

CNF Files

CNF is a data directory which contains examples of files stored using the DIMACS CNF file format. This format is used to define a Boolean expression, written in conjunctive normal form, that may be used as an example of the satisfiability problem.

The satisfiability problem considers the case in which N boolean variables are used to form a Boolean expression involving negation (**NOT**), conjunction (**AND**) and disjunction (**OR**). The problem is to determine whether there is any assignment of values to the Boolean variables which makes the formula true. It's something like trying to flip a bunch of switches to find the setting that makes a light bulb turn on.

For simplicity, it is common to require that the boolean expression be written in conjunction normal form or "CNF". A formula in conjunctive normal form consists:

1. *clauses* joined by **AND**;
2. each clause, in turn, consists of *literals* joined by **OR**;
3. each literal is either the name of a variable (*a positive literal*), or the name of a variable preceded by **NOT** (*a negative literal*).

An example of a boolean expression in 3 variables and 2 clauses:

$$\begin{aligned} & (x(1) \text{ OR } (\text{ NOT } x(3))) \\ & \text{AND} \\ & (x(2) \text{ OR } x(3) \text{ OR } (\text{ NOT } x(1))) . \end{aligned}$$

The CNF file format is an ASCII file format.

1. The file may begin with comment lines. The first character of each comment line must be a lower case letter "c". Comment lines typically occur in one section at the beginning of the file, but are allowed to appear throughout the file.
2. The comment lines are followed by the "problem" line. This begins with a lower case "p" followed by a space, followed by the problem type, which for CNF files is "cnf", followed by the number of variables followed by the number of clauses.
3. The remainder of the file contains lines defining the clauses, one by one.
4. A clause is defined by listing the index of each positive literal, and the negative index of each negative literal. Indices are 1-based, and for obvious reasons the index 0 is not allowed.
5. The definition of a clause may extend beyond a single line of text.
6. The definition of a clause is terminated by a final value of "0".
7. The file terminates after the last clause is defined.

Some odd facts include:

- The definition of the next clause normally begins on a new line, but may follow, on the same line, the "0" that marks the end of the previous clause.
- In some examples of CNF files, the definition of the last clause is not terminated by a final '0';
In some examples of CNF files, the rule that the variables are numbered from 1 to N is not followed. The file might declare that there are 10 variables, for instance, but allow them to be numbered 2 through 11.

Some of the examples in this directory come from a server at Rutgers University

<ftp://dimacs.rutgers.edu/pub/challenge/satisfiability>

Example:

Here is the CNF file that corresponds to the simple formula discussed above:

```
c  simple_v3_c2.cnf
c
p cnf 3 2
1 -3 0
2 3 -1 0
```

Licensing:

The computer code and data files described and made available on this web page are distributed under [the GNU LGPL license.](#)

Reference:

1. Rina Dechter,
Enhancement Schemes for constraint processing:
Backjumping, learning, and cutset decomposition,
Artificial Intelligence,
Volume 41, Number 3, January 1990, pages 273-312.
2. Michael Quinn,
Parallel Programming in C with MPI and OpenMP,
McGraw-Hill, 2004,
ISBN13: 978-0071232654,
LC: QA76.73.C15.Q55.
3. [dimacs_cnf.pdf](#), a writeup of the DIMACS CNF file format.

Sample Files:

- [aim-100-1_6-no-1.cnf](#), 100 variables and 160 clauses.
- [aim-50-1_6-yes1-4.cnf](#), 50 variables and 80 clauses.
- [bf0432-007.cnf](#), 1040 variables and 3668 clauses.
- [dubois20.cnf](#), 60 variables and 160 clauses.
- [dubois21.cnf](#), 63 variables and 168 clauses.
- [dubois22.cnf](#), 66 variables and 176 clauses.

- [hole6.cnf](#), based on the pigeon hole problem, a simple example with 42 variables and 133 clauses.
- [par8-1-c.cnf](#), an example with 64 variables and 254 clauses.
- [quinn.cnf](#), an example from Quinn's text, 16 variables and 18 clauses.
- [simple_v3_c2.cnf](#), a simple example with 3 variables and 2 clauses.
- [zebra.c](#), a pseudo C file that can be run through the C preprocessor to generate the CNF file for the "Who Owns the Zebra" puzzle.
- [zebra_v155_c1135.cnf](#), a formulation of the "Who Owns the Zebra?" puzzle, with 155 variables and 1135 clauses.

You can go up one level to [the DATA page](#).

Last revised on 22 May 2008.