Satisfiability test of clauses and its application

Vardhineni saketh rao
R11840584

Vishwasena Raidu Nyaramneni
R11849314

Farooq Abdulla Shaik
R11842924

November 8, 2022

1 N-Queens

The n-queens puzzle is the problem of arranging n queens on a n x n chessboard so that no two queens attack each other.
1) No two queens in the same row
2) No two queens in the same column
3) No two queens in the same diagonal.

The N-Queens contain N queens where the size of the chessboard is N X N (i.e it contain N rows and N columns). Let us consider a proposition letter for each position in the chessboard p_{ij} where i represent the i th row and j represent j th column of the chessboard.$1 \leq i \leq N, 1 \leq j \leq N$)
If the proposition letter p_{ij}=T then Queen is present at the location (i ,j)
If the proposition letter p_{ij}=F then Queen is not present at the location (i,j)
Consider there is an N X N chessboard. There are 5 rules in the N-queen problem (i.e)
(1) There should be a queen in every row.

$$p_{i1} \vee p_{i2} \vee p_{i3} \vee p_{i4} \vee .....\vee p_{in} \text{ where } 1 \leq i \leq N$$

(2) There should be no two queens on the same row.
If there is a Queen at a location (i,j) on the chessboard then proposition letter at that location is True i.e p_{ij} = T and there should

be no queens on the the i the row .(i.e) remaining all proposition letters in that i th row must be false

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

(3) There should be no two queens on the same column.
If there is a Queen at a location (i,j) on the chessboard then proposition letter at that location is True i.e p_{ij} = T and there should be no queens on the the j the column .(i.e) remaining all proposition letters in that j th column must be false.

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

(4) There should be no two queen on the same diagonals
(4.1) It means there should be no two queens on the right diagonals. If there is a Queen at a location (i,j) on the chessboard then proposition letter at that location is True i.e p_{ij} = T and there should be no other queen on that right diagonal.

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

(4.2) It means there should be no two queens on the left diagonals. If there is a Queen at a location (i,j) on the chessboard then proposition letter at that location is True I.e p_{ij} = T and there should be no other queen on that left diagonal.

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

2 Pseudo Code

Algorithm 1 To calculate number of clauses

rightD=((n-1)*n)/2

t=n-2

while t>0 **do**

  rightD=rightD+(2*(t*(t+1))/2)

  t=t-1

end while

Clauses=n+(2*(n*(n^2-n)/2))+(2*rightD)

print 'p cnf n^2 Clauses'

Algorithm 2 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 3 To check if every row has only one queen:

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

Algorithm 4 To check if every column has only one queen:

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

Algorithm 5 To check if every right diagonal has only one queen:

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

Algorithm 6 To check every left diagonal has only one queen:

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

3 SAT Solver

(1) Install minisat in unix by typing the following command.

”**sudo apt install minisat**”.

(2) After installing the minisat into unix change the directory to the location where DIMACS CNF format file is located. DIMACS CNF file is the output of the python program which converts the above mentioned propositional logic CNF form to DIMACS CNF form.
(3) Run the DIMACS CNF format file by typing the following command.

”**minisat CNF_file.cnf output1.txt**”

(4) The command prompt displays whether the given formula is satisfiable or unsatisfialble.
(5) The output file contains SAT and the assignment which satisfies the formula if it is satisfiable. If the given formula is unsatisfiable then output text file contains UNSAT.

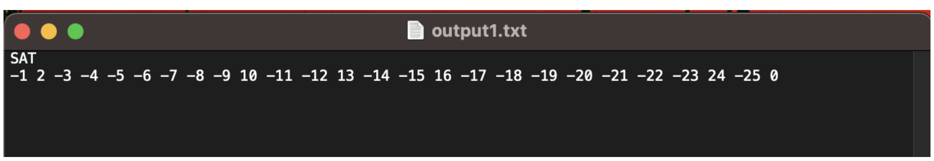

Figure 1: Satisfiable output file

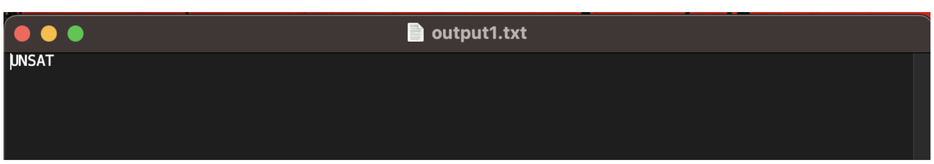

Figure 2: Unsatisfiable output file

(6) The positive literals in the assignment represent the queen location in the chessboard.

4 Output

For the input n=5, the output files consists of the following data.

SAT

-1 2 -3 -4 -5 -6 -7 -8 -9 10 -11 -12 13 -14 -15 16 -17 -18 -19 -20 -21 -22 -23 24 -25 0

In the above output, '0' is the end of line.

The positive literals in the given output are the locations of the queens. '3' is a positive literal given in the output which means, Queen at a location (1,3) on the chessboard then proposition letter at that location is True i.e p_{13} = T.
Similarly

$$p_{25} = \mathrm{T}$$
$$p_{33} = \mathrm{T}$$
$$p_{41} = \mathrm{T}$$
$$p_{54} = \mathrm{T}$$

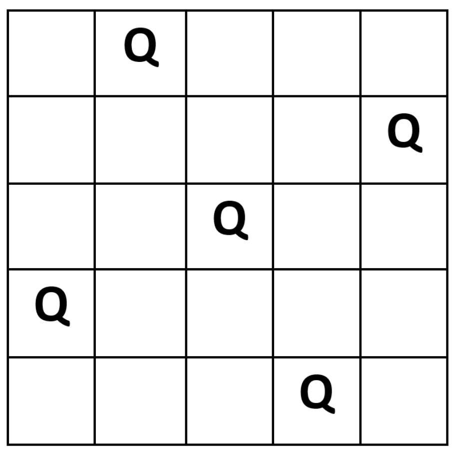

Figure 3: 5 - Queens