

CaDiCaL 2.0

International Conference on Computer Aided Verification
July 25, 2024

Armin Biere^{1,2} Tobias Faller¹ Katalin Fazekas³
Mathias Fleury^{1,2} Nils Froleyks² Florian Pollitt¹

¹University Freiburg, Freiburg, Germany

²Johannes Kepler University Linz, Austria

³TU Wien, Vienna, Austria



FWF

JKU

DFG

universität freiburg

SAT Solving and CaDiCaL 2.0

- Open source incremental SAT solver in C++ since 2017
- CaDiCaL: Conflict-Driven-Clause-Learning
- 2.0: Very wide range of new features
- Competitive performance
 - BVA + CaDiCaL 1.5.3: Winner of SAT Competition Main track in 2023
 - Competitive incremental performance as well

Main Components of CaDiCaL 2.0

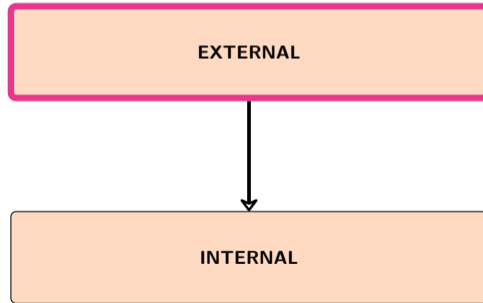


INTERNAL

Main Components: Internal Module

- The core class of the solver
- Implements several search improving techniques
 - Chronological backtracking, rephasing, shrinking, . . .
- Implements several formula simplification techniques
 - BVE, ELS, CCE, BCE, vivify, instantiate, probing, subsume, lucky phases, random walk, gate elimination, transitive reduction, ternary resolution, globally blocked clauses, . . .
- Supports inprocessing during incremental SAT solving without the need of freezing variables
 - Helpful in Bounded Model Checking (see CAMICAL)

Main Components of CaDiCaL 2.0

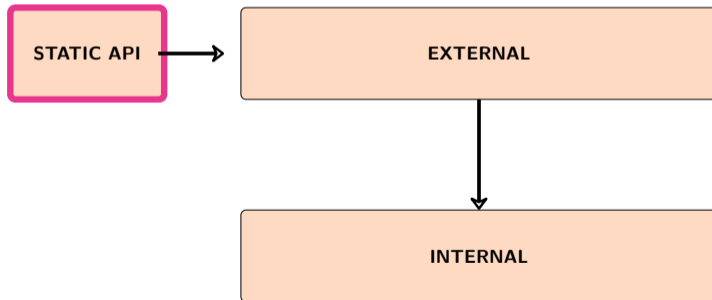


Main Components: External Module

- Hides internals of the solver

- Maps between the internal and external problem representations
 - Compacting: Map active variables into a consecutive sequence of integers
 - Solution reconstruction
 - Proof generation (natively supported proof formats: DRAT, FRAT, LRAT, VeriPB)

Main Components of CaDiCaL 2.0



Main Components: Static API

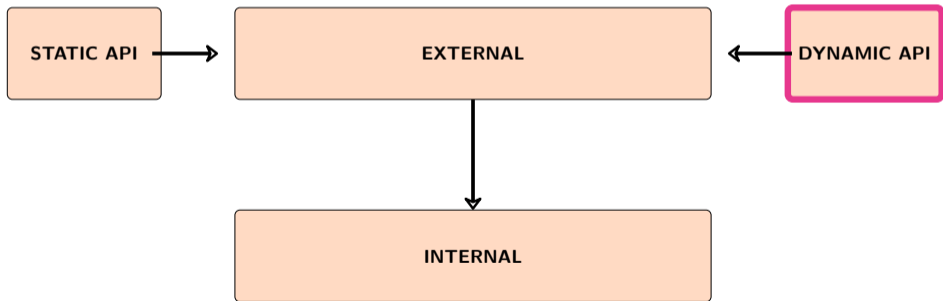
Provides access to standard solver functionalities **between** SAT solving calls:

- Parsing DIMACS or iCNF files
- IPASIR functions
- Freezing and melting *Don't Touch*-variables
- ...

Many interesting features:

- Flipping solution literals
 - Backbone extraction (see CADIBACK)
- Constrain: clausal assumption
 - Bounded Model Checking (see CAMICAL)
- Incremental Lazy Backtracking between SAT calls
- ...

Main Components of CaDiCaL 2.0

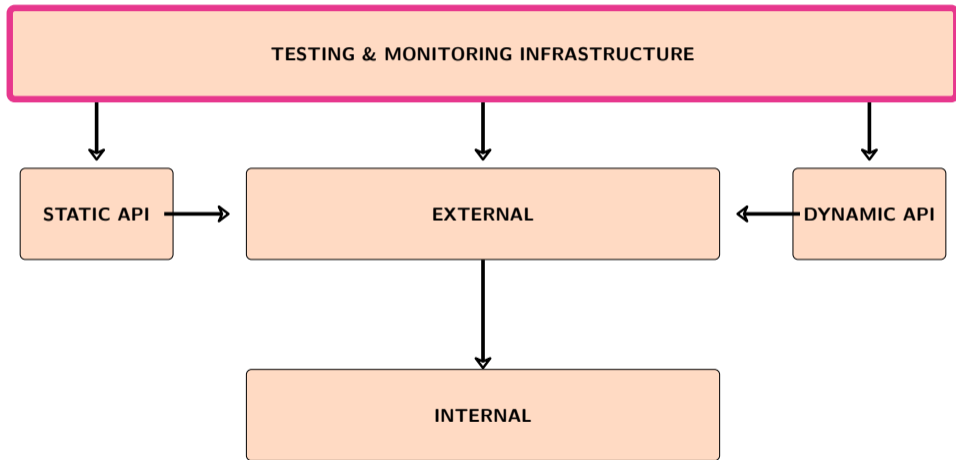


Main Components: Dynamic API

Provides an interface to interact with the solver **during** solving calls:

- Proof Tracer
 - In-memory interpolation through online in-memory LRAT-Proof tracing
→ CADICRAIG
- Learner & Iterators
 - Real-time tracing of learned clauses of the solver
 - Allows to iterate current (simplified) clauses of problem
- Terminator
 - Allows to interrupt the search of the solver
- External Propagator
 - IPASIR-UP functions: Fine-grained external control
 - Many possibilities: collect implied literals on-the-fly, force specific decision order, guide search externally, add clauses during solving, ...
→ Integrates CADICAL in CVC5 SMT solver

Main Components of CaDiCaL 2.0

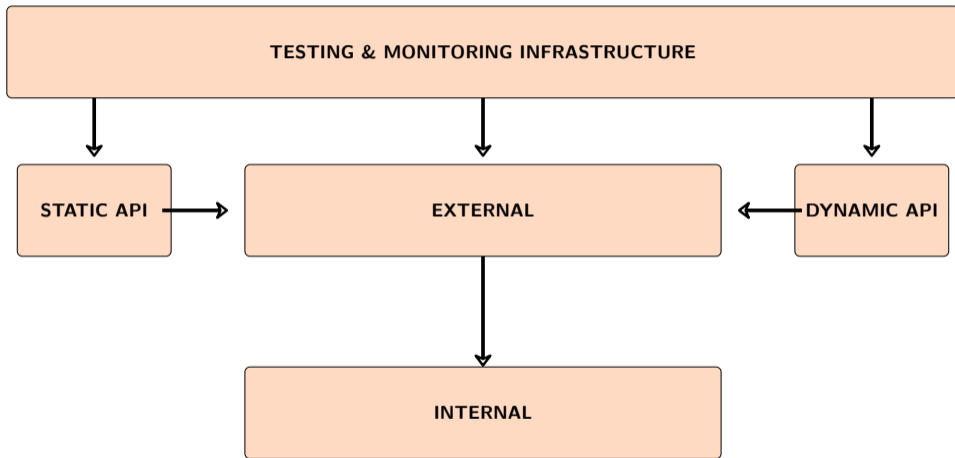


Main Components: Testing & Monitoring Infrastructure

- Comprehensive logging
- Several internal assertions
- Built-in proof and solution checkers
- MOBICAL: built-in API fuzzer
 - Generates random (incremental) API calls
 - Minimizes failing runs

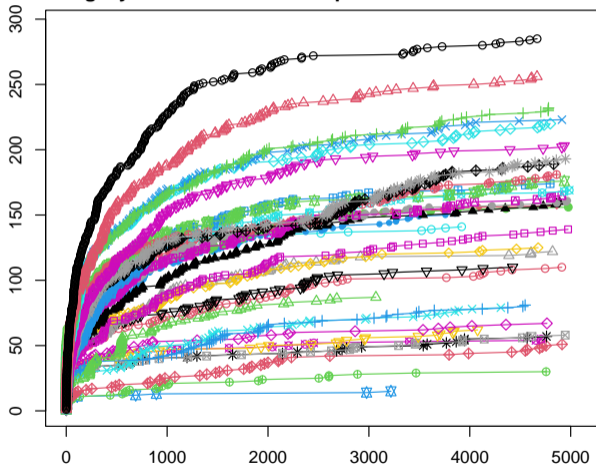
```
m start seed 13117766510884613014
m
m count          seed/buggy/reducing/reduced          calls vars clauses
m -----
m 100470         16906432094431841057          1715  85 293  37
m -----
m generated 100470 traces: 100469 ok 100%, 0 failed 0%, 100470 executed
m solved 250598: 33237 sat 13%, 136193 unsat 54%, 150128 incremental 60%
```

Main Components of CaDiCaL 2.0



Experiments – Stand-Alone

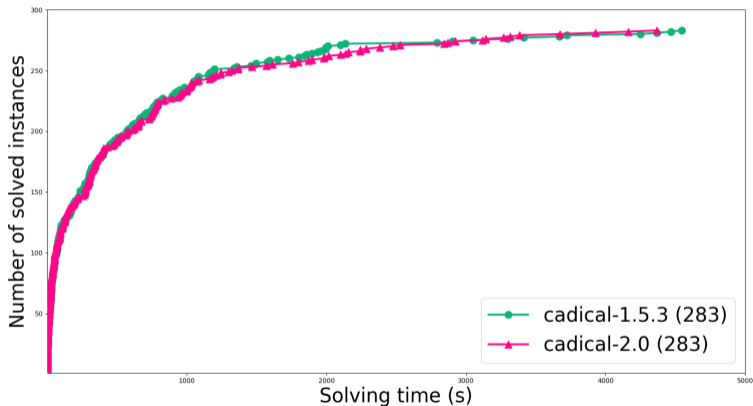
Legacy Solvers on SAT Competition 2023 Benchmarks



- 285 sbva-radical-2023
- △ 256 radical-2019
- + 232 kissat-mab-hywalk-2022
- × 223 kissat-mab-2021
- ◇ 220 kissat-2020
- ▽ 203 maple-lcm-disc-cb-dl-v3-2019
- * 193 maple-lcm-dist-2017
- ◆ 189 maple-lcm-dist-cb-2018
- 181 maple-comsps-drup-2016
- ⊠ 176 lingeling-2014
- 174 abcdsat-2015
- ⊞ 169 glucose-2016
- ⊞ 165 glucose-2011
- 161 lingeling-2010
- ▲ 160 cryptominisat-2010
- 158 lingeling-2013
- 156 glucose-2012
- 151 lingeling-2011
- 141 minisat-2008
- 139 precosat-2009
- ◇ 125 minisat-2006
- △ 122 rsat-2007
- ▽ 110 satellite-gti-2005
- 110 minisat-2005
- △ 87 chaff-2001
- + 81 siege-2003
- × 79 limmat-2002
- ◇ 67 picosat-2007
- ▽ 62 berkmin-2003
- 58 zchaff-2007
- * 57 zchaff-2004
- 55 march-2011
- ◇ 51 Boehm1-1992
- 30 posit-1995
- ⊠ 15 grasp-1997

■ Cumulative distribution function (CDF), time out 5000 seconds, Intel Xeon E5-2620 v4 CPUs running at 2.10 GHz

Experiments – Stand-Alone



■ Cumulative distribution function (CDF), time out 5000 seconds, Intel Xeon E5-2620 v4 CPUs running at 2.10 GHz

Experiments – Incremental

	CaDiBack	CaMiCaL	Bones	LSP	Max	Ijtihad	Total	
CADICAL	2.0	3297 ₁₁	2606 ₁₈	494 ₄₅	1898 ₂₇	1976 ₂₆	2980 ₁₃	2209 ₁₄₀
	2020	3409 ₉	2677 ₁₇	622 ₄₃	1955 ₂₆	2015 ₂₅	2986 ₁₃	2277 ₁₃₃
	1.0	3495 ₇	2627 ₁₈	595 ₄₄	2011 ₂₆	2028 ₂₅	2989 ₁₃	2291 ₁₃₃
CMS	3491 ₈	2701 ₁₇	397 ₄₆	1773 ₂₉	2021 ₂₅	3057 ₁₂	2240 ₁₃₇	
MINISAT	3678 ₅	2807 ₁₆	687 ₄₃	1993 ₂₆	2094 ₂₄	3123 ₁₁	2397 ₁₂₅	
RISS	3665 ₆	2836 ₁₅	892 ₄₀	1835 ₂₈	2017 ₂₅	3140 ₁₁	2398 ₁₂₅	
ABCDSAT	3582 ₇	2966 ₁₃	535 ₄₆	2493 ₂₁	2037 ₂₆	3207 ₁₀	2470 ₁₂₃	
GLUCOSE	3778 ₄	2981 ₁₃	948 ₄₀	2078 ₂₅	2117 ₂₄	3206 ₁₀	2518 ₁₁₆	
VBS	3127 ₁₄	2546 ₁₉	257 ₄₈	1765 ₂₉	1856 ₂₈	2896 ₁₄	2075 ₁₅₂	

■ PAR2 score over 50 benchmarks per application, time out 5000 seconds, Intel Xeon E5-2620 v4 CPUs running at 2.10 GHz

Summary

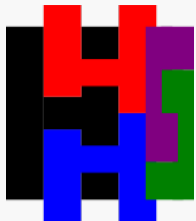


<https://github.com/arminbiere/cadical>



- Open source (MIT License)
- Very wide feature set
- Good performance both in stand-alone and incremental use cases
- Thorough testing infrastructure
- Extensible codebase

Thank you!



Hardware Model Checking Competition

HWMCC'24

Call for Benchmarks and Solvers

<https://hwmcc.github.io/2024>

affiliated to FMCAD'24, Oct. 14–18, 2024
Prague, Czech Republic

Three Single Safety Property Tracks

1. **World-Level** Track **without Arrays** (BTOR2)
2. **Word-Level** Track **with Arrays** (BTOR2)
3. **Bit-Level** with mandatory **Certificates** (AIGER)

Bit-Level Certificates checked by **Certifaiger**
<https://github.com/Froleyks/certifaiger>

Important Dates

<u>Benchmark</u> Submission	August 18
<u>Solver</u> Submission	September 1

Organizers

Armin Biere *University of Freiburg* Germany
Nils Froleyks *Johannes Kepler University Linz* Austria
Mathias Preiner *Stanford University* USA