LearnSAT: A SAT Solver for Education
Tool Paper

Mordechai (Moti) Ben-Ari
Department of Science Teaching
Weizmann Institute of Science
http://www.weizmann.ac.il/sci-tea/benari/

SAT for the Rest of Us

- SAT solving in the syllabus of an **undergraduate course in logic**.
- Generate **detailed traces** of the algorithms: DPLL, CDCL, NCB.
- Generate **graphical representations** of implication graphs and assignment trees.
- Written in Prolog for **clarity and conciseness** (core algorithms are about 200 lines). SWI-Prolog compiler runs on **vanilla computers**: Windows and Mac.
- Learning materials: non-artificial examples, user's guide, software documentation, **tutorial**.

Readable Notation

```
%  Pigeonhole principle
%  pij - pigeon i is in hole j

hole2 :-
  dpll([
    % Each pigeon in hole 1 or 2
  [p11, p12],   [p21, p22],   [p31, p32],

    % No pair is in hole 1
  [~p11, ~p21], [~p11, ~p31], [~p21, ~p31],

    % No pair is in hole 2
  [~p12, ~p22], [~p12, ~p32], [~p22, ~p32],
  ], _).
```

DIMACS conversion supported.

Using LearnSAT

dpll	Run the DPLL algorithm
set_mode	Set the algorithmic mode (dpll, cdcl, ncb)
set_order	Set variable assignment order
usage	Show the predicates, modes and display options
show_config	Show the current mode and display options
set_display	Set display options
clear_display	Clear display options
set_decorate_mode	Set decoration (color, bw)

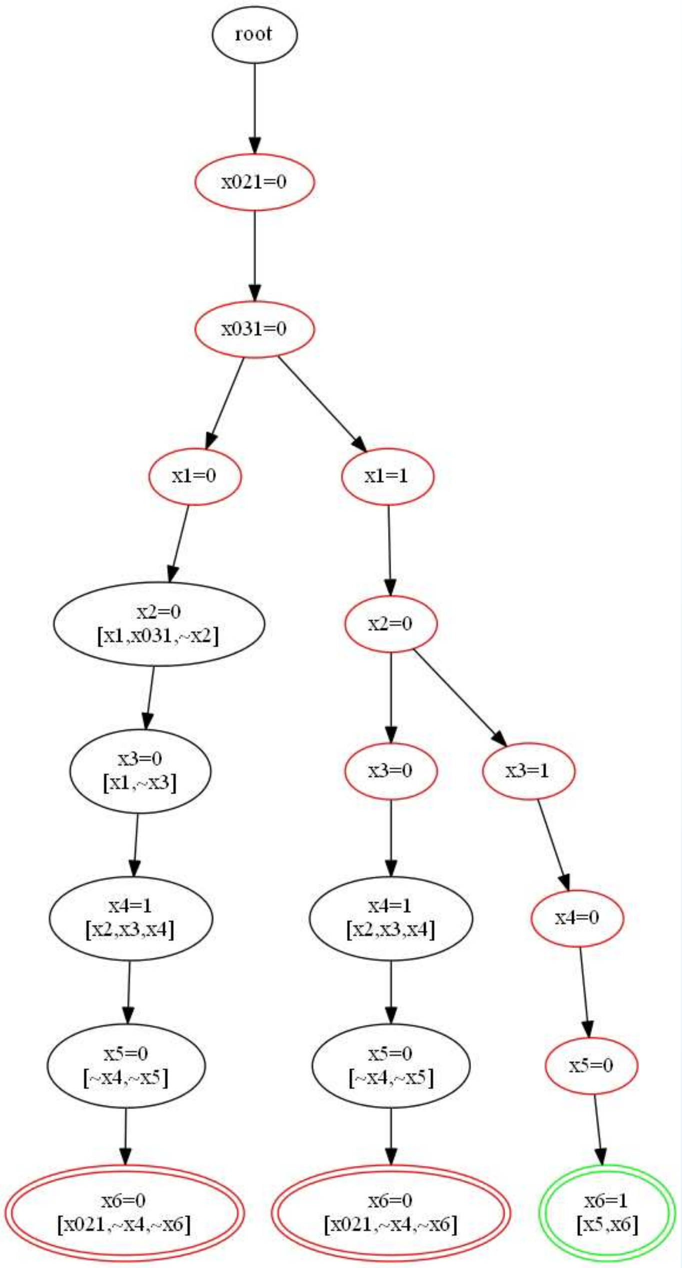
Detailed Trace

```
LearnSAT v1.4.3. Copyright 2012-13 by Moti Ben-Ari. GNU GPL.
Decision assignment: x021=0@1
Decision assignment: x031=0@2
Decision assignment: x1=0@3
Propagate unit: ~x2 (x2=0@3) derived from: 1. [x1,x031,~x2]
  ...
Conflict clause: 6. [x5,x6]
Resolvent: of [x5,x6] and antecedent [x021,~x4,~x6] is [x5,x021,~x4]
Resolvent: of [x5,x021,~x4] and antecedent [~x4,~x5] is [x021,~x4]
UIP: one literal ~x4 is assigned at level: 3
Learned clause from resolution: [x021,~x4]
Non-chronological backtracking to level: 1
Skip decision assignment: x1=1@3
Skip decision assignment: x031=1@2
Decision assignment: x021=1@1
  ...
Satisfying assignments:
[x021=1@1,x031=0@2,x1=0@3,x2=0@3,x3=0@3,x4=1@3,x5=0@3,x6=1@3]
Statistics: clauses=6, variables=8, units=10, decisions=6, conflicts=1
```

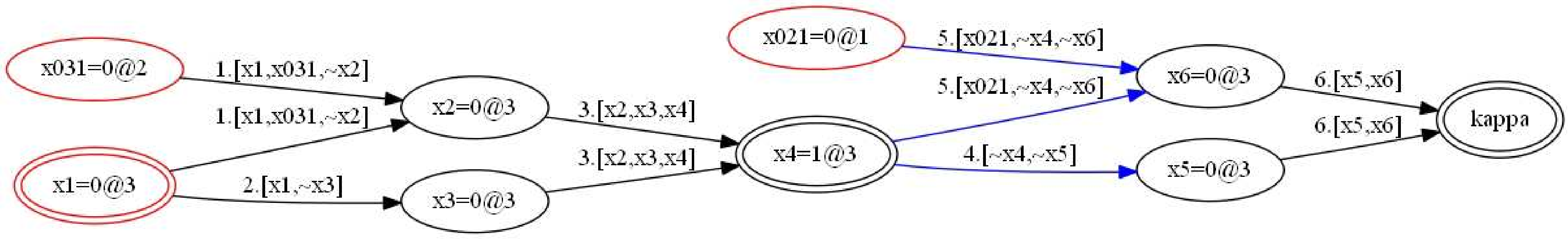
Display Options

antecedent	antecedents of the implied literals
assignment	assignments that caused a conflict
backtrack	level of non-chronological backtracking
clause	clauses to be checked for satisfiability
conflict	conflict clauses
decision	decision assignments
dominator	computation of the dominator
dot	implication graphs (final) in dot format
dot_inc	implication graphs (incremental) in dot format
evaluate	evaluation of clauses for an assignment
graph	implication graphs (final) in textual format
incremental	implication graphs (incremental) in textual format
label	graphs and trees labeled with clauses
learned	learned clause by resolution
partial	partial assignments so far
resolvent	resolvents created during CDCL
result	result of the algorithm with statistics
skipped	assignments skipped when backtracking
sorted	assignments displayed in sorted order
tree	trees of assignments (final) in dot format
tree_inc	trees of assignments (incremental) in dot format
uip	unique implication points
unit	unit clauses
variables	variables that are not assigned so far

Tree of Assignments



Implication Graph



Examples and Documentation

- Examples:
  - Published papers (MLM, MZ, MS)
  - Pigeonhole principle
  - Grid pebbling
  - Tseitin clauses for graphs
  - Four-queens
  - Bounded model checking
- Documentation
  - User's guide
  - Software documentation
  - Tutorial

Olkaa Hyvä Auttakaa Minua! (Please Help Me!)

- Bug reports (although I'm sure that there aren't any...).
- Reports of teaching experience.
- Suggestions for additional features, especially concerning graphics.
- Additional examples, preferably from applications.

Kiitos paljon (many thanks)