DM509 Programming Languages

Fall 2006, 2nd quarter

Project 2

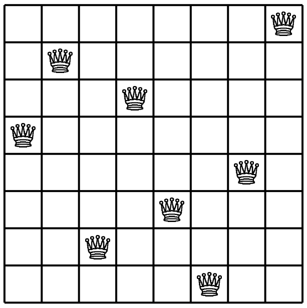
Department of Mathematics and Computer Science
University of Southern Denmark

December 14, 2006

The purpose of this project is to implement in Prolog a program finding solutions to the N-queens chess problem. The project is to be done in groups of two persons.

The N-Queens Problem

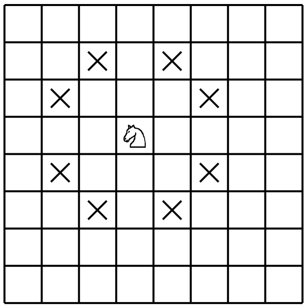
The 8-queens problem is a classic chess puzzle. The goal is to position 8 queens on a standard 8×8 chess board such that no queen threatens another. A queen can move unrestricted horizontally, vertically, and diagonally, hence a queen threatens any piece on the same row, column, or diagonal line as itself. The figure below shows one possible solution to the 8-queens problem.



The N-queens problem is the obvious generalization of positioning N queens on an $N \times N$ board with no queen threatening another.

Super-Queens

The allowed movements of a queen is a superset of the allowed movements of any other chess piece, except the knight. A knights move is one step horizontally and two steps vertically, or vice versa. The positions threatened by a knight is illustrated below.



Let us define a *super-queen* as a new chess piece whose allowed movements are those of the standard queen *and* those of the knight.

The N-super-queens problem is defined analogously to the N-queens problem: positioning N super-queens on an $N \times N$ board with no super-queen threatening another.

Task

The task is to implement a Prolog program which can find the number of solutions as well as the actual solutions of the N-queens problem and the N-super-queens problem. More specifically, implement in Prolog predicates

```
queenCount(N,Result)
queenSolutions(N,Result)
superqueenCount(N,Result)
superqueenSolutions(N,Result)
```

such that the first is true iff `Result` is the number of solutions to the `N`-queens problem, the second is true iff `Result` is an actual solution to the `N`-queens problem, and the last two are defined similarly, but for the N-super-queens problem.

Here, a solution is a list where the *i*th element gives the row of the (super-)queen in the *i*th column (this format is based on the observation that there must be exactly one queen in each column). For example, the list `[4,2,7,3,6,8,5,1]` describes the solution in the figure above.

Formalities

The report should be given as part of the comments of the source code for your solution. I.e., the source code should contain extensive comments equivalent to a report of around two to three pages. These comments should describe the general structure of the code, as well as the purpose of each predicate, including the role of all arguments.

Also, the comments part should contain a table showing the number of solutions to the N-queens problem for N ranging from 1 to 10, and also state the smallest N for which the N-super-queens problem has a solution.

No paper version should be handed in.

The Prolog code/report should be handed in using the `aflever` command on the Imada system: Move to the directory containing your code and issue the command `aflever DM509`. This will copy the contents of the directory to a place accessible by the lecturer. Repeated use of the command is possible (later uses overwrites the contents from earlier uses).

You must hand in the code/report by

Friday, December 29, 2006