

Amit Srivastav

RA1911003010633

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:

- $D = ?$, $E = ?$, $Y = ?$, $N = ?$, $R = ?$, $O = ?$, $S = ?$, $M = ?$
- $C1 = ?$, $C2 = ?$, $C3 = ?$
- $C1, C2, C3$ stands for carry variables 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:

- $D = 7$, $E = 5$, $Y = 2$, $N = 6$, $R = 8$, $O = 0$, $S = 9$, $M = 1$
- $C1 = 1$, $C2 = 1$, $C3 = 0$

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 $S=9$, $M=1$

$$\begin{array}{r} S \\ + M \\ \hline M O \end{array} \longrightarrow \begin{array}{r} 9 \\ + 1 \\ \hline 10 \end{array}$$

We get O as 0.

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

Considering $E=5$

$$\begin{array}{r} E \\ + O \\ \hline N \end{array} \not\rightarrow \begin{array}{r} 5 \\ + 0 \\ \hline 5 \end{array} \text{ This is not possible as E and N} \\ \text{cannot be assigned to same digit.}$$

We Assume that $C_2(\text{carry})=1$

$$\begin{array}{r} C_2(\text{carry}) \\ E \\ + O \\ \hline N \end{array} \rightarrow \begin{array}{r} \textcircled{1} \\ 5 \\ 0 \\ \hline 6 \end{array}$$

We get N as 6.

- Further Adding next two terms N and R

$$\begin{array}{r} N \\ + R \\ \hline E \end{array} \not\rightarrow \begin{array}{r} 6 \\ + 8 \\ \hline 14 \end{array} \text{ As E is already assigned value 5.}$$

$$\begin{array}{r} \text{We consider } C_3=1 \\ C_3 \\ N \\ + R \\ \hline E \end{array} \rightarrow \begin{array}{r} \textcircled{1} \\ 6 \\ + 8 \\ \hline 15 \end{array}$$

We get $R=8$

- On Adding last two terms 1 carry must be produced

$$\begin{array}{r} D \\ + E \\ \hline Y \end{array} \rightarrow \begin{array}{r} 07 \\ 5 \\ \hline 12 \end{array}$$

- Keeping all constraints in mind the final state is!

$$\begin{array}{rcccc} & c_3(0) & c_2(0) & c_1(1) & \\ & s(9) & E(5) & N(6) & D(7) \\ + & M(1) & O(0) & R(8) & E(5) \\ \hline M(1) & O(0) & N(6) & E(5) & Y(2) \end{array}$$

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

AMIT SRIVASTAV
RA1911003010633
ARTIFICIAL INTELLIGENCE LAB
EXPERIMENT NO: 3

IMPLEMENTATION OF CONSTRAINT
SATISFACTION PROBLEMS
(CRYPTARITHMETIC 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: And check if the sum of left value i.e, SEND+MORE is equal to 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:

```
CSP_Lab3.py x +
Run Command: RA1911003010633/CSP_Lab3.py

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

Process exited with code: 0
```

Result:

Hence, the implementation of Cryptarithmic Problem is done successfully.