Yaga

MCSat-based SMT Solver

M. Blicha, D. Hanák, and J. Kofroň

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Department of
Distributed and
Dependable
Systems

What Is Yaga?

- SMT solver developed at MFF UK, mainly by D. Hanák, M. Blicha, and J. Kofroň
- Developed with the goal to investigate alternatives to the dominant DPLL(T) framework for SMT solving
- Implements the Model Constructing Satisfiability Calculus (MCSAT)
  - currently supports QF LRA logic of SMT-LIB

YAGA ARCHITECTURE

YAGA consists of several parts:

- Core solver—implements the CDCL algorithm
- Solver trail—records progress of the solver in trails
- Clause database—storing learned clauses
- Theory plugins
- Heuristics

Trail

Trail is a sequence of trail elements:

- decision—assignment of a value to a variable
- clausal propagation—propagation of a literal by a Boolean constraint propagation (unit clause)
  - e.g., $A \vee B$ and $A = \mathit{false}$
- semantic propagation—propagates fully assigned constraints
  - e.g., $x + y < 0$ and $x = 0, y = 1$

Core Algorithm

```
while true do
    conflicts ← propagate();
    if conflicts then
        learned, level ← analyze(conflicts);
        if any clause in learned is an empty clause then
            return unsat
        backtrack_with(learned, level);
    else
        if all variables have been decided then
            return sat
        else
            decide();
```

Theory Plugins

The main goals of theory plugins are to:

- **propagate**
  - propagates implied literals
  - generates conflict clauses
- **decide**
  - deciding variable values in a consistent way

Boolean Plugin

Performs Boolean constraint propagation (BCP) to exhaustion

- using *watched literals* to detect unit clauses
- cache the last checked position in each clause

Decides values of Boolean variables

- various heuristics: always true/false, phase-saving

LRA Plugin

- Decides linear real arithmetic
- Supports linear (in)equalities on linear polynomials with rational coefficients
- Caches and propagates variable bounds along the trail
- Performs conflict analysis (on variable values)
  - bound conflicts
  - dis-equality conflicts

HEURISTICS

- Variable order—just selects a variable, the value is decided by a theory plugin
- Restart policy—determines whether to restart the search
- Clause deletion—deletes unnecessary learned clauses on restart

Future Work

- Support for theories beyond Linear Real Arithmetic
  - Linear Integer Arithmetic
  - Non-Linear Arithmetic
  - Uninterpreted Functions
- Support for incremental checking and Craig interpolation

Conclusion

- YAGA is MCSat-based SMT solver
- It took part in SMTComp competition in 2023
- It is available at GitHub: `https://github.com/d3sformal/yaga`
- We still miss a logo :-)