Specification Synthesis via Generalized Abduction

Aws Albarghouthi
University of Toronto

Specification Synthesis

$$P \overset{?}{\models} \Phi$$

Program (single procedure)
has calls to unknown procedures

Safety property

Specification Synthesis

$$P[f \mapsto \varphi_f] \models \Phi$$

Specification Synthesis

```
y = 10
x = foo()
z = x + y
assert(z >= 0)
```

$$\varphi_{foo}$$

$$R_{foo} = 0$$

$$R_{foo} = 1$$

$$false$$

$$R_{foo} \geqslant -10$$

maximal

Specification Synthesis

```
x = fx()
y = fy()
assert(x+y >= 0)
```

$$\varphi_{fx} \quad \varphi_{fy}$$

$$R_{fx} \geqslant 0 \quad R_{fy} \geqslant 0$$

$$R_{fx} \geqslant -1 \quad R_{fy} \geqslant 1$$

Specification Synthesis

$$P[f \mapsto \varphi_f] \models \Phi$$

$$s.t.\ \text{maximal}(\vec{\varphi}_f)$$

Why Spec. Synthesis?

Verification in the presence of absence

Programs have calls to external libraries / proprietary code

Top-down modular verification

(1) Verify main, (2) compute specs of callees, (3) verify specs

Code synthesis

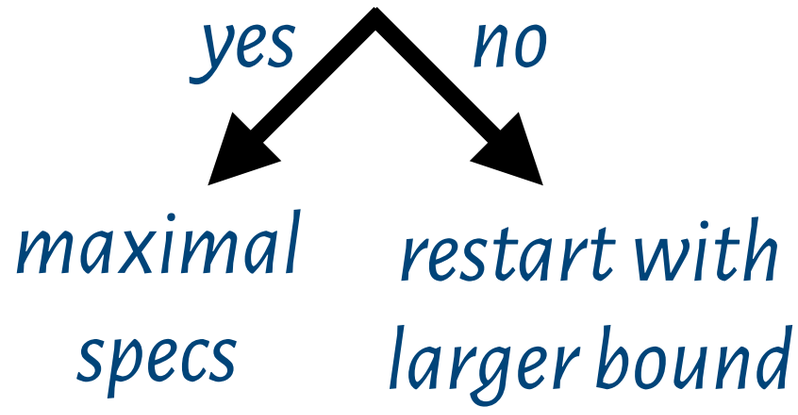
(1) Specs fill holes, (2) transform specs into code

How Spec. Synthesis?

$$P_n = \mathsf{unroll}(P, n)$$

$$\mathcal{M} = \mathsf{spec}(P_n, \Phi)$$

$$\text{Is } P[\mathcal{M}] \models \Phi?$$



Bounded Synthesis

$$\mathsf{enc}_{P_n} \Rightarrow \Phi$$

$$\bigwedge \bigvee \{a, b, \ldots, \boxed{K(\vec{x})}, \boxed{F(\vec{y})}\} \Rightarrow \Phi$$

CNF formula, where some atoms are unknown predicates

Generalized Abduction

Classical abduction

$$B \not\Rightarrow C$$

$$A(\vec{x}) \wedge B \Rightarrow C$$

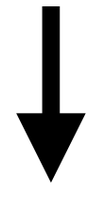
Abducible

Maximum solution

$$\forall v \notin \vec{x}(B \Rightarrow C)$$

Generalized abduction

$$\bigwedge \bigvee \{a, b, \ldots, K(\vec{x}), F(\vec{y})\} \Rightarrow \Phi$$



Multiple abducibles

Abducibles appear under arbitrary connectives

Same abducible can appear multiple times with different arguments

Generalized Abduction

Step 1: flatten and compute upper bound

$$\bigwedge \bigvee \{a, b, \ldots, K(\vec{x}), F(\vec{y})\} \Rightarrow \Phi$$

Transformations!

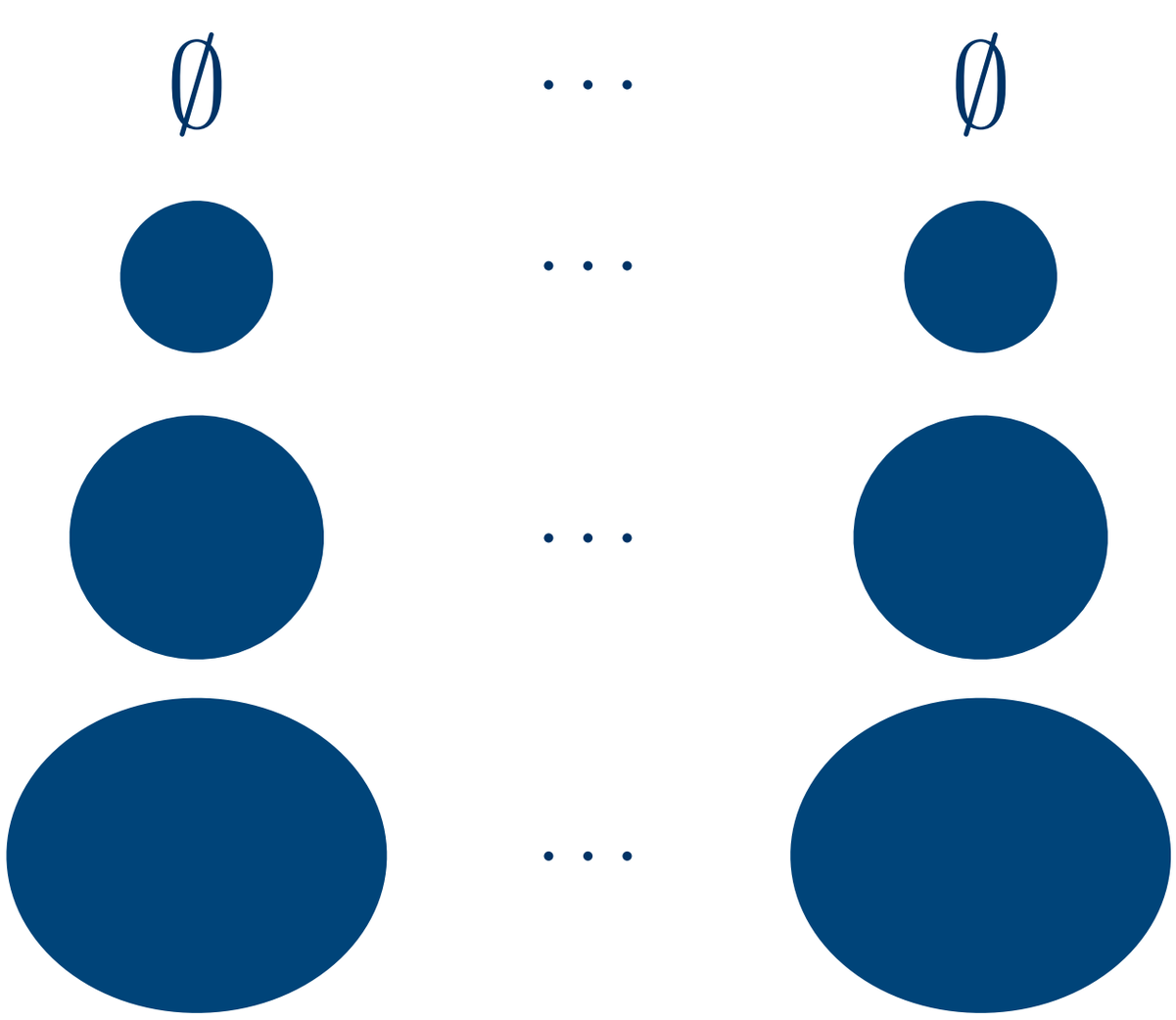
$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \Rightarrow \Psi$$

Generalized Abduction

Step 2: cartesian decomposition

For now, assume all abducibles are unique

$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \Rightarrow \Psi$$



Generalized Abduction

Step 2: cartesian decomposition

For now, assume all abducibles are unique

$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \Rightarrow \Psi$$

$$\begin{array}{ccc} false & \cdots & false \\ \vec{x_1} = \vec{m}_1 & \cdots & \vec{x_n} = \vec{m}_n \\ \bigvee \begin{array}{l} \vec{x_1} = \vec{m}_1 \\ \vec{x_1} = \vec{m'}_1 \end{array} & \cdots & \bigvee \begin{array}{l} \vec{x_n} = \vec{m}_n \\ \vec{x_n} = \vec{m'}_n \end{array} \end{array}$$

Seed relations with a model of Ψ

Generalized Abduction

Step 2: cartesian decomposition

$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \Rightarrow \Psi$$

$$R_1^\sigma \qquad \cdots \qquad R_n^\sigma$$

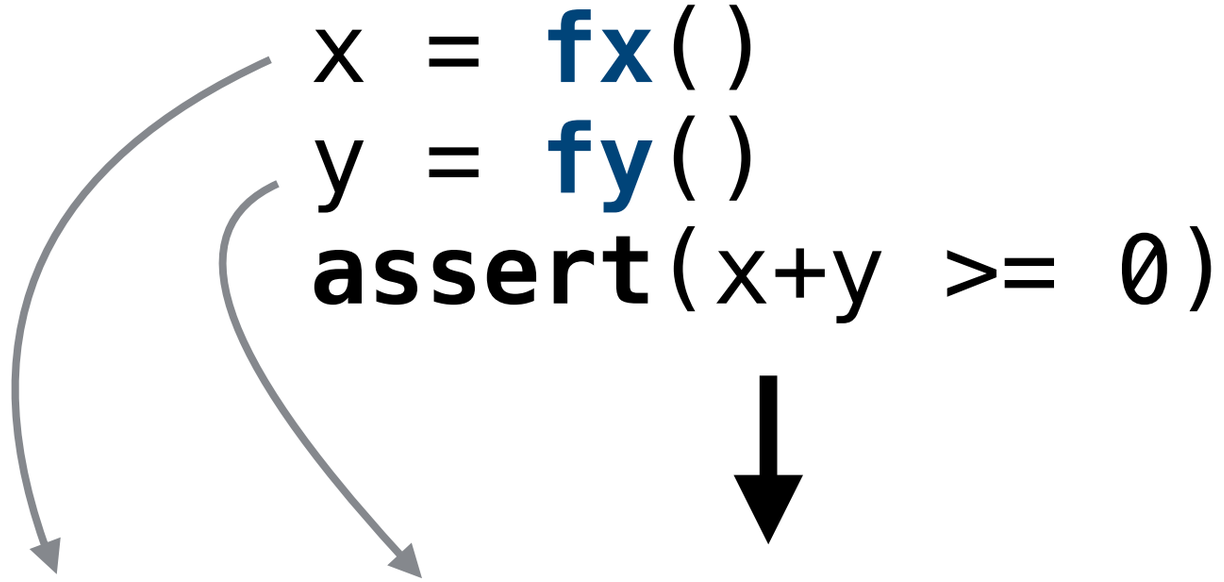
$$R_1(\vec{x_1}) \wedge B \Rightarrow \Psi$$

Solve this classical abduction problem

Solution is maximal for R_1

$$\widehat{R_1^\sigma} \qquad \cdots \qquad R_n^\sigma$$

Example



$$R_1(x) \wedge R_2(y) \Rightarrow x + y \geqslant 0$$

Example

$$R_1(x) \wedge R_2(y) \Rightarrow x + y \geqslant 0$$

$false$ $false$

$x = -1$ $y = 1$

$R_1(x) \wedge y = 1 \Rightarrow x + y \geqslant 0$ *Generalize!*

$x \geqslant -1$ $y = 1$

$R_2(y) \wedge x \geqslant -1 \Rightarrow x + y \geq 0$ *Generalize!*

$x \geqslant -1$ $y \geqslant 1$ *Maximal solution!*


WHERE ARE
THE INTERPOLANTS?
memegenerator.net

EXAMPLE (W/ INTERPOLANTS)

$$R_1(x) \wedge R_2(y) \Rightarrow x + y \geqslant 0$$

$$\underbrace{\overbrace{x = -1}^{false} \wedge \overbrace{y = 1}^{false} \wedge \neg x + y < 0}_{UNSAT}$$

$$\underbrace{x = -1}_{A} \wedge \underbrace{y = 1 \wedge \neg x + y < 0}_{B}$$

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

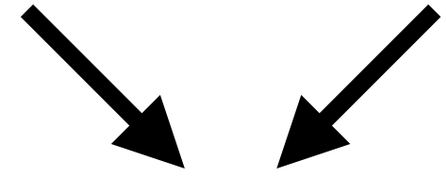
Symmetric Decomposition

$$R(\vec{x}) \wedge R(\vec{y}) \Rightarrow \Psi$$

Example

$$R(x) \wedge R(y) \Rightarrow x + y \geqslant 0$$

$$x \geqslant 0 \qquad y \geqslant 0$$



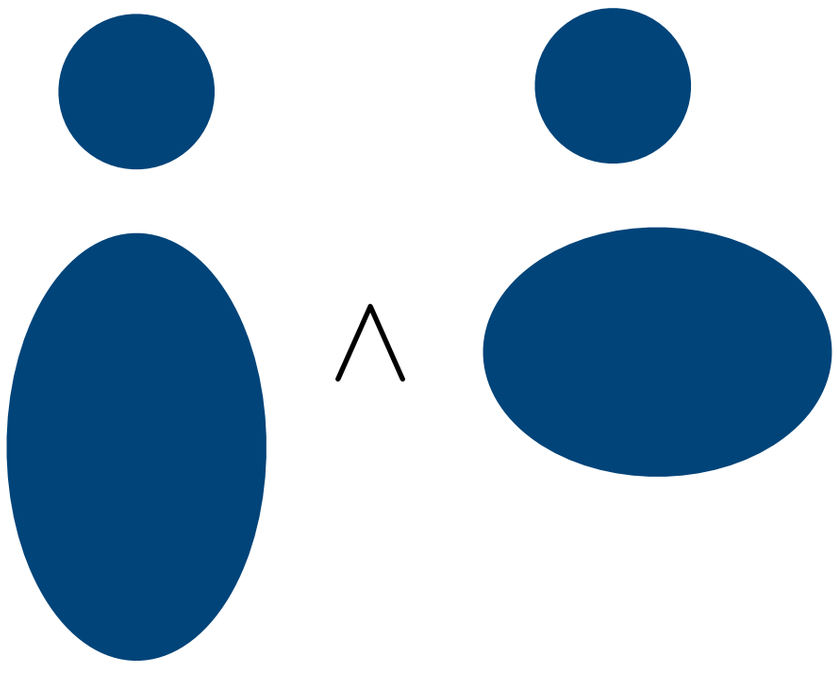
same formula (modulo renaming)

maximal solution

Symmetric Decomposition

$$R(\vec{x}) \wedge R(\vec{y}) \Rightarrow \Psi$$

$\emptyset$ $\emptyset$



Symmetric Decomposition

$$R(\vec{x}) \wedge R(\vec{y}) \Rightarrow \Psi$$

In the propositional case, keep adding models

In theories, there are cases with infinite ascending chains...

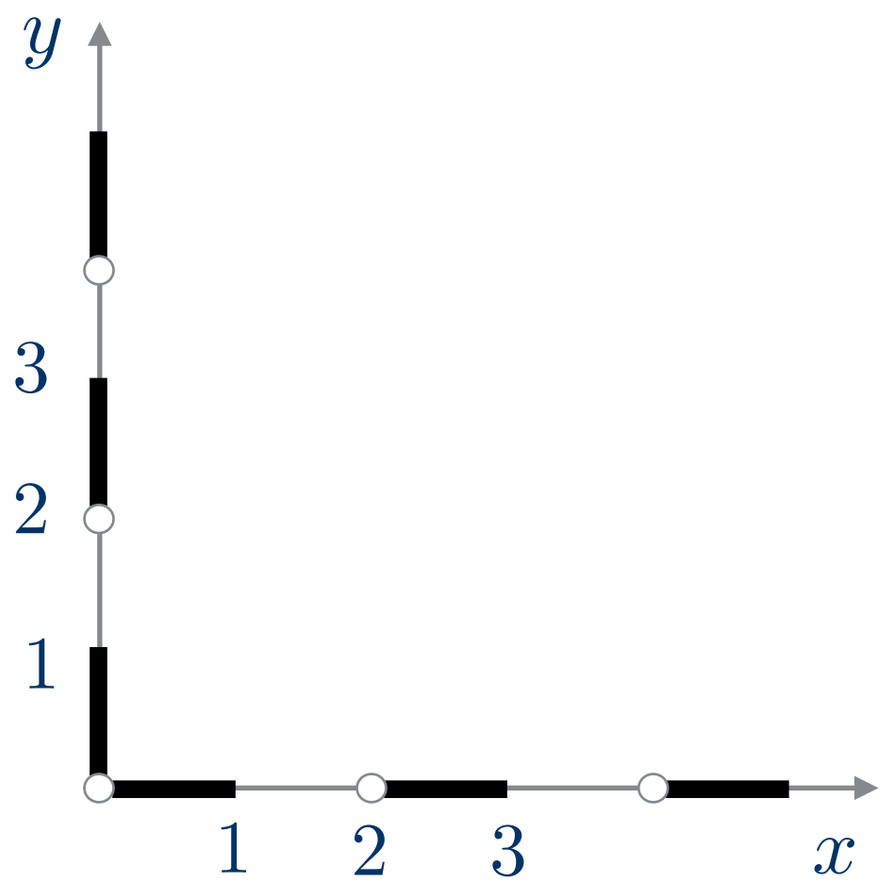
Symmetric Decomposition

Example (*thanks to Ken McMillan*)

$$R(x) \wedge R(y) \Rightarrow x \neq y - 1$$

$0 < x \leqslant 1$ $\quad$ $0 < y \leqslant 1$

$$\bigvee \begin{matrix} 0 < x \leqslant 1 \\ 2 < x \leqslant 3 \end{matrix} \qquad \bigvee \begin{matrix} 0 < y \leqslant 1 \\ 2 < y \leqslant 3 \end{matrix}$$



Conclusion

Specification synthesis

maximal specs

Reduction to generalized abduction

CEGIS loop

A technique for solving generalized abduction

tough stuff

Generalized Abduction

Step 1: flatten and compute upper bound

$$\bigwedge \bigvee \{a, b, \ldots, K(\vec{x}), F(\vec{y})\} \Rightarrow \Phi$$

syntactic transformation

$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \wedge B \Rightarrow \Phi$$

classical abduction

$$A(\vec{x}) \wedge B \Rightarrow \Phi$$

$$R_1(\vec{x_1}) \wedge \cdots \wedge R_n(\vec{x_n}) \Rightarrow \Psi$$