SAT 2022 Test of Time Award

Niklas Eén, Niklas Sörensson
An Extensible SAT-solver
published at SAT’03

25th International Conference on
Theory and Applications of Satisfiability Testing (SAT’22)
Friday, August 5, 2022
Federated Logic Conference (FLOC’22)
Haifa, Israel

presenter Armin Biere
An Extensible SAT-solver

Niklas Eén & Niklas Sörensson

Conference paper

3091 Accesses | 861 Citations

Part of the Lecture Notes in Computer Science book series (LNCS, volume 2919)

Abstract

In this article, we present a small, complete, and efficient SAT-solver in the style of conflict-driven learning, as exemplified by CHAFF. We aim to give sufficient details about implementation to enable the reader to construct his or her own solver in a very short time. This will allow users of SAT-solvers to make domain specific extensions or adaptations of current state-of-the-art SAT-techniques, to meet the needs of a particular application area. The presented solver is designed with this in mind, and includes among other things a mechanism for adding arbitrary boolean constraints. It also supports solving a series of related SAT-problems efficiently by an incremental SAT-interface.
An extensible SAT-solver

Authors  Niklas Eén, Niklas Sörensson
Publication date  2003/5/5
Conference  International conference on theory and applications of satisfiability testing
Pages  502-518
Publisher  Springer, Berlin, Heidelberg

Description  In this article, we present a small, complete, and efficient SAT-solver in the style of conflict-driven learning, as exemplified by CHAFF. We aim to give sufficient details about implementation to enable the reader to construct his or her own solver in a very short time. This will allow users of SAT-solvers to make domain specific extensions or adaptations of current state-of-the-art SAT-techniques, to meet the needs of a particular application area. The presented solver is designed with this in mind, and includes among other things a mechanism for adding arbitrary boolean constraints. It also supports solving a series of related SAT-problems efficiently by an incremental SAT-interface.

Total citations  Cited by 4006

Scholar articles  An extensible SAT-solver
N Eén, N Sörensson - International conference on theory and applications of ..., 2003
Cited by 4006  Related articles  All 20 versions
Hi Armin,

It was great to catch up with you today. This was the short summary I wrote to Olaf:

"The paper pertaining to this award, started off as an invited paper for SAT 2003. My solver 'satnik', slightly improved and submitted under the name 'satzoo', had won one of the categories of the SAT-competition. I decided I wanted to use the opportunity to write a tutorial on implementing SAT solvers, and drafted the first version of the paper, later soliciting help from Niklas to co-author the final version. When we submitted the paper, 25 pages in total, the organizers informed us for the first time that the page limit was 6 pages. We pointed out that the error was on their part, not ours. To resolve the situation, the program committee decided to include the paper with the regular submissions (page limit 20, which we edited the paper down to), despite being weeks after the deadline. It was reviewed as a regular paper, as far as we know by reviewers unaware of its origin story. Their score came back "Reject". The sentiment was that the paper contained nothing novel. One reviewer stated, in his first sentence, that he disliked the paper, "and not only for the pointless exercises" (quote from memory). Despite this, the paper was included in the proceedings, for reasons never relayed to us. But it seems the tutorial found its audience and was eventually appreciated."

// N.
How the authors entered SAT

- connection to CAV’97 in Haifa (25 years ago)
  - Stålmarck’s company had an invited talk at CAV’97 in Haifa!
  - Ed Clarke saw potential of SAT in model checking
  - BMC at TACAS’99

- two master theses around 2001 / 2002 on SAT-based model checking
  - Niklas Eén as intern at Prover Technology
  - Niklas Sörensson at Chalmers but also using prover

- Limmat SAT solver
  - winner in one of the tracks of the SAT competition 2002
  - attempt to understand CHAFF by implementing the paper (DAC’01)
  - open source and worked well

- both thought about to implement a SAT solver too but first hesitated
  - Limmat even though simpler than Chaff still hard to ”grasp”
  - I met with Niklas Eén at FLOC’02 (Copenhagen) to ”go over the code”
### SAT 2005 Competition

**Organizing committee**
Daniel Le Berre and Laurent Simon

**Judges**
Armin Biere, Oliver Kullmann and Alwen Van Gelder

**Reference**

**Benchmarks**
- Random (.tar.bz2, 25MB), Crafted (.tar.bz2, 360MB), Industrial (.tar.bz2, 205MB)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT+UNSAT</td>
<td>SatEliteGTi</td>
<td>MiniSAT.1.13</td>
<td>ZChaff, rand and HiffesAT</td>
<td>Valis</td>
<td>SatEliteGTi</td>
<td>March dl</td>
<td>kent-2004</td>
<td>March dl</td>
<td>Day-SatB1</td>
</tr>
<tr>
<td>SAT</td>
<td>SatEliteGTi</td>
<td>MiniSAT.1.13</td>
<td>Jerusat-3.1.9 and HiffesAT</td>
<td>Valis</td>
<td>SatEliteGTi</td>
<td>March dl</td>
<td>g2wsat</td>
<td>VV</td>
<td></td>
</tr>
<tr>
<td>UNSAT</td>
<td>SatEliteGTi</td>
<td>ZChaff, rand</td>
<td>MiniSAT</td>
<td>SatEliteGTi</td>
<td>MiniSAT.1.13</td>
<td>Valis and March dl</td>
<td>kent-2004</td>
<td>March dl</td>
<td>Day-SatB1</td>
</tr>
</tbody>
</table>

**Special tracks**
- zChaff
- TTSP-3.0

**NON CLAUSAL**
No solver submitted

**PSEUDO BOOLEAN**
Go to [official website](http://www.officialwebsite.com)

### SAT 2004 Competition

**Organizing committee**
Daniel Le Berre and Laurent Simon

**Judges**
Fahiem Bacchus, Hans Kleine Büning and João Marques Silva

**Reference**

**Benchmarks**
- Random (.tar.bz2, 11MB), Crafted (.tar.bz2, 36MB), Industrial (.tar.bz2, 2GB)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Industrial</th>
<th>handmade</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL (SAT+UNSAT)</td>
<td>ZChaff 2004</td>
<td>March eq</td>
<td>AdaptNovelty</td>
</tr>
<tr>
<td>SAT</td>
<td>Jerusat</td>
<td>Satzoo</td>
<td>AdaptNovelty</td>
</tr>
<tr>
<td>UNSAT</td>
<td>ZChaff 2004</td>
<td>March eq</td>
<td>Kcnfs</td>
</tr>
</tbody>
</table>

### SAT 2003 Competition

**Organizing committee**
Daniel Le Berre and Laurent Simon

**Judges**
John Franco, Hans van Maaren and Toby Walsh

**Reference**

**Benchmarks**
available from SATLIB (.tar.bz2, 345MB)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Industrial</th>
<th>handmade</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete on all</td>
<td>Forklift</td>
<td>Satzoo</td>
<td>Kcnfs</td>
</tr>
<tr>
<td>ALL on SAT</td>
<td>Forklift</td>
<td>Satzoo</td>
<td>Unitwalk</td>
</tr>
</tbody>
</table>

### SAT 2002 Competition

**Organizing committee**
Edward A. Hirsch, Daniel Le Berre and Laurent Simon

**Judges**
N/A

**Reference**

**Benchmarks**
available from SATLIB (.tgz, 147MB)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Industrial</th>
<th>handmade</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete on all</td>
<td>zChaff</td>
<td>zChaff</td>
<td>OKSolver</td>
</tr>
<tr>
<td>ALL on SAT</td>
<td>Limmat</td>
<td>Berkmin</td>
<td>OKSolver</td>
</tr>
</tbody>
</table>
2003 Winners now known!!!

The online report will be updated with all the details of the second stage, plus an analysis of the competition. For now, one can take a look at our slides (PPT, PDF) from the SAT2003 conference and the detailed results of the first stage.

Industrial category

We have a local copy of the industrial benchmarks: sc2003benches.zip.

Forklift, from Eugene Goldberg and Yakov Novikov (complete deterministic) won both the "best complete solver on industrial benchmarks" and "best solver on satisfiable industrial benchmarks" awards. Forklift is a new solver that can be viewed as an extension of Berkmin62 with limited resolution at each leaf.

Handmade category

satzoo from Niklas Een (complete deterministic) won both the "best complete solver on handmade benchmarks" and "best solver on satisfiable handmade benchmarks" awards. Satzoo is a Chaff-like SAT-solver based on watched literals and clause recording. It was developed mainly to be able to experiment with new model checking techniques which requires a tighter integration with the SAT-solver. As such, Satzoo takes a more general view on what a "SAT" problem could be, and supports solving a number of related SAT-problems by an incremental SAT interface. Satzoo is also meant to encompass more types of constraints than just clauses. To prototype this, linear constraints over boolean variables were added to the solver. The front end thus supports a subset of the CPLEX lp-format along with DIMACS cnf-format. For lp-files optimization is performed towards the goal function rather than just finding a satisfying assignment.

Random category

kcnf from Gilles Dequen and Olivier Dubois won the "best complete solver on random benchmarks" award.

unitwalk from Edward Hirsch and Arist Kojevnikov (Incomplete randomized) won the "best solver on satisfiable random benchmarks" award. UnitWalk is a combination of unit clause elimination (particularly, the idea of Paturi, Pudlak and Zane's randomized unit clause elimination algorithm) and local search. The solver participated in the competition extends this basic algorithm with adding some of 2-resolvents using incBinSat, and mixes its random walks with WalkSAT-like walks.

benchmarks awards

The benchmarks from the SAT2002 competition not solved during the first stage were used in this year competition as "SAT2002 competition challenging benchmarks". Those benchmarks were submitted last year, so they cannot be awarded this year.

Smallest unsatisfiable benchmark:
random/hrich/hgen8/hgen8-n260-01-S1597732451 (260 variables, 391 clauses, 888 literals)

Smallest satisfiable benchmark:
random/simon/sat02-random/hgen2-v400-s161064952 (400 variables, 1400 clauses, 4200 literals). There is no award for that benchmark since it is one of the SAT2002 competition challenging benchmarks. Note that last year, unitwalk solved it during the second stage in 20199s on a PIII 450 (this year the timeout was 7200s on an Athlon 1800+).

Last update: 19 may 2003
All Time Winners on SAT Competition 2021 Benchmarks

- kissat–mab–2021
- kissat–2020
- maple–lcm–dist–2017
- maple–lcm–dist–cb–2018
- abcdsat–2015
- lingeling–2014
- glucose–2011
- glucose–2012
- lingeling–2013
- minisat–2008
- precosat–2009
- cryptominisat–2010
- minisat–2006
- rsat–2007
- satelite–gti–2005
- berkmin–2003
- zchaff–2004
- limmat–2002
- grasp–1997
**INTRODUCTION**

MiniSat is a minimalistic, open-source SAT solver, developed to help researchers and developers alike to get started on SAT. It is released under the MIT licence, and is currently used in a number of projects (see "Links"). On this page you will find binaries, sources, documentation and projects related to MiniSat, including the Pseudo-boolean solver MiniSat+ and the CNF minimizer/preprocessor SatELite. Together with SatELite, MiniSat was recently awarded in the three industrial categories and one of the "crafted" categories of the SAT 2005 competition (see picture).

Some key features of MiniSat:

- **Easy to modify.** MiniSat is small and well-documented, and possibly also well-designed, making it an ideal starting point for adapting SAT based techniques to domain specific problems.
- **Highly efficient.** Winning all the industrial categories of the SAT 2005 competition, MiniSat is a good starting point both for future research in SAT, and for applications using SAT.
- **Designed for integration.** MiniSat supports incremental SAT and has mechanisms for adding non-clausal constraints. By virtue of being easy to modify, it is a good choice for integrating as a backend to another tool, such as a model checker or a more generic constraint solver.

We would like to start a community to help distribute knowledge about how to use and modify MiniSat. Any questions, comments, bugreports, bugfixes etc. should be directed to miniSat@googlegroups.com. The archives can be browsed here. The source code repository for MiniSat 2.2 can be found at github.

— Niklas & Niklas
Minisat papers...

"An Extensible SAT-solver"  

"MiniSat — A SAT Solver with Conflict-Clause Minimization"  

"Effective Preprocessing in SAT through Variable and Clause Elimination"  

"Translating Pseudo-Boolean Constraints into SAT"  

"Applying Logic Synthesis for Speeding Up SAT"  

"Cut Sweeping."  

Applications of Minisat...

"Temporal Induction by Incremental SAT Solving."  

"New techniques that improve MACE-style finite model finding."  
Koen Claessen, Niklas Sörensson, CADE 2003.

"A Single-Instance Incremental SAT Formulation of Proof- and Counterexample-Based Abstraction"  
Conclusion

- High impact paper
  - blue-print for high-performance SAT Solvers
  - Niklas Eén dissertation "SAT Based Model Checking" Chalmers 2005
  - Niklas Sörensson dissertation "Effective SAT Solving" Chalmers 2008

- High impact source code
  - used in many industrial SAT applications (EDA, model checking, SMT, …)
  - base line of many award winning solvers in the SAT competition

- Contributions
  - Exponential VSIDS (contributed to Niklas Eén)
  - Lazy Priority Decision Queue (contributed to Niklas Sörensson)
  - Assumption Interface for incremental SAT solving