

Expt 3:- Constraint Specification

Problem:

Problem:- SEND + MORE = "MONEY"

* (Solve the following puzzle by assigning numeral (0-9) in such a way that each letter is assigned unique digit which satisfy addition).

(Problem Formulation)

Constraint:- No two letters have the same value. (The constraints of arithmetic).

$$\begin{array}{r} \text{SEND} \\ + \text{MORE} \\ \hline \end{array}$$

MONEY

Initial State:-

• $S = ?; E = ?; N = ?; D = ?; M = ?; O = ?; R = ?; Y = ?$

Y-1

• $c_4 \ c_3 \ c_2 \ c_1$ (array)

S E N D

(+) M O R E

M O N E Y

Algorithm:-

5 4 3 2 1
S E N D

+ M O R E

C3 C2 C1

M O N E Y

- (1) From column 5, $M=1$, ~~and~~ since it is only carry over possible from sum of 2. Since single digit number in column 4.
- (2) To produce a carry from column 5, $S+M$ is at least 9 so ' $S=8$ or 9 ' so ' $S+M=9$ or 10 ' & so ' $0=0$ or 1 '. But ' $M=1$ ', so ' $0=0$ '.
- (3) If there is carry from Column 3 to 4 then ' $E=9$ ' & so ' $N=0$ '. But ' $0=0$ ' so there is no carry & ' $S=9$ ' & ' $C3=0$ '.
- (4) If there is no carry from column 2 to 3 then ' $E=N$ ' which is impossible, therefore there is carry & ' $N=E+1$ ' & ' $E=1$ '.
- (5) If there is carry from column 1 to 2 then ' $N = k + E \bmod 10$ ' & ' $N = E+1$ ' so ' $E+1+R = E \bmod 10$ ' so ' $R=9$ ' but ' $S=9$ ' so there must be carry from column 1 to 2. Therefore, ' $0=1$ ' & ' $R=8$ '.

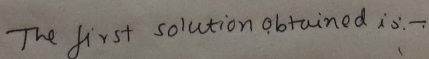
$$Y = D + \epsilon \longrightarrow C_1$$

$$E = N + R + C_1 \longrightarrow C_2$$

$$N = \varepsilon + 0 + C_2 \longrightarrow C_3$$

$$0 = S + M + c_3 \longrightarrow c_4$$

$$M = C_4$$


$$M=1, O=0, S=9, E=5, N=6, R=8, D=7, Y=2$$

The screenshot shows the AWS Cloud9 IDE interface. The browser tab is titled 'console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#'. The IDE has several tabs open: 'AWS Toolkit - Quick Start', 'GraphColouring.py', 'CamelBanana.py', and 'constraintsatisfactionprob.py'. The 'constraintsatisfactionprob.py' tab is active, showing a Python script. The script defines a function `get_value(word, substitution)` that calculates the value of a word based on a substitution mapping. It also defines a function `solve2(equation)` that solves the cryptarithm by iterating over possible mappings. The script is written in Python 3. The terminal at the bottom shows the command `bash - "ip-172-31-15-155" x` and the file `constraintsatisfactionprob.py` is selected. The status bar at the bottom indicates the runner is Python 3, CWD, and ENV.

```
1 import itertools
2 def get_value(word, substitution):
3     s = 0
4     factor = 1
5     for letter in reversed(word):
6         s += factor * substitution[letter]
7         factor *= 10
8     return s
9
10 def solve2(equation):
11     left, right = equation.lower().replace(' ', '').split('=')
12     left = left.split('+')
13     letters = set(right)
14     for word in left:
15         for letter in word:
16             letters.add(letter)
17     letters = list(letters)
18     digits = range(10)
19     for perm in itertools.permutations(digits, len(letters)):
20         sol = dict(zip(letters, perm))
21         if sum(get_value(word, sol) for word in left) == get_value(right, sol):
22             print(' + '.join(str(get_value(word, sol)) for word in left) + " = {} (mapping: {})".format(get_value(right, sol), sol))
23
24 solve2('POINT + ZERO = ENERGY')
```

The screenshot shows the AWS Cloud9 IDE interface with the same tabs as the previous image. The 'constraintsatisfactionprob.py' tab is active, and the terminal at the bottom displays the output of the script. The output shows the results of the `solve2` function, listing the sum of the left-hand side of the equation and the right-hand side, along with the mapping of letters to digits that satisfies the equation. The status bar at the bottom indicates the runner is Python 3, CWD, and ENV.

```
74285 + 6034 = 80319 (mapping: {'r': 3, 'o': 4, 't': 5, 'g': 1, 'y': 9, 'z': 6, 'i': 2, 'n': 8, 'e': 0, 'p': 7})
71385 + 9041 = 80426 (mapping: {'r': 4, 'o': 1, 't': 5, 'g': 2, 'y': 6, 'z': 9, 'i': 3, 'n': 8, 'e': 0, 'p': 7})
58369 + 2048 = 60417 (mapping: {'r': 4, 'o': 8, 't': 9, 'g': 1, 'y': 7, 'z': 2, 'i': 3, 'n': 6, 'e': 0, 'p': 5})
62479 + 8052 = 70531 (mapping: {'r': 5, 'o': 2, 't': 9, 'g': 3, 'y': 1, 'z': 8, 'i': 4, 'n': 7, 'e': 0, 'p': 6})
71582 + 9061 = 80643 (mapping: {'r': 6, 'o': 1, 't': 2, 'g': 4, 'y': 3, 'z': 9, 'i': 5, 'n': 8, 'e': 0, 'p': 7})
98504 + 3168 = 101672 (mapping: {'r': 6, 'o': 8, 't': 4, 'g': 7, 'y': 2, 'z': 3, 'i': 5, 'n': 0, 'e': 1, 'p': 9})
38549 + 2068 = 40617 (mapping: {'r': 6, 'o': 8, 't': 9, 'g': 1, 'y': 7, 'z': 2, 'i': 5, 'n': 4, 'e': 0, 'p': 3})
42659 + 8072 = 50731 (mapping: {'r': 7, 'o': 2, 't': 9, 'g': 3, 'y': 1, 'z': 8, 'i': 6, 'n': 5, 'e': 0, 'p': 4})
51762 + 9081 = 60843 (mapping: {'r': 8, 'o': 1, 't': 2, 'g': 4, 'y': 3, 'z': 9, 'i': 7, 'n': 6, 'e': 0, 'p': 5})
31745 + 9081 = 40826 (mapping: {'r': 8, 'o': 1, 't': 5, 'g': 2, 'y': 6, 'z': 9, 'i': 7, 'n': 4, 'e': 0, 'p': 3})
24735 + 6084 = 30819 (mapping: {'r': 8, 'o': 4, 't': 5, 'g': 1, 'y': 9, 'z': 6, 'i': 7, 'n': 3, 'e': 0, 'p': 2})
```

Result:-

The constraint satisfaction problem(CRYPTARITHMETIC Problem)was successfully implemented using python