Satisfiability Test of Clauses and its Application

Tyler Mack Johnson || Sawanth Mythrey Bhonagiri || Venkata Revanth Kollipara

November 2021

Objective

To encode the n-queens problem into propositional logic, we need to write a program to generate a DIMACS CNF (Conjunctive Normal Form) encoding of the problem. The generated file can then be passed to a SAT solver such as MiniSat.

Software Required/Used

Linux Operating System, Visual Studio Code, MiniSat

Report

- The problem is to consider n number of rows and columns such that no two queens should be on the same row, column or diagonal and there should be at least one queen at any given row or column
- First, in our Linux Operating System, we downloaded MiniSat
- Next, we wrote a program using Python programming language to generate the DIMACS CNF encoding of the n-queen problem

The algorithm for the python script is explained below.

Step 1: Set a variable n to be the number of queens (as well as rows and columns)

Step 2: For each i (row) we append to the DIMACS CNF file a line with the clause $P_{i,0} \vee P_{i,1} \vee ... \vee ...P_{i,n}$. This assures that there must be one queen in each row of the board

Step 3: Now, the queens should not be repeated in the same columns

- Loop through every column in the same row and disallow two queens to exist with $\neg P_{i,j} \wedge \neg P_{i,j2}$ where $j2 = 0 \rightarrow n$ and $j2 \neq j$

Step 4: Also, make sure that there is no queen repeated in the same row

- Loop through every column in the same row and disallow two queens to exist with $\neg P_{i,j} \wedge \neg P_{i2,j}$ where $i2 = 0 \rightarrow n$ and $i2 \neq i$

Step 5: Now there shouldn't be any queens in any same diagonal

- Loop through each diagonal $i2, j2$ pair (centered around each position i, j) while $i2 \in 0 \rightarrow n$ and $j2 \in 0 \rightarrow n$

Step 6: After all the above conditions satisfy, write all of them into a file and save as out.sat

- After the code started working, we wrote this all out to a file and saved as out.sat, and checked its satisfiability, using MiniSat

Note:

- We encode each $P_{i,j}$ as $i + (j * n) + 1$ so that we can use the variables $1 \rightarrow n$ inclusive
- Additionally, we keep track of how many clauses are appended so that the file can be began with `p cnf <number of variables ($n^2$)> <number of clauses>`

Results

- For $n = 4$, the result is:

  `-1 -2 3 -4 5 -6 -7 -8 -9 -10 -11 12 -13 14 -15 -16`

- The positions 3, 5, 12, 14 has a queen on a 4x4 board (represented below)
- The following figure represents the positions in which the queens present when $n = 4$