + ((2 - 2) * (7 * 5)))

- numbers = {5,8,4,8,6} goal = 6
  (6 + ((8 - 8) * (5 * 4)))

Heuristic Six

Number/Name H6 – One Goal, Zero and Pair

English If the goal is one and zero and a pair are among the numbers, then add the quotient of the pair to the product of the rest of the numbers.

Pseudocode

if (the goal is one) and (a pair exists among the numbers) and (zero is among the numbers)

then [add the quotient of the pair to the product of the rest of the numbers]

Examples

- numbers = {1,1,2,0,5} goal = 1
  ((1 / 1) + (2 * (0 * 5)))

- numbers = {2,0,7,5,2} goal = 1
  ((2 / 2) + (0 * (7 * 5)))

- numbers = {5,8,4,8,0} goal = 1
  ((8 / 8) + (4 * (5 * 0)))

Heuristic Seven

Number/Name H7 – Two Goal, Pair of Ones, Zero

English If the goal is two and zero is among the numbers and a pair of ones is among the numbers, then add the sum of the pair to the product of the rest of the numbers.

Pseudocode

if (the goal is two) and (a pair exists among the numbers) and (the pair values are one) and (zero is among the numbers)

then [add the sum of the pair to the product of the rest of the numbers]

Examples

- numbers = {1,1,2,0,5} goal = 2
  ((1 + 1) + (2 * (0 * 5)))
- numbers = {2,0,1,1,2} goal = 2
  ((1 + 1) + (2 * (0 * 2)))
- numbers = {5,1,1,8,0} goal = 2
  ((1 + 1) + (5 * (8 * 0)))

Heuristic Eight

Number/Name H8 – Four Goal, Pair of Twos, Zero
English If the goal is four and zero is among the numbers and a pair of twos is among the numbers, then add the sum of the pair to the product of the rest of the numbers.
Pseudocode
if (the goal is four) and (a pair exists among the numbers) and (the pair values are two) and (zero is among the numbers)
then [add the sum of the pair to the product of the rest of the numbers]
Examples

- numbers = {6,2,2,0,5} goal = 4
  ((2 + 2) + (6 * (0 * 5)))
- numbers = {2,0,1,1,2} goal = 4
  ((2 + 2) + (0 * (1 * 1)))
- numbers = {2,1,2,8,0} goal = 2
  ((2 + 2) + (1 * (8 * 0)))

4 Applicability of the Individual Heuristics

H1

```
?- probability([situation1],50,testing,P).
Problem: numbers = {2,5,4,7,8} and goal = 1
Problem: numbers = {9,5,9,2,1} and goal = 4
Problem: numbers = {3,9,2,1,7} and goal = 9
Problem: numbers = {2,4,3,7,8} and goal = 4
Problem: numbers = {4,7,0,5,6} and goal = 4
Problem: numbers = {7,4,7,2,8} and goal = 8
Problem: numbers = {8,5,5,7,2} and goal = 1
Problem: numbers = {5,4,3,0,9} and goal = 1
Problem: numbers = {1,0,0,1,8} and goal = 2
Problem: numbers = {7,1,4,0,7} and goal = 6
Problem: numbers = {7,0,1,4,7} and goal = 8
Problem: numbers = {3,6,7,4,9} and goal = 7
Problem: numbers = {2,4,7,7,3} and goal = 3
Problem: numbers = {9,9,2,9,6} and goal = 0
Problem: numbers = {6,1,2,1,1} and goal = 4
Problem: numbers = {2,6,4,3,4} and goal = 5
Problem: numbers = {1,2,7,4,9} and goal = 5
Problem: numbers = {4,7,9,1,8} and goal = 7
Problem: numbers = {1,6,9,0,0} and goal = 1
Problem: numbers = {0,2,7,2,4} and goal = 2
Problem: numbers = {9,5,8,2,8} and goal = 0
Problem: numbers = {3,8,4,9,2} and goal = 4
Problem: numbers = {3,2,2,6,6} and goal = 1
Problem: numbers = {2,1,0,0,9} and goal = 2
Problem: numbers = {8,2,5,1,0} and goal = 3
Problem: numbers = {2,7,9,3,7} and goal = 7
Problem: numbers = {5,2,3,8,5} and goal = 9
Problem: numbers = {1,4,0,4,1} and goal = 8
Problem: numbers = {4,9,3,2,3} and goal = 2
Problem: numbers = {6,4,3,5,3} and goal = 1
Problem: numbers = {7,5,3,6,2} and goal = 9
Problem: numbers = {8,3,4,2,1} and goal = 1
Problem: numbers = {4,9,1,8,6} and goal = 8
Problem: numbers = {4,6,1,6,6} and goal = 1
Problem: numbers = {2,7,1,4,3} and goal = 9
Problem: numbers = {8,6,2,7,2} and goal = 4
Problem: numbers = {9,2,3,3,2} and goal = 6
Problem: numbers = {2,7,2,3,1} and goal = 4
Problem: numbers = {2,0,9,4,7} and goal = 3
Problem: numbers = {7,4,9,2,2} and goal = 9
Problem: numbers = {3,0,3,1,1} and goal = 2
Problem: numbers = {8,1,9,9,3} and goal = 5
Problem: numbers = {4,7,5,2,1} and goal = 9
Problem: numbers = {8,3,2,5,9} and goal = 0
Problem: numbers = {2,6,8,5,4} and goal = 1
Problem: numbers = {0,2,7,9,2} and goal = 9
Problem: numbers = {6,7,7,4,9} and goal = 6
Problem: numbers = {8,3,6,0,1} and goal = 0 *
Problem: numbers = {9,0,8,9,2} and goal = 8
Problem: numbers = {2,4,1,3,9} and goal = 3

P = 0.02
?- probability([situation1],100000,result,P).

P = 0.04241
?- probability([situation1],100000,result,P).

P = 0.0407
```

H2

```
?- probability([situation2],50,testing,P).
Problem: numbers = {9,9,4,1,9} and goal = 5
Problem: numbers = {5,5,3,3,9} and goal = 1
Problem: numbers = {6,6,0,8,5} and goal = 4
Problem: numbers = {9,1,8,9,3} and goal = 4
Problem: numbers = {5,8,6,8,2} and goal = 1
Problem: numbers = {2,5,5,0,5} and goal = 7
Problem: numbers = {2,3,0,6,0} and goal = 2 *
Problem: numbers = {4,3,5,4,6} and goal = 1
Problem: numbers = {9,1,1,1,7} and goal = 5
Problem: numbers = {0,2,6,0,0} and goal = 6 *
Problem: numbers = {3,2,3,3,7} and goal = 4
Problem: numbers = {1,7,5,6,4} and goal = 4
Problem: numbers = {1,2,0,0,3} and goal = 3 *
Problem: numbers = {4,2,1,3,4} and goal = 0
Problem: numbers = {0,2,6,8,0} and goal = 7
Problem: numbers = {4,1,8,7,7} and goal = 3
Problem: numbers = {2,2,1,3,4} and goal = 0
Problem: numbers = {0,0,6,2,5} and goal = 5 *
Problem: numbers = {1,4,0,4,9} and goal = 6
Problem: numbers = {5,0,0,9,5} and goal = 5 *
Problem: numbers = {2,0,0,1,9} and goal = 7
Problem: numbers = {8,2,0,9,6} and goal = 7
Problem: numbers = {8,8,4,7,5} and goal = 8
Problem: numbers = {5,5,2,4,3} and goal = 3
Problem: numbers = {9,2,0,2,1} and goal = 9 *
Problem: numbers = {1,5,7,6,0} and goal = 5 *
Problem: numbers = {3,5,3,7,2} and goal = 3
Problem: numbers = {4,6,2,1,6} and goal = 0
Problem: numbers = {5,3,9,6,0} and goal = 8
Problem: numbers = {7,6,5,8,4} and goal = 4
Problem: numbers = {1,8,1,0,0} and goal = 6
Problem: numbers = {1,9,7,4,5} and goal = 8
Problem: numbers = {0,5,6,7,2} and goal = 8
Problem: numbers = {2,6,2,0,1} and goal = 7
Problem: numbers = {0,8,5,7,4} and goal = 6
Problem: numbers = {8,3,1,7,1} and goal = 7
Problem: numbers = {7,2,5,8,2} and goal = 1
Problem: numbers = {5,2,1,7,6} and goal = 4
Problem: numbers = {0,8,7,0,7} and goal = 1
Problem: numbers = {5,4,0,6,2} and goal = 0
Problem: numbers = {9,8,0,8,0} and goal = 6
Problem: numbers = {0,3,4,0,5} and goal = 1
Problem: numbers = {4,7,8,1,1} and goal = 9
Problem: numbers = {3,6,8,7,6} and goal = 7
Problem: numbers = {1,6,8,4,6} and goal = 1
Problem: numbers = {3,4,2,0,8} and goal = 8 *
Problem: numbers = {8,3,1,3,4} and goal = 8
Problem: numbers = {9,6,1,4,9} and goal = 8
Problem: numbers = {8,7,0,9,6} and goal = 9 *
Problem: numbers = {2,7,9,7,8} and goal = 7

P = 0.18
?- probability([situation2],100000,result,P).

P = 0.13048
?- probability([situation2],100000,result,P).

P = 0.13367
```

H3

```
?- probability([situation3],50,testing,P).
Problem: numbers = {3,4,9,9,1} and goal = 7
Problem: numbers = {3,6,4,6,7} and goal = 7
Problem: numbers = {2,6,0,8,0} and goal = 5
Problem: numbers = {8,1,3,9,4} and goal = 1
Problem: numbers = {8,5,9,6,2} and goal = 6
Problem: numbers = {6,4,0,8,1} and goal = 7
Problem: numbers = {4,7,1,2,5} and goal = 2
Problem: numbers = {5,1,0,6,5} and goal = 9
Problem: numbers = {9,0,6,1,0} and goal = 9
Problem: numbers = {4,1,6,4,9} and goal = 0 *
Problem: numbers = {9,4,4,3,5} and goal = 7
Problem: numbers = {6,3,7,7,4} and goal = 0 *
Problem: numbers = {0,1,3,8,7} and goal = 8
Problem: numbers = {8,1,3,5,2} and goal = 9
Problem: numbers = {1,3,4,7,0} and goal = 8
Problem: numbers = {7,0,4,8,3} and goal = 5
Problem: numbers = {5,1,8,4,2} and goal = 9
Problem: numbers = {0,1,0,4,3} and goal = 9
Problem: numbers = {0,9,9,2,2} and goal = 0 *
Problem: numbers = {4,9,0,9,9} and goal = 2
Problem: numbers = {8,5,9,0,5} and goal = 8
Problem: numbers = {9,6,4,7,0} and goal = 4
Problem: numbers = {5,1,5,9,4} and goal = 0 *
Problem: numbers = {2,3,5,8,8} and goal = 8
Problem: numbers = {5,3,8,3,0} and goal = 2
Problem: numbers = {9,9,1,9,2} and goal = 4
Problem: numbers = {4,0,2,8,8} and goal = 1
Problem: numbers = {0,0,3,3,1} and goal = 4
Problem: numbers = {0,1,6,6,3} and goal = 1
Problem: numbers = {7,7,4,2,0} and goal = 4
Problem: numbers = {4,3,2,8,3} and goal = 3
Problem: numbers = {5,1,7,6,2} and goal = 8
Problem: numbers = {5,6,0,4,6} and goal = 2
Problem: numbers = {2,3,8,6,2} and goal = 0 *
Problem: numbers = {9,2,6,8,4} and goal = 6
Problem: numbers = {8,6,2,4,4} and goal = 9
Problem: numbers = {0,3,6,0,1} and goal = 9
Problem: numbers = {7,9,0,9,2} and goal = 0 *
Problem: numbers = {7,2,0,3,6} and goal = 9
Problem: numbers = {2,3,8,9,9} and goal = 5
Problem: numbers = {1,0,7,4,8} and goal = 3
Problem: numbers = {1,7,6,7,6} and goal = 4
Problem: numbers = {9,1,2,4,8} and goal = 8
Problem: numbers = {0,0,2,5,1} and goal = 1
Problem: numbers = {0,9,4,9,2} and goal = 0 *
Problem: numbers = {2,1,5,9,5} and goal = 5
Problem: numbers = {1,0,7,8,8} and goal = 4
Problem: numbers = {5,1,6,5,5} and goal = 9
Problem: numbers = {8,0,1,4,9} and goal = 3
Problem: numbers = {4,8,8,2,0} and goal = 5

P = 0.14
?- probability([situation3],100000,result,P).

P = 0.06958
?- probability([situation3],100000,result,P).

P = 0.06978
```

H4

```
?- probability([situation4],50,testing,P).
Problem: numbers = {1,0,0,1,8} and goal = 2
Problem: numbers = {7,1,4,0,7} and goal = 6
Problem: numbers = {7,0,1,4,7} and goal = 8
Problem: numbers = {3,6,7,4,9} and goal = 7
Problem: numbers = {2,4,7,7,3} and goal = 3
Problem: numbers = {9,9,2,9,6} and goal = 0
Problem: numbers = {6,1,2,1,1} and goal = 4
Problem: numbers = {2,6,4,3,4} and goal = 5
Problem: numbers = {1,2,7,4,9} and goal = 5
Problem: numbers = {4,7,9,1,8} and goal = 7
Problem: numbers = {1,6,9,0,0} and goal = 1 *
Problem: numbers = {0,2,7,2,4} and goal = 2
Problem: numbers = {9,5,8,2,8} and goal = 0
Problem: numbers = {3,8,4,9,2} and goal = 4
Problem: numbers = {3,2,2,6,6} and goal = 1
Problem: numbers = {2,1,0,0,9} and goal = 2
Problem: numbers = {8,2,5,1,0} and goal = 3
Problem: numbers = {2,7,9,3,7} and goal = 7
Problem: numbers = {5,2,3,8,5} and goal = 9
Problem: numbers = {1,4,0,4,1} and goal = 8
Problem: numbers = {4,9,3,2,3} and goal = 2
Problem: numbers = {6,4,3,5,3} and goal = 1
Problem: numbers = {7,5,3,6,2} and goal = 9
Problem: numbers = {8,3,4,2,1} and goal = 1
Problem: numbers = {4,9,1,8,6} and goal = 8
Problem: numbers = {4,6,1,6,6} and goal = 1
Problem: numbers = {2,7,1,4,3} and goal = 9
Problem: numbers = {8,6,2,7,2} and goal = 4
Problem: numbers = {9,2,3,3,2} and goal = 6
Problem: numbers = {2,7,2,3,1} and goal = 4
Problem: numbers = {2,0,9,4,7} and goal = 3
Problem: numbers = {7,4,9,2,2} and goal = 9
Problem: numbers = {3,0,3,1,1} and goal = 2
Problem: numbers = {8,1,9,9,3} and goal = 5
Problem: numbers = {4,7,5,2,1} and goal = 9
Problem: numbers = {8,3,2,5,9} and goal = 0
Problem: numbers = {2,6,8,5,4} and goal = 1
Problem: numbers = {0,2,7,9,2} and goal = 9
Problem: numbers = {6,7,7,4,9} and goal = 6
Problem: numbers = {8,3,6,0,1} and goal = 0
Problem: numbers = {9,0,8,9,2} and goal = 8
Problem: numbers = {2,4,1,3,9} and goal = 3
Problem: numbers = {0,3,2,5,0} and goal = 6
Problem: numbers = {6,1,9,2,2} and goal = 0
Problem: numbers = {7,4,7,9,1} and goal = 8
Problem: numbers = {6,5,9,1,6} and goal = 5
Problem: numbers = {1,7,0,0,9} and goal = 5
Problem: numbers = {0,8,0,4,6} and goal = 1
Problem: numbers = {3,6,7,5,2} and goal = 7
Problem: numbers = {4,0,4,3,2} and goal = 5

P = 0.02
?- probability([situation4],100000,result,P).

P = 0.02686
?- probability([situation4],100000,result,P).

P = 0.02797
```

H5

```
?- probability([situation5],50,testing,P).
Problem: numbers = {0,3,2,5,0} and goal = 6
Problem: numbers = {6,1,9,2,2} and goal = 0
Problem: numbers = {7,4,7,9,1} and goal = 8
Problem: numbers = {6,5,9,1,6} and goal = 5 *
Problem: numbers = {1,7,0,0,9} and goal = 5
Problem: numbers = {0,8,0,4,6} and goal = 1
Problem: numbers = {3,6,7,5,2} and goal = 7
Problem: numbers = {4,0,4,3,2} and goal = 5
Problem: numbers = {7,8,2,1,1} and goal = 3
Problem: numbers = {3,3,4,5,5} and goal = 6
Problem: numbers = {3,3,5,9,9} and goal = 5 *
Problem: numbers = {8,6,9,1,1} and goal = 6 *
Problem: numbers = {0,1,4,4,9} and goal = 3
Problem: numbers = {0,5,6,8,0} and goal = 9
Problem: numbers = {9,8,7,8,2} and goal = 0
Problem: numbers = {0,9,8,4,4} and goal = 9
Problem: numbers = {0,2,2,9,1} and goal = 7
Problem: numbers = {7,2,1,6,7} and goal = 0
Problem: numbers = {4,5,8,1,8} and goal = 4 *
Problem: numbers = {0,8,3,4,3} and goal = 5
Problem: numbers = {8,7,4,2,5} and goal = 0
Problem: numbers = {6,7,6,3,9} and goal = 9 *
Problem: numbers = {9,7,9,8,6} and goal = 7 *
Problem: numbers = {3,1,5,7,3} and goal = 6
Problem: numbers = {2,5,6,1,0} and goal = 3
Problem: numbers = {6,3,1,7,9} and goal = 0
Problem: numbers = {9,7,0,8,5} and goal = 6
Problem: numbers = {3,5,7,9,9} and goal = 7 *
Problem: numbers = {9,1,0,3,1} and goal = 3
Problem: numbers = {3,1,6,2,0} and goal = 0
Problem: numbers = {2,5,6,9,4} and goal = 1
Problem: numbers = {1,0,0,0,7} and goal = 6
Problem: numbers = {2,8,6,4,0} and goal = 3
Problem: numbers = {6,3,0,3,0} and goal = 1
Problem: numbers = {0,9,9,9,1} and goal = 2
Problem: numbers = {6,5,8,2,5} and goal = 7
Problem: numbers = {9,8,3,2,0} and goal = 4
Problem: numbers = {2,9,1,7,4} and goal = 5
Problem: numbers = {8,1,0,2,6} and goal = 6
Problem: numbers = {2,5,2,9,0} and goal = 9
Problem: numbers = {2,7,6,0,8} and goal = 9
Problem: numbers = {9,8,8,6,3} and goal = 6 *
Problem: numbers = {0,7,6,2,5} and goal = 9
Problem: numbers = {8,1,7,0,4} and goal = 0
Problem: numbers = {3,6,0,9,5} and goal = 4
Problem: numbers = {0,9,1,2,2} and goal = 4
Problem: numbers = {8,3,5,8,3} and goal = 9
Problem: numbers = {9,1,5,2,2} and goal = 4
Problem: numbers = {1,7,4,6,7} and goal = 4 *
Problem: numbers = {8,9,3,1,7} and goal = 5

P = 0.18
?- probability([situation5],100000,result,P).

P = 0.1264
?- probability([situation5],100000,result,P).

P = 0.12494
```

H6

```
?- probability([situation6],50,testing,P).
Problem: numbers = {2,6,3,9,2} and goal = 9
Problem: numbers = {8,8,1,8,0} and goal = 4
Problem: numbers = {7,7,5,4,2} and goal = 8
Problem: numbers = {6,0,9,0,4} and goal = 0
Problem: numbers = {0,2,7,9,1} and goal = 7
Problem: numbers = {0,3,2,1,3} and goal = 1 *
Problem: numbers = {8,1,4,1,7} and goal = 1
Problem: numbers = {6,3,7,4,6} and goal = 1
Problem: numbers = {5,7,8,4,3} and goal = 8
Problem: numbers = {9,0,0,3,5} and goal = 4
Problem: numbers = {2,0,6,7,9} and goal = 7
Problem: numbers = {8,8,3,6,5} and goal = 4
Problem: numbers = {1,3,5,1,8} and goal = 2
Problem: numbers = {0,3,0,9,0} and goal = 5
Problem: numbers = {1,6,8,6,5} and goal = 3
Problem: numbers = {1,9,6,4,1} and goal = 1
Problem: numbers = {9,2,7,8,9} and goal = 6
Problem: numbers = {3,8,8,4,0} and goal = 6
Problem: numbers = {0,5,7,2,4} and goal = 8
Problem: numbers = {8,9,2,3,2} and goal = 2
Problem: numbers = {9,8,8,9,4} and goal = 0
Problem: numbers = {4,6,0,0,3} and goal = 5
Problem: numbers = {5,9,2,6,6} and goal = 6
Problem: numbers = {2,8,9,2,1} and goal = 9
Problem: numbers = {8,4,3,7,8} and goal = 6
Problem: numbers = {1,9,6,6,2} and goal = 7
Problem: numbers = {8,1,7,7,1} and goal = 4
Problem: numbers = {6,4,4,7,5} and goal = 7
Problem: numbers = {3,4,3,0,4} and goal = 3
Problem: numbers = {1,7,5,1,0} and goal = 6
Problem: numbers = {6,2,5,4,1} and goal = 9
Problem: numbers = {5,1,4,9,8} and goal = 1
Problem: numbers = {5,8,2,1,6} and goal = 0
Problem: numbers = {3,3,8,4,6} and goal = 5
Problem: numbers = {4,0,3,6,0} and goal = 2
Problem: numbers = {0,0,8,3,6} and goal = 0
Problem: numbers = {6,9,8,2,7} and goal = 2
Problem: numbers = {7,4,9,4,4} and goal = 3
Problem: numbers = {9,4,5,4,6} and goal = 8
Problem: numbers = {3,4,5,3,8} and goal = 5
Problem: numbers = {3,3,8,9,1} and goal = 5
Problem: numbers = {9,6,6,4,1} and goal = 2
Problem: numbers = {0,0,6,7,5} and goal = 9
Problem: numbers = {3,1,8,3,9} and goal = 4
Problem: numbers = {7,0,7,4,8} and goal = 6
Problem: numbers = {5,8,5,6,2} and goal = 2
Problem: numbers = {8,9,4,2,7} and goal = 5
Problem: numbers = {1,4,8,7,3} and goal = 2
Problem: numbers = {3,7,5,2,0} and goal = 5
Problem: numbers = {2,7,7,6,4} and goal = 1

P = 0.02
?- probability([situation6],100000,result,P).

P = 0.01947
?- probability([situation6],100000,result,P).

P = 0.02
```

H7

```
?- probability([situation7],50,testing,P).
Problem: numbers = {2,5,4,7,8} and goal = 1
Problem: numbers = {9,5,9,2,1} and goal = 4
Problem: numbers = {3,9,2,1,7} and goal = 9
Problem: numbers = {2,4,3,7,8} and goal = 4
Problem: numbers = {4,7,0,5,6} and goal = 4
Problem: numbers = {7,4,7,2,8} and goal = 8
Problem: numbers = {8,5,5,7,2} and goal = 1
Problem: numbers = {5,4,3,0,9} and goal = 1
Problem: numbers = {1,0,0,1,8} and goal = 2 *
Problem: numbers = {7,1,4,0,7} and goal = 6
Problem: numbers = {7,0,1,4,7} and goal = 8
Problem: numbers = {3,6,7,4,9} and goal = 7
Problem: numbers = {2,4,7,7,3} and goal = 3
Problem: numbers = {9,9,2,9,6} and goal = 0
Problem: numbers = {6,1,2,1,1} and goal = 4
Problem: numbers = {2,6,4,3,4} and goal = 5
Problem: numbers = {1,2,7,4,9} and goal = 5
Problem: numbers = {4,7,9,1,8} and goal = 7
Problem: numbers = {1,6,9,0,0} and goal = 1
Problem: numbers = {0,2,7,2,4} and goal = 2
Problem: numbers = {9,5,8,2,8} and goal = 0
Problem: numbers = {3,8,4,9,2} and goal = 4
Problem: numbers = {3,2,2,6,6} and goal = 1
Problem: numbers = {2,1,0,0,9} and goal = 2
Problem: numbers = {8,2,5,1,0} and goal = 3
Problem: numbers = {2,7,9,3,7} and goal = 7
Problem: numbers = {5,2,3,8,5} and goal = 9
Problem: numbers = {1,4,0,4,1} and goal = 8
Problem: numbers = {4,9,3,2,3} and goal = 2
Problem: numbers = {6,4,3,5,3} and goal = 1
Problem: numbers = {7,5,3,6,2} and goal = 9
Problem: numbers = {8,3,4,2,1} and goal = 1
Problem: numbers = {4,9,1,8,6} and goal = 8
Problem: numbers = {4,6,1,6,6} and goal = 1
Problem: numbers = {2,7,1,4,3} and goal = 9
Problem: numbers = {8,6,2,7,2} and goal = 4
Problem: numbers = {9,2,3,3,2} and goal = 6
Problem: numbers = {2,7,2,3,1} and goal = 4
Problem: numbers = {2,0,9,4,7} and goal = 3
Problem: numbers = {7,4,9,2,2} and goal = 9
Problem: numbers = {3,0,3,1,1} and goal = 2 *
Problem: numbers = {8,1,9,9,3} and goal = 5
Problem: numbers = {4,7,5,2,1} and goal = 9
Problem: numbers = {8,3,2,5,9} and goal = 0
Problem: numbers = {2,6,8,5,4} and goal = 1
Problem: numbers = {0,2,7,9,2} and goal = 9
Problem: numbers = {6,7,7,4,9} and goal = 6
Problem: numbers = {8,3,6,0,1} and goal = 0
Problem: numbers = {9,0,8,9,2} and goal = 8
Problem: numbers = {2,4,1,3,9} and goal = 3

P = 0.04
?- probability([situation7],100000,result,P).

P = 0.00214
?- probability([situation7],100000,result,P).

P = 0.0024
```

H8

```
?- probability([situation8],50,testing,P).
Problem: numbers = {0,9,3,6,6} and goal = 2
Problem: numbers = {9,8,9,3,6} and goal = 9
Problem: numbers = {0,4,6,7,9} and goal = 7
Problem: numbers = {7,5,3,4,7} and goal = 8
Problem: numbers = {7,6,9,0,1} and goal = 2
Problem: numbers = {6,2,6,7,0} and goal = 0
Problem: numbers = {1,5,5,4,2} and goal = 6
Problem: numbers = {5,5,5,0,2} and goal = 7
Problem: numbers = {1,9,4,7,2} and goal = 8
Problem: numbers = {1,6,8,6,2} and goal = 6
Problem: numbers = {2,6,3,4,6} and goal = 7
Problem: numbers = {5,2,9,0,8} and goal = 9
Problem: numbers = {7,3,3,4,7} and goal = 4
Problem: numbers = {8,1,2,2,1} and goal = 2
Problem: numbers = {9,1,6,2,2} and goal = 7
Problem: numbers = {0,8,3,0,2} and goal = 7
Problem: numbers = {0,9,1,3,7} and goal = 5
Problem: numbers = {4,8,7,5,9} and goal = 4
Problem: numbers = {1,3,6,8,0} and goal = 5
Problem: numbers = {5,3,1,0,3} and goal = 1
Problem: numbers = {1,7,1,7,1} and goal = 1
Problem: numbers = {5,2,2,9,0} and goal = 6
Problem: numbers = {4,3,7,9,2} and goal = 6
Problem: numbers = {4,7,3,3,1} and goal = 4
Problem: numbers = {9,8,5,0,5} and goal = 0
Problem: numbers = {9,9,7,6,4} and goal = 1
Problem: numbers = {8,5,7,0,9} and goal = 8
Problem: numbers = {6,0,5,2,4} and goal = 3
Problem: numbers = {0,2,7,6,7} and goal = 3
Problem: numbers = {9,2,6,9,8} and goal = 9
Problem: numbers = {0,9,4,8,2} and goal = 4
Problem: numbers = {5,1,6,4,9} and goal = 8
Problem: numbers = {1,7,6,4,6} and goal = 1
Problem: numbers = {3,6,0,9,9} and goal = 5
Problem: numbers = {9,7,8,0,1} and goal = 2
Problem: numbers = {2,6,1,9,1} and goal = 6
Problem: numbers = {8,1,2,8,5} and goal = 5
Problem: numbers = {6,4,9,8,3} and goal = 0
Problem: numbers = {3,6,3,1,7} and goal = 5
Problem: numbers = {1,7,7,8,9} and goal = 2
Problem: numbers = {1,3,0,6,0} and goal = 6
Problem: numbers = {7,2,7,6,0} and goal = 8
Problem: numbers = {1,4,0,9,5} and goal = 6
Problem: numbers = {1,6,9,6,7} and goal = 1
Problem: numbers = {3,1,8,1,0} and goal = 1
Problem: numbers = {7,8,5,0,1} and goal = 8
Problem: numbers = {8,2,9,6,7} and goal = 4
Problem: numbers = {8,3,3,4,1} and goal = 7
Problem: numbers = {0,3,9,5,2} and goal = 9
Problem: numbers = {3,0,9,6,4} and goal = 6

P = 0
?- probability([situation8],100000,result,P).

P = 0.00244
?- probability([situation8],100000,result,P).

P = 0.00217
```

Theoretical applicability computation for H1

Number of possible Crypto problems $= 10^6 = 1,000,000$.
Number of problems where H1 is applicable $= 1 \times 1 \times 10^4 = 10^4 = 10000$.
Proportion of problems where H1 is applicable $= \frac{10^4}{10^6} = \frac{10000}{1,000,000} = \frac{1}{100} = 1\%$.
This result does not match the empirical results (showing roughly 4% applicability). This may be due to the way repeated digits factor in, or due to my poor grasp of how this calculation should be performed.

Theoretical applicability computation for H7

Number of possible Crypto problems $= 10^6 = 1,000,000$.
Number of problems where H7 is applicable $= 1 \times 1 \times 1 \times 1 \times 10^2 = 10^2 = 100$.
Proportion of problems where H1 is applicable $= \frac{10^2}{10^6} = \frac{100}{1,000,000} = \frac{1}{1000} = 0.1\%$.
This result does not match the empirical results (showing roughly 0.2% applicability). This may be due to the way repeated digits factor in, or due to my poor grasp of how this calculation should be performed.

5 Pairwise Applicability of the Heuristics

Empirical Pairwise Applicabilities of the Heuristics

The following data was collected using the same method as in the previous section, with 100,000 trials. A – in place of a value denotes duplicate data or inapplicable situation.

	H1	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**
H1	–	0.17457	0.08454	0.07011	0.16601	0.06166	0.04314	0.04302
H2	–	–	0.20164	0.14761	0.25685	0.14742	0.13498	0.13216
H3	–	–	–	0.09794	0.19429	0.09030	0.07252	0.07077
H4	–	–	–	–	0.15424	0.03709	0.03023	0.02997
H5	–	–	–	–	–	0.14534	0.12721	0.12760
H6	–	–	–	–	–	–	0.02288	0.02268
H7	–	–	–	–	–	–	–	0.00483
H8	–	–	–	–	–	–	–	–

Theoretical Problem-Domain Overlap Between Heuristics

The following table is strictly theoretical, and attempts to codify the degree to which any two heuristics act upon the same set of problems. The following code is used in the table: – denotes duplicate data or inapplicable, X denotes mutually exclusive, P denotes partial overlap, and A denotes that one heuristic is a subset of the other.

	H1	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**
H1	–	X	P	X	X	X	X	X
H2	–	–	X	P	X	A	A	A
H3	–	–	–	X	X	X	X	X
H4	–	–	–	–	X	X	X	X
H5	–	–	–	–	–	X	X	X
H6	–	–	–	–	–	–	X	X
H7	–	–	–	–	–	–	–	X
H8	–	–	–	–	–	–	–	–

6 Heuristic Executability

Approach for Determining the Executability of a Heuristic

The notion of heuristic executability is a rather ephemeral one, and one that cannot simply be worked out. As such, the approach outlined below and the results that follow are to be interpreted as strictly theoretical.

Determining the executability of a heuristic is broken down into two parts: determining applicability, and determining effort of application. These two scores are simply added together to form a composite score (hopefully) in the range $[0, 1]$. Each of these subscores is made up of a simple addition of component scores, representing different cognitive components towards the final action (determining applicability or applying the heuristic). These component scores are broken down as follows, with notation and value representing subjectively determined effort required.

- $G_{Non-Zero} = 0.01$ – Noticing the goal is non-zero
- $G_n = 0.01$ – Noticing the goal is some number n
- $C_n = 0.02$ – Noticing the numbers contain some number n
- $\neg C_n = 0.04$ – Noticing the numbers do not contain some number n
- $C_{pair-n} = 0.04$ – Noticing the numbers contain a pair of some number n
- $C_{adj} = 0.05$ – Noticing the numbers contain a set of two adjacent numbers
- $M_{x,y} = 0.03$ – Multiplying two numbers x and y together
- $D_{x,y} = 0.06$ – Dividing x by y
- $A_{x,y} = 0.02$ – Adding two numbers x and y together
- $S_{x,y} = 0.04$ – Subtracting y from x

Theoretical Executabilities of the Heuristics

H1

$$Executability(H1) = (G_0 + C_0) + (4 \times M_{x,y}) = (0.01 + 0.02) + (4 \times 0.03) = 0.03 + 0.12 = 0.15$$

H2

$$Executability(H2) = (G_{Non-Zero} + C_0 + C_{Goal}) + (3 \times M_{x,y} + A_{x,y}) = (0.01 + 0.02 + 0.02) + (3 \times 0.03 + 0.02) = 0.05 + 0.11 = 0.16$$

H3

$Executability(H3) = (G_0 + C_{pair-n}) + (3 \times M_{x,y} + S_{x,y}) = (0.01 + 0.04) + (3 \times 0.03 + 0.04) = 0.05 + 0.13 = 0.18$

H4

$Executability(H4) = (G_1 + C_0 + C_{adj}) + (2 \times M_{x,y} + A_{x,y} + S_{x,y}) = (0.01 + 0.02 + 0.05) + (2 \times 0.03 + 0.02 + 0.04) = 0.08 + 0.12 = 0.20$

H5

$Executability(H5) = (G_{Non-Zero} + \neg C_0 + C_{Goal} + C_{pair-non-Goal}) + (2 \times M_{x,y} + A_{x,y} + S_{x,y}) = (0.01 + 0.04 + 0.02 + 0.04) + (2 \times 0.03 + 0.02 + 0.04) = 0.11 + 0.12 = 0.23$

H6

$Executability(H6) = (G_1 + C_0 + C_{pair-n}) + (2 \times M_{x,y} + D_{x,y} + A_{x,y}) = (0.01 + 0.02 + 0.04) + (2 \times 0.03 + 0.06 + 0.02) = 0.07 + 0.14 = 0.21$

H7

$Executability(H7) = (G_2 + C_{pair-1} + C_0) + (2 \times M_{x,y} + 2 \times A_{x,y}) = (0.01 + 0.04 + 0.02) + (2 \times 0.03 + 2 \times 0.02) = 0.07 + 0.10 = 0.17$

H8

$Executability(H8) = (G_4 + C_{pair-2} + C_0) + (2 \times M_{x,y} + 2 \times A_{x,y}) = (0.01 + 0.04 + 0.02) + (2 \times 0.03 + 2 \times 0.02) = 0.07 + 0.10 = 0.17$

7 Comparing the Heuristics

Linear Ordering by Applicability, Descending

1. H2 (0.13)
2. H5 (0.12)
3. H3 (0.07)
4. H1 (0.04)
5. H4 (0.03)
6. H6 (0.02)
7. H8 (0.002)
8. H7 (0.002)

Linear Ordering by Executability, Descending

1. H1 (0.15)
2. H2 (0.16)
3. H7 (0.17)
4. H8 (0.17)
5. H3 (0.18)
6. H4 (0.20)
7. H6 (0.21)
8. H5 (0.23)

8 Programming

H1 – Zeros

Prolog code for the heuristic

```
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)))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(5,4,0,8,9),goal(0)).
Problem: numbers = {5,4,0,8,9} and goal = 0
considering rule 1 ...
application of rule 1 produces ( 5 * ( 4 * ( 0 * ( 8 * 9 ) ) ) )

Yes
?- solve(numbers(0,5,3,5,8),goal(0)).
Problem: numbers = {0,5,3,5,8} and goal = 0
considering rule 1 ...
application of rule 1 produces ( 0 * ( 5 * ( 3 * ( 5 * 8 ) ) ) )

Yes
?- solve(numbers(7,6,7,1,0),goal(0)).
Problem: numbers = {7,6,7,1,0} and goal = 0
considering rule 1 ...
application of rule 1 produces ( 7 * ( 6 * ( 7 * ( 1 * 0 ) ) ) )

Yes
```

H2 – Zero and Goal

Prolog code for the heuristic

```
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)))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(7,0,9,2,6),goal(9)).
Problem: numbers = {7,0,9,2,6} and goal = 9
considering rule 2 ...
application of rule 2 produces ( 9 + ( 7 * ( 0 * ( 2 * 6 ) ) ) )

Yes
?- solve(numbers(5,4,3,1,0),goal(4)).
Problem: numbers = {5,4,3,1,0} and goal = 4
considering rule 2 ...
application of rule 2 produces ( 4 + ( 5 * ( 3 * ( 1 * 0 ) ) ) )

Yes
?- solve(numbers(0,2,3,5,3),goal(3)).
Problem: numbers = {0,2,3,5,3} and goal = 3
considering rule 2 ...
application of rule 2 produces ( 3 + ( 0 * ( 2 * ( 5 * 3 ) ) ) )

Yes
```

H3 – Zero Goal and Pair

Prolog code for the heuristic

```
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))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(4,5,6,4,9),goal(0)).
Problem: numbers = {4,5,6,4,9} and goal = 0
considering rule 3 ...
application of rule 3 produces ( ( 4 - 4 ) * ( 5 * ( 6 * 9 ) ) )

Yes
?- solve(numbers(5,0,6,0,7),goal(0)).
Problem: numbers = {5,0,6,0,7} and goal = 0
considering rule 3 ...
application of rule 3 produces ( ( 0 - 0 ) * ( 5 * ( 6 * 7 ) ) )

Yes
?- solve(numbers(1,0,1,2,3),goal(0)).
Problem: numbers = {1,0,1,2,3} and goal = 0
considering rule 3 ...
application of rule 3 produces ( ( 1 - 1 ) * ( 0 * ( 2 * 3 ) ) )

Yes
```

H4 – One Goal, Zero and Neighbors

Prolog code for the heuristic

```
situation4 :-
  problem(_,goal(1)),
  other_numbers(special(0),others(N1,N2,N3,N4)),
  neighbors([N1,N2,N3,N4],neighbors(_,_),others(_,_)).

action4 :-
  problem(_,goal(1)),
  other_numbers(special(0),others(N1,N2,N3,N4)),
  neighbors([N1,N2,N3,N4],neighbors(A,B),others(C,D)),
  A is B+1,
  assert(solution(ex(ex(A,-,B),+,ex(0,*,ex(C,*,D))))).
action4 :-
  problem(_,goal(1)),
  other_numbers(special(0),others(N1,N2,N3,N4)),
  neighbors([N1,N2,N3,N4],neighbors(A,B),others(C,D)),
  B is A+1,
  assert(solution(ex(ex(B,-,A),+,ex(0,*,ex(C,*,D))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(4,5,0,4,9),goal(1)).
Problem: numbers = {4,5,0,4,9} and goal = 1
considering rule 4 ...
application of rule 4 produces ( ( 5 - 4 ) + ( 0 * ( 4 * 9 ) ) )

Yes
?- solve(numbers(5,0,6,0,7),goal(1)).
Problem: numbers = {5,0,6,0,7} and goal = 1
considering rule 4 ...
application of rule 4 produces ( ( 6 - 5 ) + ( 0 * ( 0 * 7 ) ) )

Yes
?- solve(numbers(1,0,1,2,3),goal(1)).
Problem: numbers = {1,0,1,2,3} and goal = 1
considering rule 4 ...
application of rule 4 produces ( ( 2 - 1 ) + ( 0 * ( 1 * 3 ) ) )

Yes
```

H5 – Nonzero Goal, Goal, No Zero, Non-goal Pair

Prolog code for the heuristic

```
situation5 :-
  problem(numbers(N1,N2,N3,N4,N5),goal(G)),
  not(G = 0),
  doubleton(doubleton(A,_),_),
  not(A = G),
  not(member(0,[N1,N2,N3,N4,N5],_)),
  member(G,[N1,N2,N3,N4,N5],_).

action5 :-
  problem(_,goal(G)),
  doubleton(doubleton(A,B),rest(G,C,D)),
  assert(solution(ex(G,+,ex(ex(A,-,B),*,ex(C,*,D))))).
action5 :-
  problem(_,goal(G)),
  doubleton(doubleton(A,B),rest(C,G,D)),
  assert(solution(ex(G,+,ex(ex(A,-,B),*,ex(C,*,D))))).
action5 :-
  problem(_,goal(G)),
  doubleton(doubleton(A,B),rest(C,D,G)),
  assert(solution(ex(G,+,ex(ex(A,-,B),*,ex(C,*,D))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(1,1,2,3,5),goal(5)).
Problem: numbers = {1,1,2,3,5} and goal = 5
considering rule 5 ...
application of rule 5 produces ( 5 + ( ( 1 - 1 ) * ( 2 * 3 ) ) )

Yes
?- solve(numbers(2,3,7,5,2),goal(3)).
Problem: numbers = {2,3,7,5,2} and goal = 3
considering rule 5 ...
application of rule 5 produces ( 3 + ( ( 2 - 2 ) * ( 7 * 5 ) ) )

Yes
?- solve(numbers(5,8,4,8,6),goal(6)).
Problem: numbers = {5,8,4,8,6} and goal = 6
considering rule 5 ...
application of rule 5 produces ( 6 + ( ( 8 - 8 ) * ( 5 * 4 ) ) )

Yes
```

H6 – One Goal, Zero and Pair

Prolog code for the heuristic

```
situation6 :-
  problem(numbers(N1,N2,N3,N4,N5),goal(G)),
  G = 1,
  doubleton(doubleton(A,_),_),
  not(A = 0),
  member(0,[N1,N2,N3,N4,N5]).

action6 :-
  doubleton(doubleton(A,B),rest(C,D,E)),
  assert(solution(ex(ex(A,/,B),+,ex(C,*,ex(D,*,E))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(1,1,2,0,5),goal(1)).
Problem: numbers = {1,1,2,0,5} and goal = 1
considering rule 6 ...
application of rule 6 produces ( ( 1 / 1 ) + ( 2 * ( 0 * 5 ) ) )

Yes
?- solve(numbers(2,0,7,5,2),goal(1)).
Problem: numbers = {2,0,7,5,2} and goal = 1
considering rule 6 ...
application of rule 6 produces ( ( 2 / 2 ) + ( 0 * ( 7 * 5 ) ) )

Yes
?- solve(numbers(5,8,4,8,0),goal(1)).
Problem: numbers = {5,8,4,8,0} and goal = 1
considering rule 6 ...
application of rule 6 produces ( ( 8 / 8 ) + ( 5 * ( 4 * 0 ) ) )

Yes
```

H7 – Two Goal, Pair of Ones, Zero

Prolog code for the heuristic

```
situation7 :-
  problem(numbers(N1,N2,N3,N4,N5),goal(G)),
  G = 2,
  doubleton(doubleton(A,_),_),
  A = 1,
  member(0,[N1,N2,N3,N4,N5]).

action7 :-
  doubleton(doubleton(A,B),rest(C,D,E)),
  A = 1,
  assert(solution(ex(ex(A,+,B),+,ex(C,*,ex(D,*,E))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(1,1,2,0,5),goal(2)).
Problem: numbers = {1,1,2,0,5} and goal = 2
considering rule 7 ...
application of rule 7 produces ( ( 1 + 1 ) + ( 2 * ( 0 * 5 ) ) )

Yes
?- solve(numbers(2,0,1,1,2),goal(2)).
Problem: numbers = {2,0,1,1,2} and goal = 2
considering rule 7 ...
application of rule 7 produces ( ( 1 + 1 ) + ( 2 * ( 0 * 2 ) ) )

Yes
?- solve(numbers(5,1,1,8,0),goal(2)).
Problem: numbers = {5,1,1,8,0} and goal = 2
considering rule 7 ...
application of rule 7 produces ( ( 1 + 1 ) + ( 5 * ( 8 * 0 ) ) )

Yes
```

H8 – Four Goal, Pair of Twos, Zero

Prolog code for the heuristic

```
situation8 :-
  problem(numbers(N1,N2,N3,N4,N5),goal(G)),
  G = 4,
  doubleton(doubleton(A,_),_),
  A = 2,
  member(0,[N1,N2,N3,N4,N5]).

action8 :-
  doubleton(doubleton(A,B),rest(C,D,E)),
  A = 2,
  assert(solution(ex(ex(A,+,B),+,ex(C,*,ex(D,*,E))))).
```

Prolog demo – three examples of the heuristic in use

```
?- solve(numbers(6,2,2,0,5),goal(4)).
Problem: numbers = {6,2,2,0,5} and goal = 4
considering rule 8 ...
application of rule 8 produces ( ( 2 + 2 ) + ( 6 * ( 0 * 5 ) ) )

Yes
?- solve(numbers(2,0,1,1,2),goal(4)).
Problem: numbers = {2,0,1,1,2} and goal = 4
considering rule 8 ...
application of rule 8 produces ( ( 2 + 2 ) + ( 0 * ( 1 * 1 ) ) )

Yes
?- solve(numbers(2,1,2,8,0),goal(4)).
Problem: numbers = {2,1,2,8,0} and goal = 4
considering rule 8 ...
application of rule 8 produces ( ( 2 + 2 ) + ( 1 * ( 8 * 0 ) ) )

Yes
```

9 Heuristic Problem Solving Demo

See my course page at `http://csc366.suspended-chord.info/` for these results.

10 Reflective Evaluation Essay

Standard Operating Procedure – A look at a Heuristic Crypto Problem Solver

Heuristically solving problems is an interesting branch of study. With applications in psychology, philosophy, mathematics, linguistics, cognitive science, artificial intelligence, human-computer interaction, and algorithmic problem decomposition, heuristic problem solving is a wide-reaching topic that can offer insights into many different facets of computing and the psychological and cognitive sciences, along with insights into your own decision-making process. The Crypto problem, wherein you are provided with a set of n numbers and a goal g, and you must reach the goal by mathematically manipulating the numbers using the four basic arithmetic operations, seems a perfect problem domain with which to analyze heuristics and their applications to problem solving. To that end, I developed (through a series of several course assignments) a rough sketch of a heuristic problem solver for Crypto problems of order 5, $[0, 9]$; that is, $n = 5$, and each number x and the goal g range between 0 and 9 inclusive. Also in the course of these assignments, a exhaustive problem solver for the same domain of Crypto problems was designed and will be compared against the heuristic method.

Heuristic problem solving is the attempt at solving problems by applying rules to situations based on experience. Note the use of the word “attempt” here—attempt is the major keyword here, as heuristic problem solving often fails, if the solver simply lacks enough rules to satisfactorily solve the problems presented to it. These rules, the heuristics, are often defined in the form of if (some verifiable condition) then [some action to perform], essentially acting as decision points in a traversal of some finite state machine—but this example is merely given for context, and should not be taken to heart as a very good, or all encompassing, analogy. A heuristic problem solver, then, is an agent which applies these heuristics to problems within its applicable domain, and by carrying out the actions encoded in the heuristics, provides a way for solving these problems.

Heuristics are a very good way of thinking of cognitive building blocks. Verifying conditionals (the first half of a heuristic) is a searching problem, followed by a comparison. The application of an action (the second half of a heuristic) is an execution and resolution event, in which the mind has to reconcile the new result with the state previously established in the initial search. This is then followed by a second, implicit search and compare to determine the efficacy of the heuristic relative to the problem just solved. All of this can be "chunked" together, as Miller would say (though appropriated here in a slightly different, but cognizant way), and thought of as one action. Indeed, it *is* one action, with several sub phases. It is this way of looking at the idea that drives many heuristic problem solving programming exercises, as it allows a direct one-to-one mapping between concept and code, given enough familiarity with whatever language the programmer is working in (the "medium") and the expressivity of the original problem description (the "source").

A standard approach to heuristic problem solving applies brute-search techniques in attempting to apply one heuristic after another after another in a deterministic way, until either the system has no more heuristics or the process is terminated by some outside force. This is not far from how the human mind attempts solving problems with heuristics. Oftentimes a human attempting to solve a problem draws from a bank of heuristics they know to have been successful in the past, and applies them (typically in order of increasing executability—humans are intrinsically either lazy or efficient... the jury is still out on this one, in my opinion) in a brute force fashion until one of them works. If they run out of heuristics, they either give up on the problem, or they roll into the failsafe mode of brute force search (killing hundreds of trees in the process, potentially, but that is another discussion). In coding these attempts, oftentimes a glimpse of the workings if the human mind is captured, and insights are found into personal thought processes.

My heuristic problem solver (as presented in the previous 30 pages) is applicable to around 37% of Crypto problems of order 5, $[0, 9]$ (empirically determined). This no great result in terms of absolute power and general applicability, however it represents a large chunk of problems which are able to be solved with a rather small set of rules (only the eight detailed in the previous sections of this exam). These rules are all rather simple rules, as well, with rather low theoretical executabilities (in fact, the executability metric may very well be over-exaggerating the difficulties of a few heuristics, based upon personal experience!). However simple the rules, a very important look at cognition is appropriated, however, as instead of just acting like a dumb brute, the machine approaches patterns that resemble thought, even if at an extremely low level. Compared to the exhaustive problem solver (which is applicable to 100% of the solvable order 5, $[0, 9]$ Crypto problems, which represents approximately 98.3% of the entire domain, empirically determined), the heuristic problem solver provides sub-optimal performance, however in returning results, more often than not the heuristic problem solver's approach is more consistent with the line of human reasoning that would be occurring during the problem solving process. While the exhaustive solver returns results more often, the heuristic problem solver makes more sense conceptually when the results are analyzed through the lens of human cognition.

In working with this project, I gained several glimmers of thoughts that I hadn't consciously paid attention to previously. Heuristic problem solving has been something I've been involved in for about a year and a half now, through various courses, but has been something that I've failed to see as anything other than large, branching if/then/else statements. Granted, to a large degree heuristics are simply that, but they offer views into mental processes and capacities for understanding in a new light. They redefine methodical and meticulous, they approach divine order in their absolution, and most of all, they are oblivious to the existence of other heuristics. They are atomic, thread-safe, immutable, players and roles, observers and listeners, and they just work. Lastly, the offer shortcuts and optimizations in existing algorithms, a very important aspect of computing in modern day applications, where literally thousands of things may happen at once. Heuristics are the key to solving many problems, and yet they're so delightfully simple so as to be overly obvious. In a word, heuristics are beautiful.