RA1911026010029
Thejaswin S

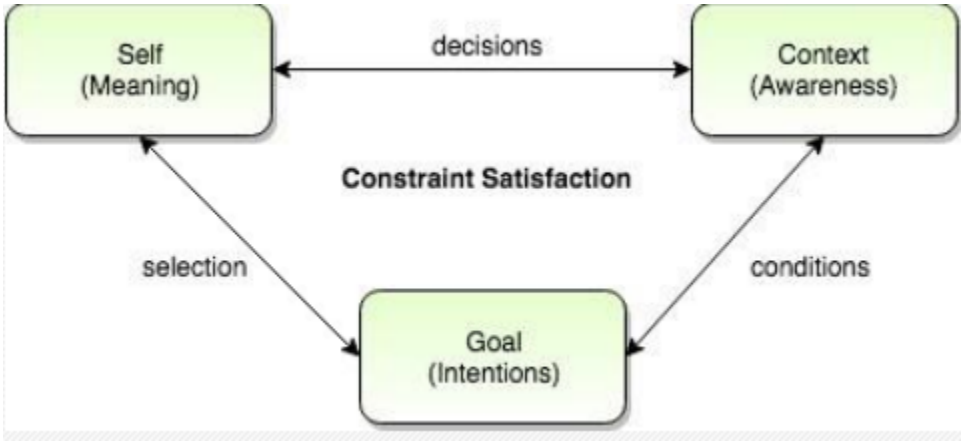
EXP-3 Constraint satisfaction problem

AIM

To implement **CryptArithmetic Problem** in python which is a Constraint satisfaction problem and verify its output.

Constraint satisfaction problem

- states and goal test conform to a standard, structured and simple representation
- general-purpose heuristic functions
- Solve: design the variable, domain, and constraints set.
- Then, look for an optimal solution. The optimal solution should satisfy all constraints



CSP is defined by 3 components (X, D, C):

- state: a set of variables X, each Xi, with values from domain Di
- goal test: a set of constraints C, each Ci involves some subset of the variables and specifies the

allowable combinations of values for that subset

- Each constraint Ci consists of a pair <scope, rel>, where the scope is a tuple of variables and rel is the

relation, either represented explicitly or abstractly

- X1 and X2 both have the domain {A, B}
- Constraints:
- <(X1, X2), [(A, B), (B, A)]>, or
- <(X1, X2), X1 ≠ X2>

RA1911026010029
Thejaswin S

Algorithm

Problem:

```
  SEND
+ MORE
.....
 MONEY
```

Initial State:

No two letters have the same value.
The sums of the digits must be as shown in the problem.

A Cryptarithmetic Problem

1. **SEND + MORE = MONEY**

```
 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, since it is only carry-over possible from the sum of 2 single-digit number in column 4.
2. To produce a carry from column 4 to column 5 'S + M' is at least 9 so 'S=8or9' so 'S+M=9or10' & so 'O = 0 or 1'. But 'M=1', so 'O = 0'.
3. If there is c+rry from Column 3 to 4 then 'E=9' & so 'N=0'. But 'O = 0' so there is no carry & 'S=9' & 'c3=0'.
4. If there is no carry from columns 2 to 3 then 'E=N' is impossible, therefore there is carry & 'N=E+1' & 'c2=1'.
5. If there is carry from column 1 to 2 then 'N+R=E mod 10' & 'N=E+1' so 'E+1+R=E mod 10', so 'R=9' but 'S=9', so there must be c+rry from column 1 to 2. Therefore 'c1=1' & 'R=8'.

14

6. To produce carry 'c1=1' from column 1 to 2, we must h+ve 'D+E=10+Y' as Y cannot be 0/1 so D+E is at least 12. As D is at most 7 & E is At least 5 (D cannot be 8 or 9 as it is already assigned). N is at most 7 & 'N=E+1' so 'E=5or6'.
7. If E were 6 & D+E at least 12 then D would be 7, but 'N=E+1' & N would also be 7 which is impossible. Therefore 'E=5' & 'N=6'.
8. D+E is at least 12 for that we get 'D=7' & 'Y=2'.

SOLUTION:	VALUES:
9 5 6 7 + 1 0 8 5 ---------- 1 0 6 5 2	S=9 E=5 N=6 D=7 M=1 O=0 R=8 Y=2

2. BASE + BALL = GAMES

```
  B A S E
+ B A L L
----------
G A M E S
```

1. Assuming numbers can't start with 0, G is 1 because two four-digit numbers can't sum to 20000 or more.
SE+LL=ES or 1ES.
2. If it is ES, then LL must be a multiple of 9 because SE and ES are always congruent mod 9. But LL is a multiple of 11, so it would have to be 99, which is impossible.
3. So SE+LL=1ES. LL must be congruent to 100 mod 9. The only multiple of 11 that work is 55, so L is 5.
4. SE+55=1ES. This is possible when E+5=S. The possibilities for ES are 27, 38, or 49.
5. BA+BA+1=1AM. B must be at least 5 because B+B (possibly +1 from a carry) is at least 10.
6. If A is less than 5, then A+A+1 does not carry, a and A must be even. Inversely, if A is greater than 5, it must be odd. The possibilities for A are 0, 2, 4, 7, or 9.
0 does not work because M would have to be 1.
2 and 7 don't work because M would have to be 5.
9 doesn't work because M would also have to be 9.
So A is 4, M is 9, and B is 7. This leaves 38 as the only possibility for ES. The

SOLUTION:	VALUES:
7 4 8 3 + 7 4 5 5 ---------- 1 4 9 3 8	B=7 A=4 S=8 E=3 G=1 M=9 L=5

RA1911026010029
Thejaswin S

3. TWO + TWO = FOUR

```
  T W O
+ T W O
---------------
 F O U R
```

1. F = 1 for carry over T >= 5.
'O' can't be 0 as R will be 0. So T can't be 5 so let T>=6
2. If T = 6, O = 2 and R = 4 and W + W = U for W can't be 1,2,6,4. W < 4 +s to avoid carry-over. W can't be 3 as U will be 6.
3. So T = 7, so, O can be 4 or 5 depending on whether or W + W > 10. If O is 4 then R = 8. W can't be 1, 2. So W = 3
If W = 3 then U = 6 hence

SOLUTION:	VALUES:
7 3 4 + 7 3 4 ---------- 1 4 6 8	T=7 W=3 O=4 F=1 U=6 R=8

4. N O + G U N + N O = H U N T

```
  N O
  G U N
+ N O
---------------
 H U N T
```

Here H = 1, from the NUNN column we must have "carry 1," so G = 9, U = zero. Since we have "carry" zero or 1 or 2 from the ONOT column, correspondingly we h+ve N + U = 10 or 9 or 8. But duplication is not allowed, so N = 8 with "carry 2" from ONOT. Hence, O + O = T + 20 – 8 = T + 12. Testing for T = 2, 4 or 6, we find only T = 2 +accept+able, O = 7. So we h+ve 87 + 908 + 87 = 1082.
HUNT = 1082

SOLUTION:	VALUES:
8 7 + 9 0 8 8 7 ---------- 1 0 8 2	N=8 T=2 U=0 O=7 G=9 H=1

RA1911026010029
Thejaswin S

Program:

```
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))

solve2('NO + GUN + NO = HUNT ')
```

RA1911026010029
Thejaswin S

bash - "ip-172-31-2-88" | exp3.py | bash - "ip-172-31-2-88"

```python
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))

solve2('NO + GUN + NO = HUNT ')
```

Output

SEND + MORE = MONEY

```
RA1911026010029:~/environment/RA1911026010029/exp3 $ python3 exp3.py
6419 + 724 = 7143 (mapping: {'n': 1, 'y': 3, 'd': 9, 's': 6, 'e': 4, 'r': 2, 'o': 7, 'm': 0})
2817 + 368 = 3185 (mapping: {'n': 1, 'y': 5, 'd': 7, 's': 2, 'e': 8, 'r': 6, 'o': 3, 'm': 0})
3719 + 457 = 4176 (mapping: {'n': 1, 'y': 6, 'd': 9, 's': 3, 'e': 7, 'r': 5, 'o': 4, 'm': 0})
2819 + 368 = 3187 (mapping: {'n': 1, 'y': 7, 'd': 9, 's': 2, 'e': 8, 'r': 6, 'o': 3, 'm': 0})
3712 + 467 = 4179 (mapping: {'n': 1, 'y': 9, 'd': 2, 's': 3, 'e': 7, 'r': 6, 'o': 4, 'm': 0})
6415 + 734 = 7149 (mapping: {'n': 1, 'y': 9, 'd': 5, 's': 6, 'e': 4, 'r': 3, 'o': 7, 'm': 0})
7316 + 823 = 8139 (mapping: {'n': 1, 'y': 9, 'd': 6, 's': 7, 'e': 3, 'r': 2, 'o': 8, 'm': 0})
7429 + 814 = 8243 (mapping: {'n': 2, 'y': 3, 'd': 9, 's': 7, 'e': 4, 'r': 1, 'o': 8, 'm': 0})
8324 + 913 = 9237 (mapping: {'n': 2, 'y': 7, 'd': 4, 's': 8, 'e': 3, 'r': 1, 'o': 9, 'm': 0})
3829 + 458 = 4287 (mapping: {'n': 2, 'y': 7, 'd': 9, 's': 3, 'e': 8, 'r': 5, 'o': 4, 'm': 0})
3821 + 468 = 4289 (mapping: {'n': 2, 'y': 9, 'd': 1, 's': 3, 'e': 8, 'r': 6, 'o': 4, 'm': 0})
6524 + 735 = 7259 (mapping: {'n': 2, 'y': 9, 'd': 4, 's': 6, 'e': 5, 'r': 3, 'o': 7, 'm': 0})
7539 + 815 = 8354 (mapping: {'n': 3, 'y': 4, 'd': 9, 's': 7, 'e': 5, 'r': 1, 'o': 8, 'm': 0})
7531 + 825 = 8356 (mapping: {'n': 3, 'y': 6, 'd': 1, 's': 7, 'e': 5, 'r': 2, 'o': 8, 'm': 0})
8432 + 914 = 9346 (mapping: {'n': 3, 'y': 6, 'd': 2, 's': 8, 'e': 4, 'r': 1, 'o': 9, 'm': 0})
5731 + 647 = 6378 (mapping: {'n': 3, 'y': 8, 'd': 1, 's': 5, 'e': 7, 'r': 4, 'o': 6, 'm': 0})
5732 + 647 = 6379 (mapping: {'n': 3, 'y': 9, 'd': 2, 's': 5, 'e': 7, 'r': 4, 'o': 6, 'm': 0})
7534 + 825 = 8359 (mapping: {'n': 3, 'y': 9, 'd': 4, 's': 7, 'e': 5, 'r': 2, 'o': 8, 'm': 0})
7649 + 816 = 8465 (mapping: {'n': 4, 'y': 5, 'd': 9, 's': 7, 'e': 6, 'r': 1, 'o': 8, 'm': 0})
8542 + 915 = 9457 (mapping: {'n': 4, 'y': 7, 'd': 2, 's': 8, 'e': 5, 'r': 1, 'o': 9, 'm': 0})
5849 + 638 = 6487 (mapping: {'n': 4, 'y': 7, 'd': 9, 's': 5, 'e': 8, 'r': 3, 'o': 6, 'm': 0})
7643 + 826 = 8469 (mapping: {'n': 4, 'y': 9, 'd': 3, 's': 7, 'e': 6, 'r': 2, 'o': 8, 'm': 0})
6853 + 728 = 7581 (mapping: {'n': 5, 'y': 1, 'd': 3, 's': 6, 'e': 8, 'r': 2, 'o': 7, 'm': 0})
6851 + 738 = 7589 (mapping: {'n': 5, 'y': 9, 'd': 1, 's': 6, 'e': 8, 'r': 3, 'o': 7, 'm': 0})
9567 + 1085 = 10652 (mapping: {'n': 6, 'y': 2, 'd': 7, 's': 9, 'e': 5, 'r': 8, 'o': 0, 'm': 1})
```

BASE + BALL = GAMES

```
RA1911026010029:~/environment/RA1911026010029/exp3 $ python3 exp3.py
2461 + 2455 = 4916 (mapping: {'e': 1, 'a': 4, 'l': 5, 'g': 0, 's': 6, 'b': 2, 'm': 9})
2483 + 2455 = 4938 (mapping: {'e': 3, 'a': 4, 'l': 5, 'g': 0, 's': 8, 'b': 2, 'm': 9})
7483 + 7455 = 14938 (mapping: {'e': 3, 'a': 4, 'l': 5, 'g': 1, 's': 8, 'b': 7, 'm': 9})
```

TWO + TWO = FOUR

```
RA1911026010029:~/environment/RA1911026010029/exp3 $ python3 exp3.py
132 + 132 = 264 (mapping: {'t': 1, 'w': 3, 'o': 2, 'u': 6, 'f': 0, 'r': 4})
173 + 173 = 346 (mapping: {'t': 1, 'w': 7, 'o': 3, 'u': 4, 'f': 0, 'r': 6})
193 + 193 = 386 (mapping: {'t': 1, 'w': 9, 'o': 3, 'u': 8, 'f': 0, 'r': 6})
234 + 234 = 468 (mapping: {'t': 2, 'w': 3, 'o': 4, 'u': 6, 'f': 0, 'r': 8})
346 + 346 = 692 (mapping: {'t': 3, 'w': 4, 'o': 6, 'u': 9, 'f': 0, 'r': 2})
357 + 357 = 714 (mapping: {'t': 3, 'w': 5, 'o': 7, 'u': 1, 'f': 0, 'r': 4})
418 + 418 = 836 (mapping: {'t': 4, 'w': 1, 'o': 8, 'u': 3, 'f': 0, 'r': 6})
428 + 428 = 856 (mapping: {'t': 4, 'w': 2, 'o': 8, 'u': 5, 'f': 0, 'r': 6})
438 + 438 = 876 (mapping: {'t': 4, 'w': 3, 'o': 8, 'u': 7, 'f': 0, 'r': 6})
459 + 459 = 918 (mapping: {'t': 4, 'w': 5, 'o': 9, 'u': 1, 'f': 0, 'r': 8})
469 + 469 = 938 (mapping: {'t': 4, 'w': 6, 'o': 9, 'u': 3, 'f': 0, 'r': 8})
479 + 479 = 958 (mapping: {'t': 4, 'w': 7, 'o': 9, 'u': 5, 'f': 0, 'r': 8})
734 + 734 = 1468 (mapping: {'t': 7, 'w': 3, 'o': 4, 'u': 6, 'f': 1, 'r': 8})
765 + 765 = 1530 (mapping: {'t': 7, 'w': 6, 'o': 5, 'u': 3, 'f': 1, 'r': 0})
836 + 836 = 1672 (mapping: {'t': 8, 'w': 3, 'o': 6, 'u': 7, 'f': 1, 'r': 2})
846 + 846 = 1692 (mapping: {'t': 8, 'w': 4, 'o': 6, 'u': 9, 'f': 1, 'r': 2})
867 + 867 = 1734 (mapping: {'t': 8, 'w': 6, 'o': 7, 'u': 3, 'f': 1, 'r': 4})
928 + 928 = 1856 (mapping: {'t': 9, 'w': 2, 'o': 8, 'u': 5, 'f': 1, 'r': 6})
938 + 938 = 1876 (mapping: {'t': 9, 'w': 3, 'o': 8, 'u': 7, 'f': 1, 'r': 6})
```

NO + GUN +NO = HUNT

```
RA1911026010029:~/environment/RA1911026010029/exp3 $ python3 exp3.py
16 + 781 + 16 = 813 (mapping: {'n': 1, 't': 3, 'u': 8, 'o': 6, 'g': 7, 'h': 0})
12 + 891 + 12 = 915 (mapping: {'n': 1, 't': 5, 'u': 9, 'o': 2, 'g': 8, 'h': 0})
13 + 891 + 13 = 917 (mapping: {'n': 1, 't': 7, 'u': 9, 'o': 3, 'g': 8, 'h': 0})
21 + 782 + 21 = 824 (mapping: {'n': 2, 't': 4, 'u': 8, 'o': 1, 'g': 7, 'h': 0})
39 + 453 + 39 = 531 (mapping: {'n': 3, 't': 1, 'u': 5, 'o': 9, 'g': 4, 'h': 0})
34 + 563 + 34 = 631 (mapping: {'n': 3, 't': 1, 'u': 6, 'o': 4, 'g': 5, 'h': 0})
31 + 673 + 31 = 735 (mapping: {'n': 3, 't': 5, 'u': 7, 'o': 1, 'g': 6, 'h': 0})
38 + 563 + 38 = 639 (mapping: {'n': 3, 't': 9, 'u': 6, 'o': 8, 'g': 5, 'h': 0})
42 + 564 + 42 = 648 (mapping: {'n': 4, 't': 8, 'u': 6, 'o': 2, 'g': 5, 'h': 0})
58 + 235 + 58 = 351 (mapping: {'n': 5, 't': 1, 'u': 3, 'o': 8, 'g': 2, 'h': 0})
56 + 345 + 56 = 457 (mapping: {'n': 5, 't': 7, 'u': 4, 'o': 6, 'g': 3, 'h': 0})
57 + 345 + 57 = 459 (mapping: {'n': 5, 't': 9, 'u': 4, 'o': 7, 'g': 3, 'h': 0})
69 + 126 + 69 = 264 (mapping: {'n': 6, 't': 4, 'u': 2, 'o': 9, 'g': 1, 'h': 0})
61 + 346 + 61 = 468 (mapping: {'n': 6, 't': 8, 'u': 4, 'o': 1, 'g': 3, 'h': 0})
74 + 127 + 74 = 275 (mapping: {'n': 7, 't': 5, 'u': 2, 'o': 4, 'g': 1, 'h': 0})
76 + 127 + 76 = 279 (mapping: {'n': 7, 't': 9, 'u': 2, 'o': 6, 'g': 1, 'h': 0})
71 + 237 + 71 = 379 (mapping: {'n': 7, 't': 9, 'u': 3, 'o': 1, 'g': 2, 'h': 0})
87 + 908 + 87 = 1082 (mapping: {'n': 8, 't': 2, 'u': 0, 'o': 7, 'g': 9, 'h': 1})
```

RA1911026010029
Thejaswin S

Result

The constraint satisfying problem SEND + MORE = MONEY, BASE + BALL = GAMES, TWO + TWO = FOUR, NO + GUN +NO = HUNT solved using the carryover technique and values for the alphabets obtained successfully.