Tseitin transformation

download patch.txt
download aim-100-1_6-no-1.cnf

Clone MiniSat repository
git clone https://github.com/niklasso/minisat
cd minisat
// copy patch.txt to the current directory
patch < patch.txt
make
// copy aim-100-1_6-no-1.cnf to the current directory
./build/release/bin/minisat aim-100-1_6-no-1.cnf

If you see an output somewhat similar to the following, then you have successfully installed MiniSat.

Read about DIMACS CNF format from:
https://jix.github.io/varisat/manual/0.2.0/formats/dimacs.html

Your goal is to generate an equisatisfiable CNF formula in the DIMACS CNF format from an input formula in propositional logic. The input formula may use parentheses. You need to interpret the formula using the following precedence and associativity rules, as discussed in class.

Precedence from high to low: (),¬, $\wedge$, $\vee$,→, ↔
Boolean connectives →, ↔ are right-associative.
Boolean connectives $\wedge$, $\vee$ are left-associative.

A sample input formula is: a->b | (c <-> d & !a) | a<->b | c <-> d & !a
Here, -> is the → operator.
| is the $\vee$ operator.
& is the $\wedge$ operator.
<-> is the ↔ operator.
! is the ¬ operator.

The output file must be in DIMACS CNF format. MiniSat solver should be able to correctly parse your file and check the satisfiability of your CNF formula.

You can use any programming language of your choice. Submit your implementation and instructions on how to compile and run your implementation. You can do it in a group of up to three students. All students in a group need to understand the implementation thoroughly. The marks will be given based on the average understanding of the group.