Visualization for Explanations in Decision Support

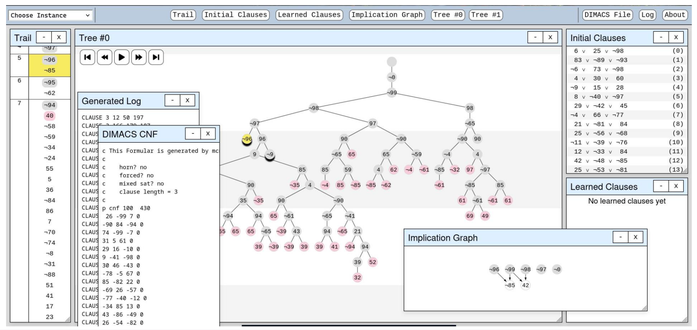
Carston Wiebe, Raegan Scheet, and Berthe Y. Choueiry
Constraint Systems Laboratory • School of Computing

Summary

Boolean satisfiability (SAT) is at the foundation of Computer Science. SAT solvers are widely used in the industry today. We built an online system to visualize and animate the key mechanisms at the heart of the MiniSAT solver. Our system is useful for teaching and outreach.

Task: Instrument MiniSAT to display its operations in a human-digestible format.

Contributions: Built an interactive online tool to visualize and animate the operations of MiniSAT.



Methodology

We studied and instrumented the MiniSAT code by placing 'hooks' to log the operations of its most important mechanisms.

When our modified MiniSAT is executed on a problem (SAT theory), it generates a detailed log of MiniSAT's operations.

```
PROP_START 14
ASSIGN 11 5
FROM 2 11 15
ASSIGN 5 5
FROM 2 5 15
CONFLICT 2 21 15
BACKTRACK 5 4
UNASSIGN 5
```

Our system parses this log to build two primary data structures, namely:

1. A linear decision-trail
2. A binary search tree

Our visualizations use the information stored in these structures to illustrate the operations of MiniSAT and allow users to interact with them.

Main Visualizations

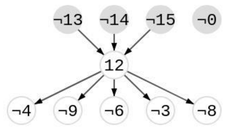
Initial Theory and Learned Clauses

6 ∨ 25 ∨ ¬98	(0)
83 ∨ ¬89 ∨ ¬93	(1)
¬6 ∨ 73 ∨ ¬98	(2)
4 ∨ 30 ∨ 60	(3)
¬9 ∨ 15 ∨ 28	(4)
8 ∨ ¬40 ∨ ¬97	(5)
29 ∨ ¬42 ∨ 45	(6)
¬4 ∨ 66 ∨ ¬77	(7)
21 ∨ ¬81 ∨ 84	(8)
25 ∨ ¬56 ∨ ¬68	(9)
¬11 ∨ ¬39 ∨ ¬76	(10)
12 ∨ ¬33 ∨ 84	(11)
42 ∨ ¬48 ∨ ¬85	(12)
25 ∨ ¬53 ∨ ¬81	(13)

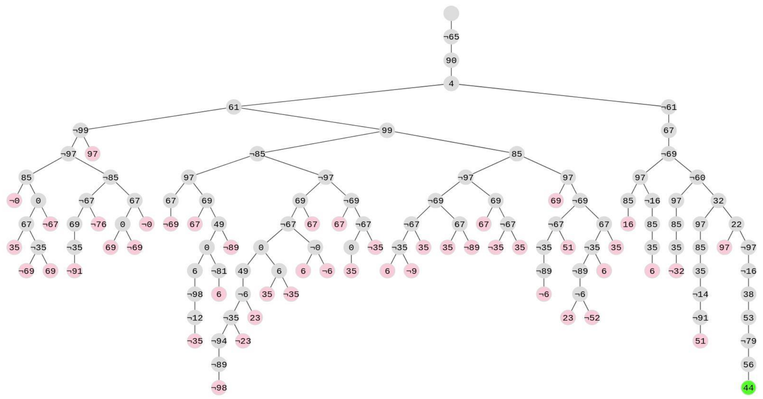
Decision Trail

Lvl	Assign
1	¬0
2	¬99
3	¬98
	42
4	¬97
5	¬96
	¬85
6	¬95
	¬62

Implication Graph



Search Tree



MiniSAT Overview

- Boolean Satisfiability (SAT) models many decision problems as a formulae in Propositional Logic and is at the foundation of Theoretical Computer Science.
- Today, SAT solvers are popular in the industry, such as air-traffic control, circuit design (EDA), software and hardware verification, and scheduling and resource allocation.
- MiniSAT, created by Niklas Eén and Niklas Sörensson in 2003, is a highly efficient and lightweight SAT Solver.

Motivating Example

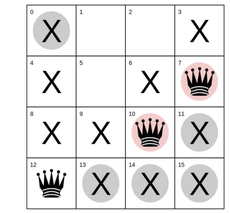
The 4 Queens puzzle aims to place 4 Queens on a 4x4 board so that no queen attacks another.

$c_0 \vee c_1 \vee c_2 \vee c_3$
$\neg c_0 \vee \neg c_1$
$\neg c_0 \vee \neg c_2$
⋮
$c_0 \vee c_4 \vee c_8 \vee c_{12}$
$\neg c_0 \vee \neg c_4$
$\neg c_0 \vee \neg c_8$
⋮

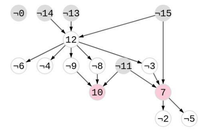
Sample of the CNF theory

The SAT theory models the board positions (0—15) as Boolean variables where true indicates that a queen is present in the position.

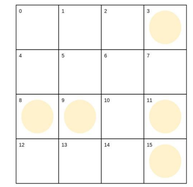
SATVIZ displays how MiniSAT solves the puzzle and allows the user to interact with the graphical representations.



Board for the first conflict



Implication graph for the first conflict

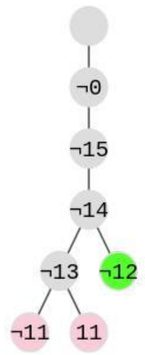


11 ∨ 8 ∨ 9 ∨ 3 ∨ 15 (0)

Learned clause for the first conflict

Lvl	Assign
1	¬0
2	¬15
3	¬14
4	¬13
	12
	¬4
	¬8
	¬3
	¬6
	¬9
5	¬11
	7
	10
	¬5
	¬2
5	11
	¬10
	¬7
	¬1
	2
	5
4	¬12
	13
	¬1
⋮	

A sample of the trail



The search tree

Future Work

- Create more graphs to display finer details of the algorithm such as propagations and satisfied clauses per decision made.
- Create an animate instance-specific visualization tools, such as a Chess or Sudoku board.

Acknowledgment: Supported by UCARE

April 9, 2025