# Generating Extended Resolution Proofs with a BDD-Based SAT Solver Demonstration Artifact

Randal E. Bryant and Marijn J. H. Heule

Computer Science Department
Carnegie Mellon University, Pittsburgh, PA, United States
{Randy.Bryant, mheule}@cs.cmu.edu

This artifact contains the complete implementation of a BDD-based SAT solver that generates checkable proofs of unsatisfiability. It also contains programs that will generate and test benchmarks based on the mutilated chessboard and pigeonhole problems. The solver generates proofs in the LRAT format [3]. An LRAT checker is included as part of the artifact.

This artifact is submitted in conjunction with a research paper describing the solver and proof generator, as well as experimental data for which the artifact demonstrates a subset of the results.

## Downloading and running the demonstration

The solver is written entirely in Python, as are the benchmark generators. The LRAT checker is written in C. All programs have been tested in the TACAS virtual environment.

A demonstration of the program can be performed with the following sequence of commands:

```
> wget http://www.cs.cmu.edu/~bryant/download/pgbdd-artifact.zip
> unzip pgbdd-artifact.zip
> cd pgbdd-artifact
> make chess
> make pigeon
```

The latter two operations will cause benchmarks to be generated and run. A lot of data will be printed, but for each benchmark, the lines `UNSAT` and `VERIFIED` should occur. The final summary data will be stored in files `chess-results.txt` and `pigeon-results.txt`. Each line of the files indicates the problem size $N$, the number of clauses generated (a measure of the proof complexity), and the runtime of the solver (including proof generation.)

## Benchmark problems

The mutilated chessboard problem considers an $N \times N$ chessboard, with the corners on the upper left and the lower right removed. It attempts to tile the board with dominos, with each domino covering two squares. Since the two removed squares had the same color, and each domino covers one white and one black square, no tiling is possible. This problem has been well studied in the context of resolution proofs, for which it can be shown that any proof must be of exponential size [1]. Our demonstration shows that the combination BDDs, a careful sequencing of conjunction and quantification operations, and a carefully chosen variable ordering yields polynomial performance, handling up to $N = 40$. In the submitted paper, we show results up to $N = 124$.

The pigeonhole problem is one of the most studied problems in propositional reasoning. Given a set of $N$ holes and a set of $N + 1$ pigeons, it asks whether there is an assignment of pigeons to holes such that 1) every pigeon is in some hole, and 2) every hole contains at most one pigeon. The answer is no, of course, but any resolution proof for this must be of exponential length [4]. Our demonstration shows that the combination BDDs, a careful sequencing of conjunction and quantification operations, and a carefully chosen variable ordering yields polynomial performance, handling up to $N = 40$. In the submitted paper, we show results up to $N = 160$.

## Interpreting the results

Here are the results obtained for the mutilated chessboard problem, running within the TACAS virtual environment and having as host machine a a 4.2 GHz Intel Core i7 processor running the OS X operating system. It uses the default virtual memory limit of 2 GB. For comparison, we show the results for KISSAT, the winner of the 2020 SAT competition [2][1]. KISSAT represents the state of the art in search-based SAT solvers. The KISSAT benchmarks were run on the same machine, but not in a virtual environment. KISSAT was configured to abort when the proof length exceeds 200 million clauses. These cases are indicated in the table with entries "—".

| $N$ | PGBDD | | KISSAT | |
|---|---|---|---|---|
| | Clauses | Seconds | Clauses | Seconds |
| 004 | 1,484 | 0.02 | 58 | 0.01 |
| 008 | 11,932 | 0.15 | 942 | 0.01 |
| 012 | 34,310 | 0.42 | 48,296 | 50.92 |
| 016 | 71,533 | 0.96 | 4,503,635 | 83.14 |
| 020 | 126,568 | 1.89 | 65,376,268 | 1,556.78 |
| 024 | 202,375 | 3.40 | — | — |
| 028 | 301,914 | 6.12 | — | — |
| 032 | 428,145 | 9.82 | — | — |
| 036 | 584,028 | 15.47 | — | — |
| 040 | 772,523 | 23.96 | — | — |

These results show PGBDD scaling as $O(N^{2.7})$, and KISSAT scaling exponentially.

Here are the results obtained for the pigeonhole problem with the same experimental conditions.

[1] Version 1.0.3. Downloaded from `https://github.com/arminbiere/kissat` on 17 August, 2020

| $N$ | PGBDD | | KISSAT | |
|---|---|---|---|---|
| | Clauses | Seconds | Clauses | Seconds |
| 004 | 2,445 | 0.03 | 73 | 0.01 |
| 008 | 17,528 | 0.22 | 3,898 | 0.05 |
| 012 | 56,385 | 0.71 | 419,734 | 12.20 |
| 016 | 130,766 | 1.64 | 196,155,696 | 38,198.01 |
| 020 | 253,055 | 3.34 | — | — |
| 024 | 436,276 | 5.86 | — | — |
| 028 | 694,093 | 9.40 | — | — |
| 032 | 1,040,810 | 14.60 | — | — |
| 036 | 1,491,371 | 21.80 | — | — |
| 040 | 2,061,360 | 30.69 | — | — |

These results show PGBDD scaling as $O(N^3)$, and KISSAT scaling exponentially.

## Archival version

The artifact is also available on GITHUB as:

```
> git clone https://github.com/rebryant/pgbdd-artifact
```

We intend to maintain this repository and continue to make it publicly available.

## References


1. Alekhnovich, M.: Mutilated chessboard problem is exponentially hard for resolution. Theoretical Computer Science **310**(1-3), 513–525 (Jan 2004)
2. Biere, A., Fazekas, K., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba, Plingeling, and Treengeling entering the SAT competition 2020 (2020), unpublished
3. Cruz-Filipe, L., Heule, M.J.H., Hunt, W.A., Kaufmann, M., Schneider-Kamp, P.: Efficient certified RAT verification. In: de Moura, L. (ed.) Automated Deduction – CADE-26. LNCS, vol. 10395, pp. 220–236 (2017)
4. Haken, A.: The intractability of resolution. Theoretical Computer Science **39**, 297–308 (1985)