Assignment 1: Encoding the n-Queens Problem into SAT

CS 6110 – Formal Methods in System Design

January 21, 2015

Deadline: **Wednesday, Jan 28, 2015 at 11:59pm.**

The n-Queens Problem: Given an integer n, find a way to place n queens on an $n \times n$ chessboard so that no two queens attack each other. Since a queen attacks along her row, column, and diagonals, a solution requires that no two queens share the same row, column, or diagonal. For more information about the problem, and to see all solutions for $n = 8$, refer to its WIKIPEDIA page at: http://en.wikipedia.org/wiki/Eight_queens_puzzle.

Your Task: Given that n-queens in an NP-complete problem, it is amenable to encoding into SAT. Your task is to write a program that takes an integer n as input and finds a solution to the n-queens problem by encoding it into SAT.
Program input: An integer $n > 0$.
Program output: A solution to the n-queens problem. (Note that I am looking for one solution and not all of them.) The output should be in the following format (example given for $n = 4$):

```
.X..
...X
X...
..X.
```

In the output, queens are marked with X. Here are some additional notes to get your started:

- I recommend you use MINISAT (http://minisat.se/) as your SAT solver. It is popular, fast, small, and easy to compile and use. You can either invoke it as an external command from your program (probably easier route), or use it as a library. If you prefer to use a different SAT solver, submit its binary with your solution.

Assignment Deliverables: Source code and a Linux (preferably) or Windows binary of your solution (sorry, no Mac OS). Also, a brief summary (at most one page, PDF format) explaining how your encoding works and how to invoke your solution. I want to be able to run your solution on my machine. You can assume that MINISAT is in the PATH variable on my machine. Email me the deliverables.

Bonus Points: Instead of generating just one solution, generate all possible solutions for the n-queens problem for a given n. I will also try to organize a little internal competition, and a solution that solves the largest instance in the given amount of time will get bonus points as well.