

SAT SOLVER REPORT

Dimacs file parser

In order to test the solver using the instances given, we developed a parser that would extract the literals and the clauses. The parser returns a structure composed of a number of literals and a vector of clauses containing its literals. This parser takes a file path as input and returns a C++ struct.

At first, we open the file and get each line thanks to a loop. Then we use a function that will split the elements of a string in order to have a vector of strings. Therefore, we loop on this vector of string and then push the literals in a vector of int. Once done with this, we push the vector on the vector of clauses. Our structure is now ready to be used in the solver.

Solver implementation

The method we chose to implement is the Davis–Putnam–Logemann–Loveland (DPLL). A few improvements have been made, though. As realized after some testing, the algorithm converges to a solution much faster if we optimize how we chose the literals. First, all unit clauses are eliminated, one by one, as its literal must be true. These decisions are propagated using the same method Simplify. The clauses are evaluated to check if the propagation caused any conflicts. If no conflict is found, then the algorithm focuses on pure literal clauses elimination. As they use only one side of the equation (they are either all negations or affirmations), they tend to be weight less on the decisions. From this point on, all literals to be evaluated and propagated are randomly chosen.

The program halts if a satisfiable solution were found or if all possibilities have been tested. If a satisfiable solution is found, then the result is displayed, containing the required format, including all literals.

Build.sh

Finally, to complete our work, we needed an executable. This led us to create a new file build.sh which contains a script that can generate an executable named **sat_solver.exec**. The build also gives the user permission to execute the file. Only one parameter is expected, the path to the Dimacs file.

References:

[HTTPS://MOODLE.PARISDESCARTES.FR/PLUGINFILE.PHP/336818/MOD_RESOURCE/CONTENT/2/SAT_SOLVER.PDF](https://moodle.parisdescartes.fr/pluginfile.php/336818/mod_resource/content/2/sat_solver.pdf)
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/DPLL_ALGORITHM](https://en.wikipedia.org/wiki/DPLL_algorithm)

Team:

RODRIGUES PEREIRA Lucas 81210878

BOUZAINÉ Taha 21811356

LAHRECH Anass 21810616