Team NUP-JetBrains at IWLS2025 contest

Stanislav Alekseev, Timur Degteari, Gregory Emdin, Mikhail Goncharov, Ilia Kondakov, Fedor Kurmazov, Maksim Levitskii, Georgii Levtsov, Maksim Shevkoplias, Roman Shumilov, Alexander S. Kulikov

EPFL, JetBrains, Neapolis University Pafos

13 June 2025

Working on last year’s success

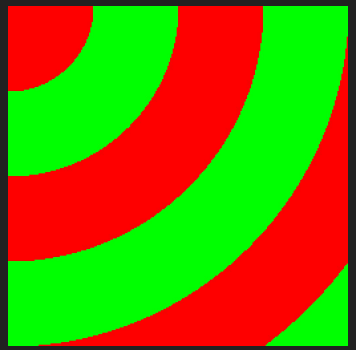
- We won last year’s edition and created an open-source tool Cirbo (https://github.com/SPbSAT/cirbo)
- It has optimized generators for basic arithmetic operations like sum, product or square root, as well as some properties check, circuit simplification and API to construct a circuit out of different parts
- Also we made a bruteforce with some clever optimizations to find the permutation of inputs for each of the old benchmarks. This way, we ensured that our solutions are no worse than the best ones from 2024.
- We used our last year’s tool to call different ABC commands
- We used our last year’s tool which iterates over small subcircuits and tries to replace them (https://github.com/SPbSAT/simplifier)

Guessing functions

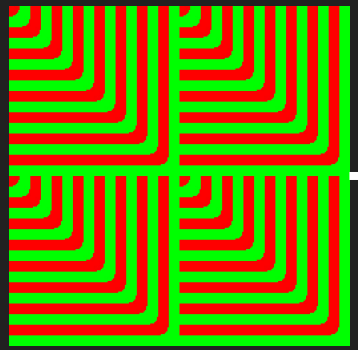
Since it is hard to generate a circuit for an arbitrary function, knowing actual meaning of a function would help, so we invested a lot of resources into that area.

We looked at different characteristics (number of zeros, injectivity), also drew a picture for each output, where we split the input into two parts, interpret each one as a number, and a pixel (i,j) gets a colour depending on an output when the first number is i and the second one is j.

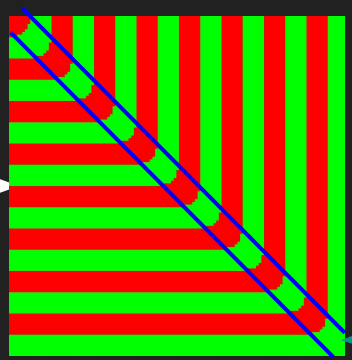
For some functions we managed to see what they do, for some we got some insights and made a final circuit using ABC.



clear



not clear



Maximum

Some truth table, ABC

Residue Number System to Binary

10 benchmarks are of the following format: inputs are split into blocks.
Each block represents a remainder some modulo. If for one of the blocks the value is at least the modulo, output zero. Otherwise, the minimum non-negative integer with such remainders.

We can write an explicit formula for this problem with some modular multiplications and additions
Since numbers in blocks have a cap (modulo), we may optimize multiplication parts

	ABC	Shennon	Generator	Smart mul	SAT
ex150	390	262	200	161	160
ex151	371	277	201	130	126
ex152	246	152	157	90	87
ex153	252	182	137	110	105
ex154	719	397	281	284	250

Automatic generation and improvement

- For our baseline, we used ABC tool.
- As was discovered this year, multiple calls of our ABC wrapper may improve the circuit. Once the circuit was improved after more than **4500** calls.
- CIOPS and eSLIM were also used, simplifying some circuits by **40%.**
- For a few benchmarks, we used our new tool which convert a python code into a circuit.
- Supports all basic operations, but doesn’t support loops and RAM access.
- Can write a function in bench language and call it as a python function.

Automatic generation and improvement

- Improved our SAT-based local improvement tool, now it hashes subcircuits to avoid repetitions, also it may take the original inputs into consideration (https://github.com/alexanderskulikov/circuit_improvement)
- Once SAT finds an improvement, make a few cheap calls to our ABC wrapper
- The best SAT-solver in our case isn't the winner of SAT competition
- Our solutions were running on an HPC-cluster of Neapolis University and on a Google Cloud provided by JetBrains Research.

copy

```
Kissat: 0.1415538787841797
cadical: 0.05524110794067383
gluecard3: 0.07761263847351074
gluecard41: 0.09806585311889648
glucose3: 0.07742595672607422
glucose4: 0.10268688201904297
lingeling: 0.18399691581726074
maplechrono: 0.41112494468688965
maplecm: 0.10690999031066895
maplesat: 0.053774118423461914
mergesat3: 0.06518983840942383
minicard: 0.05794882774353027
minisat22: 0.05624699592590332
```

Performance on our test data

Solver	Score ▲▼	Solved ▲▼
VBS	1681.3	347
kissat-sc2024	2788.13	306
Kissat_MAB-DC	3435.23	290
hKis-bva	3461.15	285
BreakID-Kissat	3461.77	284
Kissat_MAB_ESA	3490.98	287
CaDiCaL	3494.11	284
Kissat_MAB_Binary	3516.04	285
Cadical_ESA	3560.32	281
AMSAT_RGC	3631.69	279
AMSAT	3641.2	278
hCaD-bva	3728.66	271
hCaD-pbva	4056.35	261
hKis-pbva	5170.5	217
IsaSAT	6325.79	165

SAT 2024 Contest

Thank you for your attention!

- **Cirbo:** https://github.com/SPbSAT/cirbo – arithmetic generators, properties check, circuit simplification, circuit from different parts
- **SAT-based improvement:** https://github.com/alexanderskulikov/circuit_improvement
- **Simplification of small subcircuits:** https://github.com/SPbSAT/simplifier
- Guessed some functions by looking at some characteristics and pictures
- For some benchmarks, insight + ABC
- Residue Number System to Binary – optimizing modular multiplication using cap on values
- ABC baseline
- Python-to-bench
- ABC, eSLIM, CIOPS to simplify
- Most of our tools are in private repositories, but we will post them to Cirbo eventually