N queens problem solution using BDD’s

This reports contain explanation of 2 main elements of a solutions, namely: building initial BDD and updating the board, as they play a key role from authors perspective.

Building initial BDD

The build happens as soon as program is executed and a user is presented with an initial board. That allows to define the BDD which later on will be restricted each time user will add a queen to the board.

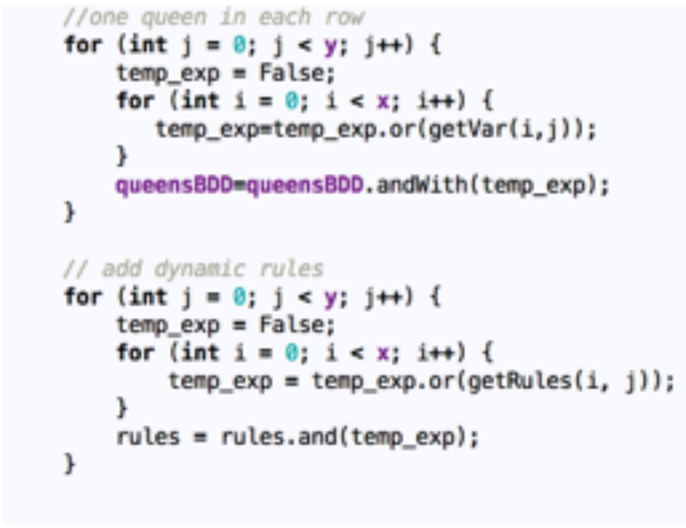
The most interesting part is applying 2 types of rules:

1. Each row has to have one queen.

2. Dynamic rules of placing a queen to the board.

The one queen in each row rule states that in each row: (col**1** or col**2** or …. or col**i)**, has to have a queen, where i is a last index of a column.

The dynamic rules are more complex and been implemented in a separate method getRules. Generally, it returns an (BDD)expression which contains binary representation of capture rules of a queen(horizontal, vertical, 2 diagonals).



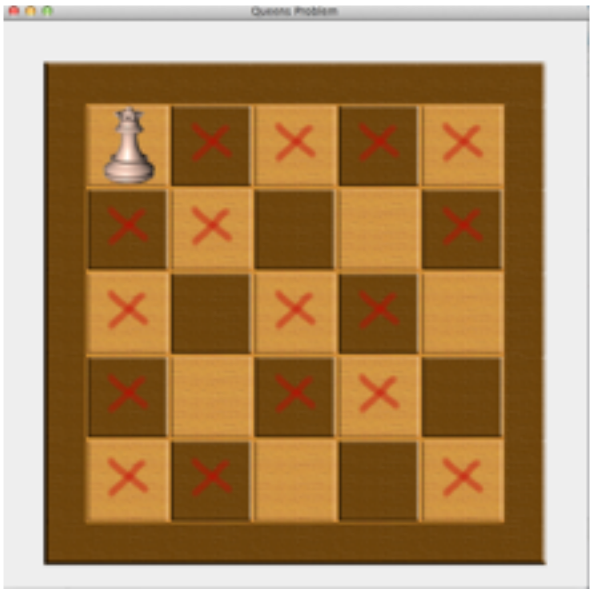
```
//one queen in each row
for (int j = 0; j < y; j++) {
    temp_exp = False;
    for (int i = 0; i < x; i++) {
        temp_exp=temp_exp.or(getVar(i,j));
    }
    queensBDD=queensBDD.andWith(temp_exp);
}

// add dynamic rules
for (int j = 0; j < y; j++) {
    temp_exp = False;
    for (int i = 0; i < x; i++) {
        temp_exp = temp_exp.or(getRules(i, j));
    }
    rules = rules.and(temp_exp);
}
```

When those rules are set, they are conjuncted together, as both has to be satisfied in order to have a solution.

Updating the board

When we start a program we want to verify if it’s generally possible to have a solution(for example if board is too small there is no solution). So we run an update method, which using restriction method tries to see if by placing a queen to every possible cell, queensBDD is satisfiable. If not the cross in a place of a cell is inserted.

The same happens when we insert a queen to the board, we update the board and verify if there are some cells which don’t lead to a solution. If those are found - they will be marked with a cross.


Queens Problem