Combinatorial Problem Solving (CPS)

Laboratory. Constraint Programming. Queens Problem.

In this session we will consider several approaches to the *Queens Problem*: given $n \geq 4$, place n queens on an $n \times n$ chess board, so that they do not attack each other. It can be seen that, as $n \geq 4$, there is always a solution. Recall that a queen attacks pieces on the same row, on the same column, and on the same diagonals.

1. Consider binary variables q_{ij} ($1 \leq i, j \leq n$) meaning: "there is a queen at cell (i, j)". Complete this model with arithmetic constraints stating that:

   - There are n queens.
   - For each row, there is at most one queen in it.
   - Idem for each colum, ascending and descending diagonal.

   Write a program `p` with Gecode that, given n from the command line, outputs a solution:

   ```
   $ ./p 4
   .X..
   ...X
   X...
   ..X.
   ```

   Use "first unassigned" for variable selection strategy, and "largest value" for value selection.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n? Use the `run` script below, which is also available at the website of CPS (http://www.cs.upc.edu/~erodri/cps.html):

   ```
   #!/bin/bash

   for i in $(seq 4 200); do
       echo "i:$i"
       time ./p $i
   done
   ```

   Recall to type `chmod +x run` on the command line to make it executable.

2. Note that, since we have to place n queens and there cannot be more than one per row, there must be exactly one per row. So consider integer variables c_i ($1 \leq i \leq n$) meaning: "the queen of row i is at column c_i". Complete this model with the corresponding arithmetic constraints, and write a program with Gecode that, given n from the command line, outputs a solution following the format above.

   Use the same variable and value selection strategies as in the previous exercise.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

3. Recall the `distinct` constraint in Gecode that, given an array of integer variables `x`, forces that they take different values (also recall its variants). Give an alternative model to the previous one by using this constraint.

   Use the same variable and value selection strategies as in the previous exercises.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

4. (*Breaking symmetries*) Which conditions can you impose on the position of the queen of the first row by symmetry? Add these constraints to your previous model.

   Use the same variable and value selection strategies as in the previous exercises.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

5. (*Tuning propagation*) Try different levels of consistency of the `distinct` constraint:

   - value consistency (default)
   - bounds consistency
   - domain consistency

   Use the same variable and value selection strategies as in the previous exercises.

   For the best setting, which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

6. (*Variable selection*) The *first fail principle* states that a good strategy for variable selection is to choose the one that "most probably" will lead to a conflict. One way to implement this is to choose dynamically the variable with the smallest domain. Add this search strategy to your previous model.

   Use the same value selection strategy as in the previous exercises.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

7. (*Redundant constraints, value selection*) Consider now integer variables r_j ($1 \leq j \leq n$) meaning: "the queen of column j is at row r_j". What is the relation between the c_i and the r_j? Add to the model this relation, together with the respective constraints for the r_j.

   The *best success principle* states that a good strategy for value selection is to choose the one that "most probably" will lead to a solution. In the queens problem, one way to implement this is, once the row (variable) has been chosen, to choose the column (value) with the least number of compatible rows.

   Add the following search strategy to your previous model: (*i*) the variable selection strategy chooses the row with the smallest domain; in case of a tie, it chooses the most centered one; and (*ii*), the value selection strategy takes the column with the least number of compatible rows; in case of a tie, it chooses the most centered one.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?

   Hint: Consult the document "Modeling and Programming with Gecode" for tie breaking and user-defined variable and value selection strategies.

8. Now implement a solution to the queens problem with a backtracking algorithm on your own, without using Gecode.

   Which is the least n for which it takes more than 5 secs. to produce a solution in your computer? How long does it take to solve the queens problem for this n?