Variations of the Eight Queens Puzzle

Madhulika Mitra, Jakob Weber, David Auinger

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
jku.at

Introduction

- The eight queens puzzle is a popular problem from 1848
- A manifold of solutions and implementation already exists
- Our goal is to solve a more general problem
  - any rectangular board size
  - other types of pieces: knight, bishop, rook, queen, amazon
  - allow mixing different types

Approach and implementation

- Entry point `main` function takes six lists as parameters
    - board size (length and width)
    - five lists of pieces: knight, bishop, rook, queen, amazon
    - `main([SizeX, SizeY], NS, BS, RS, QS, AS)`
- Function `possibleSolution` initializes all pieces
- Function `correctSolution` checks if initialization is a correct solution

Challenges

- Particular challenges due to mixing of types of pieces
- Cannot assume that a piece not attacking another piece is not itself attacked
  - Example: queen and knight, one might attack the other but not the other way
  - Need to check every piece against all other pieces of different types in both ways
- Problems with permutations
  - Prolog uses ordered lists, therefore permutations are considered unique solutions
  - However, this is not desired in our program
  - Define an order and only allow the permutation fulfilling the order

Pieces in detail

- Essentially two possibilities of attacks
  - moving to defined locations relative to current location (knight)
  - moving arbitrary distance in certain direction (bishop, rook, queen)
- Solve the knight by checking each knight with every other piece
  - no other piece in same location
  - no other piece in any of the eight locations possibly attacked by knight
  - use recursion to apply checks to all pieces
- Solve the bishop with similar recursion but different check
  - no other piece in same diagonals
- Similar checks for rook and queen
- Solve amazon by combining the checks of the knight and queen

Usage

- Input
  - classic eight queens puzzle

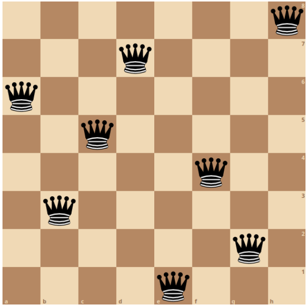
    ```
    main([8, 8], [], [], [], [Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8], []).
    ```
  - all types once on 5x5 board

    ```
    main([5,5],[N],[B],[R],[Q],[A]).
    ```
- Output prints locations of input pieces for a solution
- Also possible to count all solutions

  ```
  countSolutions([8, 8], [], [], [], [Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8], []).
  ```

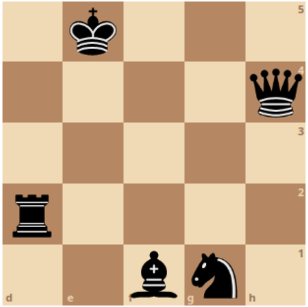
Classic eight queens problem

```
?- main([8, 8], [], [], [], [Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8], []).
Q1 = [1, 4],
Q2 = [2, 2],
Q3 = [3, 7],
Q4 = [4, 3],
Q5 = [5, 6],
Q6 = [6, 8],
Q7 = [7, 5],
Q8 = [8, 1]
```



All types once

```
?- main([5,5],[N],[B],[R],[Q],[A]).
N = [1, 2],
B = [1, 3],
R = [2, 5],
Q = [4, 1],
A = [5, 4]
```



Conclusion

- Successfully solves already solved problems
- Generalizes to different board sizes, other pieces, mixing of pieces
- Multitude of new problems and experiments possible

JKU

JOHANNES KEPLER
UNIVERSITY LINZ