Uncovering and Classifying Bugs in MaxSAT Solvers through Fuzzing and Delta Debugging

Tobias Paxian, Armin Biere

paxiant|biere@informatik.uni-freiburg.de

14th Pragmatics of SAT international workshop

Demonstration

Toolchain: WCNFuzz
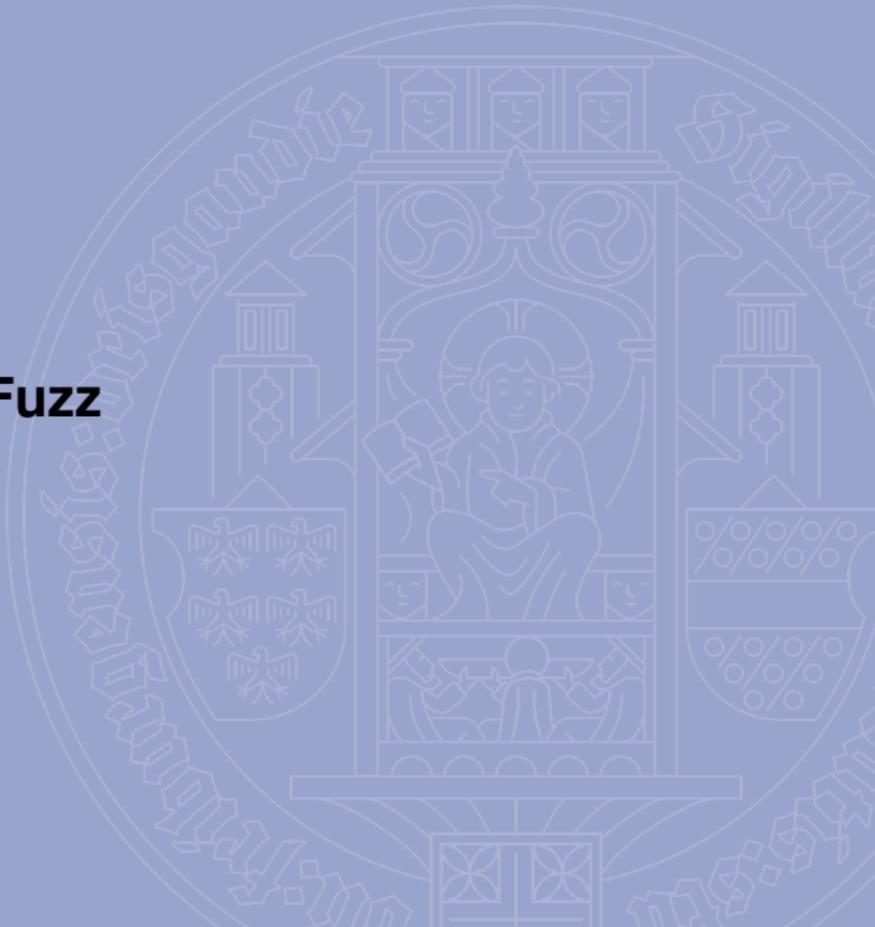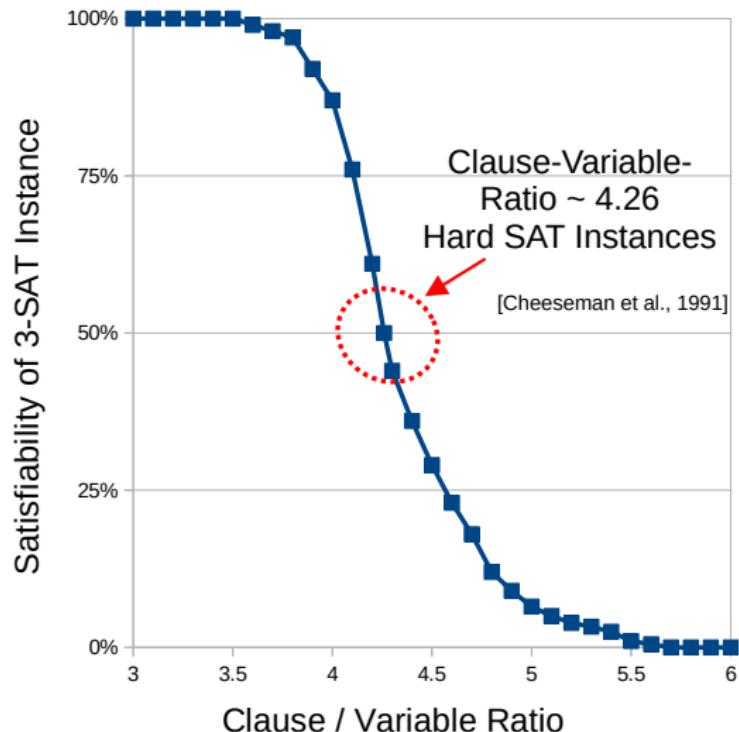
Fuzzer: Where the really hard problems are.

Toolchain: WCNFuzz
Fuzzer: Where the really hard problems are.
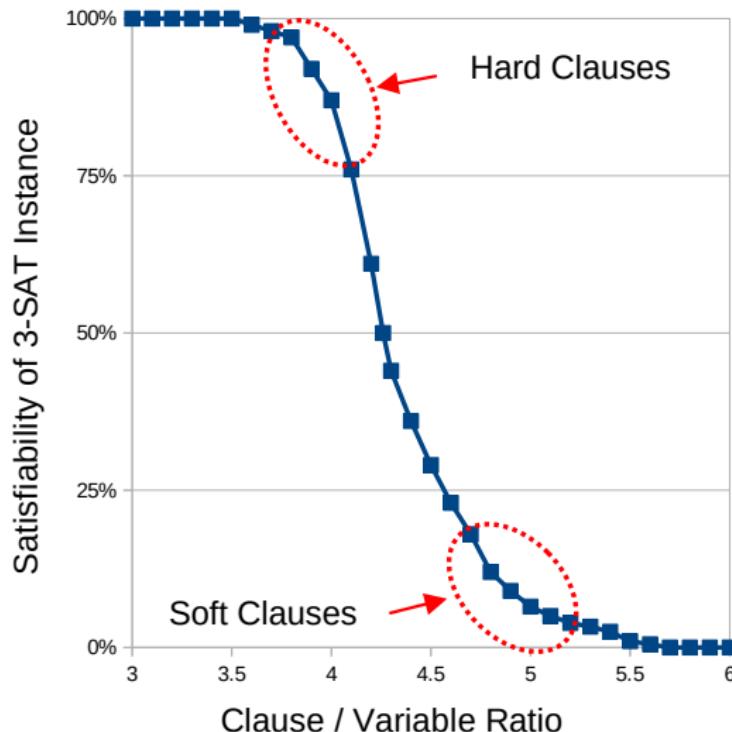
Statistics (1,000,000 instances)

- on average 516 (424 hard / 92 soft) clauses
- on average 2251 literals
- $\Rightarrow$ average clause size 4.36
- 13% are without soft clauses (pure SAT)
- 0.6% are without hard clauses (pure MaxSAT)
- 86% are satisfiable and 41.9% with o-value 0
- 5.6% have a solution $> 2^{32}$

Toolchain: WCNFcompare
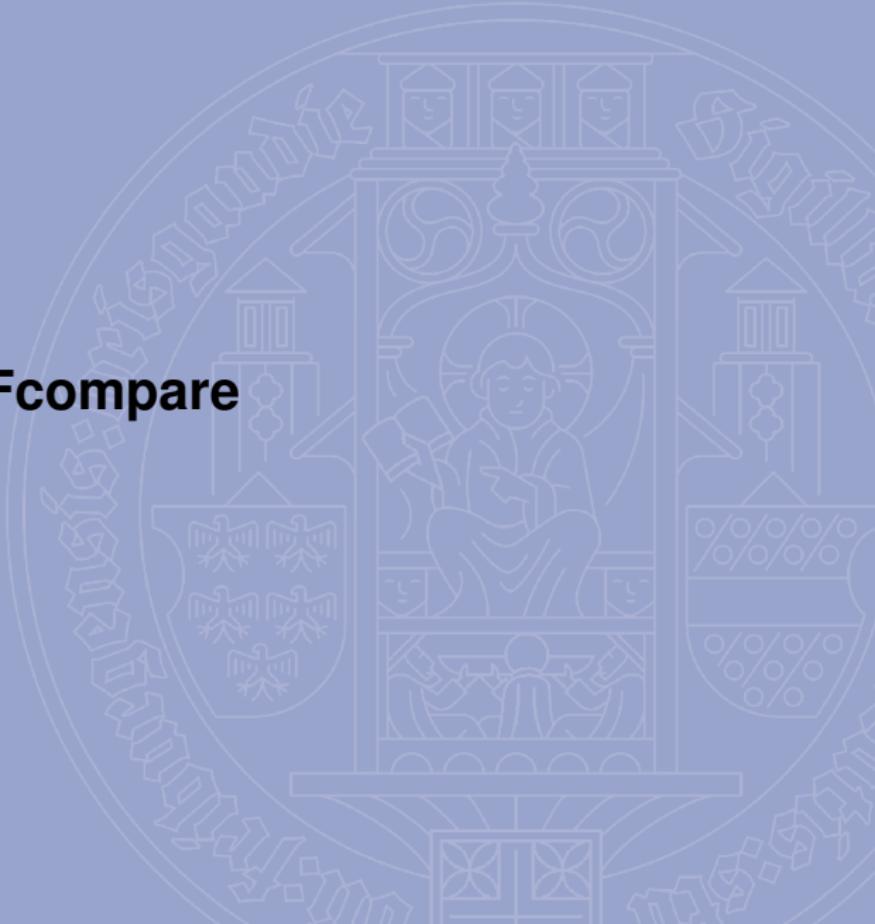
Toolchain: WCNFcompare
Comparing MaxSAT Solver

| Rank	MaxSAT Solver	
1. & 2.	CASHWMAXSAT	Lei et al. [2022], Wang et al. [2022]
3. & 7.	UWRMAXSAT	Piotrów [2022]
4.	MAXHS	Bacchus [2022]
5. & 6.	WMAXCDCL	Coll et al. [2022a,b]
8.	EVALMAXSAT	Avellaneda et al. [2022]
9.	CGSS	Ihalainen et al. [2022b]
10.	EXACT	Devriendt [2022]
6.	PACOSE	Paxian et al. [2018]
	Z3RC2	De Moura and Bjørner [2008], Bjørner et al. [2015]
	Z3MAXRES	De Moura and Bjørner [2008], Bjørner et al. [2015]
	Z3WMAX	De Moura and Bjørner [2008], Bjørner et al. [2015]
	MAXPRE 2.0	Ihalainen et al. [2022a], Korhonen et al. [2017]

Toolchain: WCNFcompare
Fault Classification

1. Crashes (SIGABRT, SIGSEGV, SIGKILL, ...)
2. Bound Violations:
   2.1 $o_{min} < o_{solver}$ and $o_{solver} == o_{model}$
   2.2 $o_{solver} \neq o_{model}$ and $o_{model} \neq o_{min}$ and $o_{solver} \neq o_{min}$.
   2.3 Either o_{model} or o_{solver} unequals o_{min}.
   2.4 Model is UNSAT.
   2.5 Solver states UNSAT, but instance is SAT.
3. Performance Regressions:
   3.1 Potential Fault: Timeout > 50 times of successful solvers.
4. Other Issues:
   4.1 Solver has an error either stated in stdout or stderr.
   4.2 Inconsistency in status line and output.

$$o_{min} := \text{minimal value of all solvers}$$
$$o_{solver} := \text{o value solver}$$
$$o_{model} := \text{o value of the model}$$
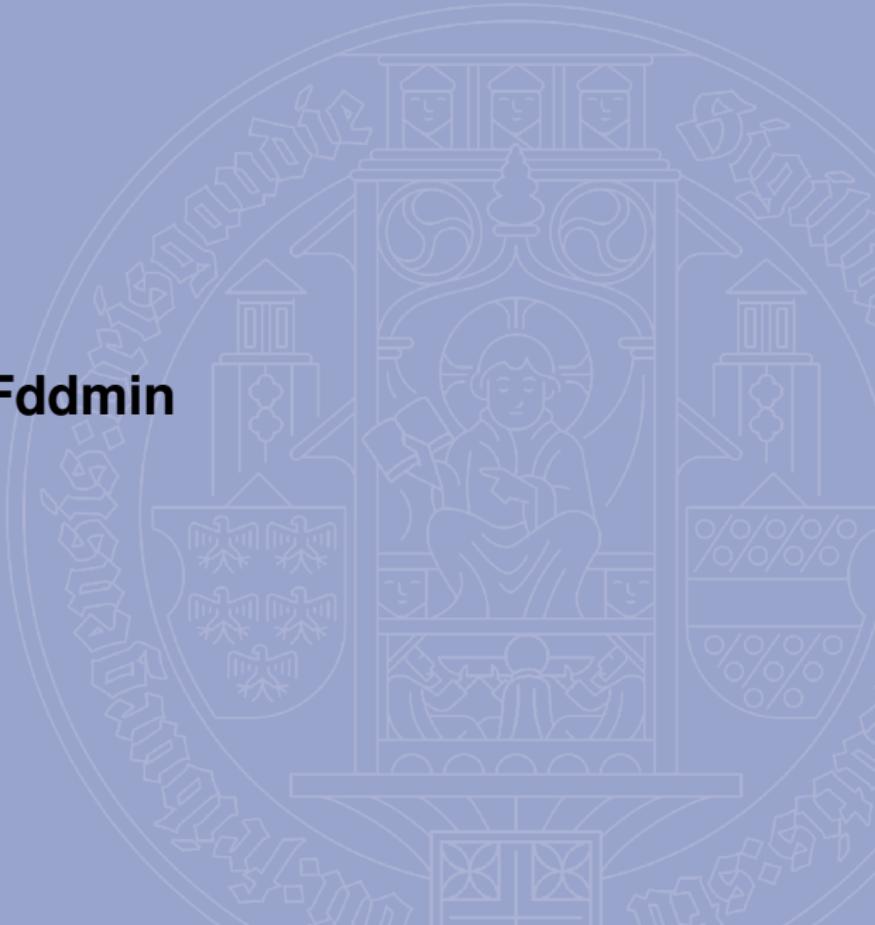
Toolchain: WCNFddmin
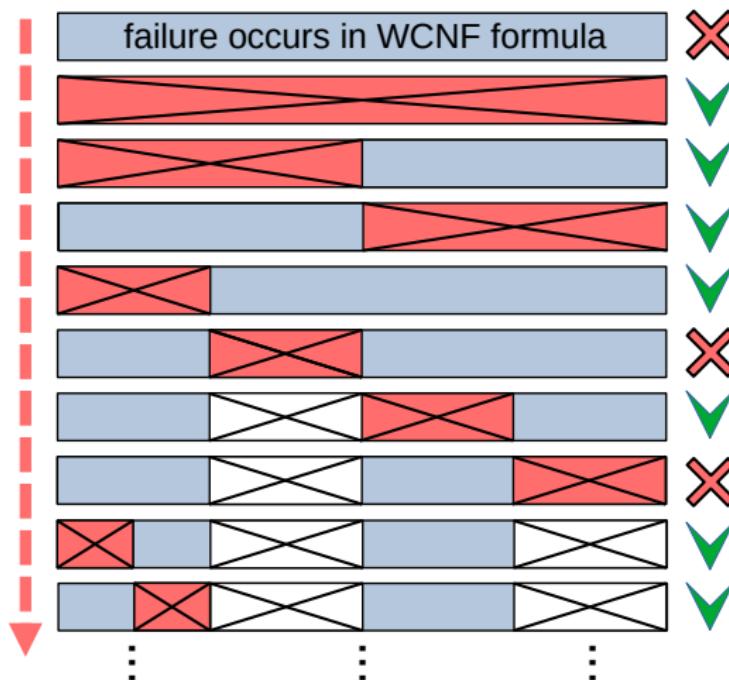
[Zeller and Hildebrandt, 2002, Zeller, 2009]

Toolchain: WCNFddmin
WCNF Delta Debugging

WCNFddmin functionality

- Removing Clauses
- Removing Variables
- Removing Literals
- *NEW* Converting soft to hard clauses
- *NEW* Weight reduction to 1
- *NEW* Binary weight reduction phase
  ——
- *NEW* Multiple Rounds
- *NEW* Shuffling Clauses
- *NEW* Shuffling Literals
- *NEW* Reorder Variables

universität freiburg

Toolchain: WCNFddmin
WCNF Delta Debugging

Statistics (555 instances)

	clauses	hard clauses	soft clauses	literals
median	97.59%	99.13%	90.40%	99.26%
average	91.67%	94.39%	74.46%	95.72%
#0%	0	0	46	0
#100%	12	74	12	12
no clauses		4	13	

Demonstration: Toolchain

Results

Fuzzing	Cr.	Bound Violations					Perf	Others		sum
$1 < \textbf{sow} < 2^{32}$	1.	2.1.	2.2.	2.3.	2.4.	2.5.	3.	4.1.	4.2.	faults
CASHWMS-CP		1e4	1e4	2e4	1					4e4
CASHWMS-Plus		1e4	1e4	2e4	1					4e4
UWrMaxSat-Scip		3e4	5e4	5e4	266					1e5
MaxHS	1	6e4	3e4	5e4			2e4	1e4	8e3	2e5
WMaxCDCL				9	2e5			2		2e5
WMaxCDCL-BA				9	2e5			2		2e5
UWrMaxSat		2								2
EvalMaxSAT										
CGSS	3e5	39								3e5
Exact		1					4			5
Pacose	3e5		11					12	23	3e5
Z3-maxres		78	12	5						95
Z3-wmax									2e6	2e6
Z3-rc2		7e5	5e4	2						8e5
maxpre2+EvalMS	2e5			9e4	2e6					2e6
# faults	7e5	9e5	2e5	2e5	3e6		2e4	1e4	2e6	7e6
# faulty solver	4	9	7	9	6		2	4	3	44

Delta Debugging	Cr.	Bound Violations					Perf	Others		sum
$1 < $ **sow** $ < 2^{32}$	1.	2.1.	2.2.	2.3.	2.4.	2.5.	3.	4.1.	4.2.	faults
CASHWMS-CP	822	6e3	9e3	2e4	260			82		3e4
CASHWMS-Plus	822	6e3	9e3	2e4	260			82		3e4
UWrMaxSat-Scip	822	7e3	1e4	2e4	505					4e4
MaxHS	249	2e4	1e4	2e4			1e4	2e3	9e3	7e4
WMaxCDCL	48	3		2e3	3e4			932		3e4
WMaxCDCL-BA	48	3		2e3	3e4			932		3e4
UWrMaxSat	822	792					55			2e3
EvalMaxSAT	822									822
CGSS	2e4	9e3								2e4
Exact	2	1e3					2e3			3e3
Pacose	2e4		889					1e3	2e3	3e4
Z3-maxres		5e3	1e3	2e3			359			8e3
Z3-wmax				281	481				3e5	3e5
Z3-rc2		1e5	1e4	1e3						1e5
maxpre2+EvalMS	4e4			4e4	2e5					2e5
# faults	8e4	2e5	6e4	1e5	2e5		2e4	5e3	3e5	1e6
# faulty solver	12	11	7	10	7	0	4	6	3	60

Fuzzing $2^{32}<$**sow**$<2^{64}-1$	Cr. 1.	Bound Violations					Perf 3.	Others		sum faults
		2.1.	2.2.	2.3.	2.4.	2.5.		4.1.	4.2.	
CASHWMS-CP	8	3e3	5e3	8e4			13	20		9e4
CASHWMS-Plus	8	3e3	5e3	8e4			13	20		9e4
UWrMaxSat-Scip	7	8e3	2e4	9e4	56		16	21		1e5
MaxHS	1	2e4	8e3	3e4			5e3	4e3	3e3	7e4
WMaxCDCL	2	78	138	3e3	5e4	2e4	61	3e4		1e5
WMaxCDCL-BA	25	78	138	3e3	5e4	2e4	61	3e4		1e5
UWrMaxSat		14								14
EvalMaxSAT										
CGSS		17	4	3e4						3e4
Exact							12			12
Pacose	9e4				2		8		10	9e4
Z3-maxres		7e3	3e4	6e3		1e5	3			1e5
Z3-wmax		8	185	206	32	1e5			4e5	5e5
Z3-rc2		1e5	4e4	4e3		1e5	2			3e5
maxpre2+EvalMS	8e3			2e4	4e5					5e5
# faults	1e5	2e5	1e5	4e5	5e5	3e5	6e3	6e4	4e5	2e6
# faulty solver	8	11	10	11	6	5	10	6	3	70

Delta Debugging $2^{32}<$**sow**$<2^{64}-1$	Cr. 1.	Bound Violations 2.1.	 2.2.	 2.3.	 2.4.	 2.5.	Perf 3.	Others 4.1.	 4.2.	sum faults
CASHWMS-CP	2e3	1e4	7e4	3e5			1e4	1e4		4e5
CASHWMS-Plus	2e3	1e4	7e4	3e5			1e4	1e4		4e5
UWrMaxSat-Scip	1e3	1e4	1e5	3e5	265		5e3	6e3		4e5
MaxHS	395	3e4	3e4	1e5			3e4	9e3	5e3	2e5
WMaxCDCL	2e3	3e3	7e3	3e4	7e4	4e4	1e4	2e5		3e5
WMaxCDCL-BA	3e3	3e3	7e3	3e4	7e4	4e4	1e4	2e5		3e5
UWrMaxSat		609					4			613
EvalMaxSAT										
CGSS			5e3	2e3	1e5					1e5
Exact							4e3			4e3
Pacose	3e5		9e3	1	3		3	268	3e3	3e5
Z3-maxres		3e4	8e4	3e4		4e5	2e3			5e5
Z3-wmax		374	2e3	9e3	1e3	4e5			7e5	1e6
Z3-rc2		2e5	1e5	3e4		4e5	646			8e5
maxpre2+EvalMS	5e4			7e4	9e5					1e6
# faults	3e5	3e5	5e5	1e6	1e6	1e6	9e4	4e5	7e5	6e6
# faulty solver	8	11	11	12	6	5	11	7	3	74

- The **empty instance**
  –> should yield a weight "o 0"
  –> status line "s OPTIMUM FOUND".
- An **empty hard clause** results in an unsatisfiable instance.
- An **empty soft clause** should be unsatisfiable.
- An **unsatisfiable** instance
  –> status line "s UNSATISFIABLE".
- **Exit code** of a solver is 0 for all results but "s UNKNOWN".

- **Empty instance, empty soft/hard clauses**.
- Non trivial reducible maximal weight instances with a **maximum single weight** $2^{63} - 1$ and a **maximal sum of weights** $2^{64} - 2$.
- Simple unsatisfiable instances.
- Tautology soft/hard clauses.
- With our fuzzer created and delta debugged instances for each fault class-solver combination, **causing each at least one solver to crash**.

```
https://cca.informatik.uni-freiburg.de/maxsatfuzz/
```

Conclusion

Conclusion

- Uncovering **bugs in almost all** MaxSAT evaluation solver.
- **New delta debugging techniques** further shrinking.
- Logging while delta debugging uncovering additional faults.
- MaxSAT **Regression suite**.
- Proposal for additional MaxSAT Evaluation rules.

Future Work

- Unweighted / **incremental** MaxSAT fuzzing.
- Testing existing MaxSAT instance generation tools.

```
https://cca.informatik.uni-freiburg.de/maxsatfuzz/
```

References I

Florent Avellaneda, Carl-Elliott Bilodeau-Savaria, and Lancelot Normand. *Weighted version of EvalMaxSAT 2022*, page 12. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

Fahiem Bacchus. *MaxHS in the 2022 MaxSAT Evaluation*, pages 17–18. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

References II

Fahiem Bacchus, Jeremias Berg, Matti Järvisalo, Ruben Martins, and Andreas (eds) Niskanen. MaxSAT evaluation 2022: Solver and benchmark descriptions. Technical report, Department of Computer Science, University of Helsinki, Helsinki, 2022. URL `http://hdl.handle.net/10138/347396`.

Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. νz-an optimizing SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems: 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings 21*, pages 194–199. Springer, 2015.

References III

Jordi Coll, Shuolin Li, Chu-Min Li, Felip Manyà, Djamal Habet, and Kun He. *MaxCDCL and WMaxCDCL in MaxSAT Evaluation 2022*, pages 15–16. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022a. URL `http://hdl.handle.net/10138/347396`.

Jordi Coll, Shuolin Li, Chu-Min Li, Felip Manyà, Djamal Habet, Jiongzhi Zheng, and Kun He. *MaxCDCL-BandAll in MaxSAT Evaluation 2022*, page 23. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022b. URL `http://hdl.handle.net/10138/347396`.

References IV

Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems: 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings 14*, pages 337–340. Springer, 2008.

Jo Devriendt. *Exact: evaluating a pseudo-Boolean solver on MaxSAT problems*, pages 13–14. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo. Clause redundancy and preprocessing in maximum satisfiability. In *Automated Reasoning: 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8–10, 2022, Proceedings*, pages 75–94. Springer, 2022a.

Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo. *CGSS in the 2022 MaxSAT Evaluation*, pages 10–11. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022b. URL `http://hdl.handle.net/10138/347396`.

Tuukka Korhonen, Jeremias Berg, Paul Saikko, and Matti Järvisalo. Maxpre: an extended MaxSAT preprocessor. In *Theory and Applications of Satisfiability Testing–SAT 2017: 20th International Conference, Melbourne, VIC, Australia, August 28–September 1, 2017, Proceedings 20*, pages 449–456. Springer, 2017.

Zhendong Lei, Yiyuan Wang, Shiwei Pan, Shaowei Cai, and Minghao Yin. *CASHWMaxSAT-CorePlus: Solver Description*, page 8. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

References VII

Tobias Paxian, Sven Reimer, and Bernd Becker. Dynamic polynomial watchdog encoding for solving weighted MaxSAT. In *Theory and Applications of Satisfiability Testing–SAT 2018: 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9–12, 2018, Proceedings 21*, pages 37–53. Springer, 2018.

Marek Piotrów. *UWrMaxSat Entering the MaxSAT Evaluation 2022*, pages 21–22. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

Yiyuan Wang, Shiwei Pan, Zhendong Lei, Shaowei Cai, Minghao Yin, Shuli Hu, and Yupeng Zhou. *CASHWMaxSAT-Plus: Solver Description*, page 9. Volume B-2022-2 of *Department of Computer Science Series of Publications B* Bacchus et al. [2022], 2022. URL `http://hdl.handle.net/10138/347396`.

Andreas Zeller. *Why programs fail: a guide to systematic debugging*. Elsevier, 2009.

Andreas Zeller and Ralf Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering*, 28(2):183–200, 2002.