Institut für Informatik
der Universität München
Dr. J. Johannsen
Leah Neukirchen

WS 2018/2019

October 22, 2018

Exercises on
SAT Solving

Problem set 1

Exercise 1: Setting up PicoSAT and learning about the DIMACS CNF format. (Not to be handed in.)

1. Download and compile PicoSAT 965 from `http://fmv.jku.at/picosat/`, which we will use in further exercises.

2. The *DIMACS CNF file format* is the standard format for feeding SAT solvers with formulas. DIMACS CNF files are simple text files looking like this:

   ```
   p cnf 3 2
   1 -3 0
   2 3 -1 0
   ```

   The line starting with `p` is the header line and says the formula consists of 3 variables and 2 clauses.

   Each of the following lines specifies a clause: positive literals are denoted by the corresponding number, and negative literals by the corresponding negative number. Each clause is terminated with a zero.

   Hence above formula is equivalent to $(v_1 \vee \neg v_3) \wedge (v_2 \vee v_3 \vee \neg v_1)$.

3. Now let PicoSAT solve this formula:

   ```
   % ./picosat-965/picosat simple.cnf
   s SATISFIABLE
   v 1 2 -3 0
   ```

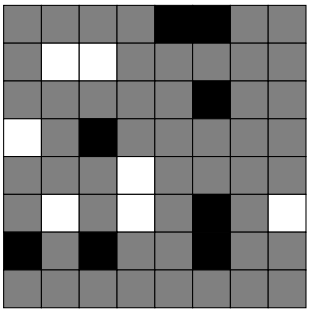
   PicoSAT computed that the formula is satisfiable and that we can assign $\{v_1 := \textbf{true}, v_2 := \textbf{true}, v_3 := \textbf{false}\}$ to fulfill it.

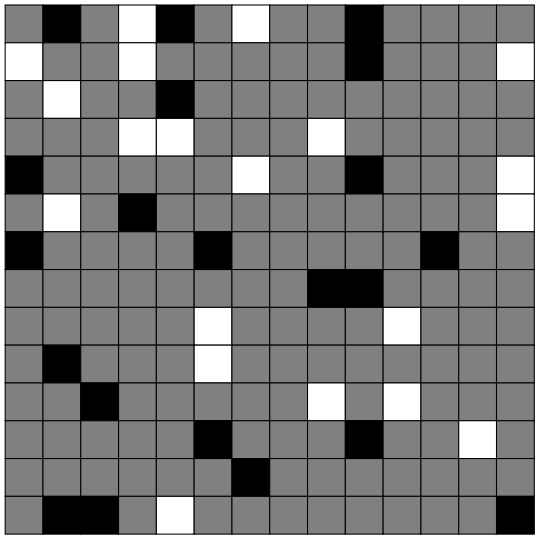
Practical Exercise 1:

The goal of the puzzle "Unruly" (by Adolfo Zanellati) is to paint a $n \times m$ matrix in black and white color, such that the following conditions are satisfied:
(1) No row or column may contain three consecutive squares of the same color.
(2) Each row and column must contain the same number of black and white fields.

Your task is to implement a program to solve these puzzles using a SAT solver. You can use any general-purpose programming language for this task. You may hand in your solution in groups of one to three people. Everyone in the group is expected to be able to explain all the code! Larger groups are expected to do more bonus exercises.

In below puzzles, the gray fields are to be determined.





We will use the following data format to store these puzzles: first a line containing the number of columns and rows as a decimal number, then a line for each row, consisting of the letters `?` for an unknown field, `B` for a black field, and `W` for a white field. For example, the first puzzle looks like this:

```
8 8
????BB??
?WW?????
?????B??
W?B?????
???W????
?W?W?B?W
B?B??B??
```

This exercise consists of four parts. The first part is neccessary to pass the problem sheet. The other parts will get you a better grade.

1. Compulsory part: Implement a program that can read the above file format (sample files are included), convert the problem into an SAT instance, run a SAT solver to solve it, and display the result *graphically*.

   Hint: Think of a good mapping of the cells to variables first. DIMACS CNF variables start at 1 and should be used contiguously, else you'll find too many solutions!

   Hint: Use `picosat --all` to list all possible solutions. Above puzzles have exactly one solution.

   Hint: Use `picosat -f` and a header of `p cnf 1 1` if you don't want to count the variables and clauses.

2. Bonus part: Add a feature to your program to enter new puzzles manually using a *graphical user interface.*

3. Bonus part: You are given inputs of various sizes, how well does your encoding work? Can you devise a more efficient encoding of the puzzle that can tackle the large sizes? (**Hint:** remember what you learned in computer architecture.)

4. Extra Bonus part: Add a feature to generate new puzzles automatically. Ensure they have exactly one solution! (**Hint:** start from an empty puzzle and randomly fill it while there is more than one solution.)

You can play Unruly online at `http://www.chiark.greenend.org.uk/~sgtatham/puzzles/js/unruly.html`.

Hand in as groups of one to three people on UniWorX until Thursday, November 15, 2018, 2pm.