1.9 CONSTRAINT SATISFACTION PROBLEM

The goal is to discover some problem state that satisfies a given set of constraints.

Basic workflow of constraint satisfaction problem

- We need to analyze the problem perfectly
- We need to derive the constraints given in the problem
- Then we need to derive a solution form a given constraint.
- Find whether we have reached the foal state, If we have not reached the goal state, we need to make a guess and that guess has to be added as the new constraint.
- After adding new constraint, again we go for the evaluation of the solution. Now we need to solve the problem by using this added new constraint, again we have to check, whether we have reached the goal state, if yes, solution found

Algorithm : Constraint satisfaction

1. Propagate available constraints. To do this , first set OPEN to the set of all objects that must have values assigned to them in a complete solution. Then do until an inconsistency is detected or until OPEN is empty:
    a. Select an object OB from OPEN. Strengthen as much as possible the set of constraints that apply to OB
    b. If this is different from the set that was assigned the last time OB was examined or if it this is the first time OB has been examined, then add to OPEN all objects that share any constraints with OB.
    c. Remove OB from OPEN.
2. If the union of the constraints discovered above defines a solution, then quit and report the solution.
3. If the solution of the constraints discovered above defines a contradiction, then return failure.
4. If neither of the above occurs, then it is necessary to make a guess at something in order to proceed. To do this, loop until a solution is found or all possible solutions have been eliminated

a. Select an object whose value is not yet determined and select a way of strengthening the constraints on that object.
b. Recursively invoke constraint satisfaction with the current set of constraints augmented by the strengthening constraint just selected.

Example :

Cryptarithmatic puzzles,

SEND	DONALD	CROSS
+MORE	+GERALD	+ROADS
……	……..	…….
MONEY	ROBERT	DANGER

Algorithm Working :

Consider the crypt arithmetic problem shown in the below fig.

Problem:

SEND

+MORE

……

MONEY

Assign decimal digit to each of the letters in such a way that the answer to the problem is correct to the same letter occurs more than once, it must be assign the same digit each time. No two different letters may be assigned the same digit. Consider the crypt arithmetic problem.

Constraints:-

1. No two digit can be assigned to same letter.
2. Only single digit number can be assign to a letter.
3. No two letters can be assigned same digit.
4. Assumption can be made at various levels such that they do not contradict each other.
5. The problem can be decomposed into secured constraints. A constraint satisfaction approach may be used.
6. Any of search techniques may be used.
7. Backtracking may be performed as applicable us applied search techniques.
8. Rule of arithmetic may be followed.

Initial state of problem.

D=? E=? Y=? N=? R=? O=? S=? M=? C1=? C2=?

C1 ,C 2, C3 stands for the carry variables respectively.

Goal State: the digits to the letters must be assigned in such a manner so that the sum is satisfied.

- The Solution process proceeds in cycles.
- At each cycle, 2 important things are done
    i. Constraints are propagated by using rules that correspond to the properties of arithmetic.
    ii. A value is guessed for some letter whose value is not determined

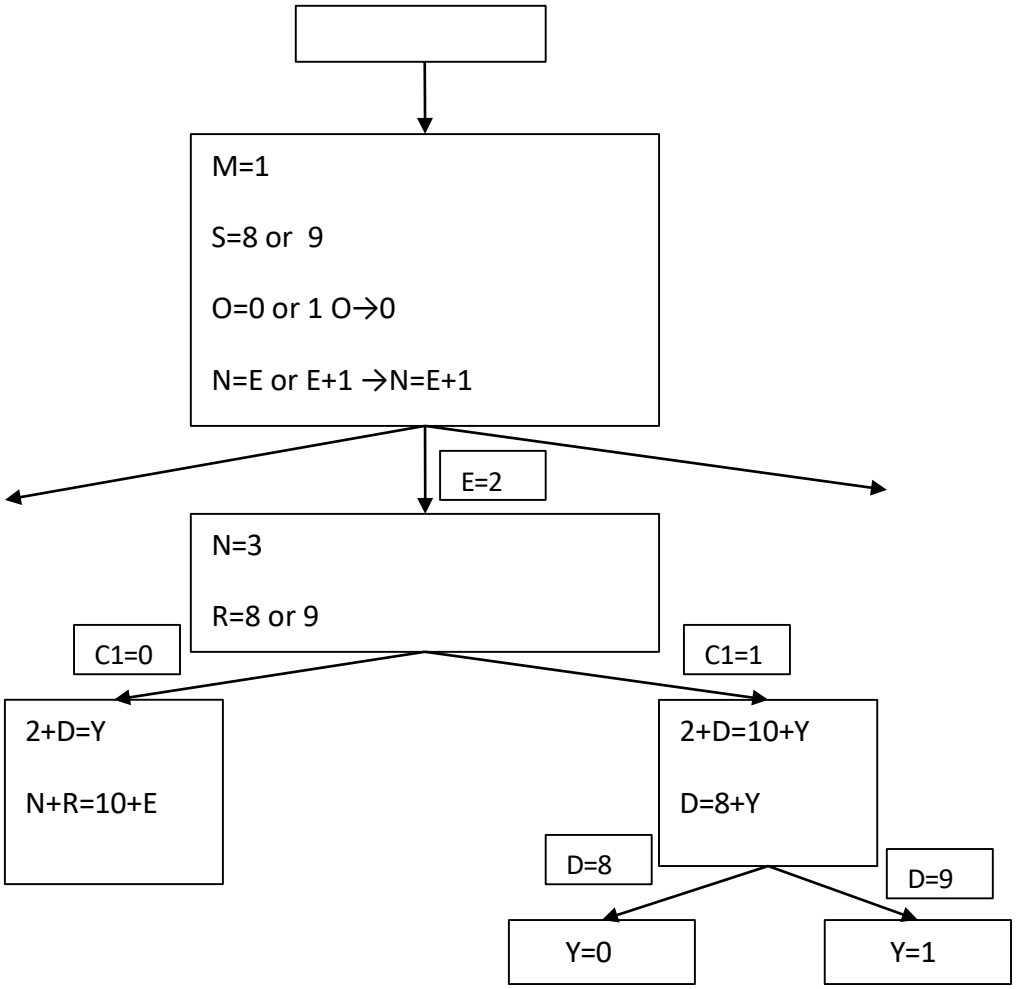

Fig Solving cryptarithmetic problem

C1,C2,C3 and C4 indicate the carry bits outs of the columns, numbering from the right.

Rules propagating constraints generate the following additional constraints:

- M=1, since two single digit numbers plus a carry cannot total more than 19.
- S=8 or 9, since S+M+C3>9 (to generate the carry) and M=1, S+1+C3>9, so S+C3>8 and C3 is atmost 1.
- O=0, since S+M(1)+C3(<=1) must be atleast 10 to generate a carry and it can be at most 11. But M is already 1, so O must be 0.

- N=E or E+1, depending on the value of C2.But N cannot have the same value as E. So N=E+1 and C2 is 1.
- In order to for C2 to be 1, the sum of N+R+C1 must be greater than 8
- N+R cannot be greater than 18, even with a carry in, so E cannot be 9.

Assume that no more constraints can be generated. To progress at this point, guessing happens. If E is assigned the value 2. Now the next cycle begins.

Constraint propagator shows that :

- N=3, since N=E+1
- R=8 or 9, since R+N(3)+C1(1 or 0)=2 or 12. But since N is already 3, the sum of these nonnegative numbers cannot be less than 3. Thus R+3+(0 or 1)=12 and R=8 or 9.
- 2+D=Y or 2+D=10+Y, from the sum in the rightmost column.
  Assuming no more constraint generation then guess is required.
  Suppose C1 is chosen to guess a value for 1, then we reach dead end as shown in the fig , when this happens , the process will be backtrack and C1=0.

Algorithm for constraint satisfaction in which chronological backtracking is used when guessing leads to an inconsistent set of constraints.

Constraints are generated are left alone if they are independent of the problem and its cause. This approach is called dependency directed backtracking (DDB).