COL876: Project 1

Due on 14th October, 2024

# 1 Objective

The goal of this project is to design a SAT solver to check satisfiability of CNF formulas.

Input Formulas should be represented in DIMACS format, see: https:
//logic.pdmi.ras.ru/~basolver/dimacs.html

**Output** Output "UNSAT" if the formula is unsatisfiable, otherwise output a satisfying assignment.

## 2 CDCL Procedure

The modern SAT solver relies on the CDCL (Conflict-Driven Clause Learning) procedure, which we will cover in detail during class. A thorough description of the algorithm is provided as Algorithm 1 on page 6 at https://satassociation.org/articles/FAIA185-0131.pdf.

It is important to note that this algorithm description leaves several design and implementation choices open. In particular, there are two key components where you will need to devise your own heuristics:

- (1) **PickBranchingVariable**: You should select a heuristic for choosing the branching variable and assigning truth values. To evaluate the effectiveness of your heuristic, compare it to the **random-choice heuristic**, where all choices are resolved randomly with a uniform distribution. Additionally, compare your heuristic to the **2-clause heuristic**, which selects propositions with the highest occurrences in 2-clauses (clauses with two literals) and resolves ties randomly. Your goal is to develop a heuristic that outperforms the 2-clause heuristic.
- (2) **ConflictAnalysis**: In class, we will cover a basic version of conflict analysis, where a new clause is created using the decision variable and literals assigned at decision levels lower than the current level. I expect you to implement a more advanced heuristic.

## Programming Language

You are free to choose your favorite programming language.

#### Advanced Resource

Once you have a basic SAT solver working, you can read description of how to design an efficient SAT solver at http://minisat.se/downloads/MiniSat.pdf. Note that the description is fairly advanced, therefore, I do not suggest you to be distracted by this material until have a basic functioning SAT solver.

## 3 Testing

You will evaluate the performance of your solver using random formulas generated according to the fixed-clause-length model. In this model, there are three parameters: the number N of variables, the number K of distinct literals per clause, and the number L of clauses. For simplicity, this project will focus on the 3-SAT problem, setting K to 3. Aim to handle values of N that are at least 150. Set L as  $\lceil rn \rceil$ , where r is a nonnegative real number. For a fixed N (ideally N = 150), use r values in the range (0,6) with intervals of 0.2 to generate the formulas.

To generate a random 3-SAT instance for given N and L, create L clauses of length three by randomly selecting three distinct propositions from the N available propositions and negating each with a probability of 0.5. You should pre-generate a set of formulas and then evaluate the performance of different implementations on this set. You can use CryptoMinisAT[https://github.com/msoos/cryptominisat] to evaluate satisfiability or unsastisfability (to see the ground truth) of generated formulas.

## 4 Final Submission

You will measure the performance of your solver by recording both the total compute time and the total number of invocations of the PickBranchingVariable subroutine needed to determine whether the input formula is satisfiable.

Your final submission should include:

- 1. The source code and your CNF formulas. Remember, when you are working as a software engineer, your code will never make to production if your peers can not review it.
- 2. In addition, a report of at most 10 pages (excluding references). It must include:

- (a) A verbal summary and an analysis of your findings. Your analysis can be intuitive, but you need to come up with an explanation of your findings.
- (b) A plot of time taken (total computing time) by your solver vs different values of r.

There will be zero tolerance for dishonest means (including copying or sharing solutions/code in projects.). Offenders will secure an F grade for themselves straight away.