

Practice/Real-Life Applications of Computational Algorithms, Spring 2021

Term Project: Yet Another SAT Solver (YaSat)

Due: 2021/04/19

1. Goal

In this project, you will need to implement your own SAT solver.

- (1) You can use the parser (parser.cpp and parser.h) we give, or write your own parser to read input which is written in CNF.

```
c this is a comment
p cnf 3 4
1 2 0
-2 3 0
1 2 -3 0
-1 3 0
```

There are 3 variables and 4 clauses
 $(a + b)$ // 0: end of a clause
 $(\bar{b} + c)$
 $(a + b + \bar{c})$
 $(\bar{a} + c)$

- (2) Then, write sat.cpp to find whether the input is satisfiable or not, and output the result in .sat file whose filename is the same as its input (.cnf).
- (3) If SAT, print “s SATISFIABLE” and a set of satisfying variable assignments.
- (4) Otherwise, print “s UNSATISFIABLE” in .sat file.
- (5) Notice: please use the Makefile we give to compile (You can modify it if additional source codes are necessary for compilation.)

2. Input / Output

Sample input 1	Sample output 1
p cnf 2 2 1 2 0 -1 -2 0	s SATISFIABLE v 1 -2 0
Sample input 2	Sample output 2
p cnf 3 4 1 -2 0 1 3 0 2 -3 0 -1 0	s UNSATISFIABLE

3. Command line

`./yasat [input.cnf]`

4. Hand in your project

Please upload the following files in a zip, specifying your ID (e.g., Student_ID.zip) to E3 by the deadline.

(1) Source codes and Makefile

(2) A **PDF** report that introduces your implementation

5. Platform

Linux

6. Q&A

For any questions regarding this term project, please contact Ning-Chi Huang (blackitty321@gmail.com) and Yu-Shin Han (yushinhan@live.com).