CS 5384
Aniruddh Chavda
Project: Program to convert N-Queens Problem into CNF

=> The n-queens problem is to find an arrangement of N queens on an NxN board such that no two queens are in the same (1)row, (2)column or (3)diagonal.

There are NxN i.e. N^2 variables, and every variable has a binary value of either true or false.
True means the queen is present and False means the queen is not present.

Logic:

(1) Rows:

If one variable in a row is set to true, then every other variable in that row must be set to false. Assume a NxN board where P_{11} , P_{12} , P_{13} , P_{14} … P_{1N} are the variables in the first row. (Denoting by P_{ij} where i is the row and j is the column)

If P_{11} is true then P_{12} , P_{13} , P_{14} … P_{1N} must be set to false. Therefore, P_{11} -> -S where S is a Set of remaining variables of that row (If P_{11} is true then S is false). Thus, it can be written as:

P_{11} -> -S

Which can be further written as:
P_{11} -> $-P_{12}$
P_{11} -> $-P_{13}$
P_{11} -> $-P_{14}$
P_{11} -> $-P_{1N}$

Now applying atomic tableau "A -> B(implies) = -A \/ B(Disjunction)", we get:
$-P_{11} \vee -P_{12}$
$-P_{11} \vee -P_{13}$
$-P_{11} \vee -P_{14}$
$-P_{11} \vee -P_{1N}$

Now we need to consider P_{12} to be true and all the other variables to be false and use the above method to get the clauses.

Similarly, we need to consider each of the variables of that row to be true one by one and every other variable to be false and then create the clauses.

★ **As it is N queen on NxN board, there must be 1 queen per row and thus we have:**
P_{11} V P_{12} V P_{13} V P_{14} V … V P_{1N}

2) Columns:

The logic remains the same as that of rows, the only difference is that instead of keeping row's value constant, here the column's value will be constant. For example:

P_{11} -> -P_{21}
P_{11} -> -P_{31}
P_{11} -> -P_{n1}

3) Diagonals:

The diagonals need to be checked from left to right and right to left. Thus, same rules should be applied diagonally. But the bounds check should be active as the elements in a diagonal range from 2 to N.

For left to right:
P_{ij} -> $P_{(i+1)\ (j+1)}$
P_{ij} -> $P_{(i+2)\ (j+2)}$
.
.
P_{ij} -> $P_{(i+(n-1)\ (j+n-1)}$

For right to left:
P_{ij} -> $P_{(i-1)\ (j-1)}$
P_{ij} -> $P_{(i-2)\ (j-2)}$
.
.
P_{ij} -> $P_{(i-(n-1))\ (j-(n-1))}$

Python program:

The Program will take the input N and output a DIMACS CNF file which can be used in minisat to determine its satisfiability and to find a solution. Also as solutions of N<4 are not possible, it will give the error message as “Invalid , The value of N should be greater than 3”.

Command to execute minisat:

```
minisat2 <CNF formula file> <output file>
cat <output file>
```

Copy the content of the output file , press enter after counting N elements and a structure would be formed with N*N variables having N true and (N*N-N) false values. That’s exactly the board with N queens placed such that it satisfies the rules.

rules for using miniSAT:

1) $\vee$ (Disjunction) is denoted by (“ ”) blank.
2) $\wedge$ (Conjunction) is denoted by (“0”) zero.
3) A comment starts with c.
4) p CNF Variables (N*N) Clauses (Here p followed by problem type is used (CNF for our case) , then followed by variables and number of clauses)