

Aman Kumar Pandey
RA1911003010685
Artificial Intelligence Lab
Lab-3

Aim - Implementation of constraint satisfaction
problem - Cryptarithmic problem (SEND + MORE
MONEY)

Problem Formulation

Given an expression where two words add to give a third word, assign some unique digit (0-9) to each letter where same letters cannot be assigned to different digit.

Display the possible mappings to each of the letters S, E, N, D, M, O, R and Y

Initial state

- $D = ?$, $E = ?$, $Y = ?$, $N = ?$, $R = ?$, $O = ?$, $S = ?$, $M = ?$
- $C_1 = ?$, $C_2 = ?$, $C_3 = ?$
- C_1, C_2, C_3 stands for carry variable respectively.

Goal state

The digit to the letters must be assigned in such a manner so that sum is satisfied.

Various possible goal state is obtained.

One of the goal state is:

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

$$\bullet C_1=1, C_2=1, C_3=0$$

Problem solving

- Starting from left hand side, the terms are S and M.

Assign a digit which could give a satisfactory result. Let's assign $S=9, M=1$

$$\begin{array}{r} S \\ + M \\ \hline MO \end{array} \rightarrow \begin{array}{r} 9 \\ + 1 \\ \hline 10 \end{array}$$

we get O as 0

- Moving ahead we get the next terms E and O to get N as its result.

Considering $E=5$

$$\begin{array}{r} E \\ + O \\ \hline N \end{array} \quad \times \quad \begin{array}{r} 5 \\ + 0 \\ \hline 5 \end{array}$$

This is not possible as E and N cannot be assigned to same digit.

we assume that $c_2(\text{carry}) = 1$

$$\begin{array}{r} c_2(\text{carry}) \quad (1) \\ E \quad \quad \quad 5 \\ + 0 \quad \quad \quad 0 \\ \hline N \quad \quad \quad 6 \end{array}$$

we get N as 6

• Further adding next two terms N and R

$$\begin{array}{r} N \quad \quad \quad 6 \\ + R \quad \quad \quad 8 \\ \hline E \quad \quad \quad 14 \end{array} \quad \begin{array}{l} \text{As } E \text{ is already} \\ \text{assigned value } 5 \end{array}$$

we consider $c_3 = 1$

$$\begin{array}{r} N \quad \quad \quad 6 \\ + R \quad \quad \quad 8 \\ \hline E \quad \quad \quad 15 \end{array} \quad \text{we get } R = 8$$

• On adding last two terms 1 carry must be produced

$$\begin{array}{r} D \quad \quad \quad 7 \\ + E \quad \quad \quad 5 \\ \hline Y \quad \quad \quad 12 \end{array}$$

• Keeping all constraints in mind the final state is

$C_3(0)$	$C_2(1)$	$C_1(1)$	
$S(9)$	$E(5)$	$N(6)$	$D(7)$
$M(1)$	$O(0)$	$R(8)$	$E(5)$
$M(1)$	$O(0)$	$N(6)$	$E(5)$
			$Y(2)$

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

AMAN KUMAR PANDEY

RA1911003010685

ARTIFICIAL INTELLIGENCE LAB

LAB: 3

Implementation of Constraint Satisfaction Problems

**(cryptarithmic problem-
send+more=money)**

Algorithm:

Step 1: Start
Step 2: Accept a expression 'SEND+MORE=MONEY'
Step 3: Extract the words SEND, MORE and MONEY.
Step 4: Permute for different combination of values for S,E,N,D,M,O,R,Y.
Step5: Check if the sum of the left value i.e, SEND+MORE is equal to the right sum i.e, MONEY or not. If the sum value matches print the mapping.
Step 6: Continue for other permutations as well.
Step 7: Stop.

Source code:

```
import itertools

def get_value(word, substitution):

    s = 0

    factor = 1

    for letter in reversed(word):

        s += factor * substitution[letter]
```

```

        factor *= 10

    return s

def solve2(equation):
    left, right = equation.lower().replace(' ', '').split('=')
    left = left.split('+')
    letters = set(right)

    for word in left:
        for letter in word:
            letters.add(letter)

    letters = list(letters)
    digits = range(10)



    for perm in itertools.permutations(digits, len(letters)):
        sol = dict(zip(letters, perm))

        if sum(get_value(word, sol) for word in left) == get_value(right, sol):
            print(' + '.join(str(get_value(word, sol)) for word in left) + " = { }"
                  (mapping: { })".format(get_value(right, sol), sol))

if __name__ == '__main__':
    solve2('SEND + MORE = MONEY')

```

Output:

```
RA1911003010685/CSP_x 
Run  Command: RA1911003010685/CSP_Lab3.py

7531 + 825 = 8356 (mapping: {'d': 1, 'r': 2, 's': 7, 'o': 8, 'e': 5, 'n': 3, 'y': 6, 'm': 0})
6851 + 738 = 7589 (mapping: {'d': 1, 'r': 3, 's': 6, 'o': 7, 'e': 8, 'n': 5, 'y': 9, 'm': 0})
5731 + 647 = 6378 (mapping: {'d': 1, 'r': 4, 's': 5, 'o': 6, 'e': 7, 'n': 3, 'y': 8, 'm': 0})
3821 + 468 = 4289 (mapping: {'d': 1, 'r': 6, 's': 3, 'o': 4, 'e': 8, 'n': 2, 'y': 9, 'm': 0})
8432 + 914 = 9346 (mapping: {'d': 2, 'r': 1, 's': 8, 'o': 9, 'e': 4, 'n': 3, 'y': 6, 'm': 0})
8542 + 915 = 9457 (mapping: {'d': 2, 'r': 1, 's': 8, 'o': 9, 'e': 5, 'n': 4, 'y': 7, 'm': 0})
5732 + 647 = 6379 (mapping: {'d': 2, 'r': 4, 's': 5, 'o': 6, 'e': 7, 'n': 3, 'y': 9, 'm': 0})
3712 + 467 = 4179 (mapping: {'d': 2, 'r': 6, 's': 3, 'o': 4, 'e': 7, 'n': 1, 'y': 9, 'm': 0})
6853 + 728 = 7581 (mapping: {'d': 3, 'r': 2, 's': 6, 'o': 7, 'e': 8, 'n': 5, 'y': 1, 'm': 0})
7643 + 826 = 8469 (mapping: {'d': 3, 'r': 2, 's': 7, 'o': 8, 'e': 6, 'n': 4, 'y': 9, 'm': 0})
8324 + 913 = 9237 (mapping: {'d': 4, 'r': 1, 's': 8, 'o': 9, 'e': 3, 'n': 2, 'y': 7, 'm': 0})
7534 + 825 = 8359 (mapping: {'d': 4, 'r': 2, 's': 7, 'o': 8, 'e': 5, 'n': 3, 'y': 9, 'm': 0})
6524 + 735 = 7259 (mapping: {'d': 4, 'r': 3, 's': 6, 'o': 7, 'e': 5, 'n': 2, 'y': 9, 'm': 0})
6415 + 734 = 7149 (mapping: {'d': 5, 'r': 3, 's': 6, 'o': 7, 'e': 4, 'n': 1, 'y': 9, 'm': 0})
7316 + 823 = 8139 (mapping: {'d': 6, 'r': 2, 's': 7, 'o': 8, 'e': 3, 'n': 1, 'y': 9, 'm': 0})
2817 + 368 = 3185 (mapping: {'d': 7, 'r': 6, 's': 2, 'o': 3, 'e': 8, 'n': 1, 'y': 5, 'm': 0})
9567 + 1085 = 10652 (mapping: {'d': 7, 'r': 8, 's': 9, 'o': 0, 'e': 5, 'n': 6, 'y': 2, 'm': 1})
7429 + 814 = 8243 (mapping: {'d': 9, 'r': 1, 's': 7, 'o': 8, 'e': 4, 'n': 2, 'y': 3, 'm': 0})
7539 + 815 = 8354 (mapping: {'d': 9, 'r': 1, 's': 7, 'o': 8, 'e': 5, 'n': 3, 'y': 4, 'm': 0})
7649 + 816 = 8465 (mapping: {'d': 9, 'r': 1, 's': 7, 'o': 8, 'e': 6, 'n': 4, 'y': 5, 'm': 0})
6419 + 724 = 7143 (mapping: {'d': 9, 'r': 2, 's': 6, 'o': 7, 'e': 4, 'n': 1, 'y': 3, 'm': 0})
5849 + 638 = 6487 (mapping: {'d': 9, 'r': 3, 's': 5, 'o': 6, 'e': 8, 'n': 4, 'y': 7, 'm': 0})
3719 + 457 = 4176 (mapping: {'d': 9, 'r': 5, 's': 3, 'o': 4, 'e': 7, 'n': 1, 'y': 6, 'm': 0})
3829 + 458 = 4287 (mapping: {'d': 9, 'r': 5, 's': 3, 'o': 4, 'e': 8, 'n': 2, 'y': 7, 'm': 0})
2819 + 368 = 3187 (mapping: {'d': 9, 'r': 6, 's': 2, 'o': 3, 'e': 8, 'n': 1, 'y': 7, 'm': 0})

Process exited with code: 0
```

Result:

Hence, the implementation of the Cryptarithmic Problem is done successfully.