Project 02

SAT approach for 8-queens problem

1. Description

Consider a chessboard of size 8x8. Florence wants to place eight queens on the board so that they cannot attack each other. Instead of applying the backtracking recursion algorithm, she would like to approach the problem as an SAT problem. In this way, she assigns a Boolean logical variable to each cell on the board, and the variable equals true if there is a queen and vice versa.

Please help Florence by completing the following tasks:

a. Formulate the problem by specifying the following points.
   -   The input(s) and output(s) of the problem
   -   The data structures that represent variables and any state of the program
   -   Correspondingly, demonstrate how the initial and goal states will look like.

b. Each logical variable matching a cell in the chessboard is named according to the row and column indices. For example, variable b[0][0] corresponds to cell [0][0] and b[0][1] for cell [0][1]. Write CNF clauses to describe restrictions required when Florence places a queen in the cell[3][3].

	0	1	2	3	4	5	6	7
0	x			x			x	
1		x		x		x		
2			x	x	x			
3	x	x	x	b[3][3]	x	x	x	x
4			x	x	x			
5		x		x		x		
6	x			x			x	
7				x				x

c. Applying the same procedure above, Florence obtains a set of CNF clauses to describe the whole chessboard when she consecutively places all the eight queens on the board. You help her create a function that returns the expected set, in which an integer v = i*8+j+1 denotes the variable b[i][j] and the value -v represents negation. For instance, b[0][0] ^ b[0][1] ^ not (b[1][1]) → 1^2^-10.

- Level 1: Each queen can only move on a single column.
- Level 2: The position of the queens is any cell on the chessboard.

d. Suppose Florence obtains n CNF clauses from Task (c). If she can find a set of values that satisfy all the CNF clauses, she will know the exact positions of eight queens. Create a program to generate CNFs clauses using Python3 and then find a set of satisfied values using the PySat library.

e. Implement a function to find a suitable assignment of variables using A* algorithm, given that some queens are placed in advance. Carefully describe in your report how you represent any state, acquire successors of a state, and how you utilize the heuristic.
Note that:

- The function receives a list of positions for m queens already placed (m < 8). These positions are guaranteed to be valid, at which the queens do not attack each other.
- The list of positions is specified by a text file whose structure is as follows.
  - The first line: the integer m indicates the number of queens the user wants to place on the chessboard.
  - The second line: the positions of m queens comply with the syntax (x, y). A single space separates two consecutive parts.
- All the logical variables are initially false, except those corresponding to the locations specified above.
- You are required to code A* by yourself without using any support libraries.

f. Implement a program to visualize the chessboard with the result found.
Note that:

- The program lets the user specify the text file and click a button to trigger the search procedure. The program calls the above function with the user-specified text file and then visualizes the state at each step of the search path.
- There is a graphical form to show the chessboard with the solution found, in which none of the eight queens attacks the others.

- A text file including the set of CNF clauses, each of which is on a separate line.

You need to submit the following materials to Moodle.

Video:

- A video recording that shows the process of running your program with given test cases and obtaining results.
- The video recording should be uploaded to YouTube. Share the link to the report.

Source code: the entire your source code (using Python 3) with running instructions.

Report:

- Group Roster: Who are the group members? What did each person do? How many percent of the work is completed?
- Your answers for a, b, e tasks.

2. Requirements

No.	Criteria	Scores
1	Task a	10%
2	Task b	10%
3	Task c includes: Level 1: Level 2:	(30%) 15% 15%
4	Task d	20%
5	Task e	10%
6	Task f	10%
7	Comply with the regulations of submission requirements	10%
Total		100%

3. References

There are some references that you may need to make your solution:

i. https://developers.google.com/optimization/cp/queens

ii. http://www.sfu.ca/~tjd/310summer2019/chp7_proplogic.html

4. Notices

- Each group has a maximum of 2 members.

- Beside the above requirements, the report must also give the following information:
  - Estimating the degree of completion level for each requirement.
  - References (if any)
- Any plagiarism, any tricks, or any lie will have a zero score for the COURSE grade.