

A new solution to the N-queens problem ?

Source: <http://coding.derkeiler.com/Archive/Prolog/comp.lang.prolog/2004-04/0183.html>

From: Clément (google_at_philosophons.com)

Date: 04/25/04

Date: 25 Apr 2004 09:59:06 -0700

Hello,

I would need help for (maybe) a new solution to the N-queens problem.

For recall, the problem is :

"Place N queens on a chessboard NxN so that no two queens are attacking each other; i.e., no two queens are in the same row, the same column, or on the same diagonal."

In my following programm, I Represent the positions of the queens as a list of numbers from 1 to N. Example: [1,3,4,2] means that the queen in the first column is in row 1, the queen in the second column is in row 3, etc...

- With this presentation, 2 queens can't be in the same column.
- For the rows, I use the classical solution :
 - a) Generation of a list of N numbers
 - b) Permutation of this list. (So, there is no duplicate, and 2 queens can't be in the same row.
- The diagonals.

I think I've found a solution for the diagonals (in theory), but I'm still having difficulties to implement it in Prolog :(
Here is my idea. In general, 2 queens attacks each others if $abs(X_i - X_j) = abs(i - j)$. No matter if my solution takes a lot of ressources, I don't really care for now; I just would like to implement this idea.

So, the programm looks like that :

```
% Qs is a solution of the N-queens problem
```

```
queens(N,Qs) :- echiquier(1,N,Rs) , permut(Rs,Qs), not(diag(Qs)).
```

```
% Generate the NxN chessboard, in two steps.
```

```
% First, echiquier(1,N,Rs) generate a list of N differents numbers.
```

```
% echiquier(A,B,L) : L is the list of numbers from A to B.
```

comp.lang.prolog: A new solution to the N-queens problem ?

```
echiquier(A,A,[A]).
echiquier(A,B,[A|L]) :- A < B, A1 is A+1, echiquier(A1,B,L).

% Second, permut\2 create the possible permutations of the generated
list.
% permut(Xs,Zs) :- the list Zs is a permutation of the list Xs.

permut([],[]).
permut(Qs,[Y|Ys]) :- del(Y,Qs,Rs), permut(Rs,Ys).

% del is needed for permut.
del(X,[X|Xs],Xs).
del(X,[Y|Ys],[Y|Zs]) :- del(X,Ys,Zs).

% Forbid the diagonals.
% In general, 2 queens attacks each others if abs(Xi-Xj)= abs(i-j).

... That's were my programm stops :(
I thought to generate all the couples of Xi first, and then compare
the list with the predicate nth1\3 (integrated in Prolog).
I've tried the following thing, but this is wrong (not all the
possibilities are eliminated):

% diag(Qs) :-
% member(X,Qs),
% member(Y,Qs),
% nth1(X,Qs,Nth),
% Y is X+1, nth1(Y,Qs,Nth_plus),
% abs(X - Y) = abs(Nth - Nth_plus).

Help would be very appreciated,
Regards,
Clement.
```