

Learning to Solve SMT Formulas

Mislav Balunović, **Pavol Bielik**, Martin Vechev

Department of Computer Science

ETH zürich

SMT Formula

Does there exist a valid assignment to **b**, **x**, **y**?

$$\varphi = (b \vee -x^2 \geq 2.3y \vee \sin(x)^3 = \cos(\log(y) \cdot x)) \wedge (-b \vee y < -34.4 \vee \exp(y) > y/x)$$

where $b \in \{\text{true}, \text{false}\}$, $x, y \in \mathbb{R}$

SMT

Theories

Booleans, Reals, Integers, Arrays, BitVectors, Strings, ...

SAT

SMT Solvers

Does there exist a valid assignment to **b**, **x**, **y**?

$$\varphi = (b \vee -x^2 \geq 2.3y \vee \sin(x)^3 = \cos(\log(y) \cdot x)) \wedge (-b \vee y < -34.4 \vee \exp(y) > y/x)$$

where $b \in \{\text{true}, \text{false}\}$, $x, y \in \mathbb{R}$

Find an assignment to all free variables in φ such that φ evaluates to true

b = true
x = 14.32
y = -37.2

First-order
logic formula

φ

→

SMT Solver

→

SAT + model

UNSAT + unsat core

SMT Solvers

Applications

Planning

Software & Hardware Verification

Type Inference

Scheduling

Neural Networks Verification

Symbolic Execution

Graph Problems

Program Synthesis

Static Program Analysis

**Find an assignment to all free variables
in φ such that φ evaluates to true**

First-order
logic formula

φ



SMT Solver

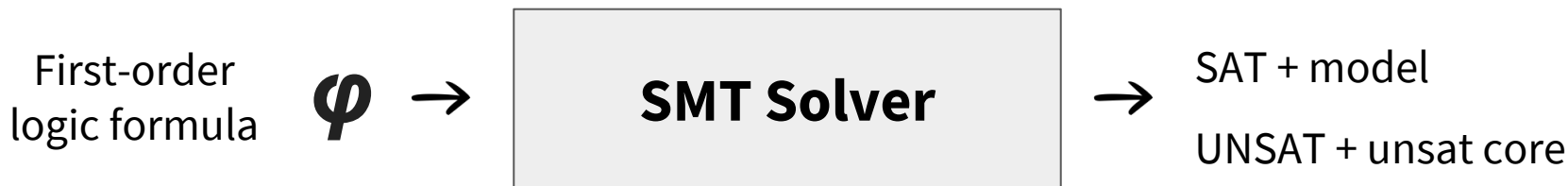


SAT + model

UNSAT + unsat core

Solving SMT Formulas is Hard

Theory	Complexity	
Quantifier Free Booleans (SAT)	NP-Complete	$O(n^k)$
Linear Real Arithmetic	2-EXPTIME	$O(2^{2^{n^k}})$
Linear Integer Arithmetic	3-EXPTIME	$O(2^{2^{2^{n^k}}})$
Non-linear Integer Arithmetic	undecidable	



Solving SMT Formulas is Hard

Theory	Complexity	
Quantifier Free Booleans (SAT)	NP-Complete	$O(n^k)$
Linear Real Arithmetic	2-EXPTIME	$O(2^{2^{n^k}})$
Linear Integer Arithmetic	3-EXPTIME	$O(2^{2^{2^{n^k}}})$
Non-linear Integer Arithmetic	undecidable	



Solving SMT Formulas is Hard

State-of-the-art SMT Solvers

- ✗ Easily perform badly on new problems
- ✗ Require expert knowledge to fix



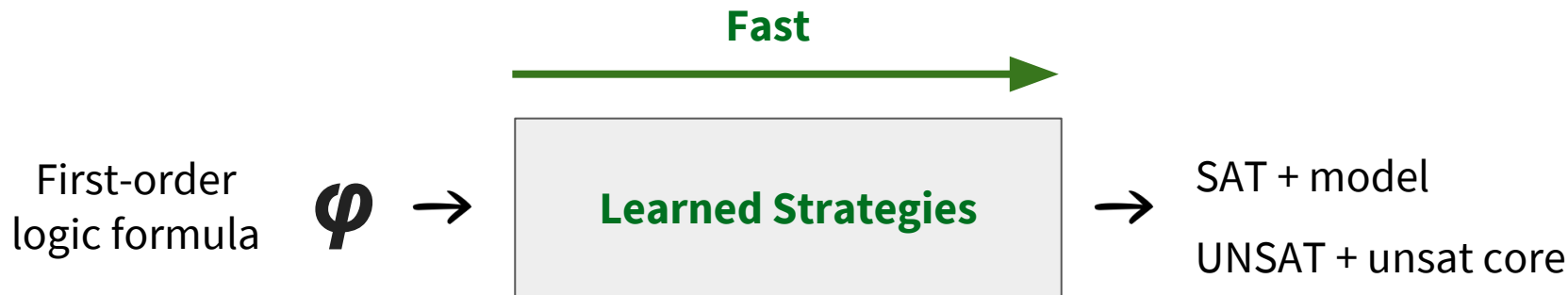
Learning to Solve SMT Formulas

State-of-the-art SMT Solvers

- ✗ Easily perform badly on new problems
- ✗ Require expert knowledge to fix

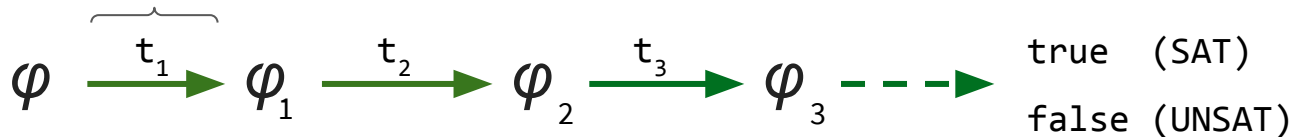
Our Work: *FastSMT*

- ✓ Learn fast strategies
- ✓ No prior knowledge



SMT Formula Solving

apply transformation
(tactic)



Action
Space

Constant Folding

$$x + 0 \longrightarrow x$$

Normalize Bounds

$$k \leq x \xrightarrow{x' = x - k} 0 \leq x'$$

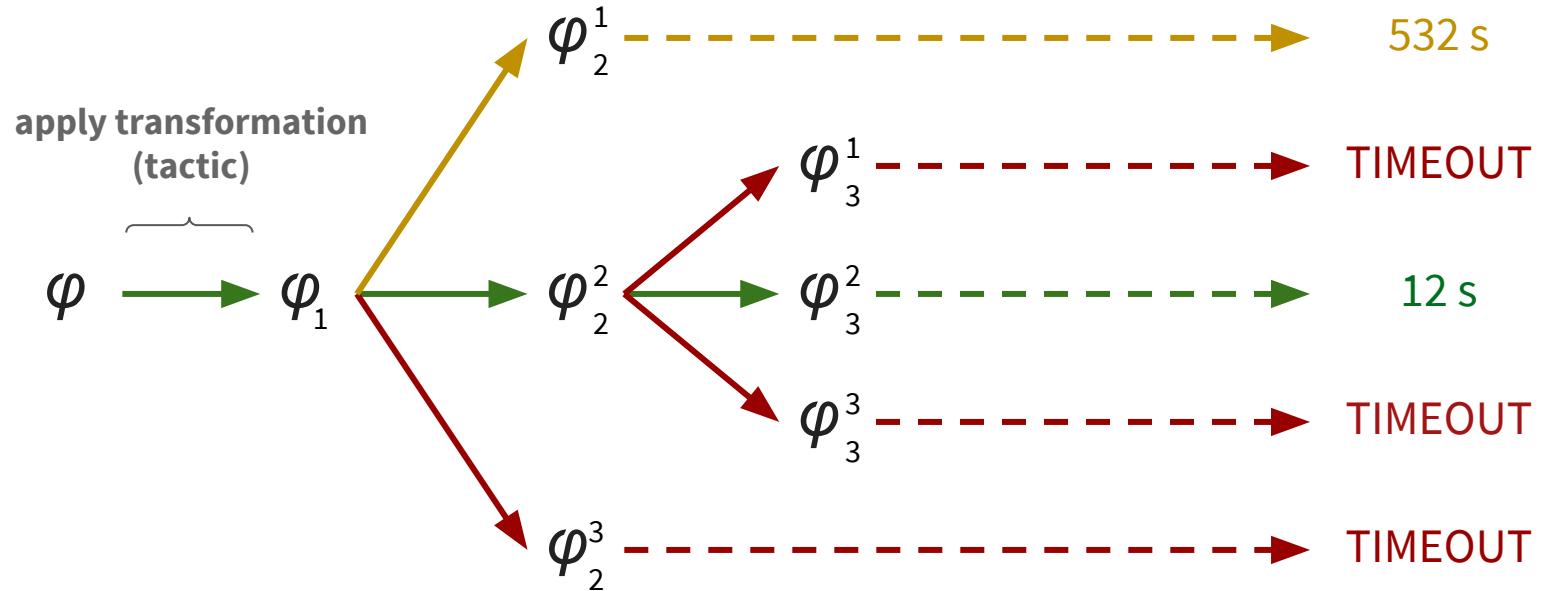
Bit Blasting

$$x = 5 \xrightarrow{0101_2} \begin{matrix} x_3 = 0 \wedge x_2 = 1 \wedge \\ x_1 = 0 \wedge x_0 = 1 \end{matrix}$$

Decision Procedure

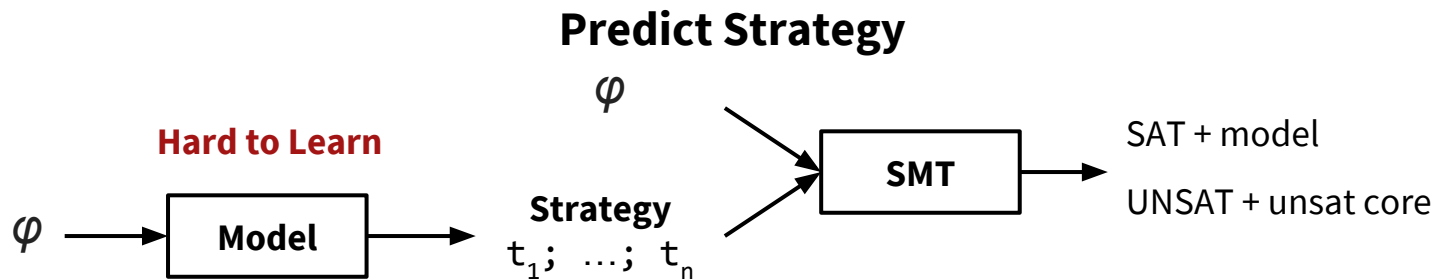
Reals, Integers, BitVectors, ...

SMT Formula Solving

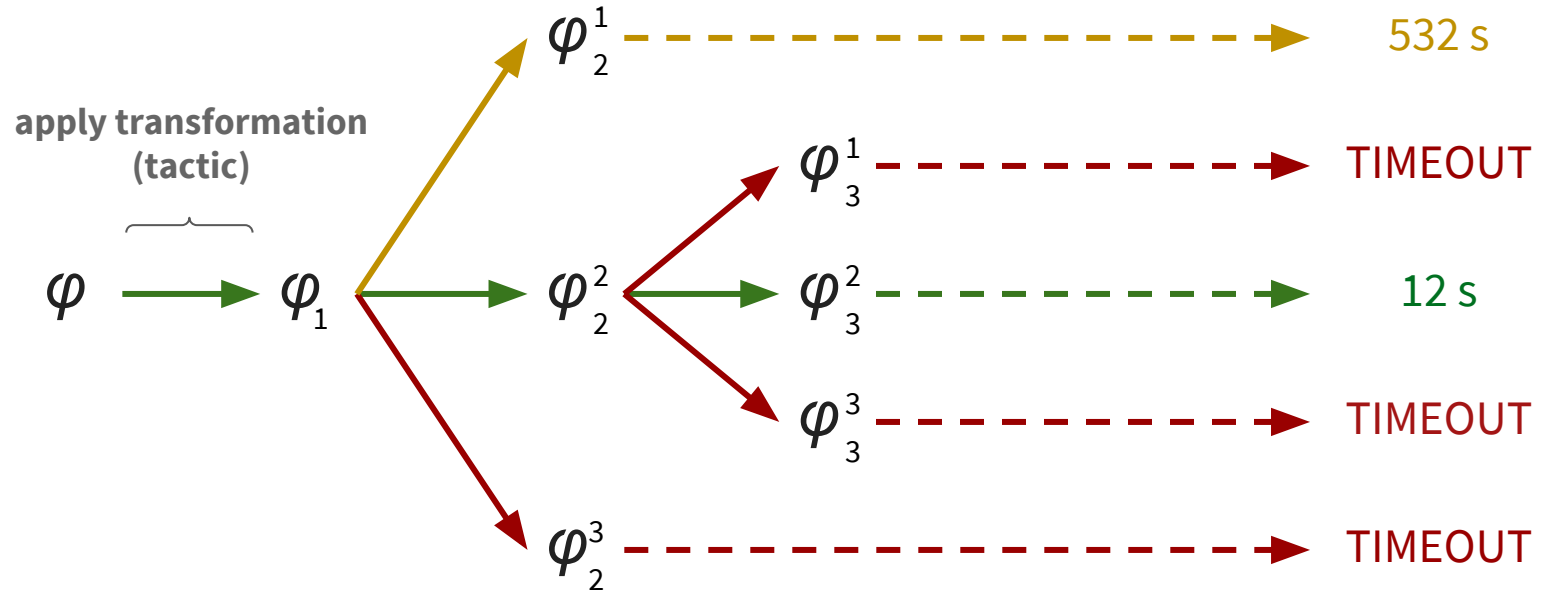


Handcrafted strategies determine which path to take

Learning to Solve Formula

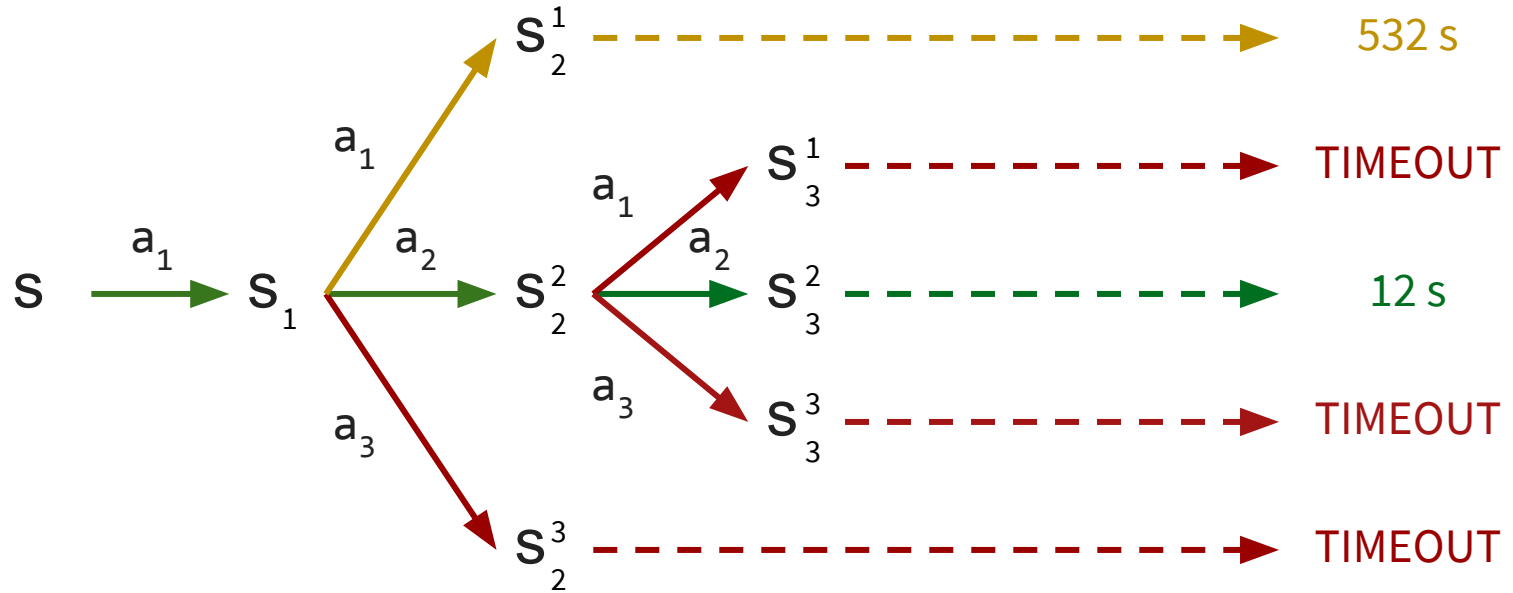


Learning to Solve Formula



Handcrafted strategies determine which path to take

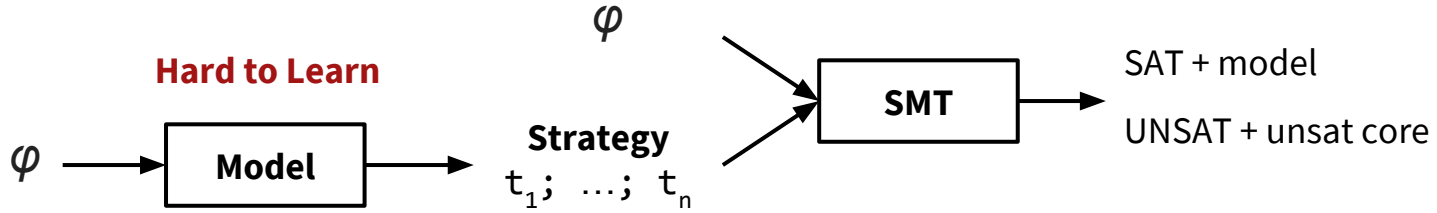
Learning to Solve Formula



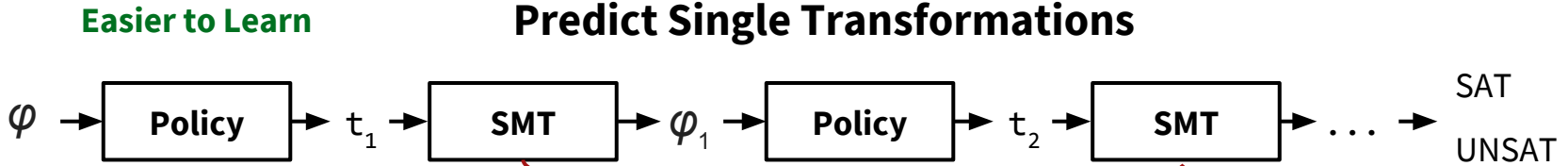
Learn path with lowest runtime

SMT Formula Solving

Predict Strategy



Predict Single Transformations



Runtime Overhead

Internal SMT State Lost

Model Not Available

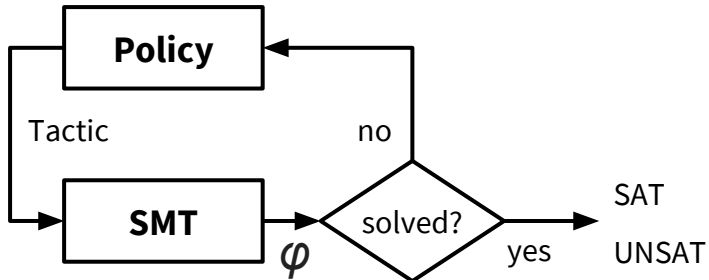
Bit Blasting

$$x = 5 \longrightarrow 0101_2 \quad \begin{matrix} x_3 = 0 \wedge x_2 = 1 \wedge \\ x_1 = 0 \wedge x_0 = 1 \end{matrix}$$

Learning to Solve SMT Formulas

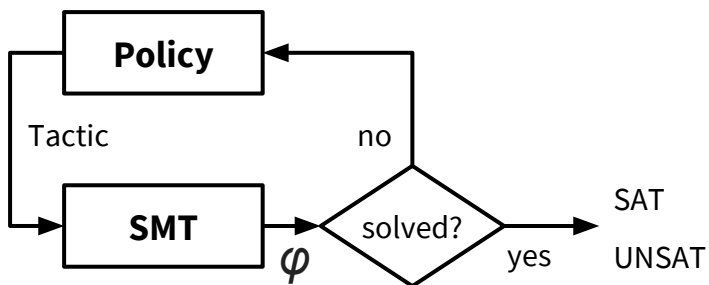
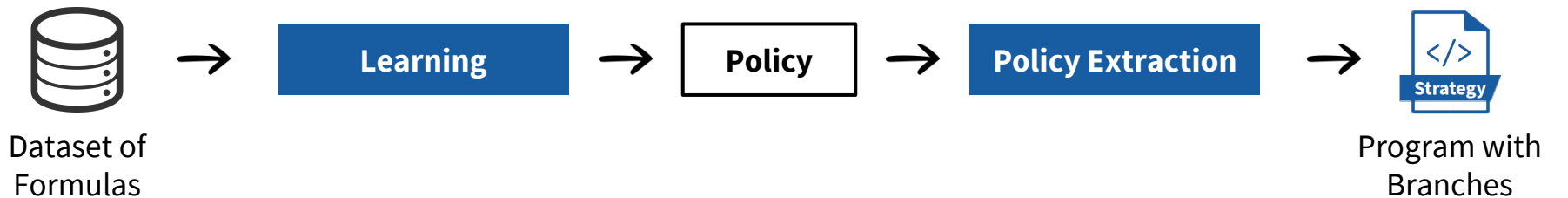


Dataset of
Formulas



**Learn a policy to
select next tactic**

Learning to Solve SMT Formulas



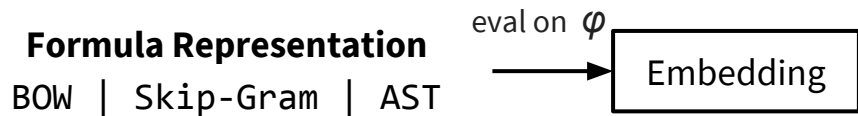
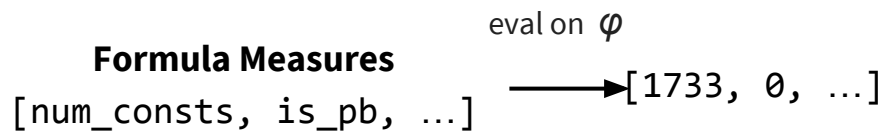
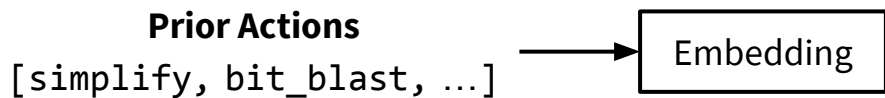
Learn a policy to select next tactic



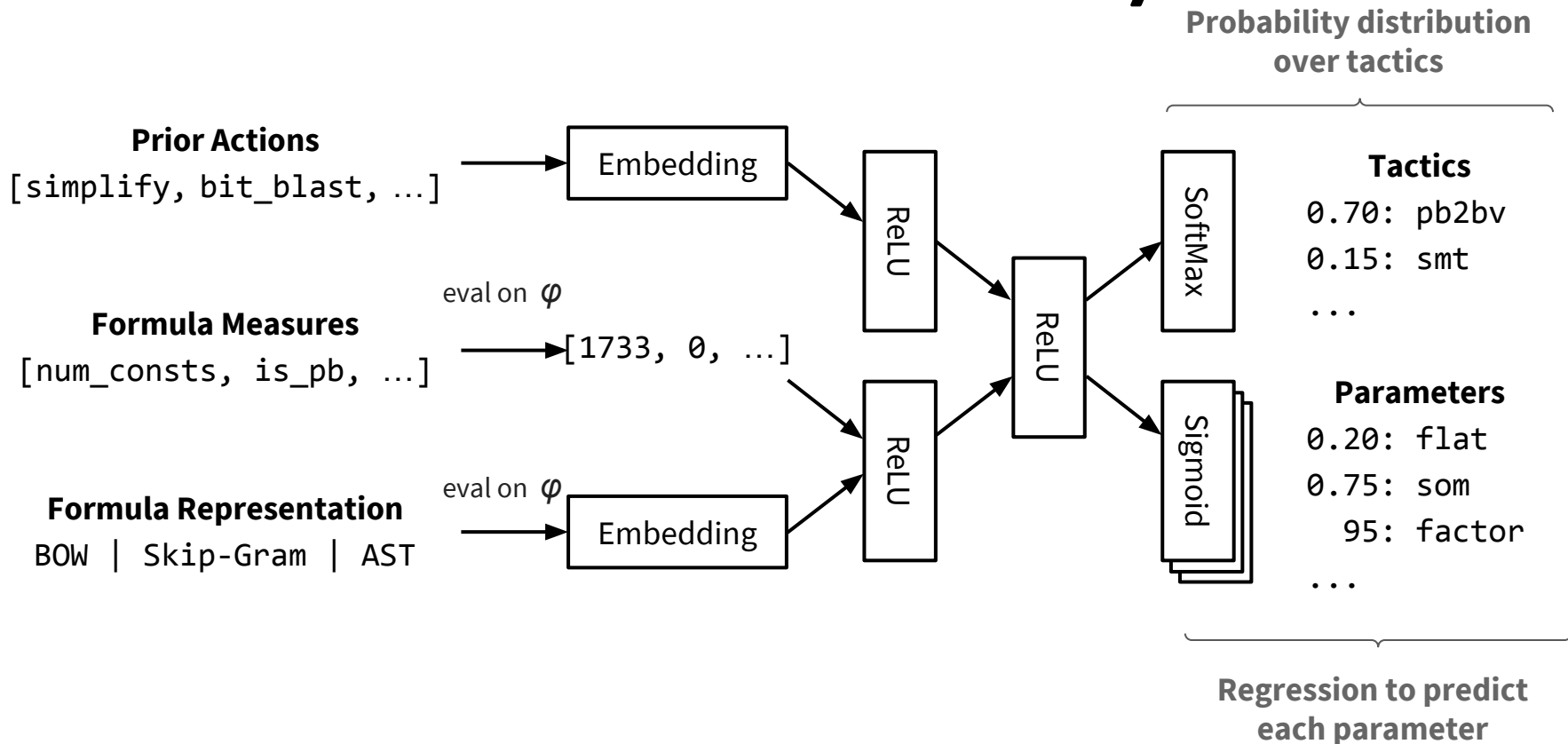
- ✓ No runtime overhead
- ✓ Integration with existing SMT Solvers

Use the learned policy to synthesize a Strategy program


Neural Network Policy



Neural Network Policy

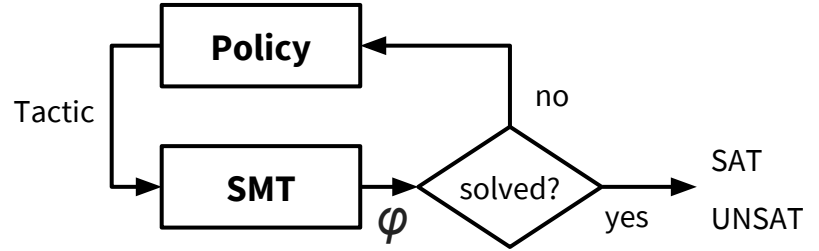
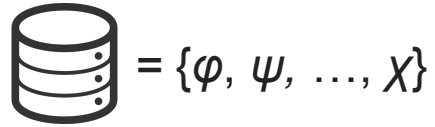


Training

 = $\{\varphi, \psi, \dots, \chi\}$

Dataset of Formulas

Training

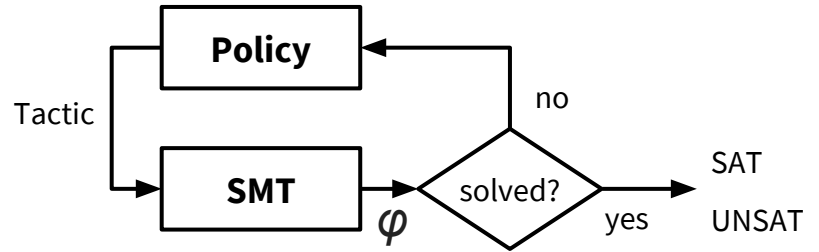
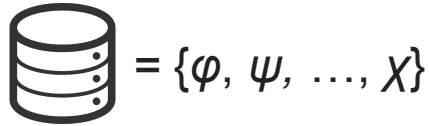


Dataset of Formulas



Sample Policy

Training



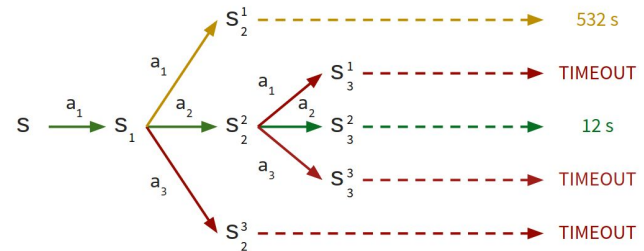
Dataset of Formulas



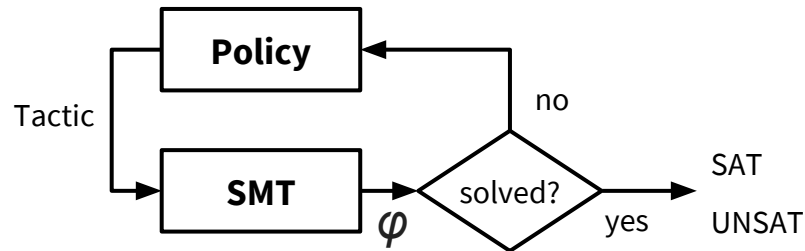
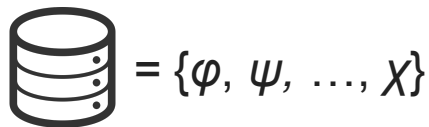
Sample Policy



Training Dataset



Training



Dataset of Formulas



Sample Policy

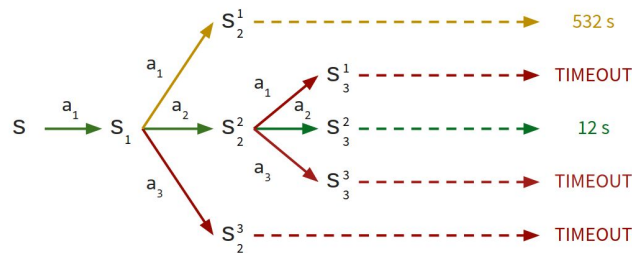


Retrain Policy

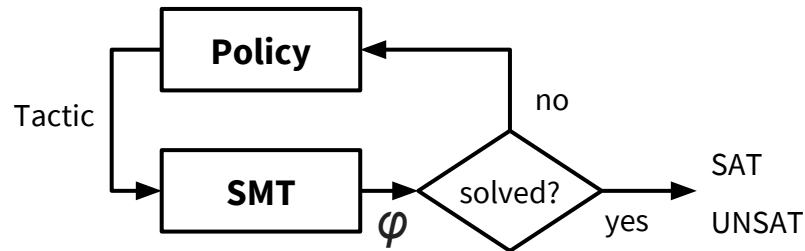
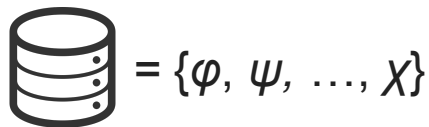


Training Dataset

$$\mathcal{L} = \text{weighted average of cross-entropy loss} + \text{mean-square-error parameters}$$



Training



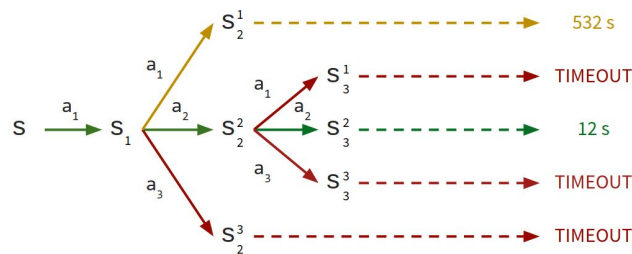
Evaluate

Retrain Policy

Sample Policy

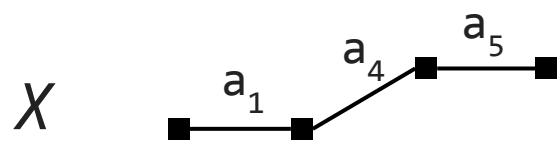
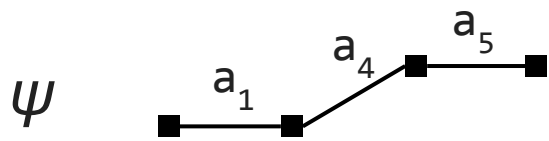
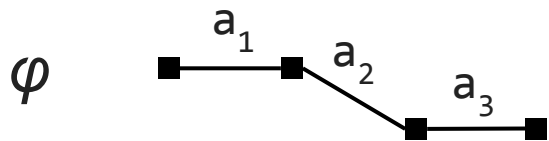
Training Dataset

$$\mathcal{L} = \text{weighted average of cross-entropy loss} + \text{mean-square-error parameters}$$



Learning

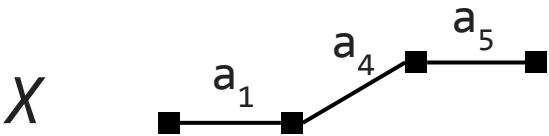
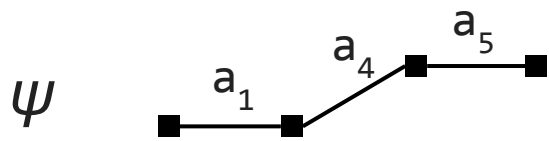
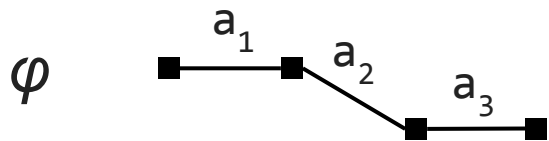
$$\pi(a \mid s)$$



Sequential Strategies

Learning

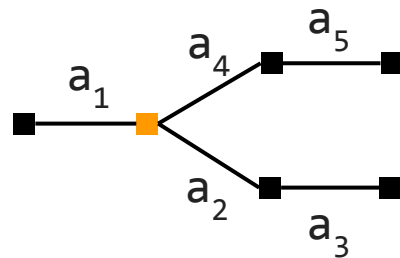
$$\pi(a \mid s)$$



Sequential Strategies

Policy Extraction

if expr then a_2 else a_4



$\{\varphi, \psi, \dots, \chi\}$

Strategy with Branches

Evaluation

state-of-the-art SMT Solver

Z3

Academic Benchmarks

Industrial Benchmarks

leipzig

core

hycomp

AProVE

Sage2



Learning

Learning

Learning

Learning

Learning

+

+

+

+

+

Policy Extraction

Policy Extraction

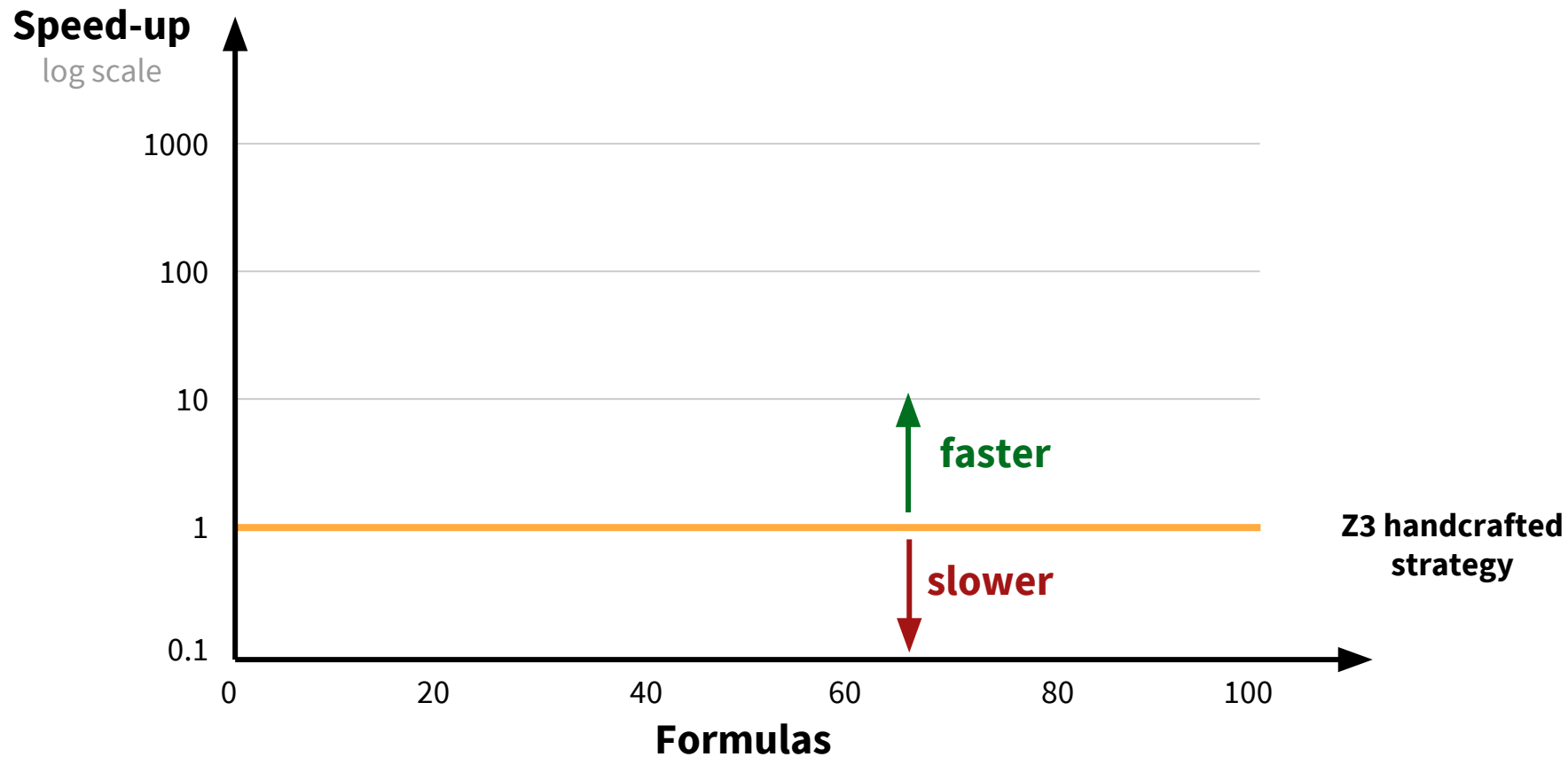
Policy Extraction

Policy Extraction

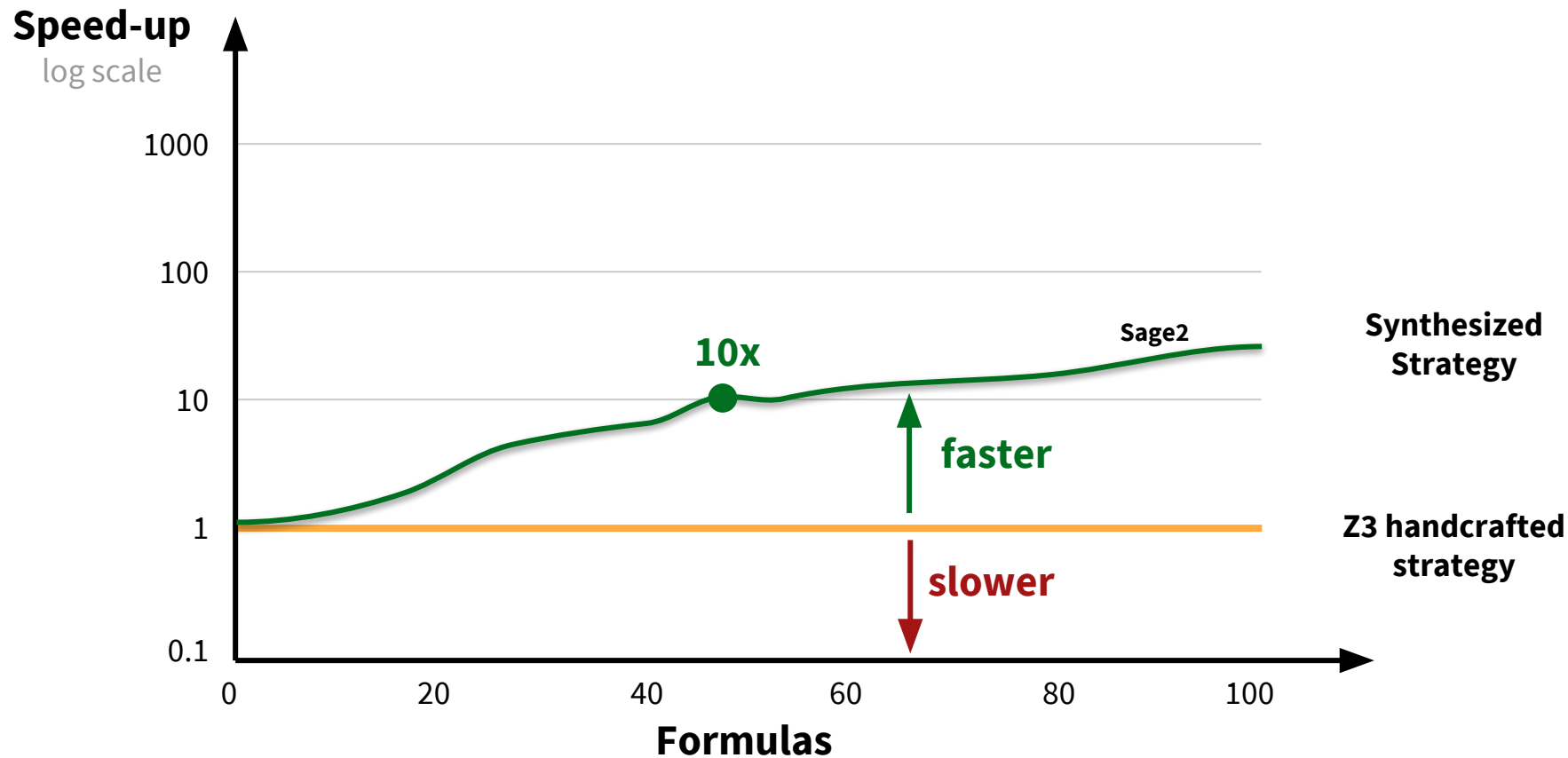
Policy Extraction



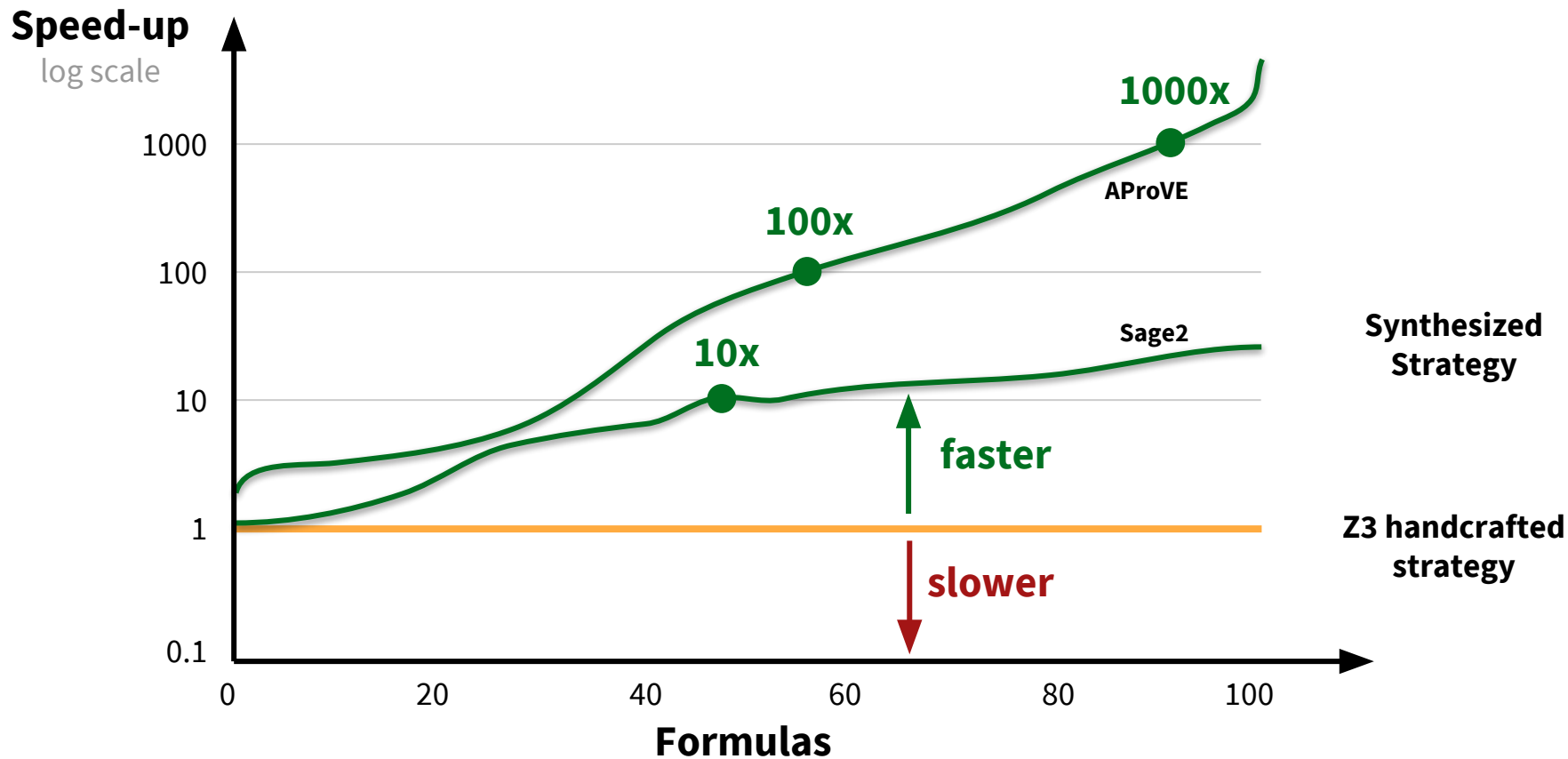
Speed-up over Z3



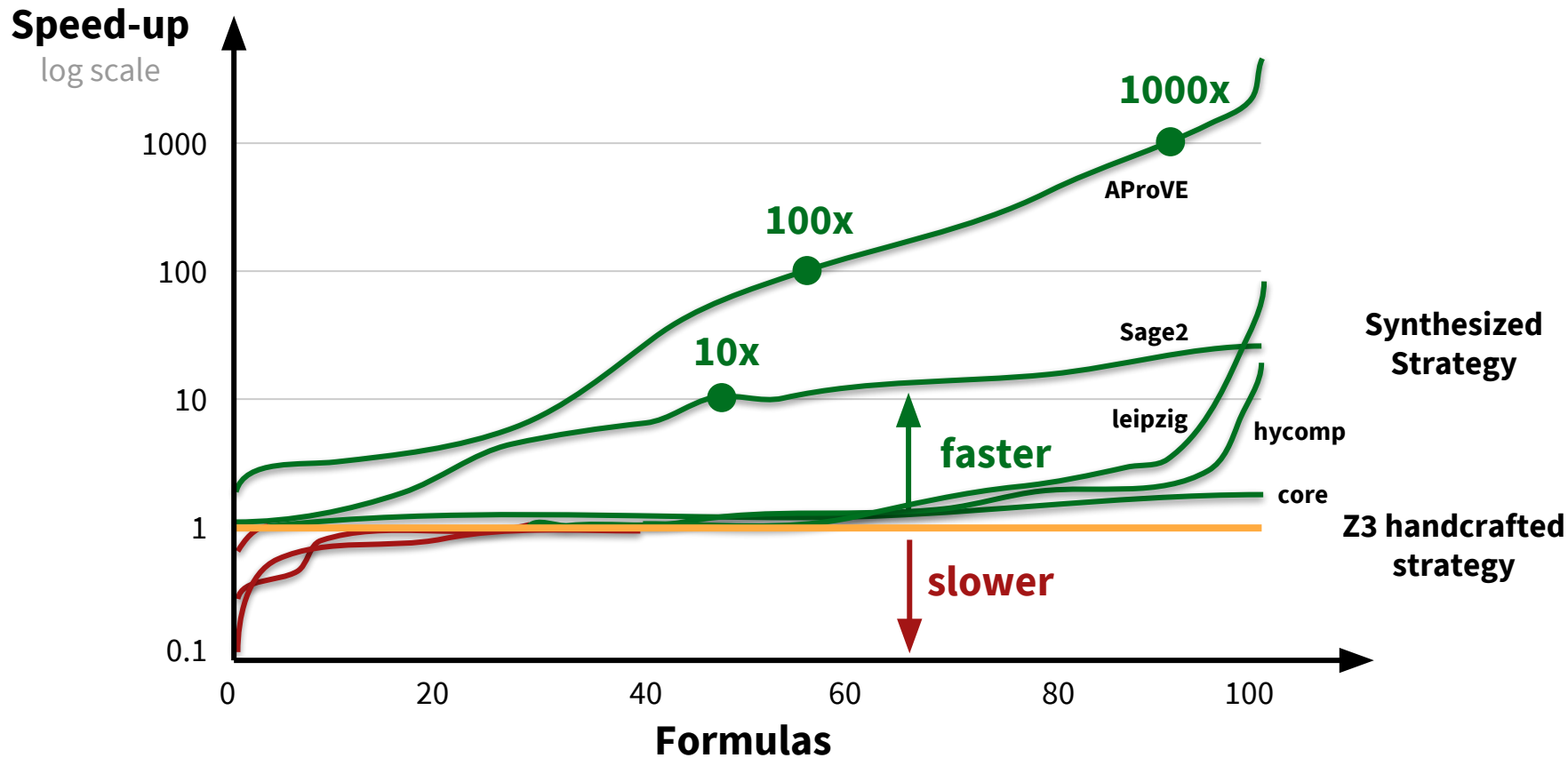
Speed-up over Z3



Speed-up over Z3



Speed-up over Z3



Learning to Solve SMT Formulas

<http://fastsmt.ethz.ch/>

