

MiniSat Tutorial

EDA, Spring 2018

Instructor: Hui-Ru Iris Jiang

2017-05-14

Presenter: Jun-Jie Wang

**The Electronic Design Automation Laboratory
Graduate Institute of Electronics Engineering
National Taiwan University
Taipei 106, Taiwan**



Outline

- MiniSat Introduction
- Input Format
- Usage

MiniSat Introduction

- MiniSat
 - The state-of-the-art SAT solver
- Boolean satisfiability problem
 - Whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE
- Satisfiable
 - Exists an assignment where the formula evaluates to TRUE
- Version
 - MiniSat_v1.14_linux
- Download website
 - <http://minisat.se/MiniSat.html>

Input Format

- DIMACS CNF format
 - Product of sums
- Format Rule
 - 'c' is for comment
 - In the line 'p', the first number is the variable number, and the second number is the clause number
 - Terminate clause line with zero '0'

- Example

```
c example
c (a + b + c) (a' + c')
p cnf 3 2
 1 2 3 0
-1 -3 0
```

Usage

- Usage:
 - `./MiniSat_v1.14_linux [*dimacs] [*output]`
- Example
 - The output of previous example is SATISFIABLE

```
jjwang@eda33 ~/LSV2017/hw2 $ ./MiniSat_v1.14_linux example.dimacs example.output
=====
[MINISAT]
=====
| Conflicts | ORIGINAL | LEARNT | Progress | | | | |
|           | Clauses  | Literals | Limit  | Clauses | Literals | Lit/Cl |           |
|=====|=====|=====|=====|=====|=====|=====|=====|
|           0 |           2 |           5 |           0 |           0 |           0 | nan | 0.000 % |
|=====|=====|=====|=====|=====|=====|=====|=====|
restarts      : 1
conflicts     : 0 (nan /sec)
decisions     : 3 (inf /sec)
propagations  : 3 (inf /sec)
conflict literals : 0 (nan % deleted)
Memory used   : 1.67 MB
CPU time      : 0 s

SATISFIABLE
```

Online Resources

- Minisat
 - <http://minisat.se/MiniSat.html>
 - <https://willyc20.github.io/2016/12/18/sat-problem-2/>
- Input Format
 - <http://www.satcompetition.org/2009/format-benchmarks2009.html>