Universitat Pompeu Fabra
Barcelona

Computational Logic

Programming Assignment 1

Introduction

In this assignment, we consider a problem that can be modeled in propositional logic and use a SAT solver to solve it. Your task will be to write a Python program, which can model the problem as a Propositional Logic formula in Conjunctive Normal Form (CNF) in DIMACS format. You will then use either the SAT solver provided or a solver of your choice to find a solution for the problem. You will be divided into groups of at most 2 students, as per your choice and each group must complete the assignment according to the parameters given to them

CNF Formulas In DIMACS Format

The DIMACS (http://www.satcompetition.org/2009/format-benchmarks2009.html) format is widely used by most SAT solvers, an example is given below

```
c
c start with comments
c
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

The file can start with comments, that is lines begining with the character c

Right after the comments, there is the line p cnf nbvar nbclauses indicating that the instance is in CNF format; nbvar is the exact number of variables appearing in the file; nbclauses is the exact number of clauses contained in the file.

Then the clauses follow. Each clause is a sequence of distinct non-null numbers between -nbvar and nbvar ending with 0 on the same line; it cannot contain the opposite literals i and -i simultaneously. Positive numbers denote the corresponding variables. Negative numbers denote the negations of the corresponding variables.

In the example shown above, there are 5 variables and 3 clauses. For the ease of presentation, let

us denote the variables by X1=1, X2=2, X3=3, X4=4 and X5=5.
Then the three clauses shown in the example are:

X1 OR NOT X4 OR X4
NOT X1 OR X5 OR X3 OR X4
NOT X3 OR NOT X4

The N Queens Problem

The N Queens problem is a historically famous mathematical problem, which mathematicians have been working on since the 19th century. (https://en.wikipedia.org/wiki/Eight_queens_puzzle). The problem consists of an NxN chessboard on which one has to place N queens such that no two queens are directly capable of attacking each other, as shown in Figure 1 for N=8.

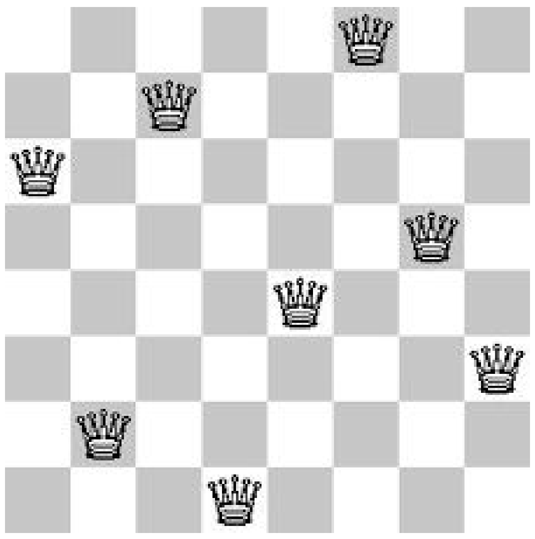
Figure 1: A solution for the N queens problem, for N = 8.

For those of you who are not familiar with the rules of chess, a queen piece on a chess board is capable of attacking other pieces which are in the same line either horizontally or vertically. The queen is also able to attack pieces which are on the same diagonal as it. Therefore the solution to the N queens problem can be written in plain text as:

Place each of the N queens on the NxN chessboard such that no two queens are on the same horizontal line, vertical line or on a diagonal. Since there are N queens and the chessboard is NxN, it follows that each row and each column must have one queen, but can have no more than one. However, this is not true for diagonals, each diagonal can have at most one queen. The conditions for the N Queens problem can be written as:

1. Each row contains a board cell occupied by a queen
2. Each column contains a board cell occupied by a queen
3. For each pair of board cells occupied by queens, the cells
    a. are not in the same row
    b. are not in the same column
    c. are not in a same diagonal

Conventions

- The variables corresponding the cells of the chessboard will be represented by integers in the DIMACS format, starting from the top left going to the bottom right. This is shown below for a 4x4 grid. Each of these integers can be thought of as a variable.

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

▪

- The meaning of variable i is that i is true when it contains a queen

- Following the DIMACS format, the negation of a variable is represented by a - sign.

To illustrate, one of the solutions of the N Queens problem (shown in Figure 2), for N=4 is represented in DIMACS format as

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

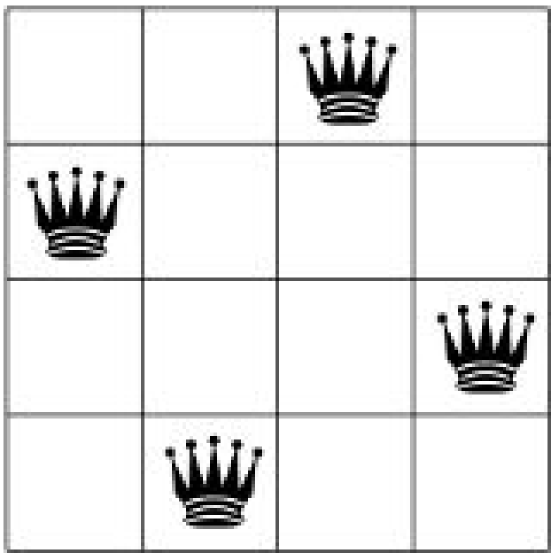

Figure 2: A solution to the 4 Queens problem

Task

You are asked to produce python code that outputs a CNF formulation of the N queens problem in DIMACS format. To guide this programming, you are provided with a template named nqueens.py. It must be called with the integer value of n and will initially print the chessboard in the required format when called.

For example, the call

```
$ python nqueens.py 4
```

Will initially display

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

You will have to complete the following functions, corresponding to the conditions listed in the first section:

AtLeastOne(VariableList):

AtMostOne(VariableList):

These can then be applied to the rows, columns and diagonals of the board to obtain the final solution of the problem in the function *nqueens(n, ready=False):*

A helper function, *createDiagonals(board)*, has been provided, which returns a list containing the diagonals of the board used. Remember that each column and each row must have at least one queen but can have no more than one queen. Also, each each diagonal can have at most one queen, but it is not necessary for each diagonal to have a queen.

Once you have implemented the functions, you can set the flag ready to True in the *main()* function and run the program from the terminal. This will write the clauses to the file nqueens_sol.in. This file must be in the same format illustrated above, which is accepted by SAT solvers.

Solver

You are provided picosat SAT solver, which works on Linux and Mac operating systems. However, you are free to use a SAT solver of your choice. There are some options available on http://baldur.iti.kit.edu/sat-competition-2016/solvers/main/

To install these solvers,

Go to the picosat directory

```
$ sh configure.sh
$ make
```

- picosat
  ./picosat/picosat nqueens_sol.cnf determines if the formula is satisfiable and returns the first solution

  ./picosat/picosat --all nqueens_sol.cnf print all the truth assignments that makes the formula true. Not recommended to try for large problems.

Submission and Evaluation

You must submit the completed code in nqueens.py as well as a document with

1. A visual representation of the solution on a chessboard. You are free to use any program to generate the visual representation and it may be in text format if required. No extra marks will be given for a good representation.
2. Time to solve problem for n=10, 30, 50.
3. What is the largest n that can be solved in a few minutes?
4. What is the number of propositional symbols required for 3?
5. What is the number of clauses in 3?

Submission deadline : 16/05/2017 at 23:59.

Submit using an email to pritish.chandna@upf.edu. Each submission must have the subject Computational Logic Lab 1 Group <Group Number>