Hochschule
München
University of
Applied Sciences

Faculty of Computer Science
and Mathematics

parSAT: Parallel Solving of Floating-Point Satisfiability

FROM 2025

Markus Krahl, Matthias Güdemann, and Stefan Wallentowitz

Motivation

- Embedded Systems control safety-critical processes by calculations of Floating Point (FP) numbers
- Most often FP-Implementations based on the IEEE 754 Standard are used
- FP numbers only approximate real values and do not fulfill mathematical rules (associativity, addition)
- Due to this behavior, FP algorithms might produce unexpected or incorrect results
- SMT solvers play a crucial role in verifying FP computations
- Current state-of-the-art SMT solvers convert FP arithmetic into propositional logic through bit- or word-blasting
- This overhead may result in limited capacity for reasoning about non-linear FP constraints
- Ideas for ***parSAT***:
  - Exploit dedicated hardware support for FP computations (FP Units in CPUs)
  - Improve parallel solving of FP SMT equations

Related Work

- ***parSAT*** extends the concepts presented in XSat and goSAT:
  - XSat:
    - Foundations for translating FP constraints into Global Optimization (GO) problem
    - Emits the optimization function as C-code, compiles it, and it reloads it as Python-Module
    - Applies Basin Hopping (BH) from the Scipy-Package to find the global minimum
  - goSAT:
    - Based on XSat
    - Applies De-Morgan's Law when translating an SMT equation into a GO problem
    - Performs a Just-In-Time compilation of the optimization function
    - Employs various GO routines of the NLOpt library for finding the global minimum
- Research investigating portfolio-settings with SMT solvers

Contributions

- ***parSAT***, an integrated tool that
  - represents a semi-decision procedure for SMT equations in FP theory
  - supports a broader scope of the SMTLIB2-standard
  - models the effect of infinite FP numbers in the generated optimization function
  - performs a portfolio-based minimization by using a multicore implementation
  - enables the integration of any GO algorithm
- Evaluation of ***parSAT*** with current state-of-the-art SMT solvers on different benchmarks

Theoretical Background

- Convert a quantifier-free FP SMT equation ***F*** into an optimization function ***G***
  - If x corresponds to a satisfiable assignment of ***F*** than ***G***(x) must be zero
  - Otherwise, ***G*** returns a positive distance value to the global minimum at zero
- Translation of negated comparison operators for FP numbers must be considered:

$$\neg(r_a < r_b) \quad = \quad (r_a \geq r_b)$$

$$\neg(f_a < f_b) \quad \neq \quad (f_a \geq f_b) \quad \text{where} \quad f_b = NaN$$

- ***F*** in conjunctive normal form with removed negations (De-Morgan) is translated to ***G***:

$$F_{CD}(\vec{x}) = \bigwedge_{i \in I} \bigvee_{j \in J} e_{i,j} \bowtie_{i,j,n} e'_{i,j} \quad \Rightarrow \quad G(\vec{x}) = \sum_{i \in I} \prod_{j \in J} d(\bowtie_{i,j,n}, e_{i,j}, e'_{i,j})$$

- Subscript *n* indicates if a negation of comparison operator (bowtie) occurred

Theoretical Background

- Definitions for $d(\bowtie_{i,j,n}, e_{i,j}, e'_{i,j})$

$$
\begin{aligned}
d(==_0, e_1, e_2) &= d(\neq_1, e_1, e_2) = \theta(e_1, e_2) \\
d(\neq_0, e_1, e_2) &= d(==_1, e_1, e_2) = e_1 \neq e_2 \; ? \; 0 : 1 \\
d(<_0, e_1, e_2) &= e_1 < e_2 \; ? \; 0 : \theta(e_1, e_2) + 1 \\
d(<_1, e_1, e_2) &= isnan(e_1) \vee isnan(e_2) \; ? \; 0 : d(\geq_0, e_1, e_2) \\
d(\leq_0, e_1, e_2) &= e_1 \leq e_2 \; ? \; 0 : \theta(e_1, e_2) \\
d(\leq_1, e_1, e_2) &= isnan(e_1) \vee isnan(e_2) \; ? \; 0 : d(>_0, e_1, e_2) \\
d(>_0, e_1, e_2) &= e_1 > e_2 \; ? \; 0 : \theta(e_1, e_2) + 1 \\
d(>_1, e_1, e_2) &= isnan(e_1) \vee isnan(e_2) \; ? \; 0 : d(\leq_0, e_1, e_2) \\
d(\geq_0, e_1, e_2) &= e_1 \geq e_2 \; ? \; 0 : \theta(e_1, e_2) \\
d(\geq_1, e_1, e_2) &= isnan(e_1) \vee isnan(e_2) \; ? \; 0 : d(<_0, e_1, e_2)
\end{aligned}
$$

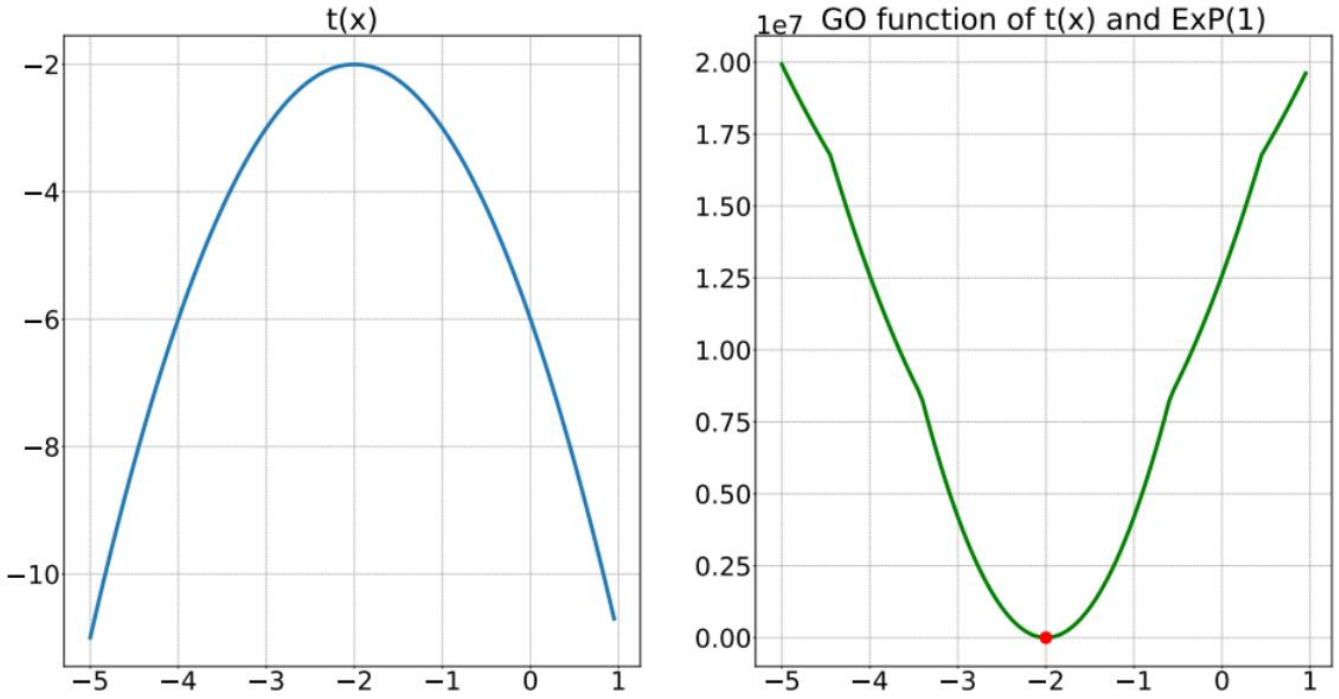
where

$$\theta(e_1, e_2) = isnan(e_1) \vee isnan(e_2) \; ? \; 1 : (e_1 == e_2 \; ? \; 0 : (|bits(e_1) - bits(e_2)|))$$

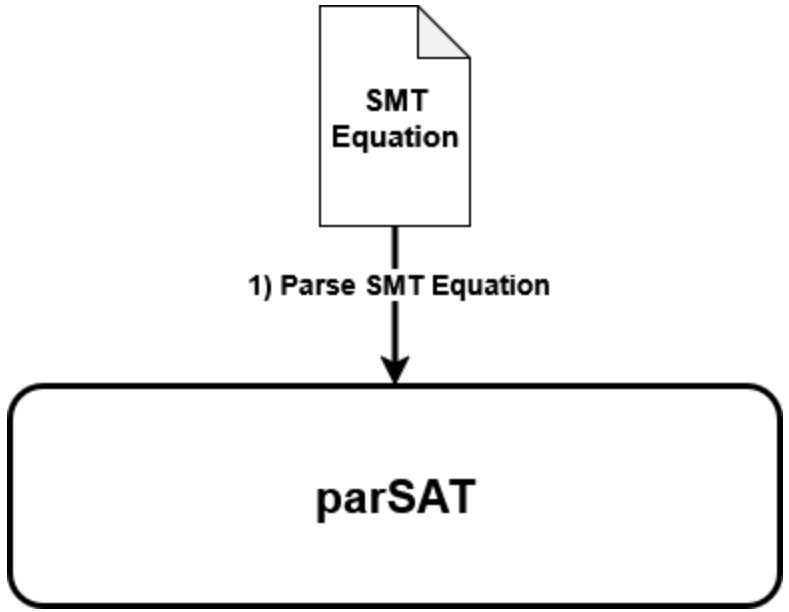
Example

$$t(x) = -1\dot{(}x+2)^2 - 2$$

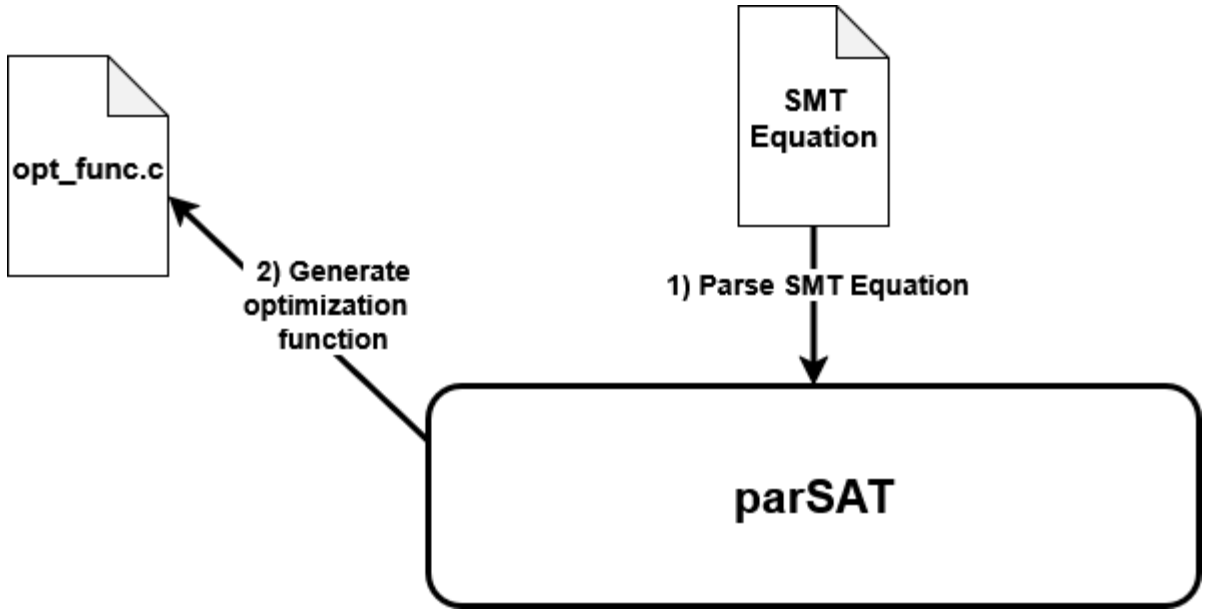
ExP(1): $x \in FP \wedge t(x) \geq -2.$



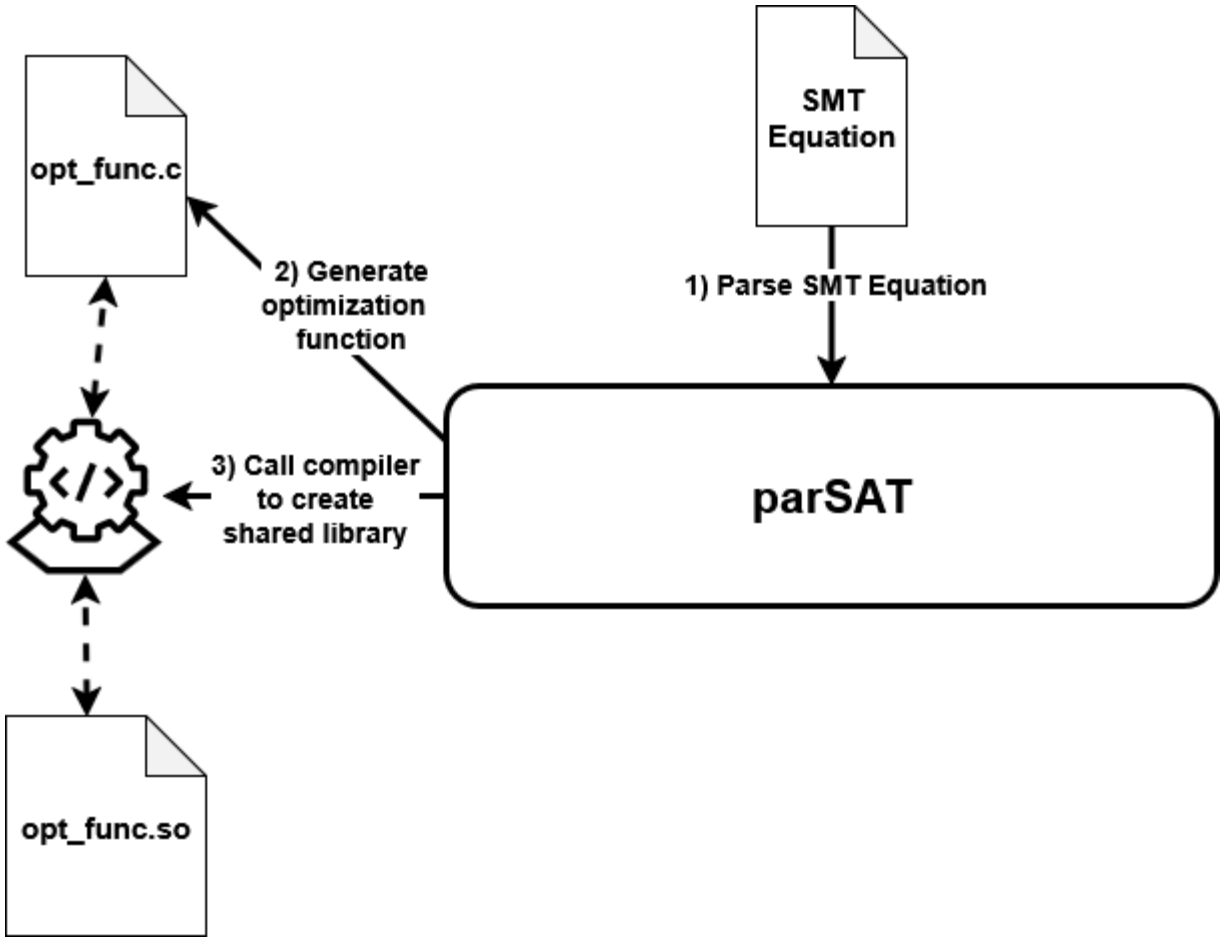
parSAT‘s execution flow



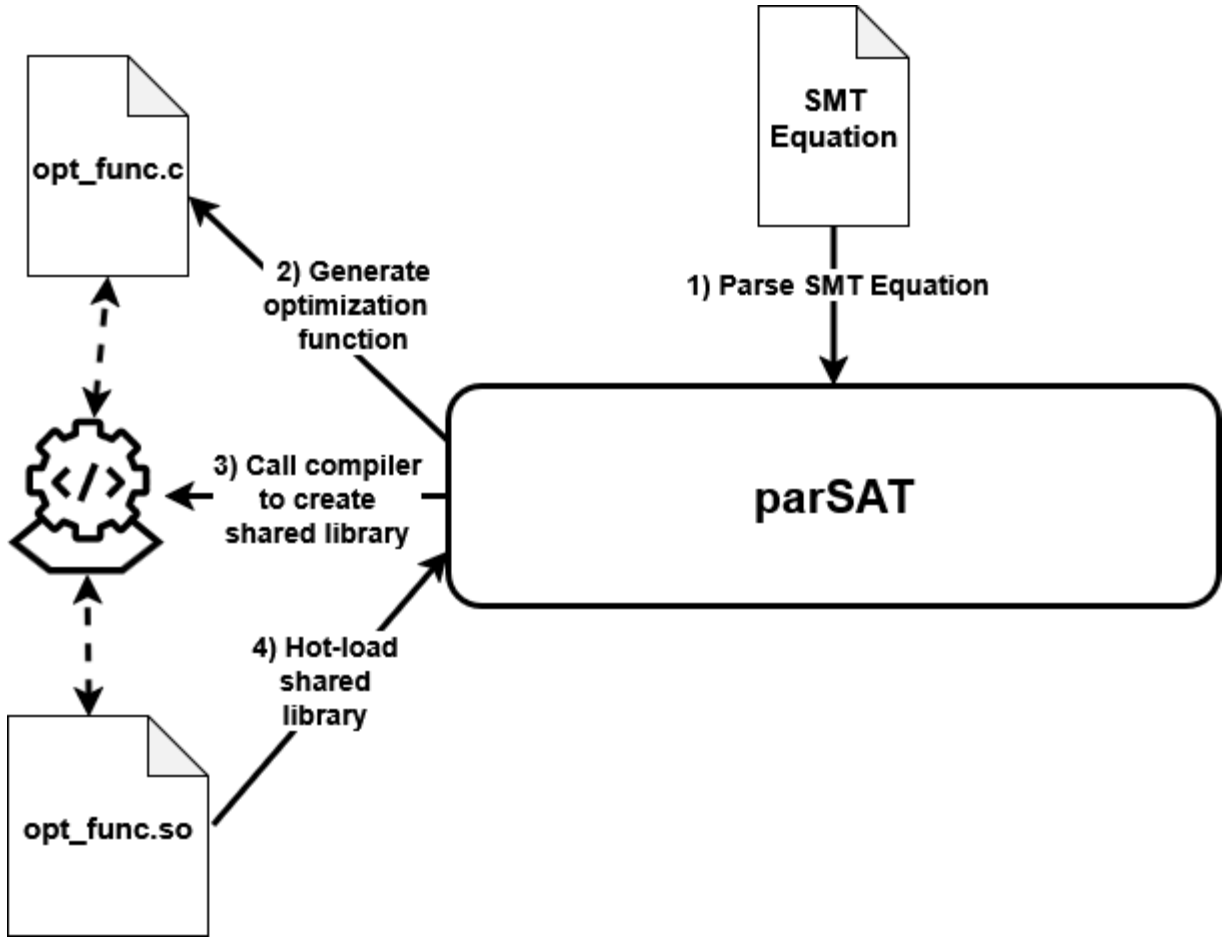
parSAT‘s execution flow



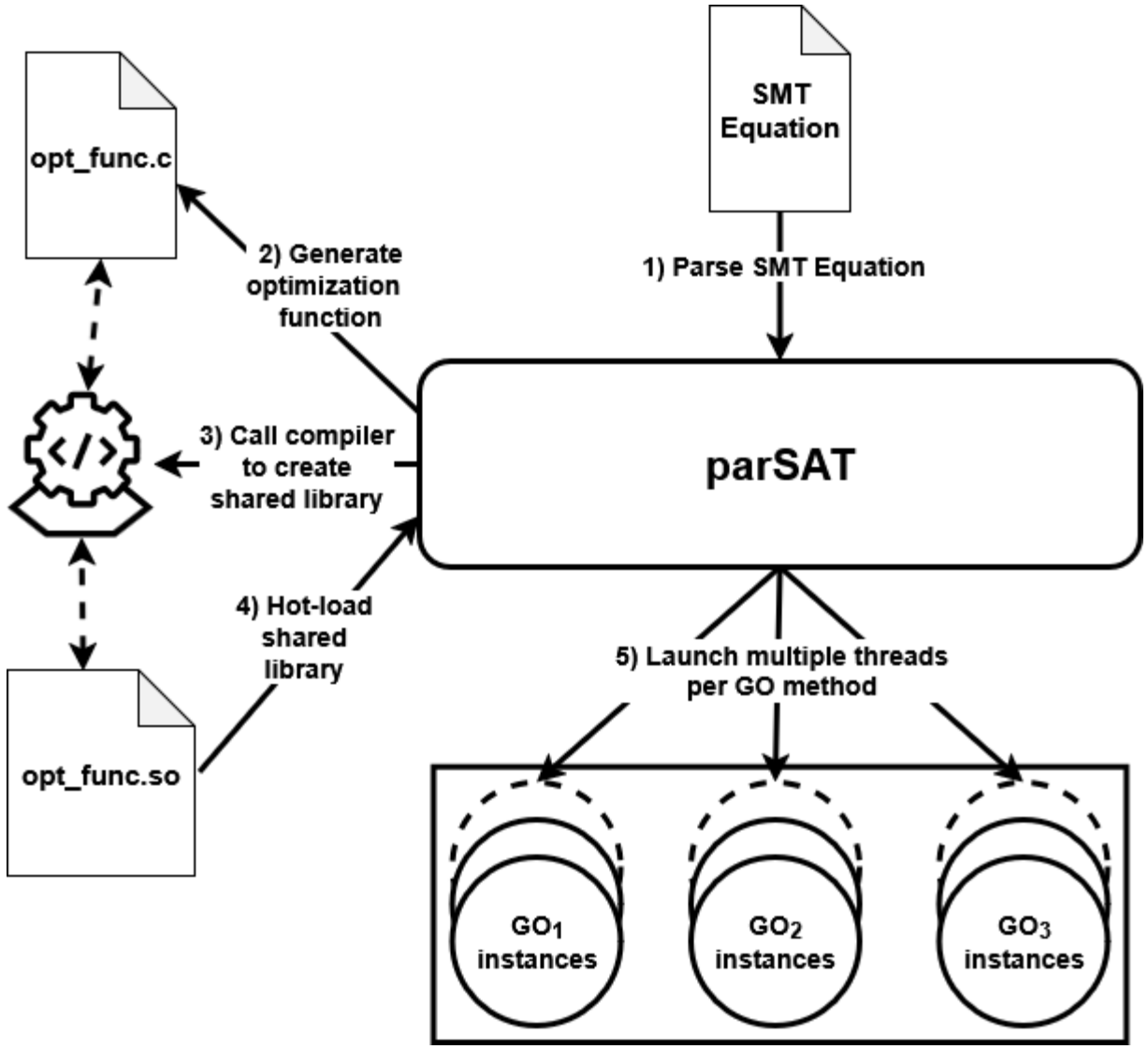
parSAT‘s execution flow



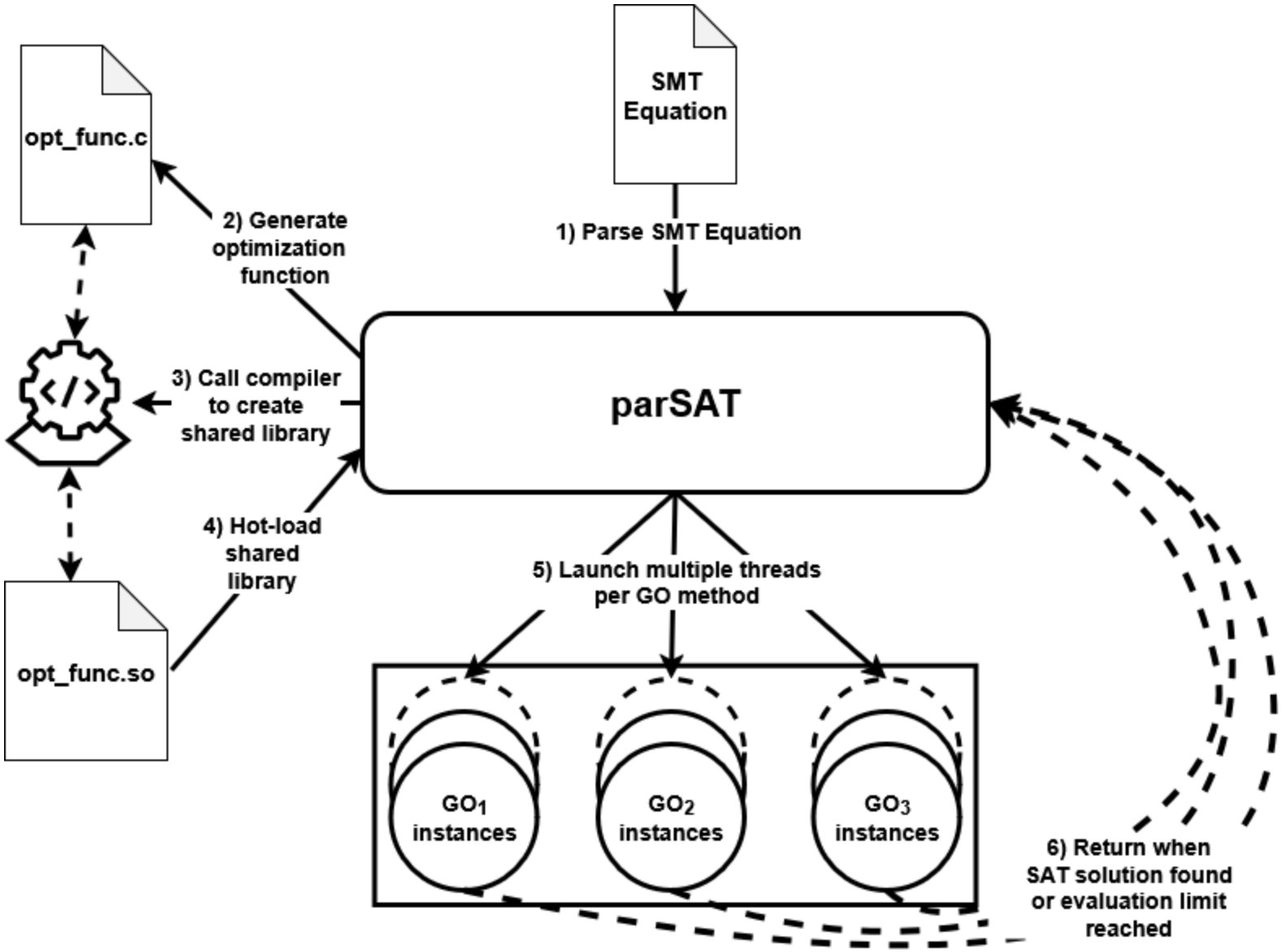
parSAT‘s execution flow



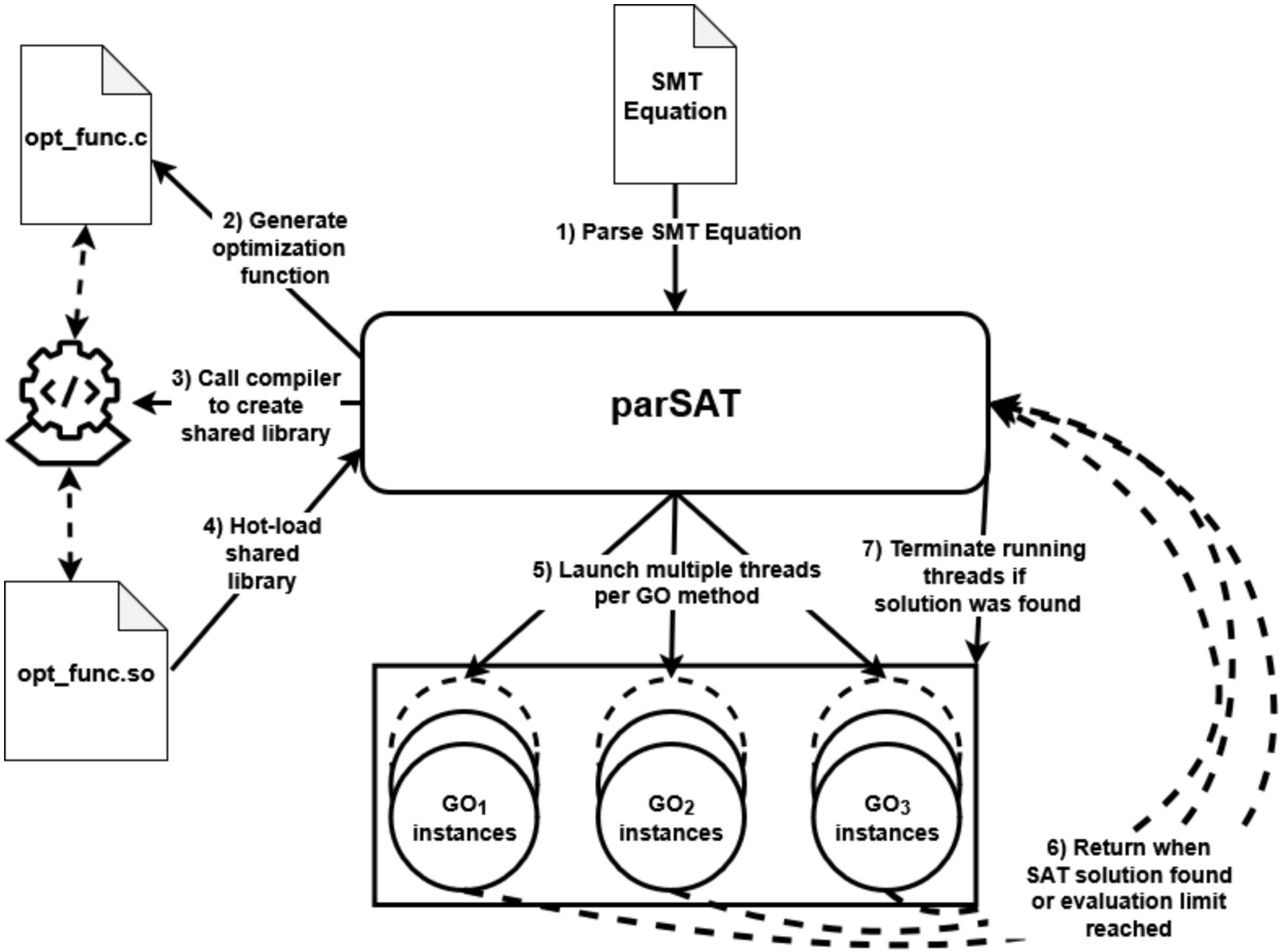
parSAT‘s execution flow



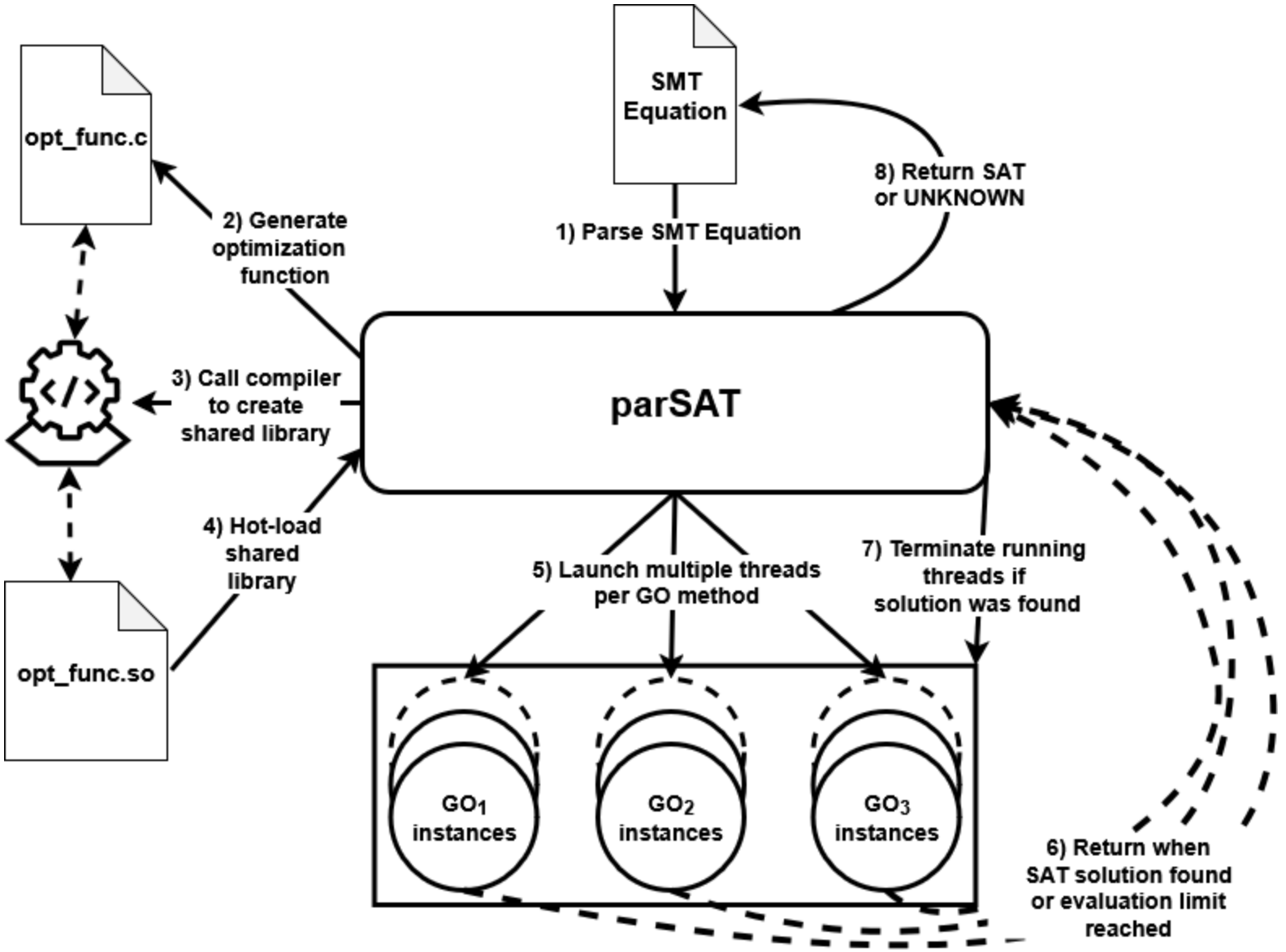
parSAT‘s execution flow



parSAT‘s execution flow



parSAT‘s execution flow



Incompleteness of parSAT

- If ***parSAT*** finds a global minimum at zero, these coordinates represent a satisfiable solution to the initial SMT equation
- If ***parSAT*** reaches an evaluation limit (maximal number of calling the evaluation function exceeded) it is unknown whether the optimization function has a global minimum at zero (satisfiable) or not (unsatisfiable)
- The generated optimization function may not be smooth and therefore not differentiable
- Only derivative-free GO methods can be used
- Only when all possible inputs were evaluated and no solution was found, ***parSAT*** could safely determine UNSAT

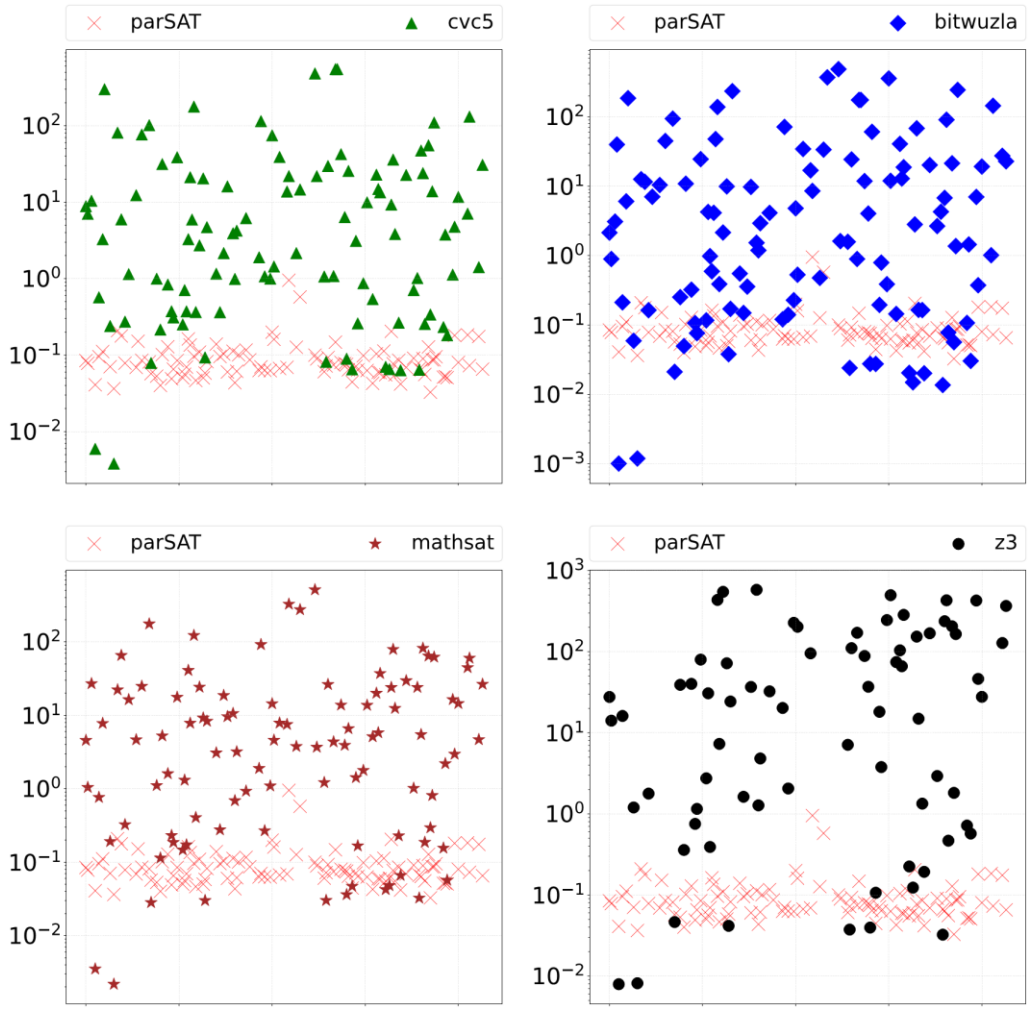
Distribution of fastest GO methods when evaluating the Griggio Benchmark

	BH	CRS2	ISRES
parsat $BH_1, CRS2_1, ISRES_1$	46.6%	27.9%	25.5%
parsat $BH_2, CRS2_2, ISRES_2$	59.1%	25.2%	15.7%
parsat $BH_3, CRS2_3, ISRES_3$	71.6%	17.3%	11.1%
parsat $BH_4, CRS2_4, ISRES_4$	74.0%	17.3%	8.7%
parsat $BH_5, CRS2_5, ISRES_5$	77.5%	15.5%	7.0%
parsat $BH_6, CRS2_6, ISRES_6$	83.6%	10.0%	6.4%
parsat $BH_7, CRS2_7, ISRES_7$	87.0%	7.7%	5.3%
parsat $BH_8, CRS2_8, ISRES_8$	87.5%	7.0%	5.5%
parsat $BH_9, CRS2_9, ISRES_9$	89.6%	5.1%	5.3%
parsat $BH_{10}, CRS2_{10}, ISRES_{10}$	91.2%	4.1%	4.7%

Full comparison of parSAT with state-of-the-art SMT solvers (Griggio Benchmark)

	SAT	*UNSAT*	timeout / evaluation limit	errors	average *SAT* runtime
$parSAT_{best} BH_{14}, CRS2_3, ISRES_3$	102	0	112	0	0.1
`bitwuzla`	108	76	30	0	32.71
`cvc5`	104	76	30	4	34.01
`MathSAT`	100	69	45	0	25.86
`Z3`	75	56	54	29	88.7

Solving time of parSAT and state-of-the-art SMT solvers for SAT equations (Griggio Benchmark)



Introduction to the 2019-Guedemann Benchmark

- Re-Implementation of natural logarithm (ln‘) and exeponential (exp‘) function

- The 2019-Guedemann Benchmark includes SMT equations to verify that the following properties hold for the re-implemented algorithms

$$\forall x \in \mathbb{R}: \quad x > 0 \implies \exp'(\ln' x) \approx x$$
$$\forall x \in \mathbb{R}: \quad x > 0 \implies \ln'(\exp' x) \approx x$$
$$\forall x, y \in \mathbb{R}: \quad x, y \in [0,1] \implies \exp'(x+y) \approx \exp'(x) \cdot \exp'(y)$$
$$\forall x \in \mathbb{R}: \quad x, y > 0 \implies \ln'(x \cdot y) \approx \ln'(x) \cdot \ln'(y)$$
$$\forall x \in \mathbb{R}: \quad x, y \in [0,1] \implies img(x^y) = img(\exp'(y \cdot \ln'(x))) \approx [0,1]$$

Full comparison of parSAT with state-of-the-art SMT solvers (2019-Guedemann Benchmark)

	SAT	*UNSAT*	timeout / evaluation limit	errors	average *SAT* runtime
$parSAT_{best}BH_{14}, CRS2_3, ISRES_3$	10	0	3	0	0.1
`bitwuzla`	10	2	1	0	37.88
`cvc5`	4	1	8	0	180.09
`MathSAT`	4	1	8	0	22.8
`Z3`	0	0	4	9	-

Potential effect of combining parSAT with bitwuzla (Griggio Benchmark)

	SAT	*UNSAT*	timeout / evaluation limit	errors	average *SAT* runtime
$parSAT_{best} BH_{14}, CRS2_3, ISRES_3$	102	0	112	0	0.1
`bitwuzla`	108	76	30	0	32.71
$parSAT_{best}$ + `bitwuzla`	113	76	25	0	12.75

Conclusion and Outlook

- ***parSAT*** presents a considerable complementary alternative to current state-of-the-art SMT solvers
- ***parSAT*** was able to handle a variety of SMT equations from commonly used, representative benchmarks
- The modular architecture of ***parSAT*** enables integration of further GO algorithms
- Due to the necessary evaluation limit for the optimization function, ***parSAT*** is a semi-decision procedure that underapproximates the search space and cannot prove *UNSAT*
- ***parSAT*** is limited to only find solutions consisting of finite FP values

- Extend ***parSAT*** to utilize other platforms for running concurrent GO methods such as GPUs or (cloud) computing clusters
- Enable information sharing between ***parSAT*** with SMT solvers in a complementary setting
- Investigate ***parSAT'***s potential in the problem domain of SMT model counting