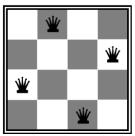
Constraint model for N-queens

Place n-queens on an $n \times n$ board so that no pair of queens attacks each other



Variables: x_1, x_2, x_3, x_4

Domains: $\{1, 2, 3, 4\}$

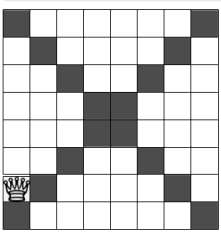
Constraints: $x_i \neq x_j$

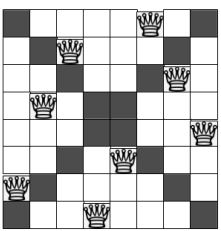
$|x_i - x_j| \neq |i - j|$

A solution: $x_1 \leftarrow 3$, $x_2 \leftarrow 1$, $x_3 \leftarrow 4$, $x_4 \leftarrow 2$

Example (Fancy queens)

I have placed a queen in one of the white squares of the board shown. Place 7 more queens in white squares so that no 2 of the 8 queens are in line horizontally, vertically, or diagonally





```
set(arithmetic).
assign(max_models,-1).
assign(domain_size,8).    %queens can be placed in one column from 0 to 7

formulas(classic_queens).
  all x exists y Q(x, y).                  %each row has at least one queen
  Q(x,y1) & Q(x,y2) -> y1 = y2.            %each row has at most one queen
  Q(x1,y) & Q(x2,y) -> x1 = x2.            %each column has at most one queen

  %each / diagonal has at most one queen
  Q(x1,y1) & Q(x2,y2) & (x2 + -x1 = y2 + -y1) -> x1 = x2 & y1 = y2.

  %each \ diagonal has at most one queen
  Q(x1,y1 ) & Q(x2,y2) & (x1 + -x2 = y2 + -y1) -> x1 = x2 & y1 = y2.
end_of_list.

formulas(fancy_queens).
   Q(x,y) -> x != y.                          %no queen on the main diag
   Q(x,y) -> x != domain_size + -y + -1.      %no queen on the sec. diag
   Q(6,0).                                    %there is a queen at (6,0)
end_of_list.
```

- $Q(x, y)$ is a predicate, not a function
- Note the usage of `domain_size`
- Note the usage of modules
- Note the usage of the minus operator