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
A model is **infinite** iff the universe is infinite.

**Example:** semigroups

\[(\forall xyz)((x \ast y) \ast z = x \ast (y \ast z))\]

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

\[ f(0) < f(1) \]

\[ f_x \triangleq (1 \text{ if } x = 1 \text{ else } 0) \]
SMT Models: Quantifiers

\[(\forall x)(fx \leq x)\]

\[fx \triangleq x\]
SMT Models: Quantifiers

\[(\forall x)(fx < x)\]

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

:!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!
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
Model-Based Guided Quantifier Instantiation

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

- candidate models $M_i$
- counterexample instantiations $\sigma_i$

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?

\[(\forall x)(fx > x)\]

- \(f(0) > 0\)
- \(f(1) > 1\)
- \(f(2) > 2\)

- \(f(0) \triangleq 1\)
- \(f(1) \triangleq 2\)
- \(f(2) \triangleq 3\)

\[f(x) \triangleq x + 1\]

[Janota et al., 2023]
Results UFLIA

Implemented in cvc5:

<table>
<thead>
<tr>
<th>solver</th>
<th>SAT</th>
<th>UNSAT</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard MBQI</td>
<td>18843</td>
<td>7863</td>
<td>26706</td>
</tr>
<tr>
<td>ours smart MBQI</td>
<td>31977</td>
<td>7863</td>
<td>39840</td>
</tr>
<tr>
<td>Z3</td>
<td>28380</td>
<td>7482</td>
<td>35862</td>
</tr>
</tbody>
</table>
- **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