Logic and Model Checking Fall 2024

Project 1: SAT

Lecturer: Jaime Ramos *Department of Mathematics, Técnico*

1 Resolution

Consider the formula

$$(\neg p_1 \vee p_3 \vee p_4 \vee p_5) \wedge (\neg p_3 \vee p_4 \vee p_5) \wedge (\neg p_1 \vee p_3 \vee \neg p_4) \wedge (p_1 \vee p_2) \wedge (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee \neg p_5) \wedge (\neg p_3 \vee \neg p_4 \vee p_5)$$

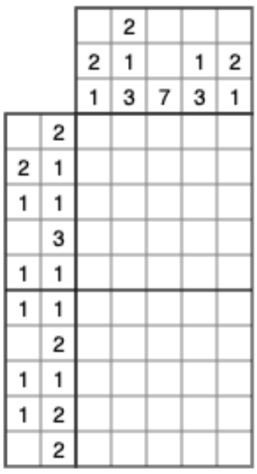
Apply the CDCL algorithm to this formula to decide if it satisfiable or not. For each conflict, draw the implication graph and learn the conflict clause corresponding to the principal cut. The literals should be decided in the order of their subscripts, that is, p_1, p_2, p_3, p_4, p_5 and choosing always the positive case first.

2 Nonogram

2.1 Description

Nonogram is a logic puzzle in which cells in a grid must be either coloured black or left blank, according to numbers at the edges of the grid, in order to reveal a hidden picture. On the left of each row of the grid are listed the lengths of the runs of black squares on that row. Above each column are listed the lengths of the runs of black squares in that column. These numbers define the runs of (consecutive) black squares in that row/column. Between any two runs of black squares in the same row/column there is always at least one blank square, and there may be blank squares before the first and after the last run.

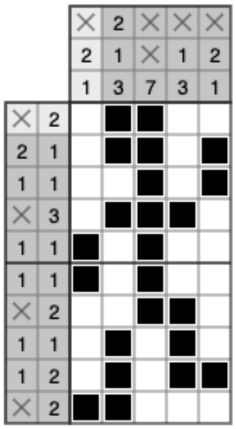
Consider the grid below.



The number 2 on the left of the first row tells us that there is a single run of 2 consecutive black squares in that row. The four possibilities for this row are:

The list of numbers 2,1 on the left of the second row tells us that there is a run of 2 black squares followed by a run of 1 black square, and these two runs must be separated by at least one blank square. In this case, the three possibilities are:

The following picture is a (unique) solution of the puzzle, that respects all horizontal and vertical constrains:



A nonogram puzzle is well posed if it has a unique solution.

2.2 Exercise

You are asked to encode *nonogram* as a SAT problem: given two lists V and H of lists of positive integers, corresponding to the lists of vertical and horizontal constraints, respectively, encode the problem of finding a solution for that nonogram as a SAT problem. Observe that the length of list V determines the number of columns of the grid, and the length of list H determines the number of rows of the grid. Based on this encoding, and using Z3, define the following functions in Python:

- `nonogram(V,H)` – function that given two lists `V` and `H` of lists of positive integers, determines a solution for the nonogram, if there is one.
- `well-posed(V,H)` – function that given two lists `V` and `H` of lists of positive integers, determines if the given constraints define a well posed puzzle.

For instance, evaluating

```
nonogram([[2,1],[2,1,3],[7],[1,3],[2,1]],[[2],[2,1],[1,1],[3],[1,1],[1,1],[2],[1,1],[1,2],[2]])
```

should return the solution of the puzzle that is represented as a list of lists of blank and black squares, as for instance,

```
[[0, 1, 1, 0, 0],
 [0, 1, 1, 0, 1],
 [0, 0, 1, 0, 1],
 [0, 1, 1, 1, 0],
 [1, 0, 1, 0, 0],
 [1, 0, 1, 0, 0],
 [0, 0, 1, 1, 0],
 [0, 1, 0, 1, 0],
 [0, 1, 0, 1, 1],
 [1, 1, 0, 0, 0]]
```

where `0` corresponds to a blank square and `1` corresponds to a black square.

3 Submission

The project is to be submitted in the *Fenix* platform by a member of the group (after the group has registered). The submission should consist of a **single compressed folder** containing:

- a report on the project, that should include the answer to the first exercise, the encoding of the nonogram problem as a SAT problem, and execution examples;
- a Jupyter notebook (or equivalent) with the implementation of the requested functions.

Submission deadline: **October 19, 2024, at 23:59.**