Scalable Algorithms for Abduction via Enumerative Syntax-Guided Synthesis

Andrew Reynolds, *Haniel Barbosa*, Daniel Larraz, Cesare Tinelli

IJCAR 2020

2020–07–04, The Internet

Overview

Abduction

Abduction

"What facts am I missing to reach a conclusion?"

The abduction problem

Given *axioms* Ax and a *goal* G, find a solution S such that:

- $\mathsf{Ax} \wedge \mathsf{S} \models \mathsf{G}$,
- $\mathsf{Ax} \wedge \mathsf{S}$ is satisfiable

Some applications of abduction

- ▷ Finding missing facts for discharging proof obligations [DDA12]
- ▷ Inferring library specifications [ZDD13]
- ▷ Synthesizing specifications for unknown subprocedures [ADG16]
- ▷ Loop invariant generation [DDLM13; EPS19]
- ▷ Compositional program verification [LDD+13]
- ▷ Synthesis of missing guards for memory safety [DDC14]
- ▷ ...

Despite numerous applications...

- Few standalone tools for abductive reasoning
  - GPID [EPS18]
  - EXPLAIN [DD13]

- Issues with
  - generality: restricted to specific logic fragments
  - flexibility: solutions within fixed criteria

Despite numerous applications...

- Few standalone tools for abductive reasoning
  - GPiD [EPS18]
  - Explain [DD13]

- Issues with
  - generality: restricted to specific logic fragments
  - flexibility: solutions within fixed criteria

Abduction via syntax-guided synthesis (SyGuS)

- Can be used with any background theory supported by SMT solvers
- Syntax restrictions can encode different criteria for solutions
- Standardized language: SyGuS-IF built on top of SMT-LIB

Redefining the abduction problem

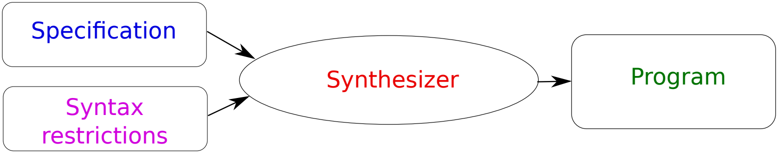
The (syntax-restricted) abduction problem for theory T

Given *axioms* Ax, *goal* G, theory T and grammar R, find a solution S such that:

- $\text{Ax} \wedge \text{S} \models_T \text{G}$,
- $\text{Ax} \wedge \text{S}$ is T-satisfiable
- S is generated by a context-free grammar R.

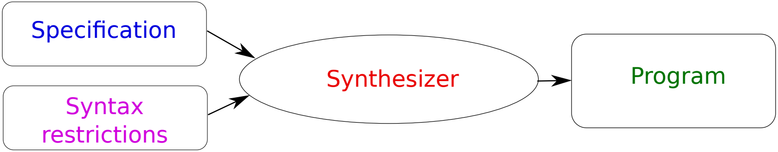
Abduction via Enumerative Syntax-Guided Synthesis (SyGuS)

Syntax-Guided Synthesis (SyGuS) [ABJ+13]



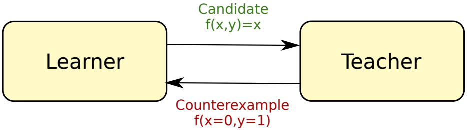
- ▷ Specification is given by (second-order) T-formula: $\exists f.\, \forall \bar{x}.\, \varphi[f, \bar{x}]$
- ▷ Syntactic restrictions given by context-free grammar R

Syntax-Guided Synthesis (SyGuS) [ABJ+13]



- ▷ Specification is given by (second-order) T-formula: $\exists f.\, \forall \bar{x}.\, \varphi[f, \bar{x}]$
- ▷ Syntactic restrictions given by context-free grammar R
- ▷ Commonly solved via enumerative CEGIS [STB+06; URD+13; RBN+19]

$I ::= 0 \mid 1 \mid x \mid y \mid I + I \mid I - I$



$\exists f.\, \forall \bar{x}.\, f(x, y) > x + 1$

Solving abduction with enumerative CEGIS

- We exploit the specification requiring that
  - $\mathsf{Ax}[\bar{x}] \land \mathsf{S}[\bar{x}] \land \neg\mathsf{G}[\bar{x}]$ be unsatisfiable

  to eagerly discard candidates

- Accumulate points (values $\bar{p}$ for $\bar{x}$) on which
  - Axioms are satisfied: $\textsc{eval}(\mathsf{Ax}[\bar{p}]) = \top$
  - Goal is falsified: $\textsc{eval}(\neg\mathsf{G}[\bar{p}]) = \top$

- Every candidate solution must be false on such points!
  - Otherwise $\textsc{eval}(\mathsf{Ax}[\bar{p}] \land \mathsf{S}[\bar{p}] \land \neg\mathsf{G}[\bar{p}]) = \top$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{\}$	

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{\}$	$x \geq 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{\}$	$x \geq 0$

$x \geq 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-SAT, $p_1 = (0, 0, -1)$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1)\}$	

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1)\}$	$x < 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1)\}$	$x < 0$

$\textsc{eval}(x < 0,\, p_1) = \bot$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1)\}$	$x < 0$

$x < 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-SAT, $p_2 = (-1, 0, 0)$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$y \geq 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0,0,-1), p_2 = (-1,0,0)\}$	$y \geq 0$

$\textsc{eval}(y \geq 0,\, p_1) = \top, \textsc{eval}(y \geq 0,\, p_2) = \top$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$y < 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$y < 0$

$\textsc{eval}(y < 0,\ p_1) = \bot, \textsc{eval}(y < 0,\ p_2) = \bot$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$y < 0$

$y < 0 \land y \geq 0$ is not T-SAT

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$z \geq 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$z \geq 0$

$\textsc{eval}(z \geq 0,\, p_1) = \bot, \textsc{eval}(z \geq 0,\, p_2) = \top$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$z < 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$z < 0$

$\text{EVAL}(z < 0,\, p_1) = \top, \text{EVAL}(z < 0,\, p_2) = \bot$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + y \geq 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + y \geq 0$

$\textsc{eval}(x + y \geq 0,\, p_1) = \top, \textsc{eval}(x + y \geq 0,\, p_2) = \bot$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + y < 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + y < 0$

$\textsc{eval}(x + y < 0,\, p_1) = \bot, \textsc{eval}(x + y < 0,\, p_2) = \top$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + z \geq 0$

Solving abduction with enumerative CEGIS

Consider T as LIA, $\mathsf{Ax} = \{y \geq 0\}$, $\mathsf{G} = \{x + y + z \geq 0\}$,

$$R = \quad A \to 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \to A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + z \geq 0$

$\text{EVAL}(x + z \geq 0,\ p_1) = \perp, \text{EVAL}(x + z \geq 0,\ p_2) = \perp$

Solving abduction with enumerative CEGIS

Consider T as LIA, Ax $= \{y \geq 0\}$, G $= \{x + y + z \geq 0\}$,

$$R = \quad A \rightarrow 0 \mid 1 \mid x \mid y \mid z \mid A + A \qquad B \rightarrow A < A \mid A \geq A$$

and let P be the set of points satisfying Ax and falsifying G.

P	Candidate
$\{p_1 = (0, 0, -1), p_2 = (-1, 0, 0)\}$	$x + z \geq 0$ ✓

$x + z \geq 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-UNSAT and
$x + z \geq 0 \wedge y \geq 0$ is T-SAT

Scalability issues

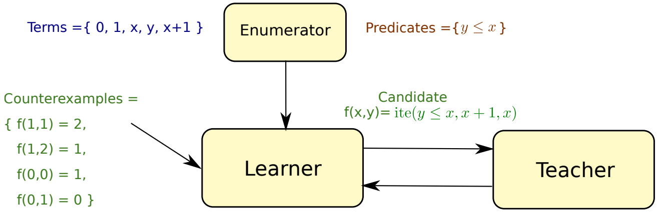
Enumerative CEGIS is effective but limited by the explosion of the enumeration space as term size increases

For this bit-vector grammar, enumerating

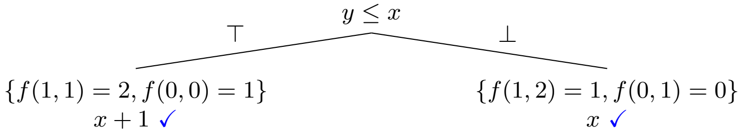
- Terms of size $= 1$: .05 seconds
- Terms of size $= 2$: .6 seconds
- Terms of size $= 3$: 48 seconds
- Terms of size $= 4$: 5.8 hours
- Terms of size $= 5$: ??? (100+ days)

```
(synth-fun f ((s (BitVec 4))
              (t (BitVec 4)))
 (BitVec 4) (
 (Start (BitVec 4) (
  s t #x0
  (bvneg  Start)
  (bvnot  Start)
  (bvadd  Start Start)
  (bvmul  Start Start)
  (bvand  Start Start)
  (bvlshr Start Start)
  (bvor   Start Start)
  (bvshl  Start Start))))
```

Divide and conquer SyGuS [ARU17; NSM18; BRLT19]



- Generate partial solutions correct on examples seen so far
- Unify partial solutions (e.g. via decision tree learning)



- D&C provides much better scalability

Scalable syntax-guided synthesis for abduction

- We extend the procedure by unifying partial solutions into conjunctions
- Besides P also maintains
  - a set E of enumerated formulas
- Candidates C are subsets of E such that
  - For every point in P at least one element of C is false
    Otherwise $\textsc{eval}(\mathsf{Ax}[\bar{p}] \wedge \mathsf{C}[\bar{p}] \wedge \neg \mathsf{G}[\bar{p}]) = \top$

Scalable syntax-guided synthesis for abduction

- We extend the procedure by unifying partial solutions into conjunctions
- Besides P also maintains
  - a set E of enumerated formulas
  - a set U of subsets of E which are inconsistent with Ax
- Candidates C are subsets of E such that
  - For every point in P at least one element of C is false

    Otherwise $\textsc{eval}(\mathsf{Ax}[\bar{p}] \wedge \mathsf{C}[\bar{p}] \wedge \neg \mathsf{G}[\bar{p}]) = \top$
  - no subset of C occurs in U

    Otherwise C is inconsistent with Ax
- Leverages unsat cores to improve eagerly discarding candidates

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$			

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$			$\{x \geq 0\}$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$			$\{x \geq 0\}$

$x \geq 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-SAT, $p_1 = (0, 0, -1)$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$			

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0\}$
$x < 0$			

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0\}$
$x < 0$			

$$\textsc{eval}(x < 0,\, p_1) = \perp$$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0\}$
$x < 0$			

$x < 0 \wedge y \geq 0 \wedge x + y + z \ngeq 0$ is T-SAT, $p_2 = (-1, 0, 0)$

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$		$p_2 = (-1, 0, 0)$	

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0, x \geq 0\}$
$x < 0$		$p_2 = (-1, 0, 0)$	

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0, x \geq 0\}$
$x < 0$		$p_2 = (-1, 0, 0)$	

$\text{EVAL}(x < 0,\ p_1) = \perp, \text{EVAL}(x \geq 0,\ p_2) = \perp$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x < 0, x \geq 0\}$
$x < 0$		$p_2 = (-1, 0, 0)$	

$x < 0 \land x \geq 0 \land y \geq 0$ is T-unsat, $u = \{x < 0, x \geq 0\}$

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{y \geq 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{y \geq 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			

$\text{EVAL}(y \geq 0,\ p_1) = \top, \text{EVAL}(y \geq 0,\ p_2) = \top$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$			

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{y < 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$			

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{y < 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$			

$\text{EVAL}(y < 0,\, p_1) = \perp, \text{EVAL}(y < 0,\, p_2) = \perp$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{y < 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$			

$y < 0 \land y \geq 0$ is T-unsat, $u = \{y < 0\}$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		

Consider again $\text{Ax} = \{y \geq 0\}$ and $\text{G} = \{x + y + z \geq 0\}$

$e \in \text{E}$	U	$p \in \text{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		
$z \geq 0$			

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x \geq 0, z \geq 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		
$z \geq 0$			

Consider again $\mathsf{Ax} = \{y \geq 0\}$ and $\mathsf{G} = \{x + y + z \geq 0\}$

$e \in \mathsf{E}$	U	$p \in \mathsf{P}$	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x \geq 0, z \geq 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		
$z \geq 0$			

$\text{EVAL}(z \geq 0,\, p_1) = \bot, \text{EVAL}(x \geq 0,\, p_2) = \bot$

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x \geq 0, z \geq 0\}$
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		
$z \geq 0$			

$x \geq 0 \wedge z \geq 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-UNSAT and
$x \geq 0 \wedge z \geq 0 \wedge y \geq 0$ is T-SAT

Consider again Ax $= \{y \geq 0\}$ and G $= \{x + y + z \geq 0\}$

$e \in$ E	U	$p \in$ P	C
$x \geq 0$		$p_1 = (0, 0, -1)$	$\{x \geq 0, z \geq 0\}$ ✓
$x < 0$	$\{x < 0, x \geq 0\}$	$p_2 = (-1, 0, 0)$	
$y \geq 0$			
$y < 0$	$\{y < 0\}$		
$z \geq 0$			

$x \geq 0 \wedge z \geq 0 \wedge y \geq 0 \wedge x + y + z \not\geq 0$ is T-UNSAT and
$x \geq 0 \wedge z \geq 0 \wedge y \geq 0$ is T-SAT

Incremental weakening

- Algorithms can be extended to incrementally generate weaker solutions

- Generate new abduct C and test if C $\wedge$ Ax $\wedge \neg S$ is T-satisfiable
  - If yes, then C has more models than S consistent with Ax
  - S is updated to S $\vee$ C

Evaluation

Setup

- Two configurations of CVC4SY

CVC4SY+B	abduction via enumerative CEGIS
CVC4SY+U	abduction via divide and conquer

- GPiD and EXPLAIN as baselines
- 300s timeout, 8gb RAM, Intel E5-2637 v4 CPUs, Ubuntu 16.04
- We had to generate our own benchmarks
  - No existing standard benchmark library for abduction
  - Integration into verification tools was beyond the scope of this work

Full data at `http://cvc4.cs.stanford.edu/papers/abduction-sygus/`

Benchmark selection

- Three relevant-to-verification SMT-LIB logics:
  - QF_LIA, QF_NIA, and QF_SLIA
- From SAT benchmarks $\varphi = \psi_1 \wedge \cdots \wedge \psi_n$:
  - $\underbrace{\psi_1 \wedge \cdots \wedge \psi_{n-1}}_{\text{Axioms}} \wedge \underbrace{\mathsf{S}}_{\text{Solution}} \models \underbrace{\neg\psi_n}_{\text{Goal}}$
- Grammars are generated based on logic and benchmark variables
- Example of grammar for QF_LIA problem with variables $x_1, \ldots, x_n$:

$$\begin{array}{lcl} A & \to & x_1 \mid \cdots \mid x_n \mid 0 \mid 1 \mid A + A \mid A - A \mid \mathsf{ite}(B,\ A,\ A) \\ B & \to & A \geq A \mid A \simeq A \mid \neg B \end{array}$$

Finding missing assumptions in SAT benchmarks

		cvc4sy+b		cvc4sy+u	
Logic	#	Solved	Unique	Solved	Unique
QF_SLIA	11954	10902	3	10980	81
QF_LIA	2025	721	261	594	134
QF_NIA	12214	1492	171	1712	391
Total	26593	13329	435	13628	606

- Can any solution be found in 300s?
  - Orthogonality in QF_{LIA,NIA} probably due to fragility of integer solvers

Finding missing assumptions in SAT benchmarks

		CVC4SY+B			CVC4SY+U		
Logic	#	Solved	Unique	Weaker	Solved	Unique	Weaker
QF_SLIA	11954	10902	3	466	10980	81	0
QF_LIA	2025	721	261	183	594	134	2
QF_NIA	12214	1492	171	671	1712	391	45
Total	26593	13329	435	1320	13628	606	47

- Can any solution be found in 300s?
  - Orthogonality in QF_{LIA,NIA} probably due to fragility of integer solvers
- Who finds weaker solution overall?
  - CVC4SY+U has better success rate but often produces stronger solutions

Comparison with Explain

- Restricted to QF_LIA as it was better supported by Explain
- Explain solves harder problem: solutions with minimal variable set

- Can any solution be found in 300s?

	Solved	Unique	Total time
cvc4sy+b	721	261	418849s
cvc4sy+u	594	125	449424s
Explain	33	0	532839s

Comparison with Explain

- Restricted to QF_LIA as it was better supported by Explain
- Explain solves harder problem: solutions with minimal variable set

- Can any solution be found in 300s?

	Solved	Unique	Total time
cvc4sy+b	721	261	418849s
cvc4sy+u	594	125	449424s
Explain	33	0	532839s

- In incremental mode cvc4sy+u finds solution with minimal number of variables to 25 of the 33 Explain solves

Comparison with GPID

- Restricted to 400 satisfiable QF_UFLIA benchmarks used in [EPS18]
  - While GPID's method is theory agnostic, their tooling restricts usage
- GPID solves similar problem: implicates for satisfiable benchmarks
  - No axioms, goal is whole original formula
  - Uses pre-computed *abduces*. GPID-1 restricts abduces to size 1
- Can any solution be found in 300s?

	Solved	Unique	Total time
CVC4SY+B	214	0	57290s
CVC4SY+U	342	0	18735s
GPID	193	0	69s
GPID-1	398	54	1188s

Comparison with GPID

- Restricted to 400 satisfiable QF_UFLIA benchmarks used in [EPS18]
  - While GPID's method is theory agnostic, their tooling restricts usage
- GPID solves similar problem: implicates for satisfiable benchmarks
  - No axioms, goal is whole original formula
  - Uses pre-computed *abduces*. GPID-1 restricts abduces to size 1
- Can any solution be found in 300s?

	Solved	Unique	Total time
CVC4SY+B	214	0	57290s
CVC4SY+U	342	0	18735s
GPID	193	0	69s
GPID-1	398	54	1188s

- GPID heavily dependent on pre-computed abduces
- CVC4SY+U 30% slower on average than GPID-1 on commonly solved

Conclusions

Conclusions

- New scalable enumerative SyGuS framework for abduction
  - General
  - Flexible

- Evaluation shows favorable comparison with abduction tools

- Future work:
  - Integration into verification engines
  - Generating conditional rewrite rules for SMT solvers
    - Synthesize most general condition under which two terms are equivalent
    - Generalizes semi-automated development of rewrite rules [NRB+19]
  - Lifting approach to interpolation

Scalable Algorithms for Abduction via Enumerative Syntax-Guided Synthesis

Andrew Reynolds, *Haniel Barbosa*, Daniel Larraz, Cesare Tinelli

IJCAR 2020

2020–07–04, The Internet

Completing UNSAT cores

- From SMT-LIB benchmarks with minimal unsat cores U:
  - $U \setminus \{\psi_{\mathsf{G}}, \psi_{\max}\} \wedge \mathsf{S} \models \neg\psi_{\mathsf{G}}$, with S as weak as $\psi_{\max}$, where
  - $\psi_{\max}$ is U's component with maximal size, used as a *reference*
  - $\psi_{\mathsf{G}} \in U \setminus \{\psi_{\max}\}$
- We chose the goal as the last formula in the core (viewed as a list) after the reference was removed
- We used Z3 to compute minimal unsat cores (120s)
- Excluded cores with less than three assertions

Completing UNSAT cores

- Can any solution at least as weak as the reference be found in 300s?

		CVC4SY+B		CVC4SY+U	
Logic	#	Solved	Unique	Solved	Unique
QF_LIA	97	6	0	6	0
QF_SLIA	2546	2546	32	2514	0
QF_NIA	781	86	49	41	4
Total	3424	2638	81	2561	4

Completing UNSAT cores

▷ Can any solution at least as weak as the reference be found in 300s?

		CVC4SY+B		CVC4SY+U	
Logic	#	Solved	Unique	Solved	Unique
QF_LIA	97	6	0	6	0
QF_SLIA	2546	2546	32	2514	0
QF_NIA	781	86	49	41	4
Total	3424	2638	81	2561	4

▷ CVC4SY+B significantly outperforms CVC4SY+U in QF_SLIA

- Small references (generally size < 3) void need for specialized procedure

▷ Overall CVC4SY+B has an advantage for finding weaker solutions

References

Rajeev Alur, Rastislav Bodík, Garvit Juniwal, et al. "Syntax-guided synthesis". In: Formal Methods In Computer-Aided Design (FMCAD). IEEE, 2013, pp. 1–8.

Aws Albarghouthi, Isil Dillig, and Arie Gurfinkel. "Maximal specification synthesis". In: Symposium on Principles of Programming Languages (POPL). Ed. by Rastislav Bodík and Rupak Majumdar. ACM, 2016, pp. 789–801.

Rajeev Alur, Arjun Radhakrishna, and Abhishek Udupa. "Scaling Enumerative Program Synthesis via Divide and Conquer". In: Tools and Algorithms for Construction and Analysis of Systems (TACAS). Ed. by Axel Legay and Tiziana Margaria. Vol. 10205. Lecture Notes in Computer Science. 2017, pp. 319–336.

Haniel Barbosa, Andrew Reynolds, Daniel Larraz, et al. "Extending enumerative function synthesis via SMT-driven classification". In: Formal Methods In Computer-Aided Design (FMCAD). Ed. by Clark W. Barrett and Jin Yang. IEEE, 2019, pp. 212–220.

Isil Dillig and Thomas Dillig. "Explain: A Tool for Performing Abductive Inference". In: Computer Aided Verification (CAV). Ed. by Natasha Sharygina and Helmut Veith. Vol. 8044. Lecture Notes in Computer Science. Springer, 2013, pp. 684–689.

References

Isil Dillig, Thomas Dillig, and Alex Aiken. "Automated error diagnosis using abductive inference". In: ACM SIGPLAN Conference on Programming Language Design and Implementation, P Ed. by Jan Vitek, Haibo Lin, and Frank Tip. ACM, 2012, pp. 181–192.

Thomas Dillig, Isil Dillig, and Swarat Chaudhuri. "Optimal Guard Synthesis for Memory Safety". In: Computer Aided Verification (CAV). Ed. by Armin Biere and Roderick Bloem. Vol. 8559. Lecture Notes in Computer Science. Springer, 2014, pp. 491–507.

Isil Dillig, Thomas Dillig, Boyang Li, et al. "Inductive invariant generation via abductive inference". In: International Conference on Object Oriented Programming Systems Languages & App Ed. by Antony L. Hosking, Patrick Th. Eugster, and Cristina V. Lopes. ACM, 2013, pp. 443–456.

Mnacho Echenim, Nicolas Peltier, and Yanis Sellami. "A Generic Framework for Implicate Generation Modulo Theories". In: International Joint Conference on Automated Reasoning (IJCAR). Ed. by Didier Galmiche, Stephan Schulz, and Roberto Sebastiani. Vol. 10900. Lecture Notes in Computer Science. Springer, 2018, pp. 279–294.

References

Mnacho Echenim, Nicolas Peltier, and Yanis Sellami. "Ilinva: Using Abduction to Generate Loop Invariants". In: Frontiers of Combining Systems (FroCoS). Ed. by Andreas Herzig and Andrei Popescu. Vol. 11715. Lecture Notes in Computer Science. Springer, 2019, pp. 77–93.

Boyang Li, Isil Dillig, Thomas Dillig, et al. "Synthesis of Circular Compositional Program Proofs via Abduction". In: Tools and Algorithms for Construction and Analysis of Systems (TACAS). Ed. by Nir Piterman and Scott A. Smolka. Vol. 7795. Lecture Notes in Computer Science. Springer, 2013, pp. 370–384.

Andres Nötzli, Andrew Reynolds, Haniel Barbosa, et al. "Syntax-Guided Rewrite Rule Enumeration for SMT Solvers". In: Theory and Applications of Satisfiability Testing (SAT). Ed. by Mikolás Janota and Inês Lynce. Vol. 11628. Lecture Notes in Computer Science. Springer, 2019, pp. 279–297.

Daniel Neider, Shambwaditya Saha, and P. Madhusudan. "Compositional Synthesis of Piece-Wise Functions by Learning Classifiers". In: ACM Trans. Comput. Log. 19.2 (2018), 10:1–10:23.

References

Andrew Reynolds, Haniel Barbosa, Andres Nötzli, et al. "cvc4sy: Smart and Fast Term Enumeration for Syntax-Guided Synthesis". In: Computer Aided Verification (CAV), Part II. Ed. by Isil Dillig and Serdar Tasiran. Vol. 11562. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 74–83.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodík, et al. "Combinatorial sketching for finite programs". In: Architectural Support for Programming Languages and Operating Systems (ASPLOS). Ed. by John Paul Shen and Margaret Martonosi. ACM, 2006, pp. 404–415.

Abhishek Udupa, Arun Raghavan, Jyotirmoy V. Deshmukh, et al. "TRANSIT: specifying protocols with concolic snippets". In: Conference on Programming Language Design and Implementation (PLDI). Ed. by Hans-Juergen Boehm and Cormac Flanagan. ACM, 2013, pp. 287–296.

Haiyan Zhu, Thomas Dillig, and Isil Dillig. "Automated Inference of Library Specifications for Source-Sink Property Verification". In: Programming Languages and Systems - 11th Asian Symposium, APLAS 2013, Melbo Ed. by Chung-chieh Shan. Vol. 8301. Lecture Notes in Computer Science. Springer, 2013, pp. 290–306.