Title: Crypto Arithmetic Problems

Theory

Crypto-arithmetic problems are mathematical puzzles where letters replace digits in arithmetic equations. The objective is to substitute the letters with digits (0–9) so that the equation holds true under the following constraints:

- Each letter represents a unique digit.
- No two letters can have the same numerical value.
- Leading letters of numbers cannot be zero.

These problems are solved using methods like logical deduction, trial and error, backtracking, and constraint satisfaction. Logical deduction involves analyzing the equation to identify possible digit assignments. Backtracking systematically explores combinations of digits, reverting assignments when conflicts arise. Constraint satisfaction narrows down potential values by eliminating invalid options.

The solution process typically involves ensuring that the arithmetic operations (addition, subtraction, or multiplication) remain valid while adhering to the constraints. For example, in addition problems, carry-overs must be considered to maintain numerical accuracy.

Crypto-arithmetic problems are widely used to test problem-solving skills and demonstrate the application of algorithms in logical reasoning and constraint-based problem solving.

PROLOG CODE:

1. BASE+BALL=GAMES

```
digit(0).
digit(1).
digit(2).
digit(3).
digit(4).
digit(5).
digit(6).
digit(7).
digit(8).
digit(9).

solution(Q):-
digit(B),B>0,
digit(A),
digit(S),
digit(E),
digit(L),
digit(A),
digit(G),G=1,
digit(M),
Q=[B,A,S,E,L,G,M],
1000*B+100*A+10*S+E+1000*B+100*A+10*L+L=:=10000*G+1000*A
+100*M+10*E+S,
different(Q).
different([X|R]):-not(member(X,R)), different(R).
different([]).
```

Output:

```
?- solution(Q).
Q = [7, 4, 8, 3, 5, 1, 9]
```

2.CROSS+ROADS=DANGER

```
digit(0).
digit(1).
digit(2).
digit(3).
digit(4).
digit(5).
digit(6).
digit(7).
digit(8).
digit(9).

solution(M):-
digit(C),C>0,
digit(R),R>0,
digit(O),
digit(S),
digit(A),
digit(D),D=1,
digit(N),
digit(G),
digit(E),
M=[C,R,0,S,A,D,N,G,E],
10000*C+1000*R+100*O+10*S+S+10000*R+1000*O+100*A+10*D+S=:=100000*D+10000*
A+1000*N+100*G+10*E+R,
different(M).
different([X|R]):-not(member(X,R)),different(R).
different([]).
```

Output:

```
?- solution(M).
M = [8, 4, 0, 7, 2, 1, 5, 3, 9]
```

3.CROSS+ROAD=DANGER

```
digit(0).
digit(1).
digit(2).
digit(3).
digit(4).
digit(5).
digit(6).
digit(7).
digit(8).
digit(9).
solution([H]) :-
    digit(C), C > 0,
    digit(R),
    digit(O),
    digit(S),
    digit(A),
    digit(D), D = 1,
    digit(N),
    digit(G),
    digit(E),
    H = [C, R, O, S, A, D, N, G, E],
    difference(H),
    10000*C + 1000*R + 100*O + 10*S + S + 10000*R + 1000*O + 100*A + 10*D + S =:=
    100000*D + 10000*A + 1000*N + 100*G + 10*E + R.
difference([]).
difference([X|R]) :-  not(member(X, R)), difference(R).
```

Output:

```
?-solution(X).
```

There is no output because the equation CROSS + ROAD = DANGER with the specified constraints does not have a valid solution.

Discussion

The Prolog code provided efficiently solves cryptarithm puzzles by leveraging its logical and constraint-solving capabilities. It uses predicates to define the rules, such as ensuring that digits are unique and satisfying arithmetic equations. The backtracking feature of Prolog systematically tests all possible combinations of digits, ensuring that the solution adheres to the constraints of the problem.

For the first problem, the solution ensures that each digit in the variables Q represents a unique value and satisfies the given equation. Similarly, in the second problem, the solution for M is obtained by adhering to constraints like non-zero leading digits, uniqueness, and a valid mathematical relationship. The results demonstrate how Prolog can handle complex logical problems with clarity and precision, showcasing its effectiveness in combinatorial tasks like cryptarithms.

Conclusion

The Prolog code successfully solves cryptarithm puzzles by leveraging logical rules and backtracking. It ensures digit uniqueness and satisfies all arithmetic constraints. The results highlight Prolog's efficiency in solving combinatorial problems and demonstrate its practical application in logical reasoning tasks.