A Benchmark for Infinite Models in SMT

Mikoláš Janota and Chad E. Brown and Cezary Kaliszyk

Czech Technical University in Prague

University of Innsbruck, Innsbruck

Intro

- A model is infinite iff the universe is infinite.
- Example: semigroups

$$(\forall xyz)((x * y) * z = x * (y * z))$$

- $(\{0,1\}, + \text{ mod } 2)$ — finite semigroup
- $(\mathbb{N}, +)$ — infinite semigroup

Motivation

- Models as counterexamples to:
  - incorrect programs
  - incorrect theorems
- Structures of interesting properties
  "Find a semigroup not a group!"
- Some properties only for infinite models
- In Satisfiability Modulo Theories infinite models often required for functions + integers + quantifiers (UFLIA).

SMT Models: Constants

```
(declare-fun c () Int)
(declare-fun d () Int)
(assert (< c d))
(check-sat)
(get-model)
```

$c < d$

```
:!z3 ex1.smt2
sat
(
  (define-fun d () Int
    1)
  (define-fun c () Int
    0)
)
```

$c = 0, d = 1$

SMT Models: Functions

```
(declare-fun f (Int) Int)
(assert (< (f 0) (f 1)))
(check-sat)
(get-model)
```

$$f(0) < f(1)$$

```
:!z3 ex2.smt2
sat
(
  (define-fun f ((x!0 Int)) Int
    (ite (= x!0 1) 1
      0))
)
```

$$fx \triangleq (1 \text{ if } x = 1 \text{ else } 0)$$

SMT Models: Quantifiers

```
(declare-fun f (Int) Int)
(assert (forall ((x Int))
            (<= (f x) x)))
(check-sat)
(get-model)
```

$$(\forall x)(fx \leq x)$$

```
:!z3 ex3.smt2
sat
(
  (define-fun f ((x!0 Int)) Int
    x!0)
)
```

$$fx \triangleq x$$

SMT Models: Quantifiers

```
(declare-fun f (Int) Int)
(assert (forall ((x Int))
             (< (f x) x)))
(check-sat)
(get-model)
```

$$(\forall x)(fx < x)$$

```
:!z3 -T:60 ex4.smt2
timeout
```

Not Solved!

A lot of work ahead of us!

Where the Problem Instances At?

- There are many many satisfiability instances
- because people make mistakes
- But these are not submitted to libraries.

Generating New Problems

- Based on existing problems (which are possibly unsatisfiable).
- Must be *very* easy!
- Focus on *fragments* of existing problems!

f, g-fragment

- Pick 2 uninterpreted functions f, g
- Keep only forall assertions containing f, g (and possibly constants)

Example

```
(declare-fun c () Int)
(declare-fun f (Int) Int)
(declare-fun g (Int) Int)
(declare-fun h (Int) Int)
(assert (forall ((x Int)) (< (f x) x)))
(assert (forall ((x Int)) (< (g x) (+c x))))
(assert (forall ((x Int)) (< (f x) (g x))))
(assert (forall ((x Int)) (< (f x) (h x))))
```

```
(declare-fun c () Int)
(declare-fun f (Int) Int)
(declare-fun g (Int) Int)
(assert (forall ((x Int)) (< (f x) x)))
(assert (forall ((x Int)) (< (g x) (+ c x))))
```

Glimpse into Future

MBQI

Model-Based Guided Quantifier Instantiation

For $\forall x \phi$ construct a sequence of:

- candidate models M_i
- counterexample instantiations σ_i
- s.t. $M_i \models \bigwedge_{j \in 1..i-1} \phi[x/\sigma_j]$
- s.t. $M_i \not\models \phi[x/\sigma_i]$

[Ge and de Moura, 2009]

Learn Infinite Models?

$$(\forall x)(fx > x)$$

$$f(0) > 0 \qquad f(0) \triangleq 1$$
$$f(1) > 1 \qquad f(1) \triangleq 2$$
$$f(2) > 2 \qquad f(2) \triangleq 3$$

$$f(x) \triangleq x + 1$$

[Janota et al., 2023]

Results UFLIA

Implemented in cvc5:

solver	SAT	UNSAT	total
standard MBQI	18843	7863	26706
ours smart MBQI	31977	7863	39840
Z3	28380	7482	35862

- Infinite models: under-explored in SMT
- Satisfiable problems mainly not in libraries.
- Despite being common during the process
- Benchmarks as fragments of existing
  - mostly wieldy
  - mostly satisfiable
  - anchored in reality
- More parameters for generation
  - number of functions
  - handling of constants

Ge, Y. and de Moura, L. M. (2009).
Complete instantiation for quantified formulas in satisfiabiliby modulo theories.
In *Computer Aided Verification CAV*, pages 306–320.

Janota, M., Piotrowski, B., and Chvalovský, K. (2023).
Towards learning infinite SMT models.
In *25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*.