Satisfiability test of clauses and its application

Venkata Hemanth R11759266
Dinesh Reddy R11782006
Praveen Reddy R11793392

November 13, 2021

1 N-Queens

The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other.

2 Propositional Logic for N-Queens

- For NxN chess board there are n^2 cells for a queen to be placed. We represent the presence of queen with the proposition $P_{i,j}$.
- i and j are parameters which represents the respective row and column, where $i = 1, 2, ...n$ and $j = 1, 2, ...n$.
- If $P_{i,j} = T$, then the queen is present in i^{th} row and j^{th} column.
- If $P_{i,j} = F$, then the queen is not present in i^{th} row and j^{th} column.
- We make sure there is a queen present in every row.

  $$P_{i,1} \vee P_{i,2} \vee .... \vee P_{i,n}$$

  According to N- Queens, there should'nt be any queen in all cells of ith row except for cell of jth column.

  $$\forall k \; P_{i,j} \rightarrow \neg P_{i,k}, \text{ where } k \in [i+1, n] \text{ and } k \neq j$$

- Similarly, there should'nt be any queen in all cells of jth column except for cell of ith row.

$$\forall k \ P_{i,j} \rightarrow \neg P_{k,j}, \text{ where } k \in [j+1, n] \text{ and } k \neq i$$

- Addition to the row and column conditions, from the definition of N-Queens, we also check for diagonal cells.
  - Right Diagonal: We iterate through all diagonal cells to the right of the current column.

    $$\forall k \ P_{i,j} \rightarrow \neg P_{i+k,j+k}, \text{ where } k \in [1, min(n-i, n-j)]$$

  - Left Diagonal: Correspondingly, We iterate through all diagonal cells to the left of the current column.

    $$\forall k \ P_{i,j} \rightarrow \neg P_{i+k,j-k}, \text{ where } k \in [1, min(n-i, j-1)]$$

- The connective '$\rightarrow$' must be written in the Conjunctive Normal Form, so it should be converted to '$\vee$' connective.

$$P \rightarrow Q \Leftrightarrow \neg P \vee Q, \text{ where P,Q are propositions.}$$

3 DIMACS CNF

DIMACS (The Center for Discrete Mathematics and Theoretical Computer Science) CNF is a standard input CNF format for SAT(Satisfiability) Solvers.

We write a program to generate the above propositional logic CNF Form into DIMACS CNF format.

Pseudo Code:

Algorithm 1 To check if there is a queen in every row:

```
for i: 1 to n^2 do
    print 'i'
    if i % n is 0 then
        print '0'
    end if
end for
```

Algorithm 2 To check there is no other queen present in a row:

```
for i: 1 to n^2 do
    for j: i to (int(i/n) + 1) × n do
        if j is not i then
            print '-i -j 0'
        end if
    end for
end for
```

Algorithm 3 To check there is no other queen present in a column:

```
for i: 1 to n^2 do
    while j: i to n^2 do
        if j is not i then
            print '-i -j 0'
        end if
        j ← j + n
    end while
end for
```

Algorithm 4 To check there is no other queen present in right diagonal:

```
for i : 1 to n^2 do
    row ← (int(i/n)+1)
    col ← i % n
    col ← col is 0 ? n : col
    while j : i to min(n^2, (n - col + row)*n + 1) do
        if j is not i then
            print '-i -j 0'
        end if
    end while
end for
```

Algorithm 5 To check there is no other queen present in left diagonal:

```
for i: 1 to n^2 do
    while j: i to n^2 do
        if j is not i then
            print '-i -j 0'
        else if int((j - (n-1)) / n) is int(j / n) then
            exit While
        end if
        j ← j + n − 1
    end while
end for
```

4 SAT Solver

- SAT Solver(Satisfiability) is used to determine if for a given propositional logic formula there exists an interpretation which satisfies the formula.
- We use a miniSAT solver that takes a simple DIMACS CNF file as input that is generated with the above algorithm(section 3) to solve the N-Queens problem.
- For n = 4, there would be 16 propositions and 80 clauses in the DIMACS CNF file that was generated. The DIMACS CNF file is given as input to the miniSAT solver which produces a output that satisfies the N-Queens.
- **Output :** -1 -2 3 -4 5 -6 -7 -8 -9 -10 -11 12 -13 14 -15 -16 0
- The above output can be made into propositional logic as:
- -1 means $P_{1,1}$ has no Queen present in the cell (1,1) making proposition $P_{1,1}$ False.

$$P_{1,1} = F$$

- Similarly, -2, -4, -6, -7, -8, -9, -10, -11, -13, -15, -16 make their corresponding propositions False.

- 12 means $P_{3,4}$ has a Queen present in the cell (3,4) making proposition $P_{3,4}$ True.

$$P_{3,4} = T$$

- 3, 5, 14 making their corresponding proposition True.
- The Queens from above output are placed as shown in Figure 1.

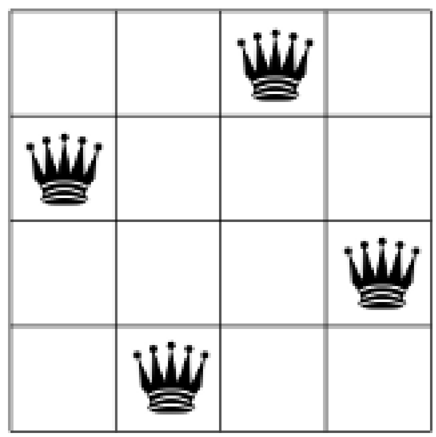

Figure 1: N-Queens