Amit Srivastav
RA1911003010633

Artificial Intelligence Lab
Lab-3

Aim: Implementation of Constraint Satisfaction Problem - Cryptarithmetic 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

$$\begin{array}{r} S \\ +\ M \\ \hline M\ O \end{array} \longrightarrow \begin{array}{r} 9 \\ +\ 1 \\ \hline 1\ 0 \end{array}$$

We get O as 0.

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

  Coneidering E = 5

$$\begin{array}{r} E \\ +\ 0 \\ \hline N \end{array} \not\longrightarrow \begin{array}{r} 5 \\ +\ 0 \\ \hline 5 \end{array}$$

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

We Assume that C2 (carry) = 1

$$\begin{array}{r} C2\,(carry) \\ E \\ +\ 0 \\ \hline N \end{array} \longrightarrow \begin{array}{r} (1) \\ 5 \\ 0 \\ \hline 6 \end{array}$$

We get N as 6.

- Further Adding next two terms N and R

$$\begin{array}{r} N \\ +\ R \\ \hline E \end{array} \not\longrightarrow \begin{array}{r} 6 \\ +\ 8 \\ \hline 14 \end{array}$$

As E is already assigned value 5.

We consider C3 = 1

$$\begin{array}{r} {\scriptstyle C3} \\ N \\ +\ R \\ \hline E \end{array} \longrightarrow \begin{array}{r} (1) \\ 6 \\ +\ 8 \\ \hline 15 \end{array}$$

We get R = 8

- On Adding last two terms 1 carry must be produced

$$\begin{array}{r} D \\ +E \\ \hline Y \end{array} \longrightarrow \begin{array}{r} 17 \\ 5 \\ \hline 12 \end{array}$$

- Keeping all constraints in mind the final state is :

	C3(0)	C2(1)	C1(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

AMIT SRIVASTAV

RA1911003010633

ARTIFICIAL INTELLIGENCE LAB

EXPERIMENT NO: 3

IMPLEMENTATION OF CONSTRAINT SATISFACTION PROBLEMS (CRYPTARITHMETIC PROBLEM- SEND+MORE=MONEY)

Algorithm:

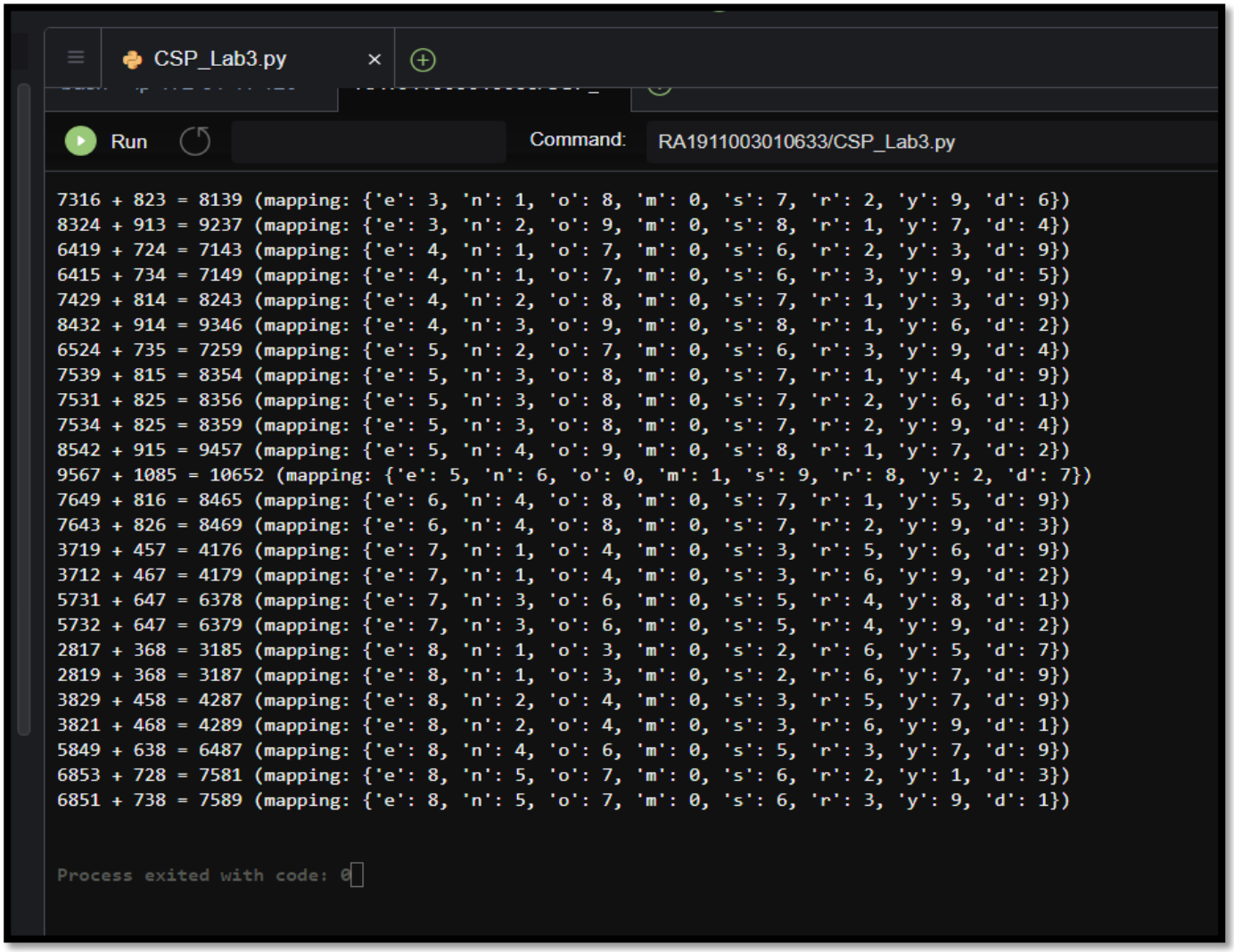
Step 1: Start
Step 2: Accept a expression 'SEND+MORE=MONEY'
Step 3: Extract the words SEND,MORE and MONEY.
Step 4: Permute for different combination of values for S,E,N,D,M,O,R,Y.
Step5: And check if the sum of left value i.e, SEND+MORE is equal to right sum i.e, MONEY or not. If the sum value matches 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:



Result:

Hence, the implementation of Cryptarithmetic Problem is done successfully.