The eight queens problem

Let us now use Prolog to solve a more difficult kind of problem. One of these famous problems is **the eight queens problem**.

It consists in placing on a (European) chess board eight queens such that they do not attack each other.

We would like to define a predicate `solution` such that

```
?- solution(Pos).
```

returns a substitution for `Pos` which corresponds to a chess board position satisfying the problem's constraints.

The eight queens problem (cont)

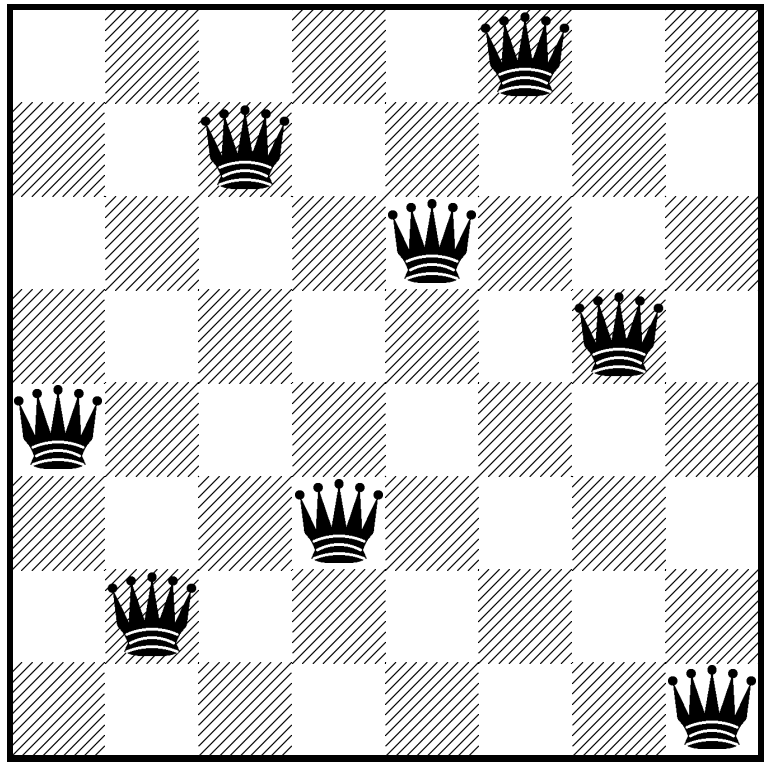

Figure: A solution to the eight queens problem.

The eight queens problem (cont)

First, we have to choose a representation for the board positions. One possibility is to model a square by two coordinates, the leftmost, down-most square being $(1, 1)$ and the rightmost, uppermost $(8, 8)$.

The example in the previous slide can be modeled by the list of queens

```
[.(1,4),.(2,2),.(3,7),.(4,3),.(5,6),.(6,8),.(7,5),.(8,1)]
```

Keeping this idea, we choose a template solution of the form

```
template([.(1,Y1),.(2,Y2),.(3,Y3),.(4,Y4),
          .(5,Y5),.(6,Y6),.(7,Y7),.(8,Y8)]).
```

because there must be a queen on each column.

The eight queens problem (cont)

There are two cases:

1. the list of queens is empty: the empty list is certainly a solution since there is no attack;
2. the list of queens is not empty: then it has the shape `[.(X,Y) | Others]`, that is, the first queen is on the square `.(X,Y)` and the others in the sub-list `Others`. If this is a solution, then the following constraints must hold.
   - 2.1 there must be no attack between the queens in `Others`, i.e. `Others` must be a solution;
   - 2.2 `X` and `Y` must be integers between 1 and 8;
   - 2.3 a queen at square `.(X,Y)` must not attack any of the queens in the list `Others`.

The eight queens problem (cont)

This is written in Prolog

```
solution([]).

solution([.(X,Y) | Others]) :-
  solution(Others),
  member(Y, [1,2,3,4,5,6,7,8]),
  no_attack(.(X,Y), Others).

member(Item, [Item | _]).

member(Item, [_ | Tail]) :- member(Item, Tail).
```

It remains to define the relation `no_attack`.

The eight queens problem (cont)

Given `no_attack(Q, Qlist)`, there are two cases.

1. if the list of queens `Qlist` is empty, then the relationship is true because there is no queen to attack or to be attacked by;
2. if the list `Qlist` is not empty, it must be of the shape `[Q1 | Qsublist]`, with the following conditions holding:
   - 2.1 the queen at `Q` must not attack the queen at `Q1`,
   - 2.2 the queen at `Q` must not attack the queens in `Qsublist`.

The eight queens problem (cont)

A queen does not attack another queen if they are on different columns, rows and diagonals.

Finally:

```
no_attack(_, []).

no_attack(.(X,Y), [.(X1,Y1) | Others]) :-
  Y =\= Y1,
  Y1 - Y =\= X1 - X,
  Y1 - Y =\= X - X1,
  no_attack(.(X,Y), Others).
```

The eight queens problem (cont)

The query has the shape

```
?- template(S), solution(S).
```

Note that

```
?- solution(S), template(S).
```

is **wrong**. Why?