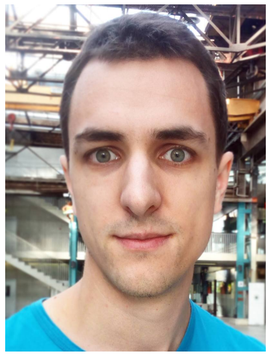
Validating SMT Solvers via Semantic Fusion

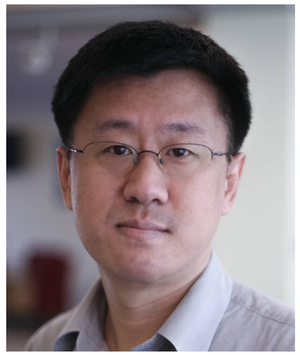
Dominik Winterer*
ETH Zurich, Switzerland

Chengyu Zhang*
East China Normal University, China

Zhendong Su
ETH Zurich, Switzerland

(*Equal contributions)





PLDI ’20, June 15–20, 2020, Online

SMT Problem

$$\varphi : x > 0 \wedge x < 0$$

SMT Problem

$$\varphi : x > 0 \wedge x < 0$$

UNSAT

SMT Problem

$$\varphi : x > 0 \wedge x < 1$$

SMT Problem

$$\varphi : x > 0 \wedge x < 1$$

SAT

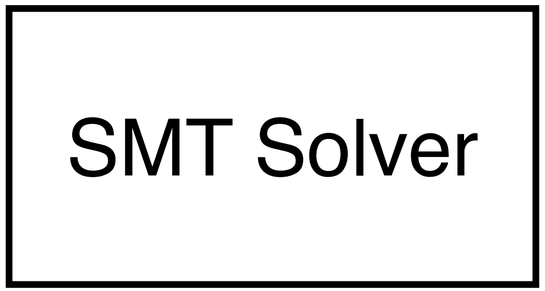
SMT Problem

$$\varphi : x > 0 \wedge x < 1$$

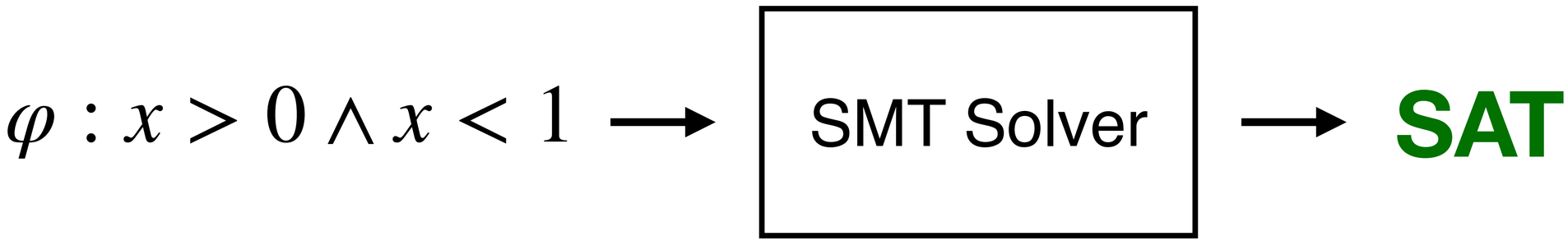
SAT

$$x = 0.5$$

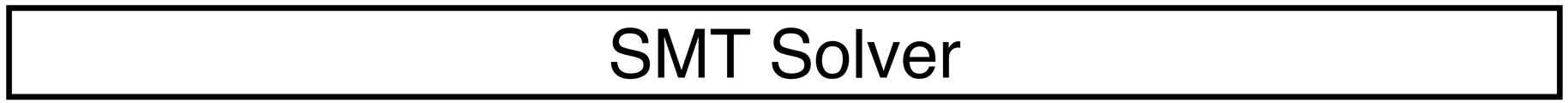
SMT Solver

$\varphi : x > 0 \wedge x < 1 \longrightarrow$ 

SMT Solver

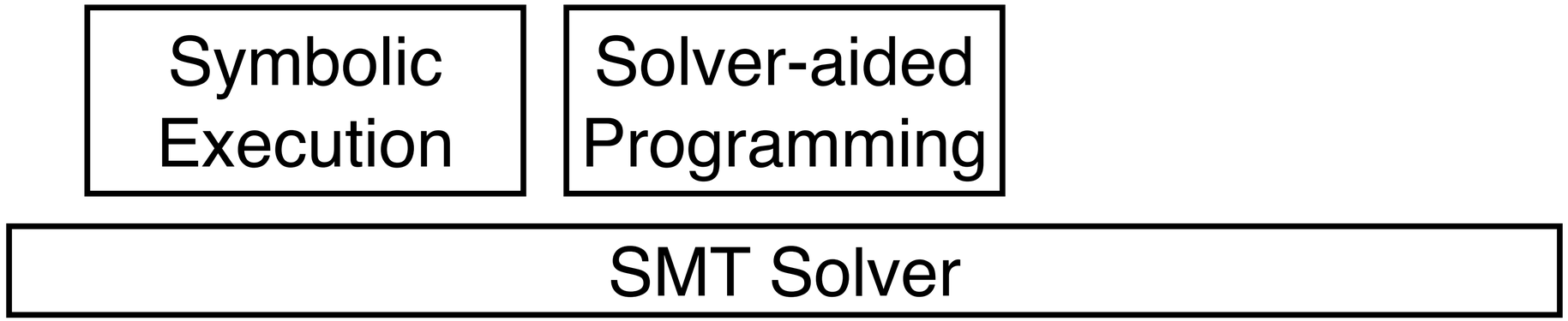


SMT Solver

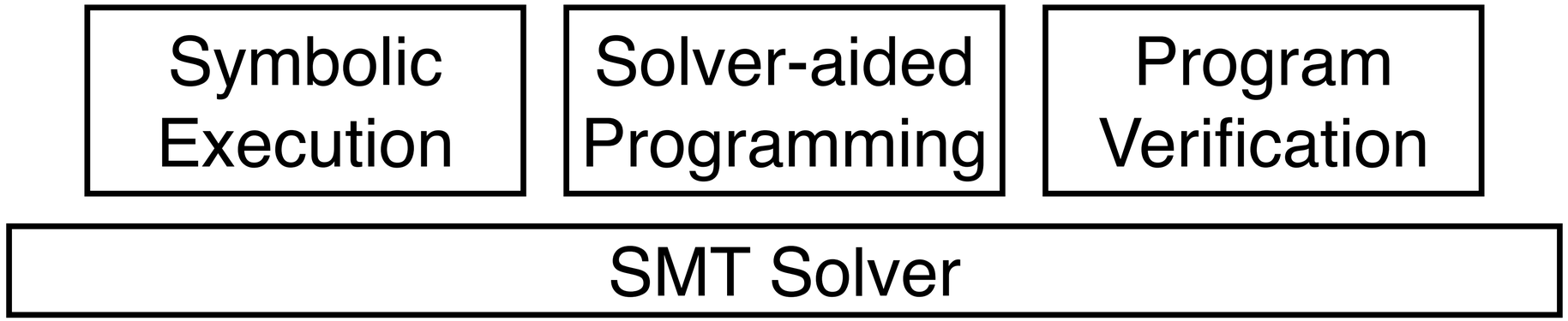


SMT Solver

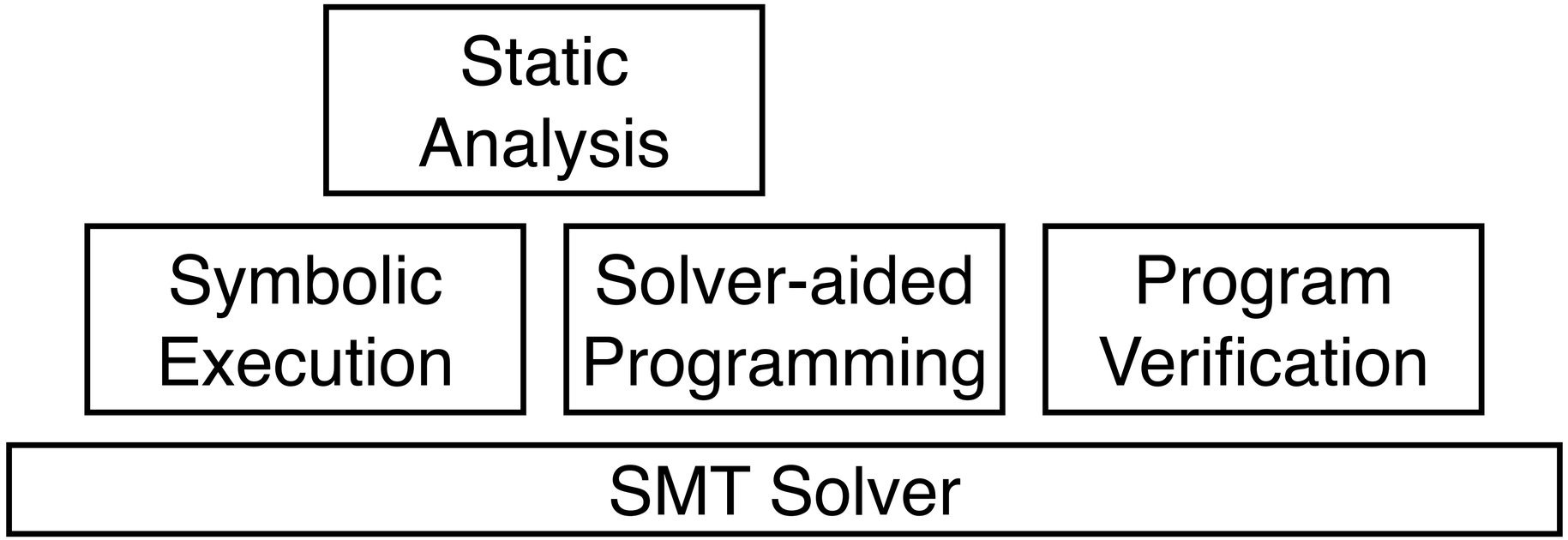
SMT Solver



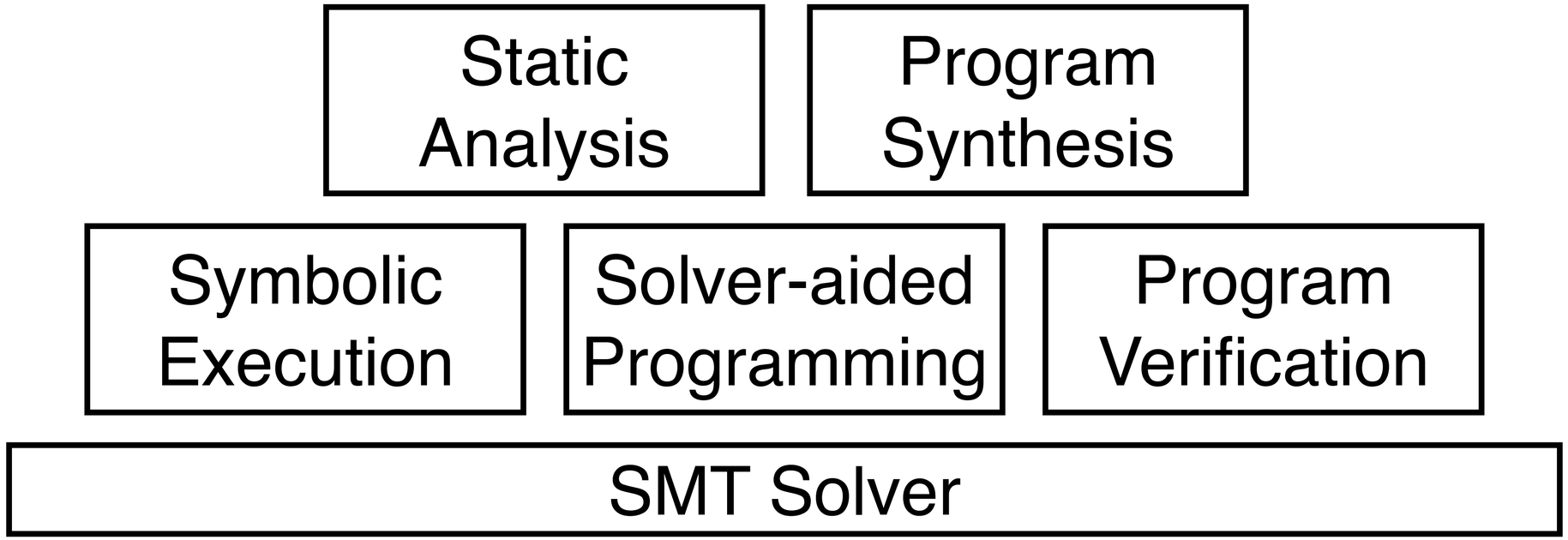
SMT Solver


Symbolic Execution
Solver-aided Programming
Program Verification
SMT Solver

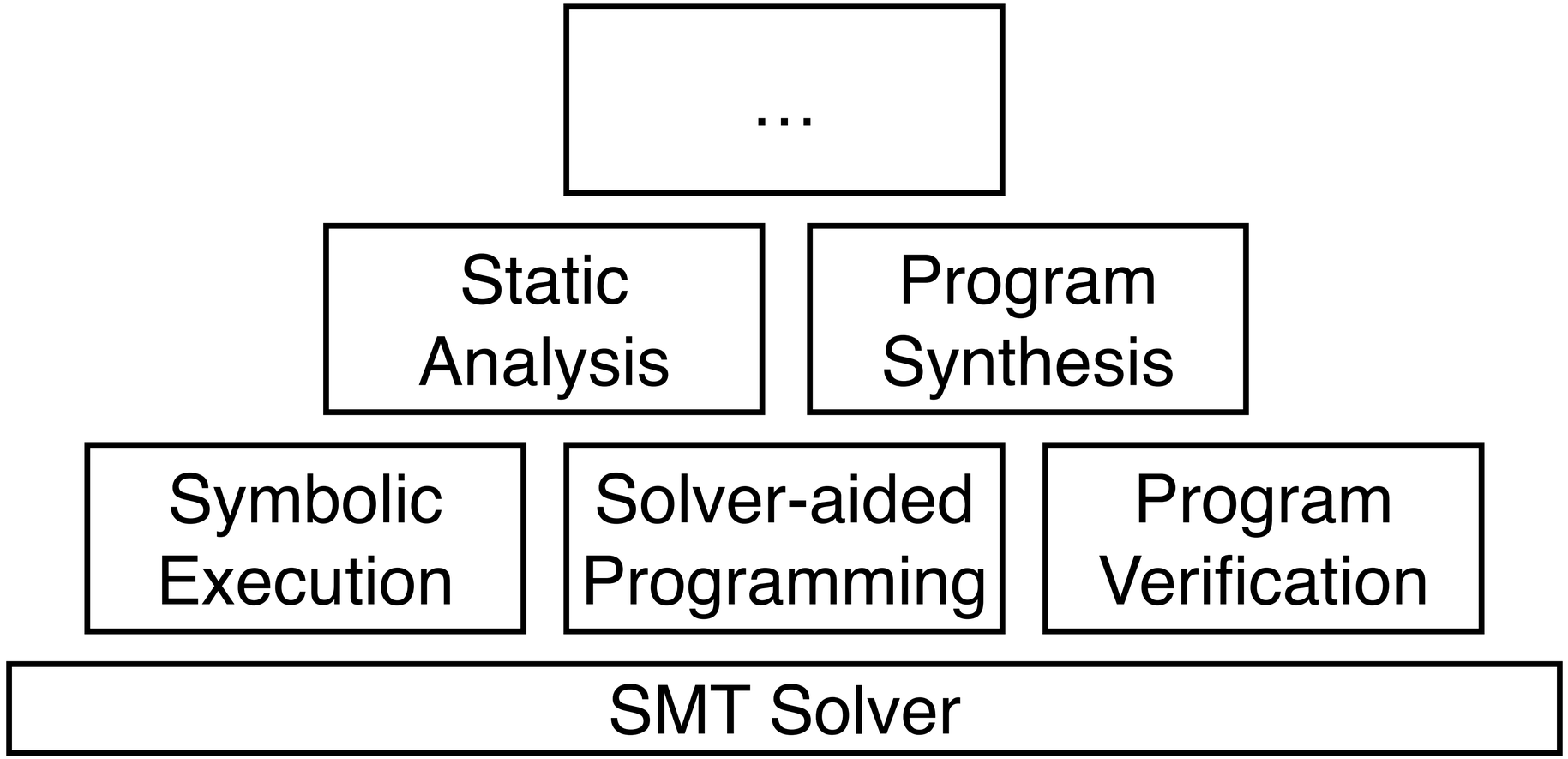
SMT Solver


Static
Analysis
Symbolic
Execution
Solver-aided
Programming
Program
Verification
SMT Solver

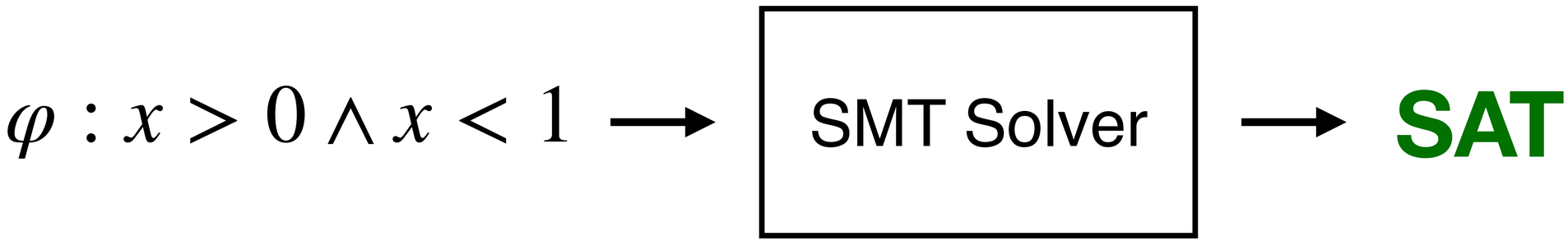
SMT Solver


Static Analysis
Program Synthesis
Symbolic Execution
Solver-aided Programming
Program Verification
SMT Solver

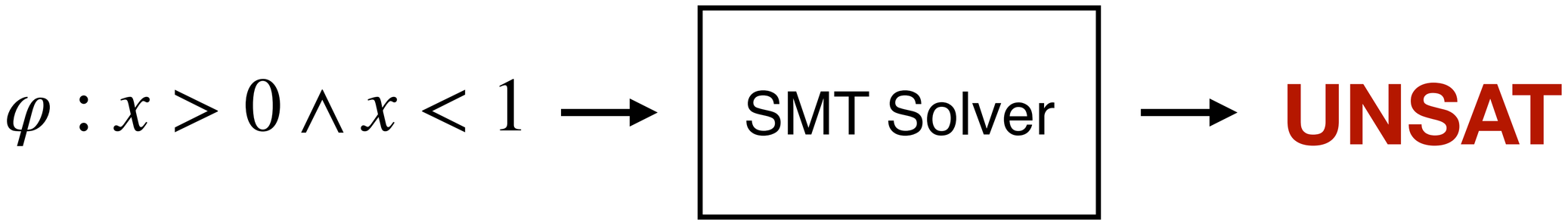
SMT Solver


...
Static Analysis
Program Synthesis
Symbolic Execution
Solver-aided Programming
Program Verification
SMT Solver

SMT Solver



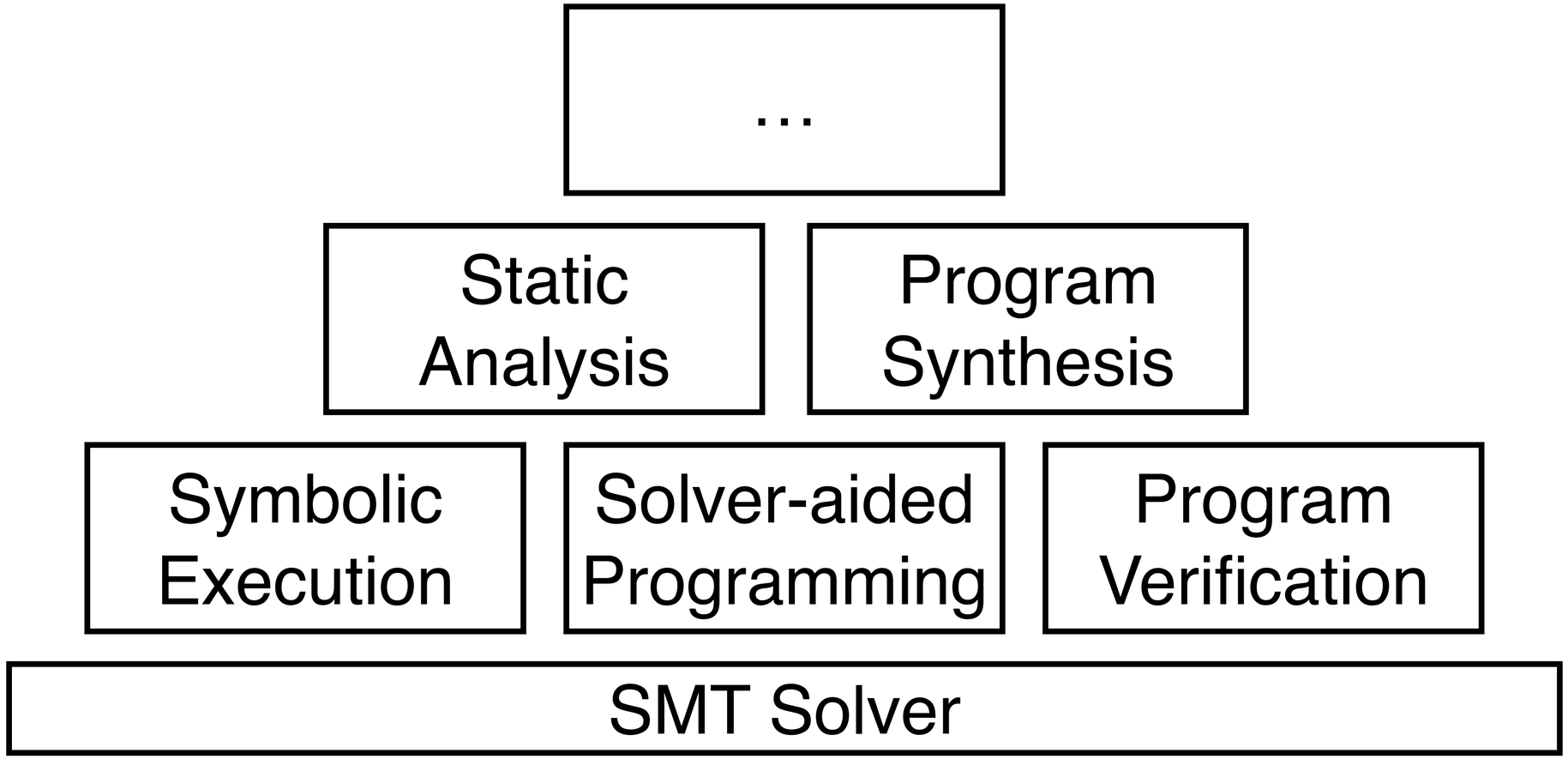
SMT Solver



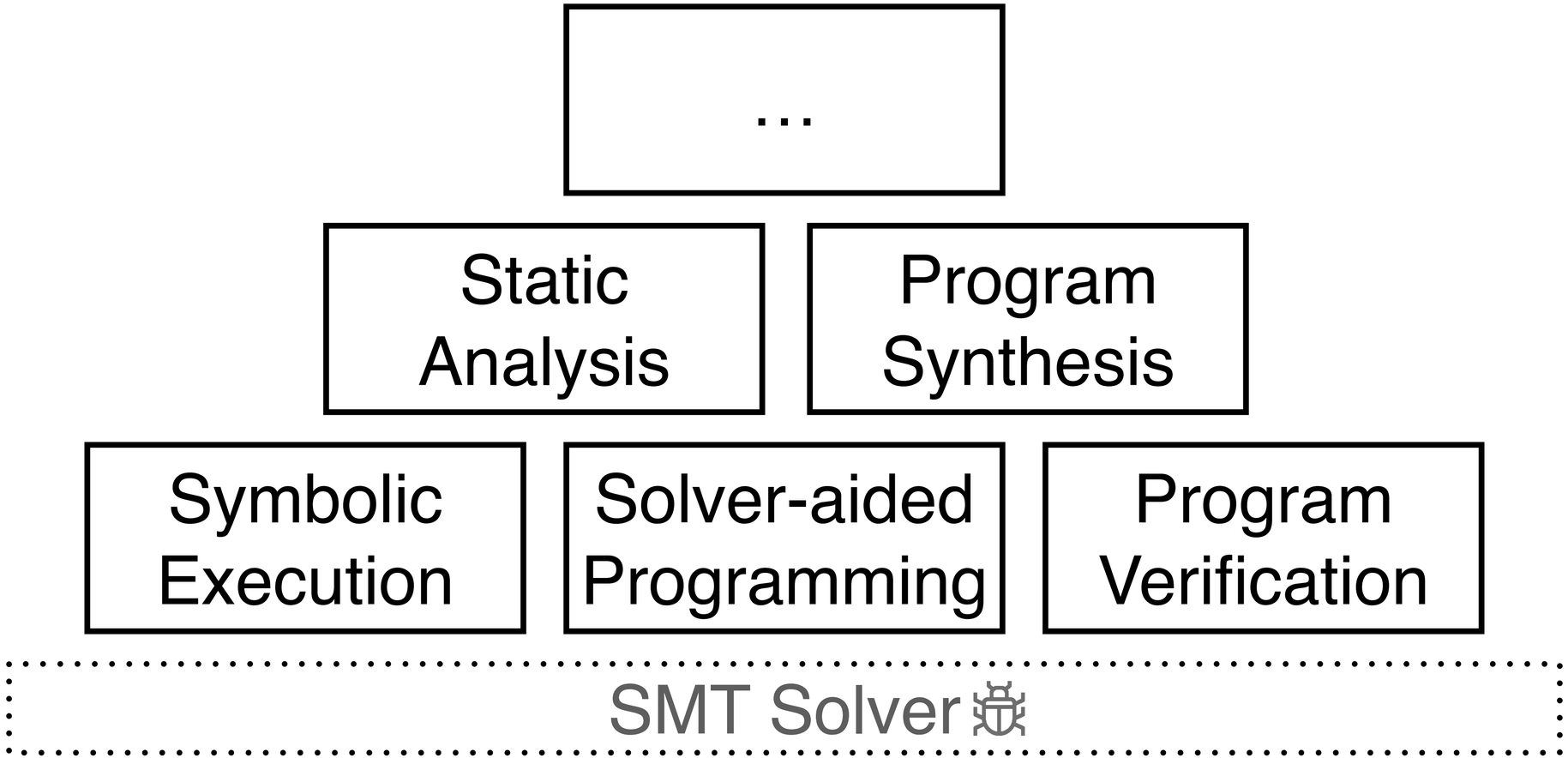
SMT Solver

$\varphi : x > 0 \wedge x < 1$ → SMT Solver → **UNSAT** 

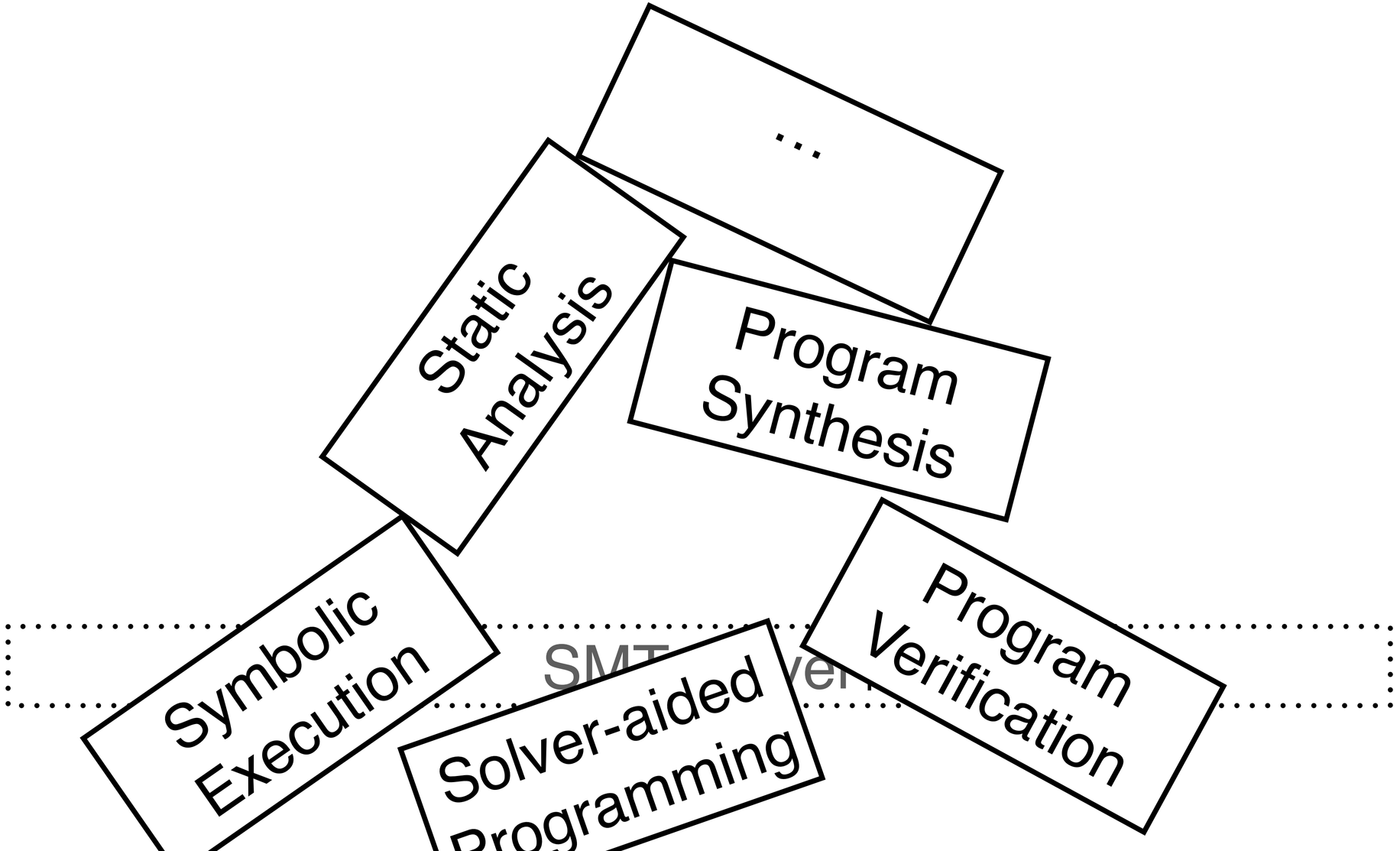
SMT Solver


...
Static Analysis
Program Synthesis
Symbolic Execution
Solver-aided Programming
Program Verification
SMT Solver

SMT Solver


...
Static Analysis
Program Synthesis
Symbolic Execution
Solver-aided Programming
Program Verification
SMT Solver

SMT Solver


...
Static Analysis
Program Synthesis
Symbolic Execution
Solver-aided Programming
Program Verification

Testing SMT solvers is challenging

Testing SMT solvers is challenging

- How to generate test formulas?

Testing SMT solvers is challenging

- How to generate test formulas?
- How to obtain the test oracles?

Testing SMT solvers is challenging

```
(declare-fun a () Real)
(declare-fun p () Real)
(declare-fun b () Real)
(declare-fun c () Real)
(declare-fun d () Real)
(declare-fun k () Real)
(declare-fun e () Real)
(declare-fun q () Real)
(assert (or
            (not (exists ((f Real))
                (=>
                (and
                    (>= c 0)
                    (> (/ b q) 2)
                    (>= (/ p q) 1)
                    (<= d 12)
                    (>= (/ p q) (- (* 1 k)))
                    (<= (/ p q) (+ 10 k)))
                (<= (+ (* (- 2) (- a e)) d) 12))))
            (exists ((o Real))
                (forall ((g Real))
                    (exists ((h Real))
                        (and
                            (or
                                (>= g (* (- 3) h) 57)
                                (and (> (* 79 o) 8 (+ g h) 0) (= h 0))
                                (< 0 (+ g h) 0))
                            (> (+ (* (- 97) o) g) 0)))))))
(assert (= a (+ c e)(* d q)(/ b q)))
(assert (= q (/ b k)))
(check-sat)
(get-model)
```

Testing SMT solvers is challenging

- How to generate test formulas?
- How to obtain the test oracles?

Testing SMT solvers is challenging

- How to generate test formulas?
- How to obtain the test oracles?
- It is challenging to find bugs.

Semantic Fusion

- Fusing test formulas while preserving satisfiability
- Finding bugs in two state-of-the-art SMT solvers

Semantic Fusion

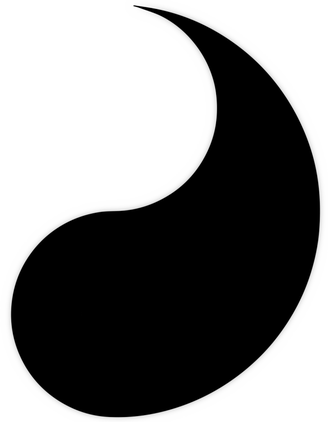
- Fusing test formulas while preserving satisfiability
- Finding bugs in two state-of-the-art SMT solvers

46 Bugs Confirmed, 42 Bugs Fixed
in Z3 and CVC4 default mode

Semantic Fusion

φ_1

φ_2



Semantic Fusion

$$\varphi_{concat}$$

Semantic Fusion

$$\varphi_{fused}$$

Semantic Fusion

$$\varphi_1 = x > 0 \wedge x > 1 \quad \textbf{SAT}$$
$$\varphi_2 = y < 0 \wedge y < 1 \quad \textbf{SAT}$$

Semantic Fusion

$$\varphi_1 \qquad\qquad \varphi_2$$

$$(x > 0 \wedge x > 1) \wedge (y < 0 \wedge y < 1)$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2}$$ **SAT**

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2} \quad \textbf{SAT}$$

$$x = 1 \qquad y = -1$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2} \quad \textbf{SAT}$$

$$\searrow \quad z \quad \swarrow$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2} \quad \textbf{SAT}$$

$$z = x + y$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2}$$

SAT

$$z = x + y$$

Fusion Function

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2} \quad \textbf{SAT}$$

$$z = x + y$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2}$$

SAT

$$z = x + y$$

$$x = z - y \qquad y = z - x$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2}$$ **SAT**

$$z = x + y$$

$$x = z - y \qquad y = z - x$$

Inversion Functions

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2}$$

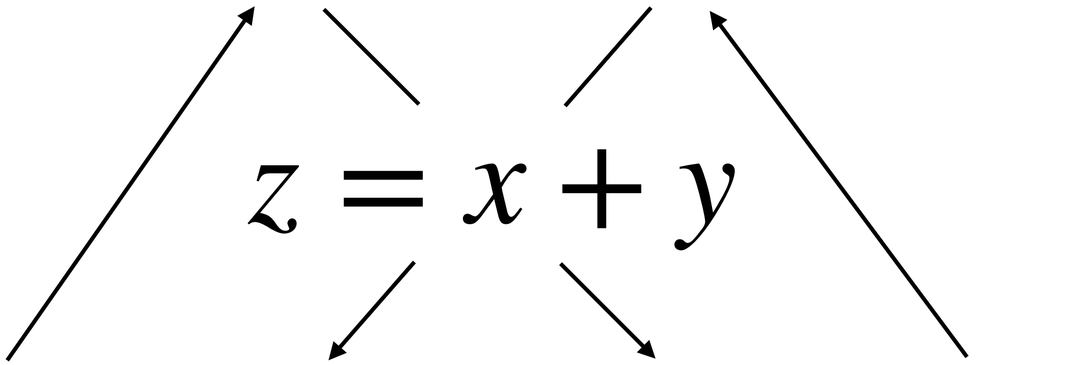
SAT

$$z = x + y$$

$$x = z - y \qquad y = z - x$$

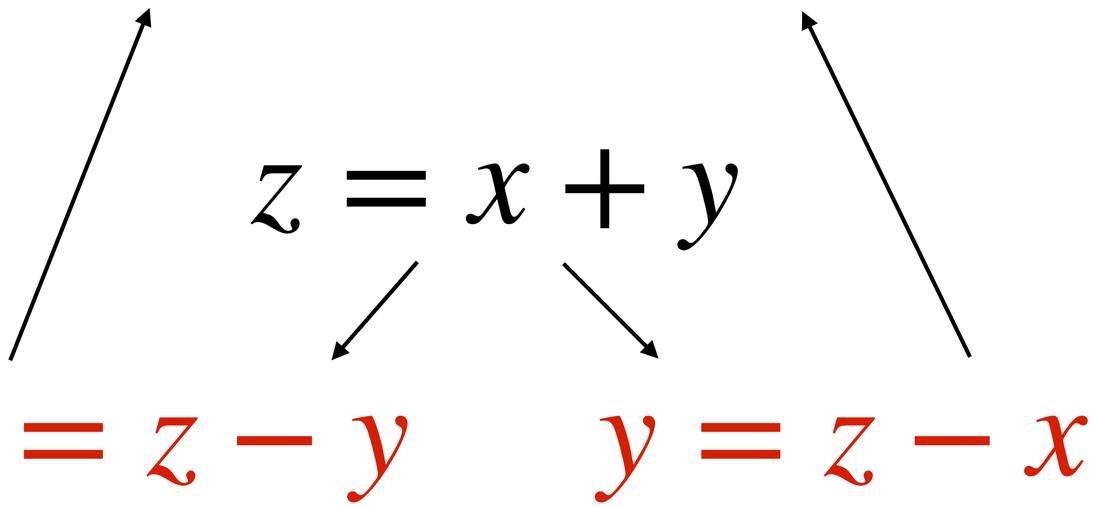
Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 0 \wedge x > 1)}_{\varphi_1} \wedge \underbrace{(y < 0 \wedge y < 1)}_{\varphi_2} \quad \textbf{SAT}$$



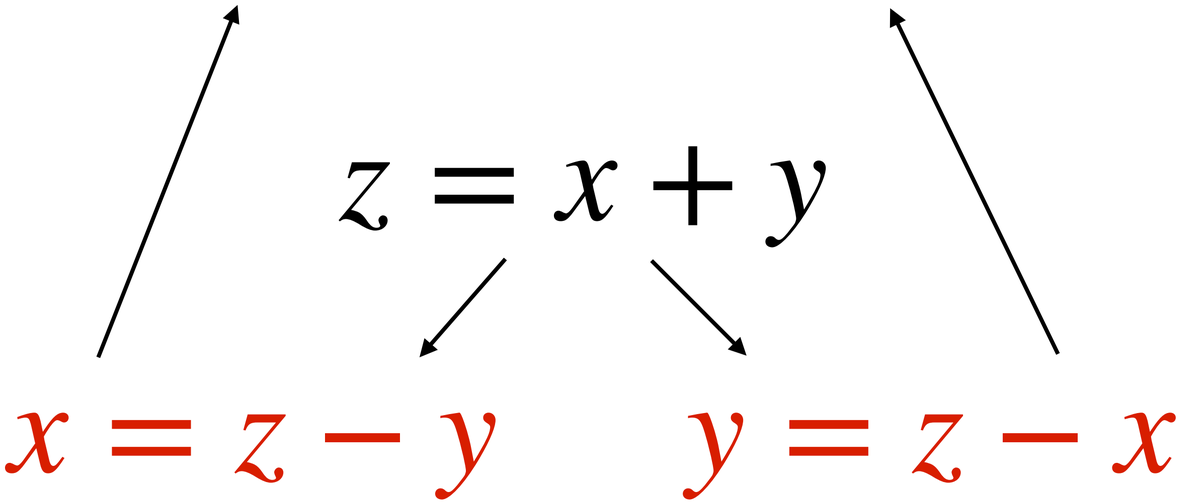
Semantic Fusion

$$\varphi_{fused} = (x > 0 \wedge \underbrace{(z - y) > 1}_{\varphi_1}) \wedge (\underbrace{(z - x) < 0}_{\varphi_2} \wedge y < 1)$$



Semantic Fusion

$$\varphi_{fused} = (x > 0 \wedge (z - y) > 1) \wedge ((z - x) < 0 \wedge y < 1)$$ **SAT**



Semantic Fusion

$$\varphi_{concat} = (x > 0 \wedge x > 1) \wedge (y < 0 \wedge y < 1)$$

$x = 1 \qquad y = -1$

$$\varphi_{fused} = (x > 0 \wedge (z - y) > 1) \wedge ((z - x) < 0 \wedge y < 1)$$ **SAT**

$$z = x + y$$

$$x = z - y \qquad y = z - x$$

Semantic Fusion

$$\varphi_{concat} = (x > 0 \wedge x > 1) \wedge (y < 0 \wedge y < 1)$$

$$x = 1 \quad z = x + y = 0 \quad y = -1$$

$$\varphi_{fused} = (x > 0 \wedge (z - y) > 1) \wedge ((z - x) < 0 \wedge y < 1)$$ **SAT**

$$z = x + y$$

$$x = z - y \qquad y = z - x$$

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
```

SAT

```
(declare-fun y () Int)
(declare-fun v () Bool)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

SAT

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
```

```
(declare-fun y () Int)
(declare-fun v () Bool)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= x (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= y (- 1))))
```

```
z = x * y
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= (div z y) (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= (div z x) (- 1))))
```

```
z = x * y
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= (div z y) (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= (div z x) (- 1))))
```

SAT

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= (div z y) (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= (div z x) (- 1))))
```

SAT

```
$ cvc4 example.smt2
unsat
```

Semantic Fusion: an Example

```
(declare-fun x () Int)
(declare-fun w () Bool)
(declare-fun y () Int)
(declare-fun v () Bool)
(declare-fun z () Int)
(assert (= (div z y) (- 1)))
(assert (= w (= x (- 1))))
(assert w)
(assert (= v (not (= y (- 1)))))
(assert (ite v false (= (div z x) (- 1))))
```

SAT

```
$ cvc4 example.smt2
unsat
```

https://github.com/CVC4/CVC4/issues/3413

Semantic Fusion

$$\varphi_1 = x > 1 \wedge x < 0 \quad \textbf{UNSAT}$$
$$\varphi_2 = y < 0 \wedge y > 1 \quad \textbf{UNSAT}$$

Semantic Fusion

$$\varphi_1 \qquad\qquad \varphi_2$$

$$(x > 1 \wedge x < 0) \vee (y < 0 \wedge y > 1)$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 1 \wedge x < 0)}_{\varphi_1} \vee \underbrace{(y < 0 \wedge y > 1)}_{\varphi_2}$$

UNSAT

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 1 \wedge x < 0)}_{\varphi_1} \vee \underbrace{(y < 0 \wedge y > 1)}_{\varphi_2}$$

UNSAT

z

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 1 \wedge x < 0)}_{\varphi_1} \vee \underbrace{(y < 0 \wedge y > 1)}_{\varphi_2}$$

UNSAT

$$z = x + y$$

Semantic Fusion

$$\varphi_{concat} = \underbrace{(x > 1 \wedge x < 0)}_{\varphi_1} \vee \underbrace{(y < 0 \wedge y > 1)}_{\varphi_2}$$

UNSAT

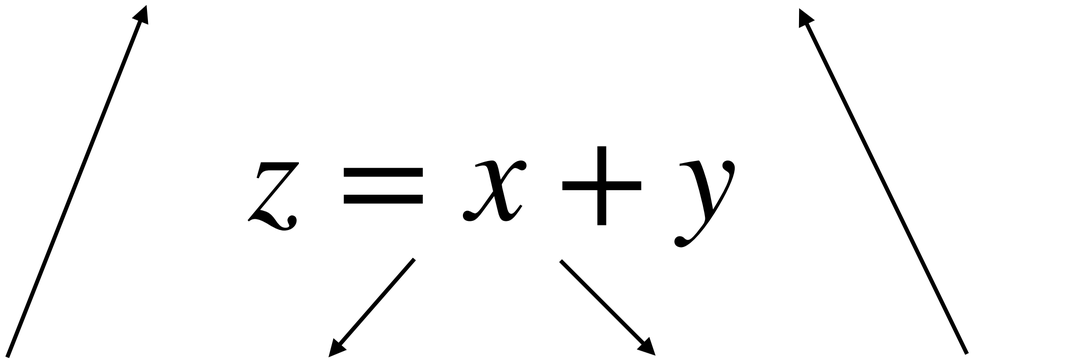
$$z = x + y$$

$$x = z - y \qquad y = z - x$$

Semantic Fusion

$$\varphi_{fused} = (x > 1 \wedge (z - y) < 0) \vee ((z - x) < 0 \wedge y > 1)$$

φ_1 φ_2



Semantic Fusion

$$\varphi_{fused} = (x > 1 \wedge (z - y) < 0) \vee ((z - x) < 0 \wedge y > 1)$$ **SAT**

Semantic Fusion

$$x = 2 \qquad z = 0 \qquad y = 2$$

$$\varphi_{fused} = (x > 1 \wedge (z - y) < 0) \vee ((z - x) < 0 \wedge y > 1)$$ **SAT**

Semantic Fusion

$$\varphi_{fused} = ((x > 1 \wedge (z - y) < 0) \vee ((z - x) < 0 \wedge y > 1)) \wedge z = x + y$$

Fusion Constraint

Semantic Fusion

$$\varphi_{fused} = ((x > 1 \wedge (z - y) < 0) \vee ((z - x) < 0 \wedge y > 1)) \wedge z = x + y$$

UNSAT

Semantic Fusion: an Example

```
(declare-fun x () Real)
(assert (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
```

UNSAT

```
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(assert (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0)))
```

UNSAT

Semantic Fusion: an Example

```
(declare-fun x () Real)
(assert (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
```

```
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(assert (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0)))
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(assert (or
        (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
        (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0))))
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(assert (or
        (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
        (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0))))
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
        (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0))))
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 x) 6.0)
                (+ 7.0 x))))
        (and (< y v) (>= w v)
             (< (/ w v ) 0) (> y 0)))))
```

```
z = x * y
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)                          z = x * y
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 (/ z y)) 6.0)
                (+ 7.0 x))))
        (and (< (/ z x) v) (>= w v)
             (< (/ w v ) 0) (> (/ z x) 0))))
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 (/ z y)) 6.0)
                (+ 7.0 x))))
        (and (< (/ z x) v) (>= w v)
             (< (/ w v ) 0) (> (/ z x) 0))))
(assert (= z (* x y)))
(assert (= x (/ z y)))
(assert (= y (/ z x)))
```

```
z = x * y
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 (/ z y)) 6.0)
                (+ 7.0 x))))
        (and (< (/ z x) v) (>= w v)
             (< (/ w v ) 0) (> (/ z x) 0))))
(assert (= z (* x y)))
(assert (= x (/ z y)))
(assert (= y (/ z x)))
```

UNSAT

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 (/ z y)) 6.0)
                (+ 7.0 x))))
        (and (< (/ z x) v) (>= w v)
             (< (/ w v ) 0) (> (/ z x) 0))))
(assert (= z (* x y)))
(assert (= x (/ z y)))
(assert (= y (/ z x)))
```

UNSAT

```
% z3 example.smt2
sat
```

Semantic Fusion: an Example

```
(declare-fun x () Real)
(declare-fun y () Real)
(declare-fun w () Real)
(declare-fun v () Real)
(declare-fun z () Real)
(assert (or
        (not (= (+ (+ 1.0 (/ z y)) 6.0)
                (+ 7.0 x))))
        (and (< (/ z x) v) (>= w v)
             (< (/ w v ) 0) (> (/ z x) 0))))
(assert (= z (* x y)))
(assert (= x (/ z y)))
(assert (= y (/ z x)))
```

UNSAT

```
% z3 example.smt2
sat
```

https://github.com/Z3Prover/z3/issues/2391

Fusion Functions

Type	Fusion Function	Variable Inversion Functions	
		r_x	r_y
Int	$x + y$	$z - y$	$z - x$
	$x + c + y$	$z - c - y$	$z - c - x$
	$x * y$	$z \ div \ y$	$z \ div \ x$
	$c_1 * x + c_2 * y + c_3$	$(z - c_2 * y - c_3) \ div \ c_1$	$(z - c_1 * x - c_3) \ div \ c_2$
Real	$x + y$	$z - y$	$z - x$
	$x + c + y$	$z - c - y$	$z - c - x$
	$x * y$	z/y	z/x
	$c_1 * x + c_2 * y + c_3$	$(z - c_2 * y - c_3)/c_1$	$(z - c_1 * x - c_3)/c_2$
String	$x \ str{+}{+} \ y$	$str.substr \ z \ 0 \ (str.len \ x)$	$str.substr \ z \ (str.len \ x) \ (str.len \ y)$
	$x \ str{+}{+} \ y$	$str.substr \ z \ 0 \ (str.len \ x)$	$str.replace \ z \ x \ ""$
	$x \ str{+}{+} \ c \ str{+}{+} \ y$	$str.substr \ z \ 0 \ (str.len \ x)$	$str.replace \ (str.replace \ z \ x \ "") \ c \ ""$

Empirical Evaluation

Empirical Evaluation

- Tool YinYang, our realization of Semantic Fusion

Empirical Evaluation

- Tool YinYang, our realization of Semantic Fusion
- Bug hunting with YinYang (July-October 2019)

Empirical Evaluation

- Tool YinYang, our realization of Semantic Fusion
- Bug hunting with YinYang (July-October 2019)
- Bug reduction with C-Reduce

Empirical Evaluation

- Tool YinYang, our realization of Semantic Fusion
- Bug hunting with YinYang (July-October 2019)
- Bug reduction with C-Reduce
- Bug reports on issue trackers of Z3 and CVC4

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

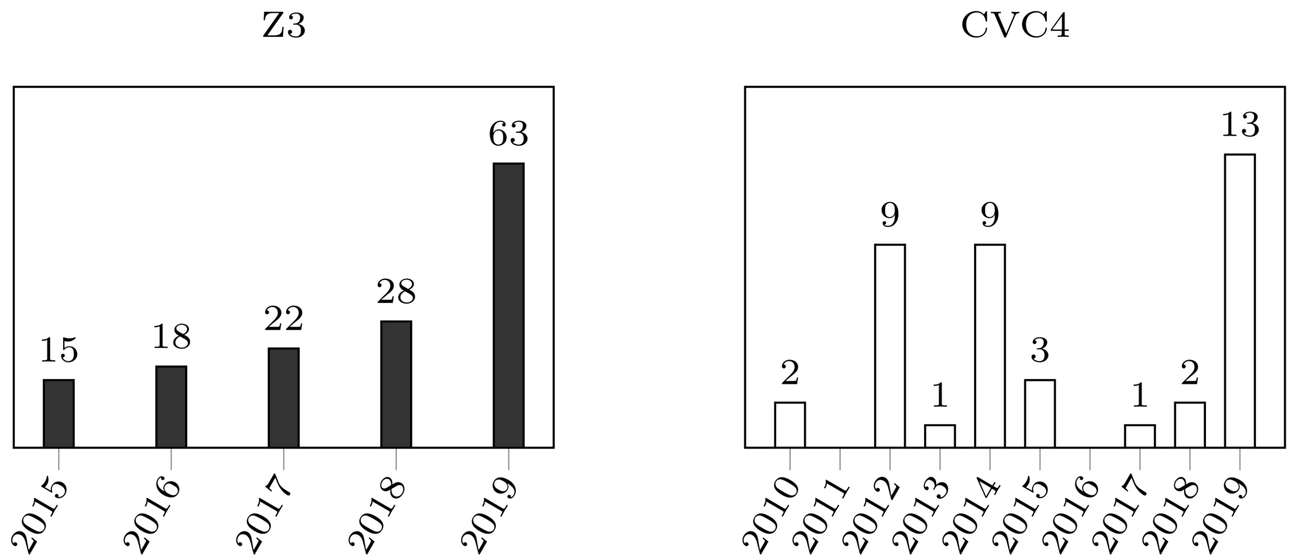
How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

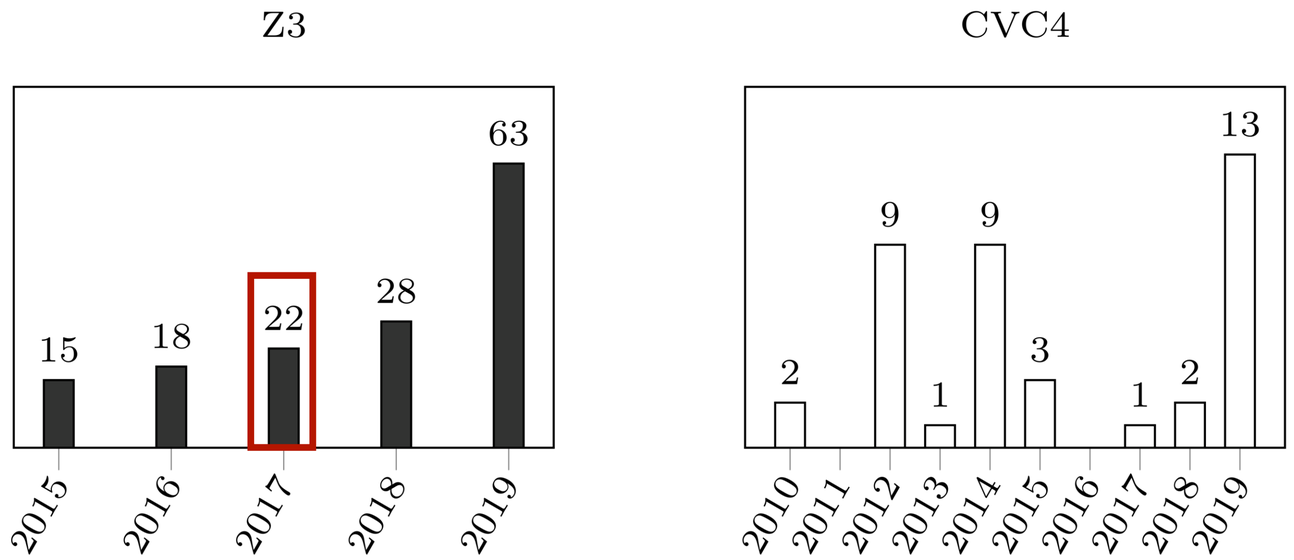
Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

Significance of the bug finding results



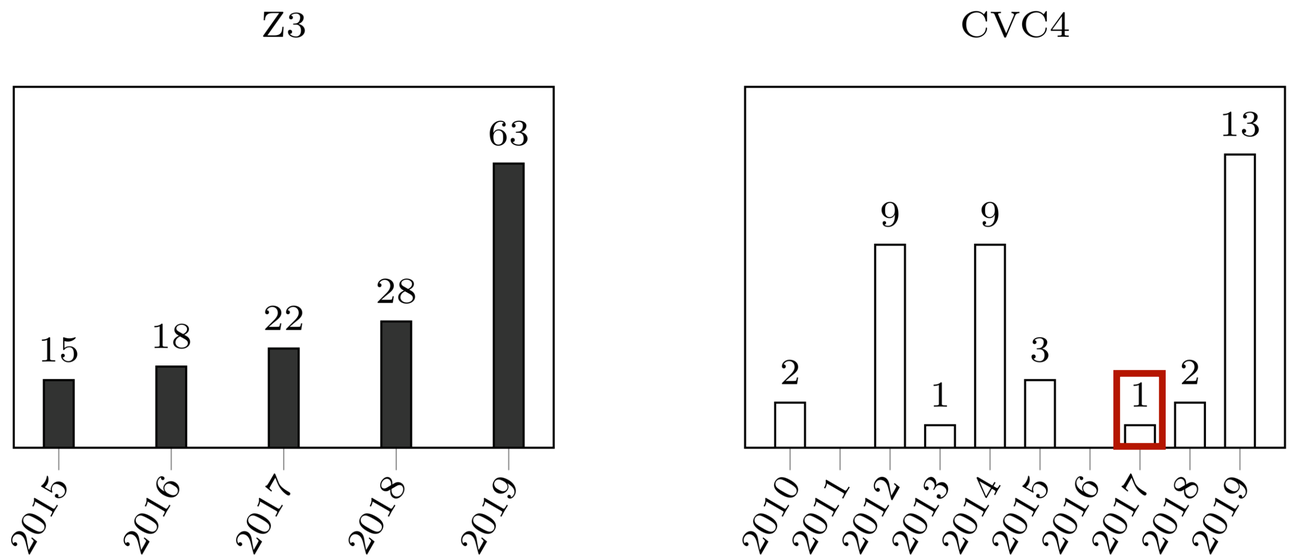
Soundness bugs per year in Z3 and CVC4.

Significance of the bug finding results



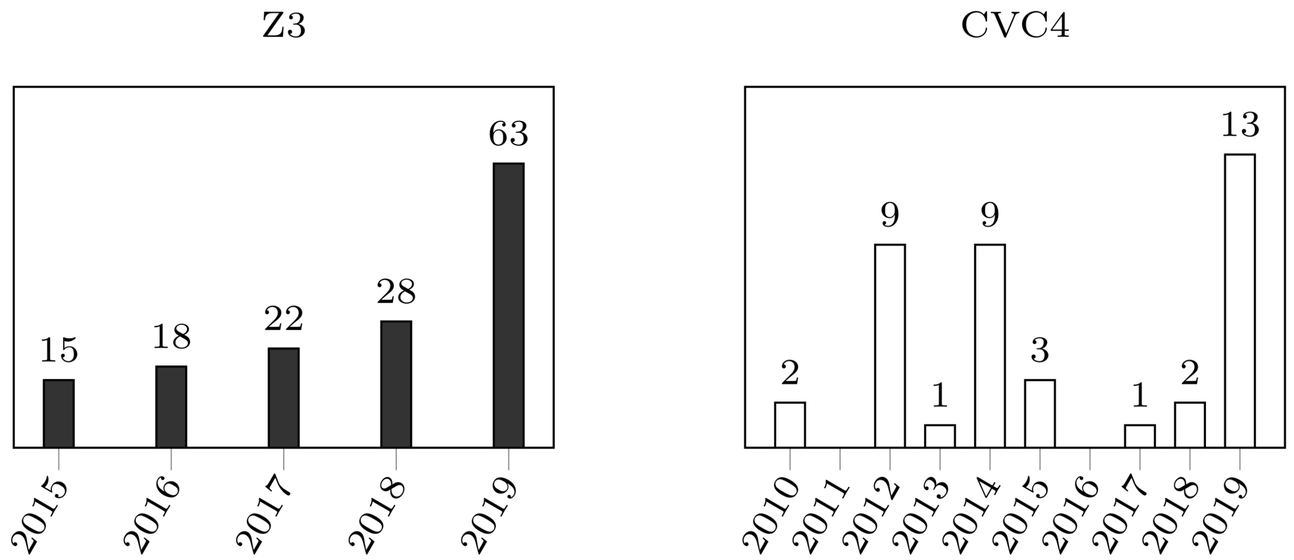
Soundness bugs per year in Z3 and CVC4.

Significance of the bug finding results



Soundness bugs per year in Z3 and CVC4.

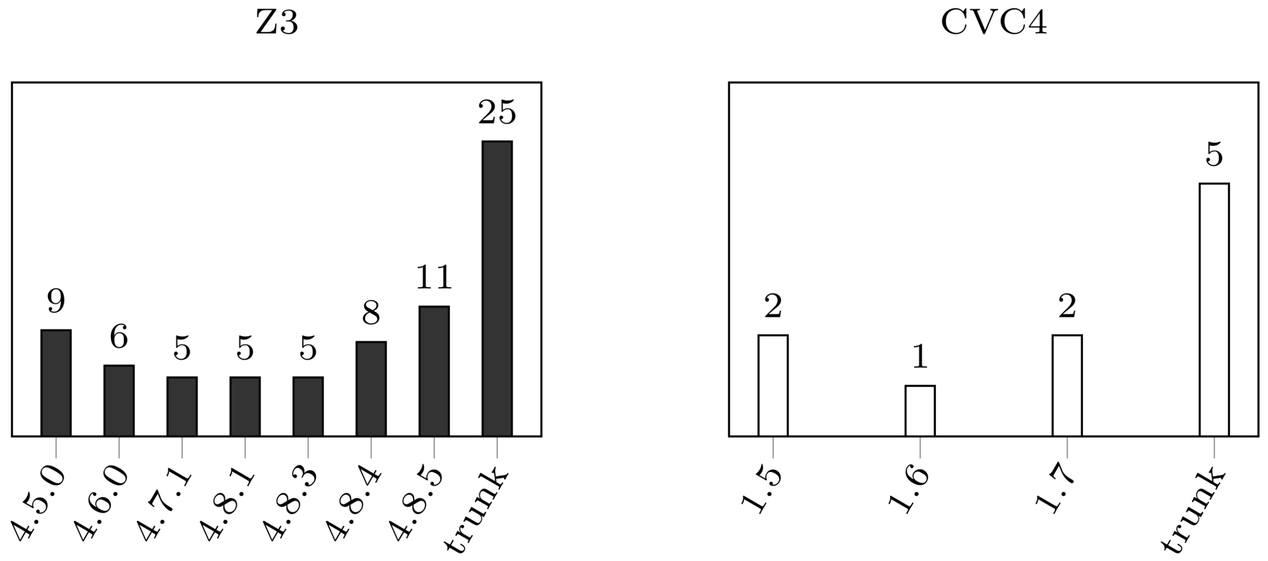
Significance of the bug finding results



Soundness bugs per year in Z3 and CVC4.

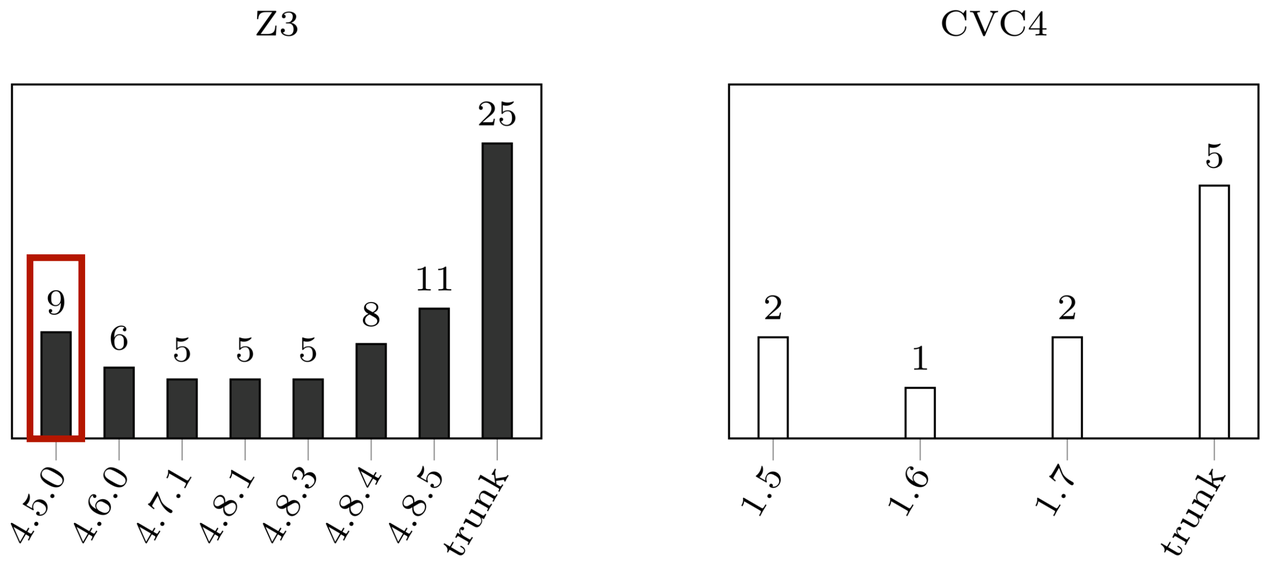
YinYang found 24 in Z3, 5 in CVC4 in 4 months

Significance of the bug finding results



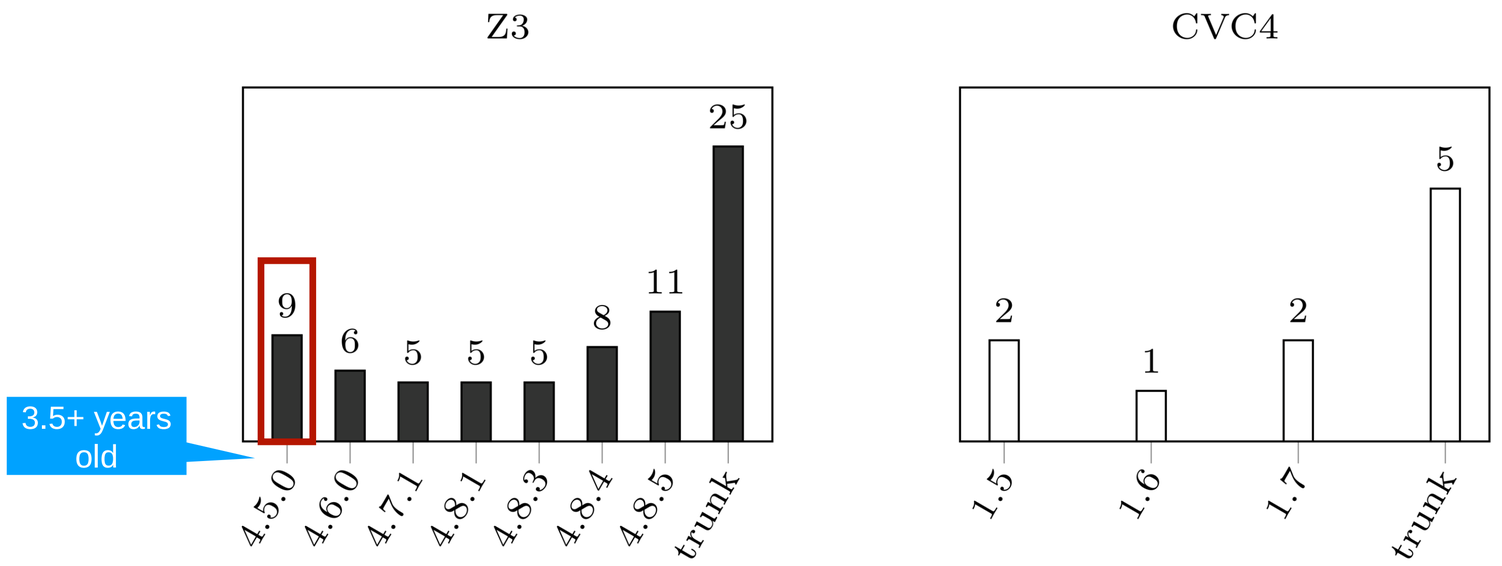
Soundness bugs in historical Z3 and CVC4 releases and the trunk.

Significance of the bug finding results



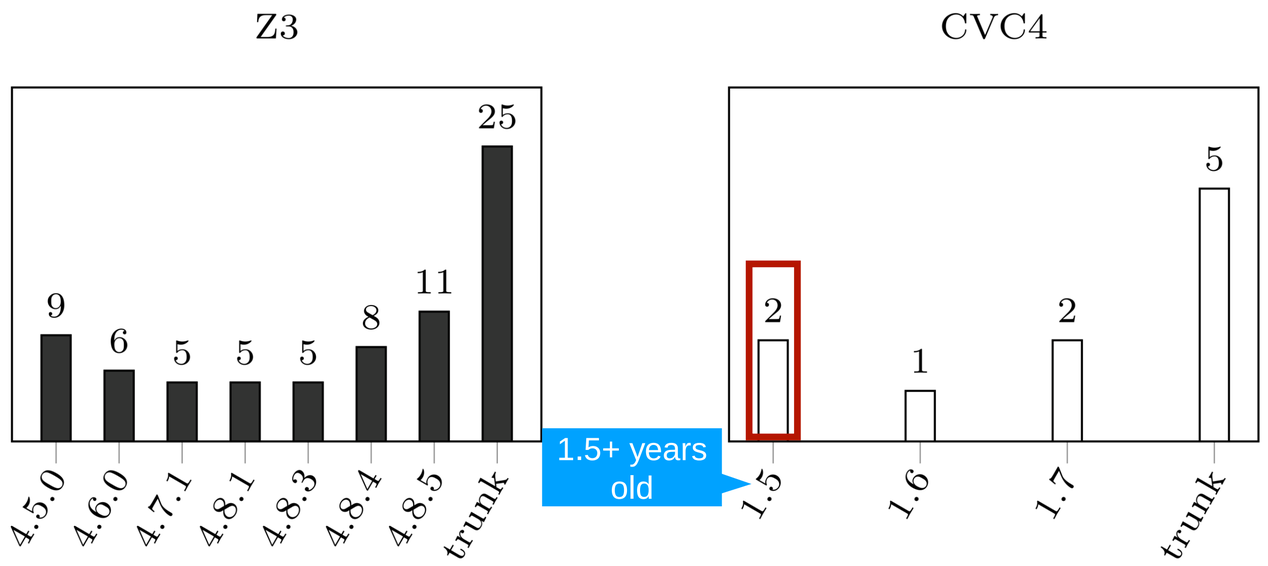
Soundness bugs in historical Z3 and CVC4 releases and the trunk.

Significance of the bug finding results



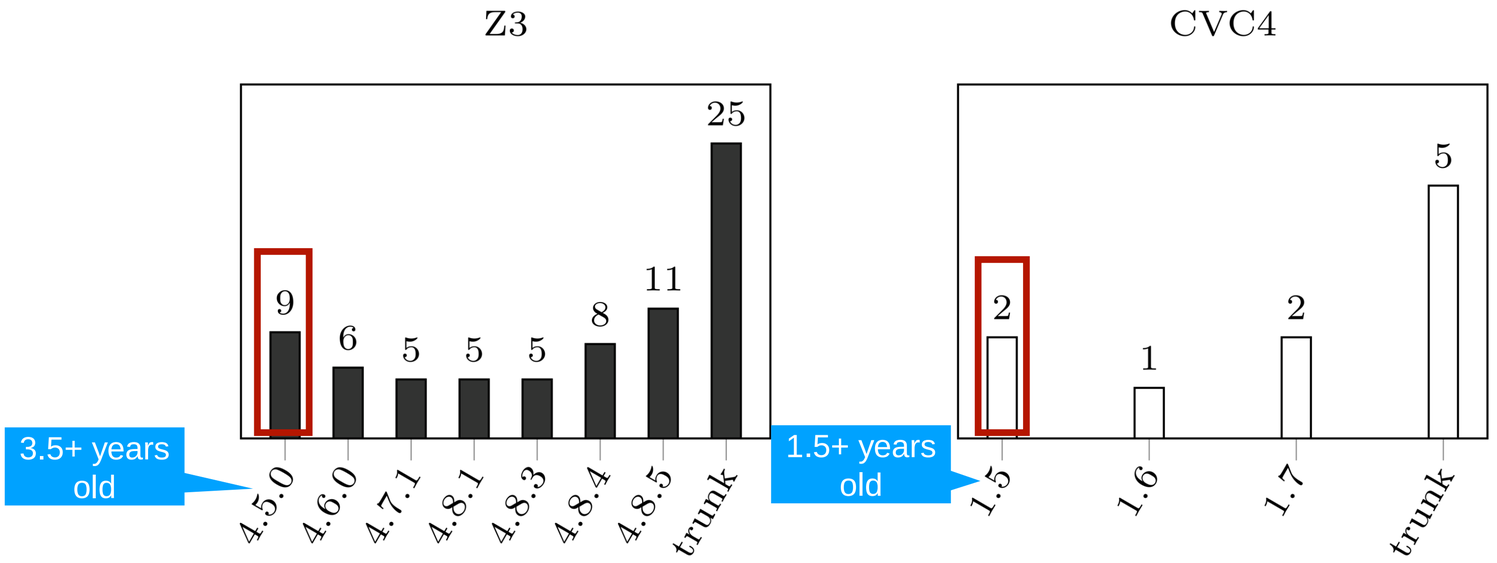
Soundness bugs in historical Z3 and CVC4 releases and the trunk.

Significance of the bug finding results



Soundness bugs in historical Z3 and CVC4 releases and the trunk.

Significance of the bug finding results



Soundness bugs in historical Z3 and CVC4 releases and the trunk.

YinYang found longstanding soundness bugs

Is Semantic Fusion necessary?

Is Semantic Fusion necessary?

ConcatFuzz

$\varphi_1 \quad \wedge \quad \varphi_2$

SAT **SAT**

$\varphi_1 \quad \vee \quad \varphi_2$

UNSAT **UNSAT**

Is Semantic Fusion necessary?

ConcatFuzz

$\varphi_1 \quad \wedge \quad \varphi_2$

SAT **SAT**

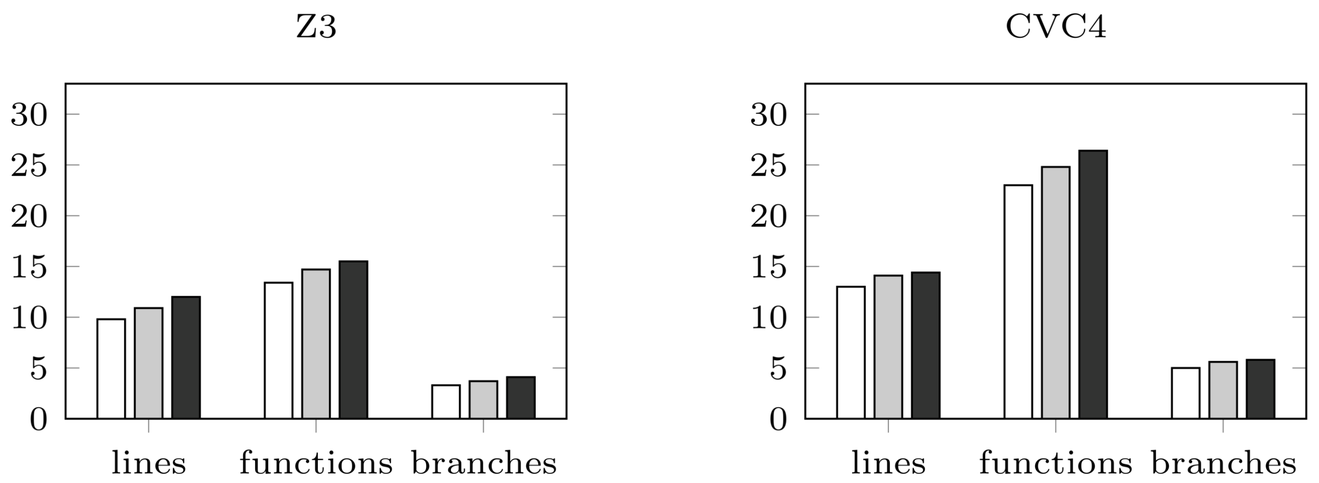
$\varphi_1 \quad \vee \quad \varphi_2$

UNSAT **UNSAT**

ConcatFuzz can only retrigger 5/50 bugs

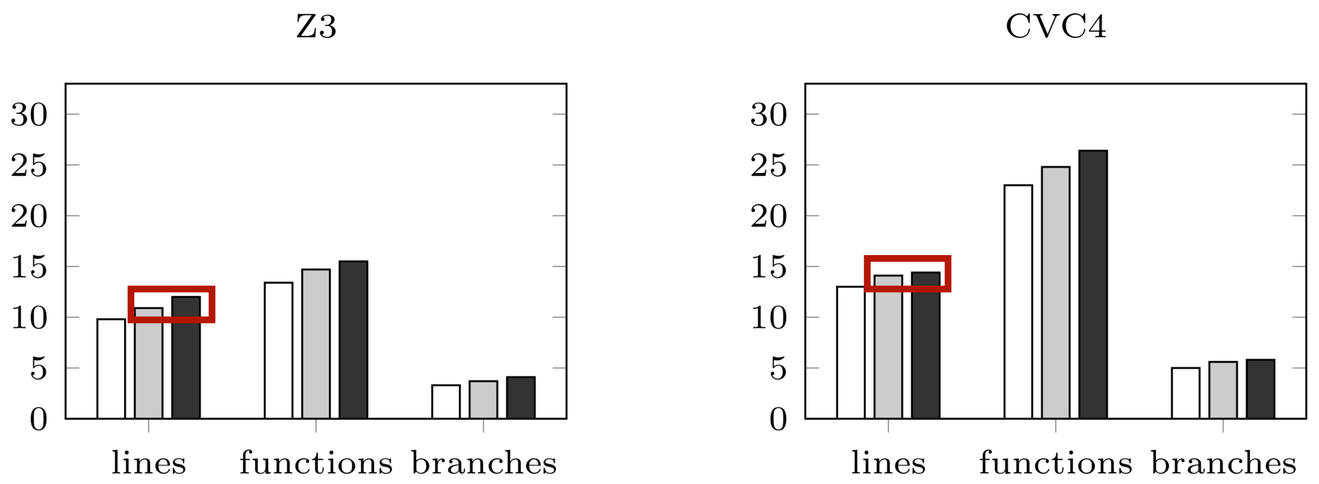
Is Semantic Fusion necessary?

Is Semantic Fusion necessary?



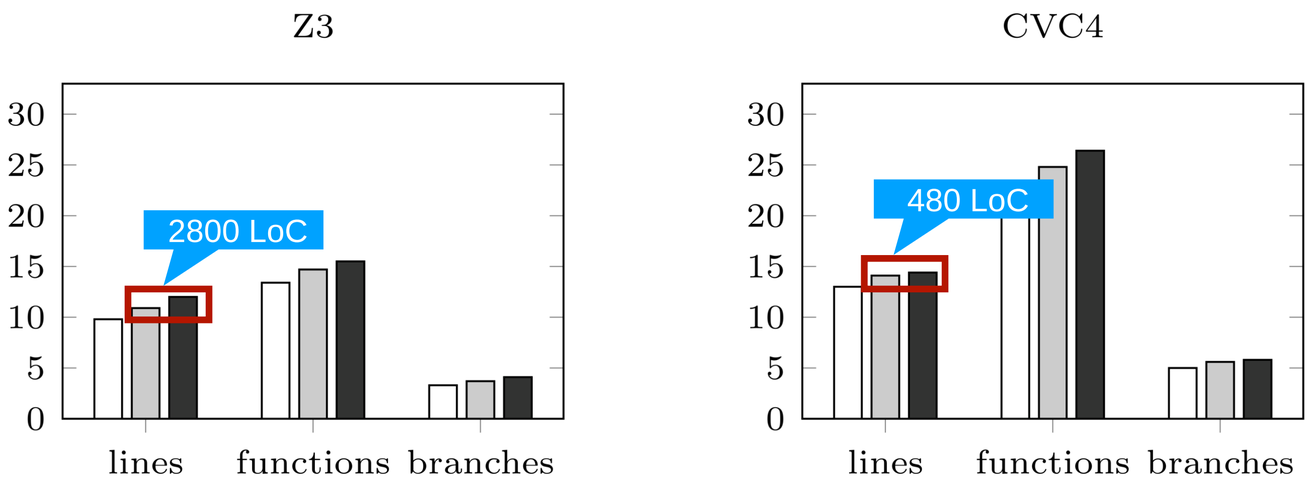
Code coverage comparison of Benchmark, ConcatFuzz and YinYang.

Is Semantic Fusion necessary?



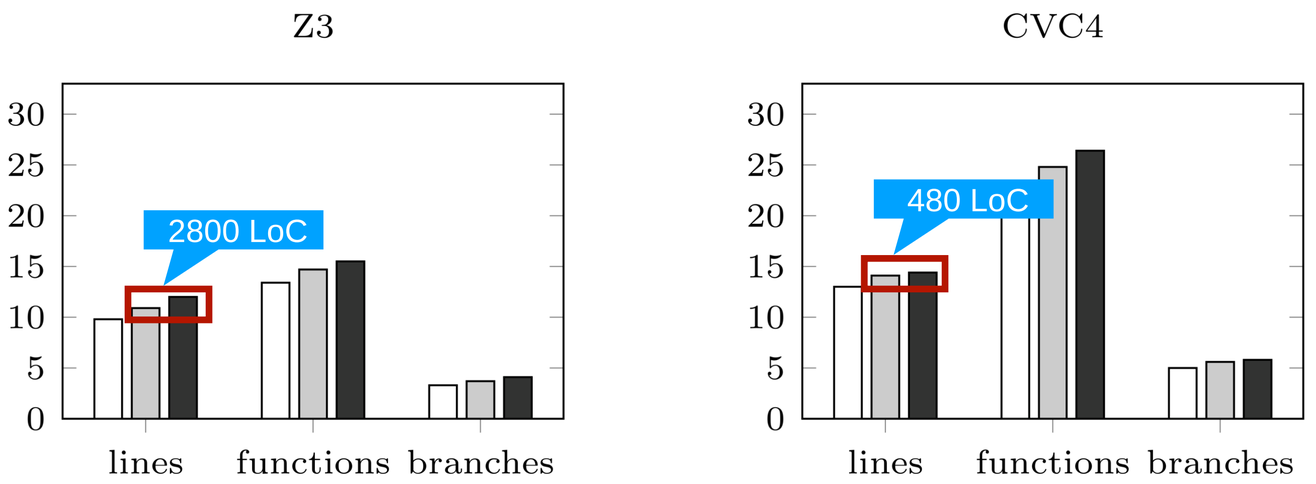
Code coverage comparison of Benchmark, ConcatFuzz and YinYang.

Is Semantic Fusion necessary?



Code coverage comparison of Benchmark, ConcatFuzz and YinYang.

Is Semantic Fusion necessary?



Code coverage comparison of Benchmark, ConcatFuzz and YinYang.

YinYang consistently achieves higher coverage

Z3 #2376

```
% cat formula.smt2
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun c () Real)
(declare-fun d () Real)
(declare-fun j () Real)
(declare-fun e () Real)
(assert (not (exists ((f Real))
(=> (and (< (/ 0 0) c) (< (/ 0 (* 2.0 b))
d))(= (= 0.0 a) (not (=> (<= f a) (<= e
j))))))))
(check-sat)

% cvc4 formula.smt2
unsat

%z3 formula.smt2
sat
```

Z3 #2376

```
% cat formula.smt2
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun c () Real)
(declare-fun d () Real)
(declare-fun j () Real)
(declare-fun e () Real)
(assert (not (exists ((f Real))
(=> (and (< (/ 0 0) c) (< (/ 0 (* 2.0 b))
d))(= (= 0.0 a) (not (=> (<= f a) (<= e
j))))))))
(check-sat)

% cvc4 formula.smt2
unsat

%z3 formula.smt2
sat
```

NikolajBjorner commented on 5 Jul 2019

thanks, fixed

CVC4 #3412

```
% cat formula.smt2
(declare-fun a () Int)
(declare-fun b () Int)
(assert (= (div a b) (- 1)))
(check-sat)

% z3 formula.smt2
sat

% cvc4 formula.smt2
unsat
```

CVC4 #3412

```
% cat formula.smt2
(declare-fun a () Int)
(declare-fun b () Int)
(assert (= (div a b) (- 1)))
(check-sat)
```

```
% z3 formula.smt2
sat
```

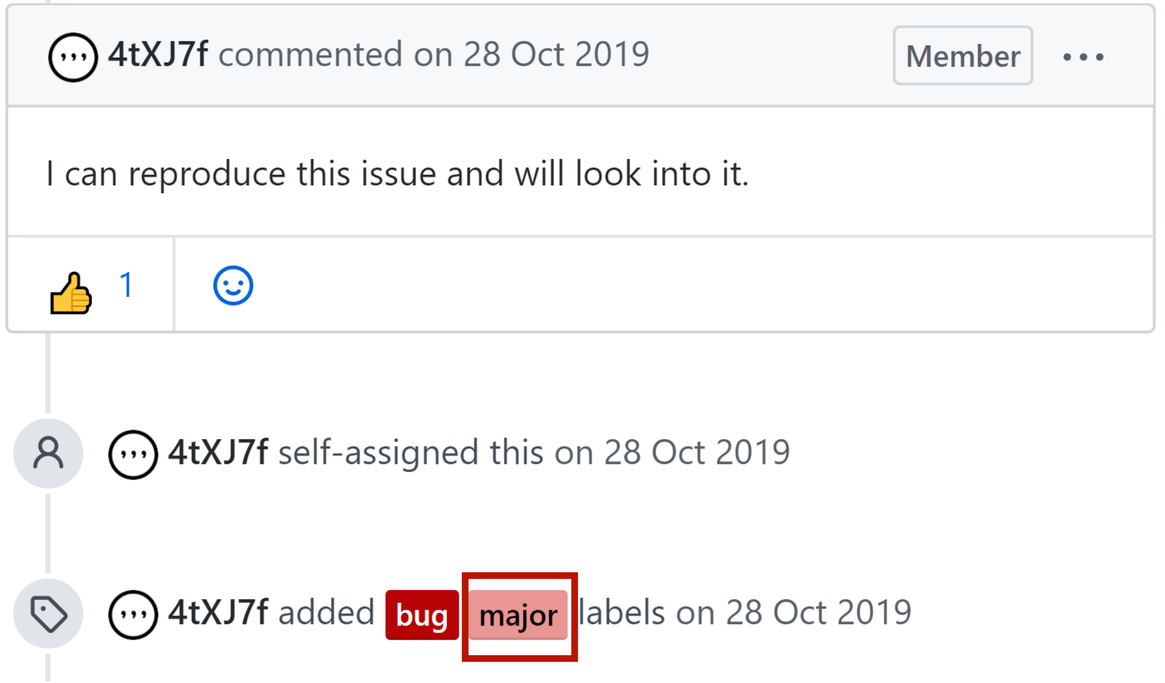
```
% cvc4 formula.smt2
unsat
```

CVC4 #3412

```
% cat formula.smt2
(declare-fun a () Int)
(declare-fun b () Int)
(assert (= (div a b) (- 1)))
(check-sat)

% z3 formula.smt2
sat

% cvc4 formula.smt2
unsat
```



Z3 #4153

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(assert (=
    (str.++ (str.substr "1" 0  (str.len a))
"0") b))
(assert (< (str.to.int b) 0))
(check-sat)

% z3-4.8-7 formula.smt2
unsat

% z3 formula.smt2
sat
```

Z3 #4153

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(assert (=
    (str.++ (str.substr "1" 0  (str.len a))
"0") b))
(assert (< (str.to.int b) 0))
(check-sat)

% z3-4.8-7 formula.smt2
unsat

% z3 formula.smt2
sat
```

NikolajBjorner commented on 29 Apr

exposed

- incomplete axiomatization of stoi
- more opportunities for rewriting

CVC4 #3217

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(declare-fun c () String)
(declare-fun d () String)
(assert
    (or (not (= (str.suffixof "B"
                (str.replace "A" b "B"))
     (= ( str.substr a 0 (str.len b)) "A")))
        (not (= (not (= c "A")) (str.suffixof "A"
        (str.replace "A" c "B"))))))
(assert (= a (str.++ (str.++ b "") d)))
(check-sat)


% z3 formula.smt2
unsat

% cvc4 formula.smt2
sat
```

CVC4 #3217

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(declare-fun c () String)
(declare-fun d () String)
(assert
    (or (not (= (str.suffixof "B"
                (str.replace "A" b "B"))
     (= ( str.substr a 0 (str.len b)) "A")))
        (not (= (not (= c "A")) (str.suffixof "A"
        (str.replace "A" c "B"))))))
(assert (= a (str.++ (str.++ b "") d)))
(check-sat)
```

ajreynol commented on 23 Aug 2019

Another excellent find, thanks a lot.

This is fixed in my latest PR.

ajreynol added **bug** **major** labels on 23 Aug 2019

```
% z3 formula.smt2
unsat

% cvc4 formula.smt2
sat
```

Z3 #2618 & CVC4 #3357

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(declare-fun c () String)
(assert (str.in.re c
(re.* (re.union (str.to.re "aa")
(str.to.re ""))))))
(assert (= 0 (str.to.int
(str.replace a b (str.at a
(str.len a))))))
(assert (= a (str.++ b c)))
(check-sat)

% cvc4 formula.smt2
unsat

% z3 formula.smt2
sat
```

Z3 #2618 & CVC4 #3357

```
% cat formula.smt2
(declare-fun a () String)
(declare-fun b () String)
(declare-fun c () String)
(assert (str.in.re c
(re.* (re.union (str.to.re "aa")
(str.to.re ""))))))
(assert (= 0 (str.to.int
(str.replace a b (str.at a
(str.len a))))))
(assert (= a (str.++ b c)))
(check-sat)

% cvc4 formula.smt2
unsat

% z3 formula.smt2
sat
```

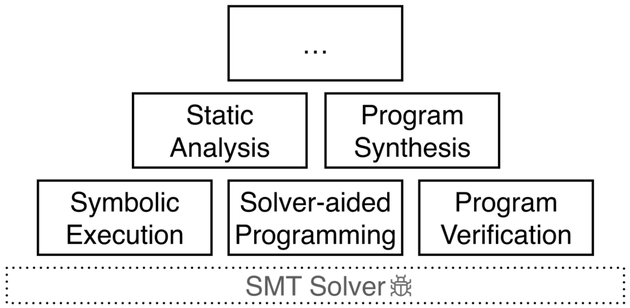
```
% z3 unreduced.smt2
sat
```

```
% cvc4 unreduced.smt2
sat
```

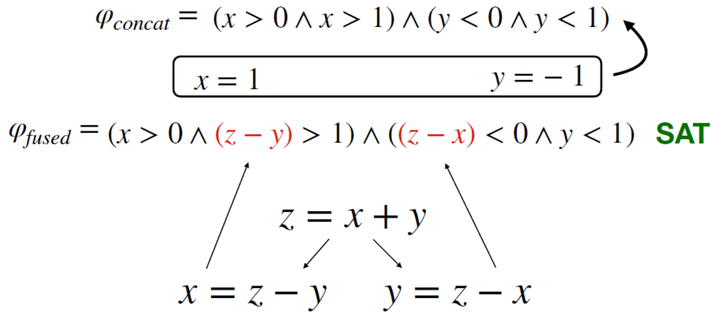
Z3 and CVC4 are both unsound on the unreduced test!

Summary

SMT Solver



Semantic Fusion



Testing SMT solvers is challenging

- How to generate test formulas?
- How to obtain the test oracles?
- It is challenging to find bugs.

How many bugs can YinYang find?

Status	Z3	CVC4	Total
Reported	45	13	58
Confirmed	38	8	46
Fixed	36	6	42
Duplicate	4	1	5
Won't fix	2	0	2

Type	Z3	CVC4	Total
Soundness	24	6	30
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	16	4	20
QF_SLIA	3	1	4

YinYang Release

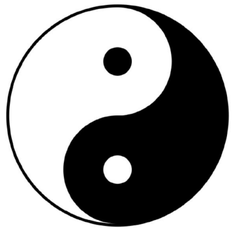
YinYang will be released this summer.
Please stay tuned!

Statistics & Report links

https://github.com/Z3Prover/z3/issues/2530 Fixed
https://github.com/Z3Prover/z3/issues/2531 Fixed
https://github.com/Z3Prover/z3/issues/2533 Fixed
https://github.com/Z3Prover/z3/issues/2546 Fixed
https://github.com/Z3Prover/z3/issues/2548 Fixed
https://github.com/Z3Prover/z3/issues/2556 Fixed
https://github.com/Z3Prover/z3/issues/2557 Fixed
https://github.com/Z3Prover/z3/issues/2562 Fixed
https://github.com/Z3Prover/z3/issues/2563 Dup
https://github.com/Z3Prover/z3/issues/2566 Dup
https://github.com/Z3Prover/z3/issues/2567 Fixed
https://github.com/Z3Prover/z3/issues/2573 Fixed
https://github.com/Z3Prover/z3/issues/2578 Fixed
https://github.com/Z3Prover/z3/issues/2580 Fixed
https://github.com/Z3Prover/z3/issues/2612 Fixed

https://testsmt.github.io

Code



github.com/testsmt