N-Queens Configuration Project

Intelligent Systems Programming 2014

Jakob Melnyk, jmel@itu.dk
Jacob Claudius Grooss, jcgr@itu.dk

April 3, 2014

Implementation

Our implementation of the n-queens configurator uses binary decision diagrams (BDDs). We have an overall BDD used for decision making and a 2D array of BDDs (the BDDBoard). The BDDBoard corresponds to the game board and is used to make it easier to access the correct positions in the decision-making BDD. It also has the benefit of making the code easier to understand.

Our implementation consists of two parts:

1. The **initialization** part which sets up the game board and initialises the BDDFactory and the BDDBoard. The decision making BDD is built by applying the rules of the N-queens problem to the BDDBoard.
2. The **insert queen** part is a repeating step. It happens when the user attempts to place a queen on the board. If the move is valid, the BDD is used to update the game board to display invalid positions.

Building the BDD

When we build the overall BDD, we start with an empty BDD and an empty BDDBoard. We then iterate over every position on the BDDBoard and build the required rules for every position (see next section). For every column, we also apply the condition that it should have at least one queen for it to be a satisfiable solution.

Adding the rules for the problem

We add the n-Queens rules to the BDD by iterating over all the positions in the BDDboard and constructing the rules for each position individually.

The rules for each position is built by constructing a BDD for the column by applying ”not and” to each other position in the same column. This process is repeated for the row, the down right diagonal line and the up right diagonal line. At the end they are combined with the overall BDD.

Inserting a queen

Whenever a queen is inserted, we restrict the overall BDD with the BDD of the position of the newly inserted queen being true. After we have restricted the overall BDD, we update the game board by getting the valid positions of the board (described below) and closing the positions that are not valid.

Getting the valid positions

A valid position on the board is a position that does not contain a queen and would not lead to a unsatisfiable solution if a queen was placed there.

We calculate the valid positions of the board by iterating over every position on the board and attempting to place a queen on that position by restricting that position to true. If the restricted BDD is not zero (false), then that position is valid.

Conclusion

We conclude that we have successfully implemented the N-Queens configurator using the supplied BDD package. We have tested our solution on board sizes 1-12. The configurator offers no solutions on sizes 2-4 (as there are none for the Queens problem), but for the other board sizes, the configurator works as intended and does so at an acceptable speed[1].

[1]On our machines.