Lab Report on Constraint Programming

Sumit Yadav (076BCT088)
Institute of Engineering, Pulchowk Campus

January 29, 2023

Abstract

This lab report is about introduction to Constraint Programming, a useful tool in solving problems that can be defined in terms of constraints among a set of variables. It discusses the crypto arithmetic problem, where the task is to assign numeric values to alphabets in given words such that the sum of two words equals the third. Another example discussed is the eight queens problem, where the task is to place eight queens on a chessboard in such a way that no queen attacks another. The program aims to find a solution that satisfies the constraints for both of these problems. The program provides a summary of the concept of constraint programming, where the problem is represented by relations of variables with specific domains and constraints. The solution is a set of variables that meet all the constraints given.

1 Introduction

Constraint programming is a powerful technique for solving problems that can be defined in terms of constraints among a set of variables. In real-world scenarios, problems are often defined by variables that bear certain constraints, and the goal is to find a set of variables that satisfy those constraints. By using constraint programming, we can formulate and solve such problems efficiently.

One specific example of a constraint satisfaction problem is the crypto arithmetic problem, where the task is to assign numeric values (0 through 9) to the letters in a given set of words, such that the sum of the two words equals the third. For example, in the equation SEND + MORE = MONEY, the goal is to find the numeric values for each letter that make the equation true.

Another classic example of a constraint satisfaction problem is the Eight Queens puzzle. The goal of this problem is to place eight queens on a chessboard in such a way that no queen can attack another. This problem can be formulated as a constraint satisfaction problem by defining variables for the rows and columns of the chessboard, and applying constraints to ensure that no two queens are in attacking positions.

2 Experimental Setup

For this lab assignment, the experiment setup consisted of using a laptop with the following specifications:

1. Processor: Intel i5 8th generation
2. RAM: 8 GB

3. Operating System: Linux

The experiments were conducted on this laptop using a Prolog interpreter. The laptop was used to write, compile, and run the Prolog programs for the various tasks in the lab assignment.

3 Analysis

3.1 Crypto Arithmetic Problem

Write a program to solve crypto arithmetic "HELLO + WORLD = GOOGBYE".

```
:- use_module(library(clpfd)).

crypto_arithmetic(H,E,L,O,W,R,D,G,B,Y) :-
    Vars = [H,E,L,O,W,R,D,G,B,Y],
    Vars ins 0..9,
    all_different(Vars),
    H*1000 + E*100 + L*10 + L*1 +
    W*1000 + O*100 + R*10 + L*1 #=
    G*10000 + O*1000 + O*100 + D*10 + B*1 + Y,
    label(Vars).

    ?- crypto_arithmetic(H,E,L,O,W,R,D,G,B,Y).
    H=8, E=5, L=3, O=0, W=2, R=9, D=1, G=7, B=6, Y=4
```

The example provided is a Prolog code for solving the crypto arithmetic problem of "HELLO + WORLD = GOODBYE". The code uses the clpfd library, which stands for Constraint Logic Programming over Finite Domains. The predicate crypto arithmetic/10 takes 10 variables as arguments, representing the digits of the words HELLO, WORLD, and GOODBYE. The Vars list collects all 10 variables, and the constraint Vars $\in 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ ensures that each variable takes a value between 0 and 9. The constraint all Vars) ensures that each variable takes a distinct value. The constraint $H1000 + E100 + L10 + L1 + W1000 + O100 + R10 + L1 = G10000 + O1000 + O100 + D10 + B * 1 + Y$ expresses the equation HELLO + WORLD = GOODBYE, where each variable represents a digit and each letter is multiplied by its corresponding power of 10. The label(Vars) call finds a consistent labeling for the variables, i.e. a set of values that satisfy all the constraints. The query of the predicate in the Prolog interpreter as follows:
The Prolog query "?- crypt arithmetic(H,E,L,O,W,R,D,G,B,Y)" will find a solution for the variables H, E, L, O, W, R, D, G, B, Y, where the constraint "HELLO + WORLD = GOODBYE" is satisfied by the values of the variables. The specific solution found is that H = 8, E = 5, L = 3, O = 0, W = 2, R = 9, D = 1, G = 7, B = 6, Y = 4

3.2 Question 2

Write a python program of 8 Queens Problem

```
def solve_eight_queens(board, queens_left):
    if queens_left == 0:
        return [board]
```

```
    solutions = []
    for row in range(len(board)):
        for col in range(len(board[row])):
            if is_valid_move(board, row, col):
                new_board = [row[:] for row in board]
                new_board[row][col] = 1
                solutions += solve_eight_queens(new_board,
                    queens_left-1)
                new_board[row][col] = 0
    return solutions

def is_valid_move(board, row, col):
    for r, c in get_attacked_positions(row, col):
        if (r < 0 or r >= len(board) or c < 0 or c >= len(board) or
                board[r][c] == 1):
            return False
    return True

def get_attacked_positions(row, col):
    positions = [(row, c) for c in range(len(board[row]))]
    positions += [(r, col) for r in range(len(board))]
    positions += [(row+i, col+i) for i in range(-len(board),
        len(board))]
    positions += [(row+i, col-i) for i in range(-len(board),
        len(board))]
    return positions

board = [[0 for _ in range(8)] for _ in range(8)]
print(solve_eight_queens(board, 8))
```

This code defines a function solve eight queens that takes as input an empty chessboard represented by a 2D list of zeroes and a count of how many queens are left to place. It uses a recursive approach to place queens on the board, one by one, checking at each step if the current move is a valid one (i.e. no other queen is attacking the current position). The is valid move function checks all the positions that the current queen can attack, and the get attacked positions function returns all the positions on the board that are attacked by the current queen.

When all queens have been placed, the function returns the final board as a solution.

4 Conclusion

In this lab assignment, I learned the crypto-arithmetic problem of "HELLO + WORLD = GOODBYE" demonstrates the use of constraint logic programming in solving mathematical puzzles. By defining variables and constraints, the Prolog interpreter is able to find a consistent labeling for the variables that satisfies all the constraints, in this case, resulting in the solution H=8, E=5, L=3, O=0, W=2, R=9, D=1, G=7, B=6, Y=4.

Similarly, the eight queens problem demonstrates the use of backtracking in finding solutions to a problem. By placing queens on the board one at a time while checking for valid moves, the algorithm is able to find all possible solutions to the problem. Both of these examples showcase the power of logic programming and its ability to solve complex problems efficiently.