Validating SMT Solvers via Semantic Fusion

[PLDI’20]

Dominik Winterer, Chengyu Zhang, Zhendong Su

张灵毓

Outline

1. Background
   - SAT/SMT
   - Motivation

2. Approach
   - Semantic Fusion

3. Evaluation

Background

SAT

SAT problem: Given a well-formed formula α in propositional logic, decide whether there exists a satisfying solution for α.

$A \land B \land C$

SAT solver

sat: $\{ A = T, B = T, C = T\}$

SMT

SMT problem: Given a well-formed formula φ in first-order logic (often disallow quantifiers), decide whether there exists a satisfying solution for φ.

$$A \wedge B \wedge C \longrightarrow f(x,y,z) \wedge g(x,y,z) \wedge h(x,y,z)$$

$f(x,y,z)$: $\quad 3x + 2y - z = 1$

$g(x,y,z)$: $\quad 2x - 2y + 4z = -2$

$h(x,y,z)$: $\quad -x + 0.5y - z = 0$

SMT

$$f(x,y,z): \quad 3x + 2y - z = 1$$
$$g(x,y,z): \quad 2x - 2y + 4z = -2$$
$$h(x,y,z): \quad -x + 0.5y - z = 0$$

$$f(x,y,z) \wedge g(x,y,z) \wedge h(x,y,z)$$

SMT solver

sat: $\{\, x = 1, y = -2, z = -2\}$

Motivation

Widely used

- SMT solvers such as Z3 and CVC4 have been used as a building block for a wide range of applications across computer science, including in automated theorem proving, program analysis, program verification, and software testing.*

Lack of effective method

- Within the last ten years, SMT solvers have greatly matured, and finding bugs in them has become more difficult.

- Yet none has targeted other SMT theories nor found bugs in recent versions of CVC4.

*https://en.wikipedia.org/wiki/Satisfiability_modulo_theories

Approach

Key Insight

Key insight: fuse two tests into a new test that combines the structures of its ancestors.

Intuitively, concatenate two tests (conjunction or disjunction).

$$\varphi_1 = x > 0 \land x < 2$$
$$\varphi_2 = \big(v = (y \neq -1)\big) \land (v \rightarrow false)$$

$$\downarrow$$

$$\varphi_{concate} = (x > 0 \land x < 2\,) \land (\big(v = (y \neq -1)\big) \land (v \rightarrow false))$$

Formula Concatenation

Concatenate two tests (conjunction or disjunction).

$$\varphi_{concate} = (x > 0 \land x < 2) \land \big((v = (y \neq -1)\big) \land (v \to false))$$

More complex and satisfiability is provable.

Free variables of φ_1 (e.g. x) and free variables of φ_2 (e.g. y) are independent with each other.

Semantic Fusion

$$\varphi_{concate} = (x > 0 \wedge x < 2\,) \wedge \big((v = (y \neq -1)\big) \wedge (v \rightarrow false))$$

φ_1 φ_2

Variable Fusion: Create fresh variables to connect the free variable sets of φ_1 and φ_2 using fusion function.

Fusion function: $z = f(x, y)$ (e.g. $z = x * y$)

Variable Inversion: Substitute some occurrences of the chosen variables in φ_1 and φ_2 by inversion functions.

Inversion function: $x = r_x(y, z)$ (e.g. $x = z\,/y$)
$y = r_y(x, z)$ (e.g. $y = z\,/x$)

Semantic Fusion

$$\varphi_{concate} = (x > 0 \land x < 2\ ) \land ((v = (y \neq -1)) \land (v \rightarrow false))$$

$$\varphi_{fused} = (((z/y) > 0 \land x < 2\ ) \land ((v = ((z/x) \neq -1)) \land (v \rightarrow false))$$

Satisfiability is still provable?

Yes, map z to the value which is calculated by fusion function.

$$M(v) = M_1(v), \quad \text{for } v \in vars(\varphi_1)$$

$$M(v) = M_2(v), \quad \text{for } v \in vars(\varphi_2)$$

$$M(z) = f(M_1(x), M_2(y))$$

Fusion constraints

Add fusion constraints while fusing UNSAT formula.

Counterexample:

$$\varphi_1 = x > 0 \land x < 0$$
$$\varphi_2 = y \neq y$$

$\downarrow$ $z = x + y$

$$\varphi_{fused} = (x > 0 \land (z - y) < 0) \lor (y \neq z - x)$$

$$sat: \{x = 1, y = 2, z = 1\}$$

Fusion constraints

Correct fusion:

$$\varphi_1 = x > 0 \wedge x < 0$$
$$\varphi_2 = y \neq y$$

$$\Big\downarrow \; z = x + y$$

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

Evaluation

Evaluation

none has targeted other SMT theories

Logic	Z3	CVC4	Total
NIA	2	1	3
NRA	15	1	16
QF_NIA	0	1	1
QF_NRA	2	0	2
QF_S	15	4	19
QF_SLIA	3	1	4

(c)

(c) Affected SMT logics of the confirmed bugs in Z3 and CVC4

Evaluation

nor found bugs in recent versions of CVC4.

Status	Z3	CVC4	Total
Reported	44	13	57
Confirmed	37	8	45
Fixed	35	6	41
Duplicate	4	1	5
Won't fix	2	0	2

(a)

Type	Z3	CVC4	Total
Soundness	24	5	29
Crash	11	1	12
Performance	1	2	3
Unknown	1	0	1

(b)

(a)Status of the reported bugs in Z3 and CVC4
(b)Types of the confirmed bugs in Z3 and CVC4

Evaluation

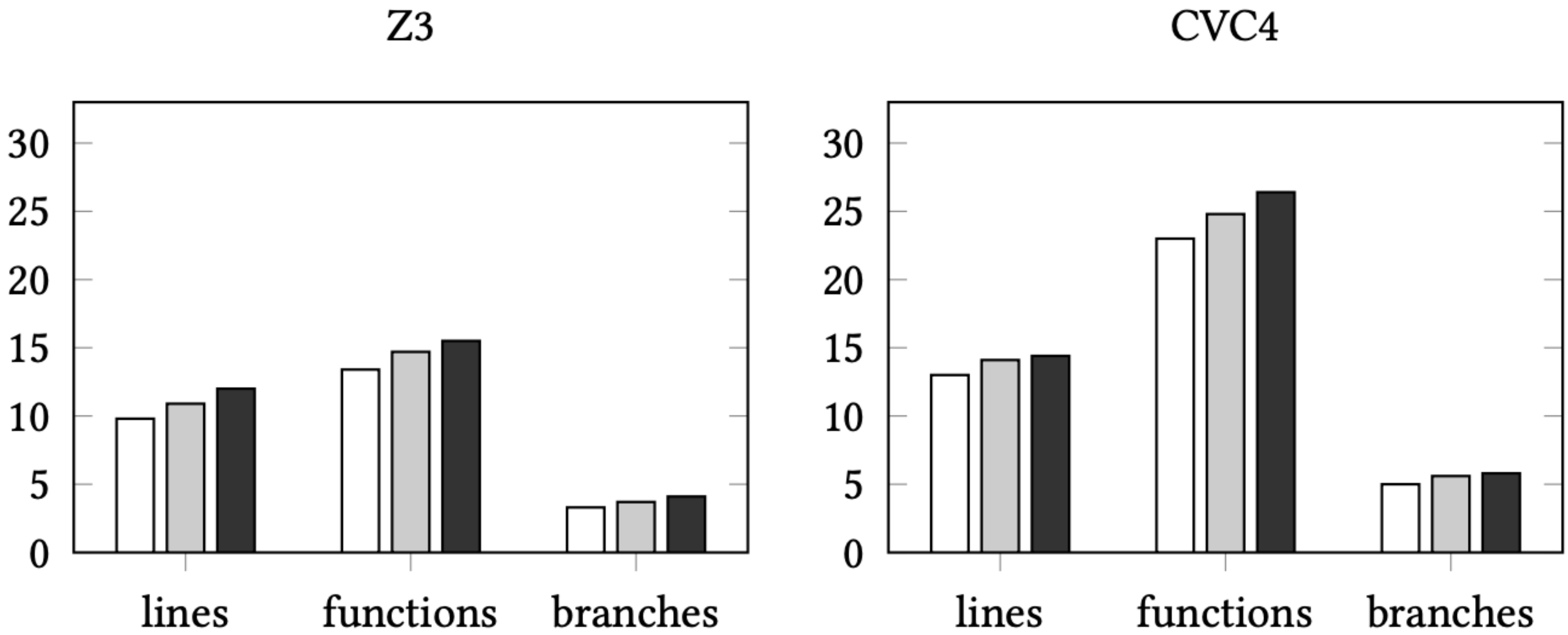

Figure 12. Coverage improvement (%) of ConcatFuzz (in gray) and YinYang (in black) over Benchmark (in white) averaged over all logics.

Thanks