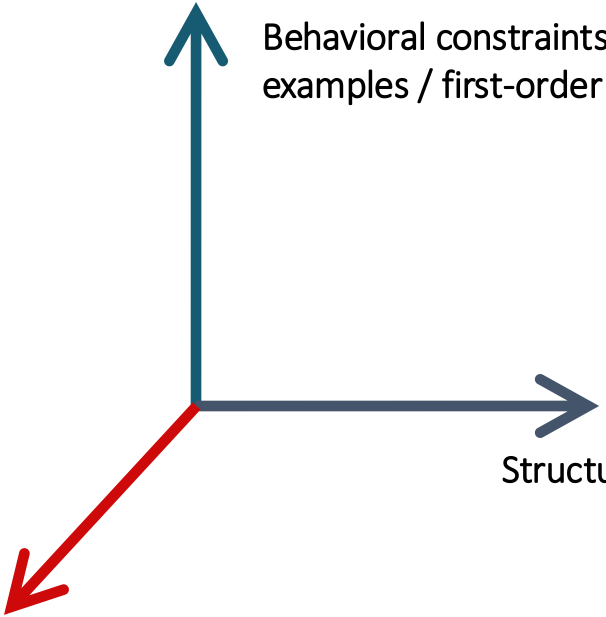
Lecture 9
Unrealizability

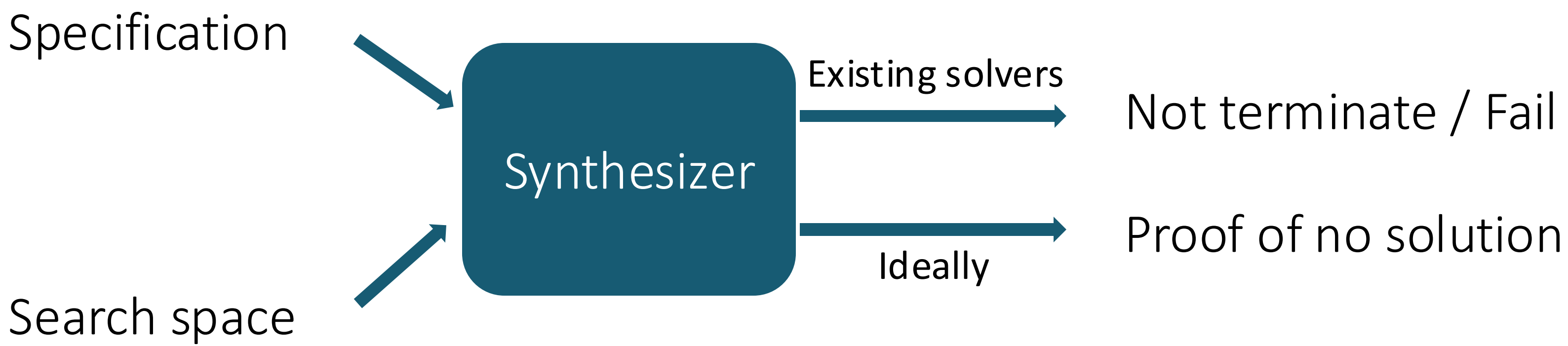
The problem statement

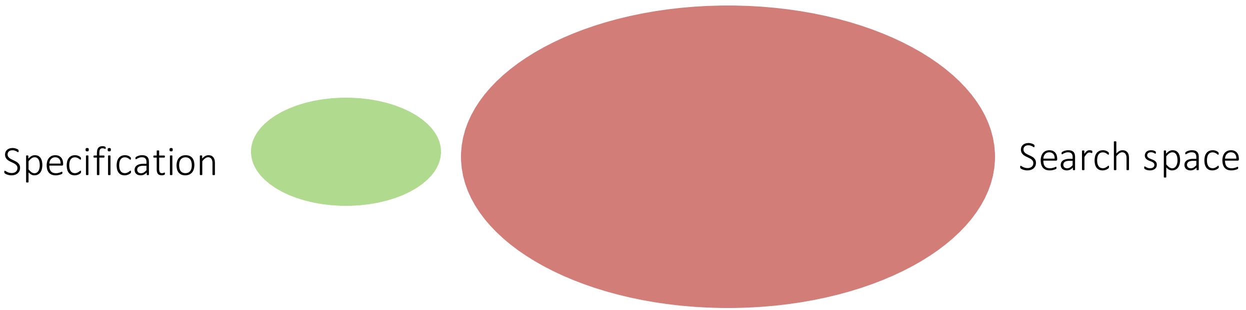


Search strategy?

Enumerative
Representation-based
Stochastic
Constraint-based
Invariant-based

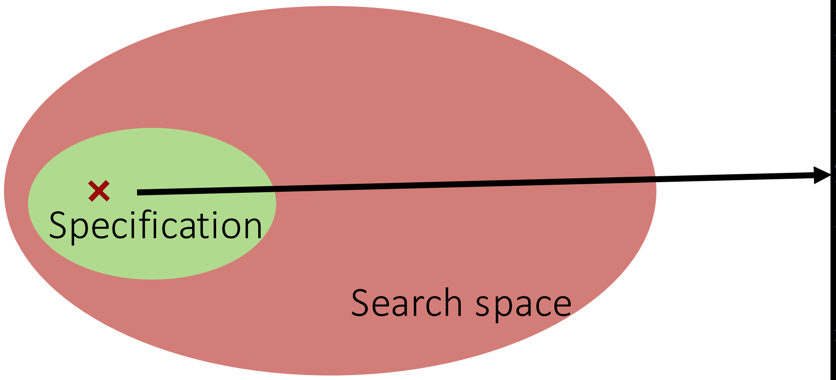
Unrealizable Synthesis Problems

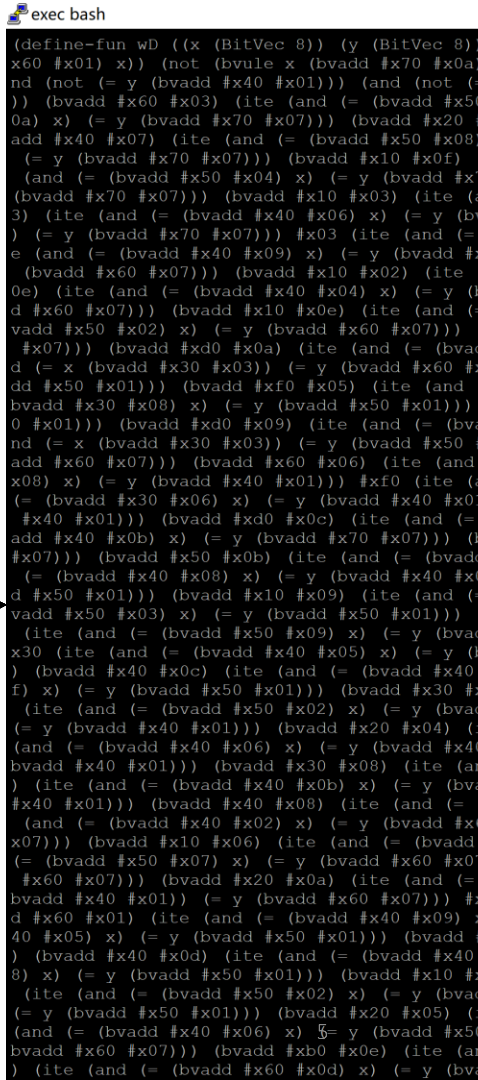




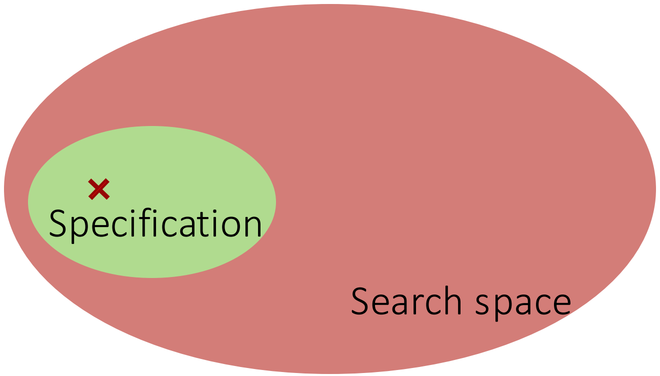
Why Prove Unrealizability?

From Proving Optimality to Proving Unrealizability [pldi17, cav18]

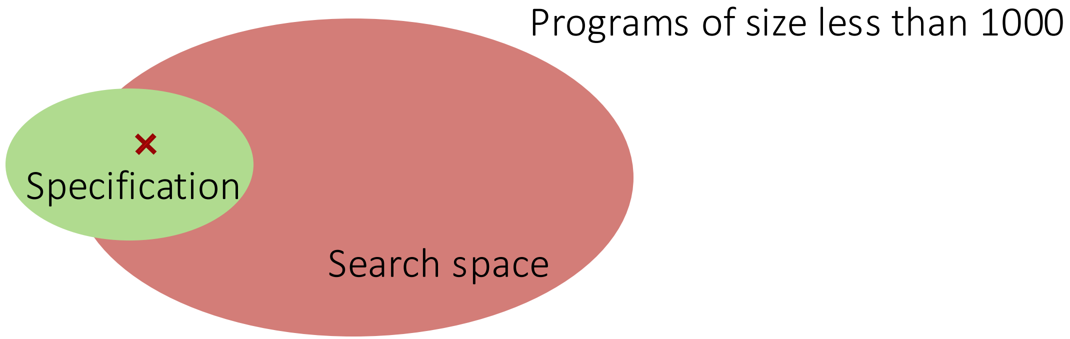




From Proving Optimality to Proving Unrealizability [pldi17, cav18]



From Proving Optimality to Proving Unrealizability [pldi17, cav18]

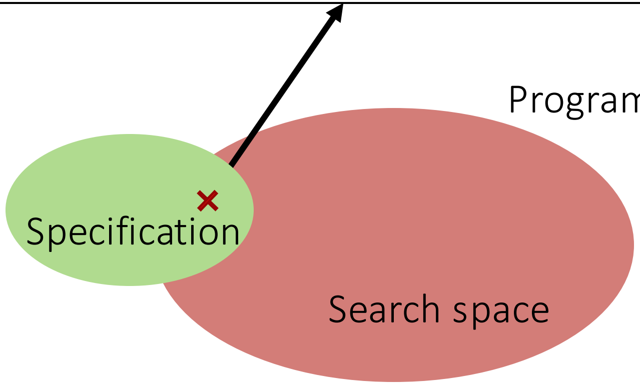


From Proving Optimality to Proving Unrealizability [pldi17, cav18]

Size = 9

```
(define-fun ((x (BitVec 8)) (y (BitVec 8))) (bvand (bvlshl (DD x) #x02) (bvlshr (DD y) #x06)))
```

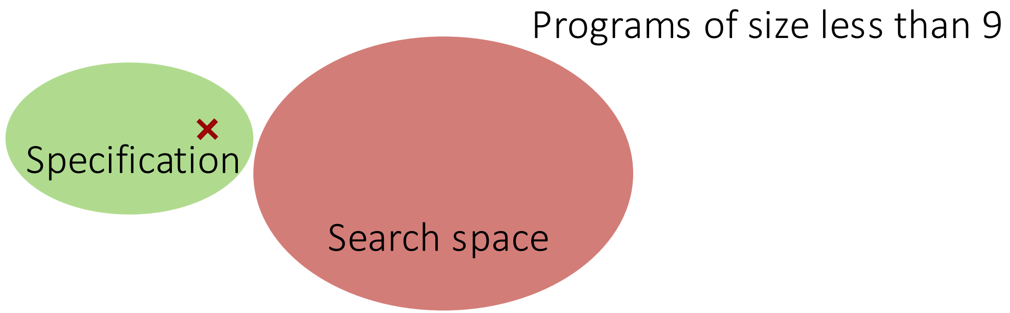
Programs of size less than 100



From Proving Optimality to Proving Unrealizability [pldi17, cav18]

Size = 9

```
(define-fun ((x (BitVec 8)) (y (BitVec 8))) (bvand (bvlshl (DD x) #x02) (bvlshr (DD y) #x06)))
```



× is optimal **iff** this synthesis problem is **unrealizable**

Why is this hard?

Specification

$$\Phi(P) : \forall x, y. P \geq x \wedge P \geq y \\ \wedge (P = x \vee P = y)$$

Search space

Start := Start+Start

| x | y | 0 | 1

SyGuS solver

Unrealizable

Proof that

$$\neg \exists Q \ s.t. \quad \Phi(Q) \\ \wedge \quad Q \in L(G)$$

Infinite grammar makes the problem undecidable

Soundness of CEGIS for unrealizability

$Sy{\uparrow}E$ unrealizable $\Rightarrow$ Sy unrealizable

No solution over E $\Rightarrow$ No solution

Proving unrealizability for SyGuS over examples

Outline of the algorithm

$$Sy{\uparrow}E := (\Phi, G, E)$$

construct

```
int[4] Start(x_0,y_0,x_1,y_1,x_2,y_2,x_3,y_3){
    if(??){return (0,0,0,0);}            // Start -> 0
    if(??){return (1,1,1,1);}            // Start -> 1
    if(??){return (x_0,x_1,x_2,x_3);}    // Start -> x
    if(??){return (y_0,y_1,y_2,y_3);}    // Start -> y
    else{                                // Start -> Start + Start
        int[4] L = Start(x_0,y_0,x_1,y_1);
        int[4] R = Start(x_0,y_0,x_1,y_1);
        return (L[0]+R[0],L[1]+R[1],L[2]+R[2],L[3]+R[3]);}
}
int[4] P = Start(0,0,0,1,1,0,2,0);
assert (P[0]!=0 || P[1]!=1 || P[2]!=1 || P[3]!=2);
```

$Sy{\uparrow}E$ unrealizable ⟷ `assert` always holds

Reachability Problem

Nondeterministic choice

```
void main(){
      int x = 0;
      while(nd()){
            x++;
      }
      assert(x<0)
}
```

Reachability solvers:

CPA-checker

Uautomizer

Seahorn

Goal: can the `assert` be falsified?

Sy^E to Re^E

Set input to E

$$\vec{x} \leftarrow E$$

f_G is non-deterministically drawn from $L(G)$

$$\vec{o} \leftarrow f_G(\vec{x})$$

Check if $\vec{o}$ doesn't satisfy φ ⟷ $f_G(\vec{x})$ satisfy φ on E

```
assert(
```
$\neg\varphi(o, x) \quad , x_i))$

$Sy{\uparrow}E$ unrealizable

Set input to E

$$\vec{x} \leftarrow E$$

Examples E:
(x0,y0)=(0,0)
(x1,y1)=(0,1)

```
x0 = 0;
y0 = 0;
x1 = 0;
y1 = 1;
```

Sy^E to Re^E

Set input to E

$$\vec{x} \leftarrow E$$

f_G is non-deterministically drawn from $L(G)$

$$\vec{o} \leftarrow f_G(\vec{x})$$

Check if $\vec{o}$ doesn't satisfy φ

assert($\neg\varphi(o,\bar{x})$ ´ $, x_i))$

Check if $\vec{o}$ doesn't satisfy φ

$$\texttt{assert}(\neg \bigwedge x_i \in E.\, \varphi(o_i, x_i))$$

```
void main(){
    …
    assert(!(spec(x0,y0,o0)&&spec(x1,y1,o1)));
}
bool spec(x,y,o){
    return (o>=x)&&(o>=y)&&(o==x||o==y);
}
```

$$\Phi(f) : \forall x, y. f(x, y) \geq x \wedge f(x, y) \geq y \wedge (f(x, y) = x \vee f(x, y) = y)$$

Sy^E to Re^E

Set input to E

$$\vec{x} \leftarrow E$$

f_G is non-deterministically drawn from $L(G)$

$$\vec{o} \leftarrow f_G(\vec{x})$$

Check if $\vec{o}$ doesn't satisfy φ

`assert(` $\neg\varphi(o, \bar{x})$ $, x_i))$

f_G is non-deterministically drawn from $L(G)$

$$\vec{o} \leftarrow f_G(\vec{x})$$

```
o0 = fStart(x0,y0);
```

```
int fStart(x0,y0){
   if(nd()){ return 0;}   \\ Start -> 0
   if(nd()){ return 1;}   \\ Start -> 1
   if(nd()){ return x0;}  \\ Start -> x
   if(nd()}{ return y0;}  \\ Start -> y
   if(nd()){              \\ Start -> +(Start,Start)
      left = fStart(x0,y0);
      right = fStart(x0,y0);
      return left + right;}
}
```

Example 0

`o0=fStart(x0,y0);`

`o0 is` f_G`(x0,y0)for some` f_G `in` $L(G)$

Example 1

`o1=fStart(x1,y1);`

`o1 is` f_G`(x1,y1)for some` f_G `in` $L(G)$

The two f_G can be different!

f_G is non-deterministically drawn from $L(G)$

$$\vec{o} \leftarrow f_G(\vec{x})$$

```
(o0,o1) = fStart(x0,y0,x1,y1);

<int,int> fStart(x0,y0,x1,y1){
   if(nd()){ return (0,0);}    \\ Start -> 0
   if(nd()){ return (1,1);}    \\ Start -> 1
   if(nd()){ return (x0,x1);}  \\ Start -> x
   if(nd()}{ return (y0,y1);}  \\ Start -> y
   if(nd()){                   \\ Start -> +(Start,Start)
      (a0,a1) = fStart(x0,y0,x1,y1);
      (b0,b1) = fStart(x0,y0,x1,y1);
      return (a0+b0,a1+b1);}
}
```

Outline of the algorithm

$$Sy{\uparrow}E := (\Phi, G, E)$$

construct

```
int[4] Start(x_0,y_0,x_1,y_1,x_2,y_2,x_3,y_3){
    if(??){return (0,0,0,0);}          // Start -> 0
    if(??){return (1,1,1,1);}          // Start -> 1
    if(??){return (x_0,x_1,x_2,x_3);}  // Start -> x
    if(??){return (y_0,y_1,y_2,y_3);}  // Start -> y
    else{                              // Start -> Start + Start
        int[4] L = Start(x_0,y_0,x_1,y_1);
        int[4] R = Start(x_0,y_0,x_1,y_1);
        return (L[0]+R[0],L[1]+R[1],L[2]+R[2],L[3]+R[3]);}
}
int[4] P = Start(0,0,0,1,1,0,2,0);
assert (P[0]!=0 || P[1]!=1 || P[2]!=1 || P[3]!=2);
```

$Sy{\uparrow}E$ unrealizable ⟷ `assert` always holds

Sy unrealizable

Nay: Illustrative Example

[PLDI20] Exact and Approximate Methods for Proving Unrealizability of Syntax-Guided Synthesis Problems

Example of an Unrealizable Problem

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

$\mathrm{Start} \rightarrow \mathrm{Expr}_1 \mid \mathrm{Expr}_2$

Solution $\overset{?}{\in}$ $\mathrm{Expr}_1 \rightarrow x + x + \mathrm{Expr}_1 \mid 1$

Solution $\underset{?}{\in}$ $\mathrm{Expr}_2 \rightarrow x + x + x + \mathrm{Expr}_2 \mid 0$

Example of an Unrealizable Problem

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

$x = 1$ $\exists \lambda.\ 2\lambda 1 + 1 = 5$

$x = 2$ $\wedge\ 2\lambda 2 + 1 = 6$

$\mathrm{Start} \rightarrow \mathrm{Expr}_1 \mid \mathrm{Expr}_2$

Solution $\overset{?}{\in}$ $\mathrm{Expr}_1 \rightarrow x + x + \mathrm{Expr}_1 \mid 1$ $\boldsymbol{2\lambda x + 1}$: $1, 2x + 1, 4x + 1, \ldots$ $\exists \lambda \forall x.\ 2\lambda x + 1$ satisfies the specification

Solution $\overset{?}{\in}$ $\mathrm{Expr}_2 \rightarrow x + x + x + \mathrm{Expr}_2 \mid 0$

Example of an Unrealizable Problem

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

$$\overbrace{f(1)}$$

$x = 1$ $\quad \exists\lambda.\, 2\lambda 1 + 1 = 5$

$x = 2$ $\quad \wedge\, 2\lambda 2 + 1 = 6$

$$f(2)$$

Start $\rightarrow$ $\text{Expr}_1 \mid \text{Expr}_2$

Solution $\in^{?}$ $\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$

$\mathbf{2\lambda x + 1}$: $\quad 1, 2x + 1, 4x + 1, \ldots$

$\exists\lambda\forall x.\, 2\lambda x + 1$ satisfies the specification

Solution $\in^{?}$ $\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$

Example of an Unrealizable Problem

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

$$\overbrace{f(1)}$$

$x = 1$ $\quad \exists\lambda.\ 2\lambda 1 + 1 = \overbrace{5}^{f(1)}$

$x = 2$ $\quad \wedge\ \underbrace{2\lambda 2 + 1}_{\text{odd}} = \underbrace{6}_{f(2)}$

$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$

Solution ∈? $\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$

$\mathbf{2\lambda x + 1}$: $\quad 1, 2x + 1, 4x + 1, \ldots$ $\quad \exists\lambda\forall x.\ 2\lambda x + 1$ satisfies the specification

Solution ∈? $\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$

Example of an Unrealizable Problem

$f(1) = 5$
$x \neq 1 \rightarrow f(x) = 3x$

divisible by 3

$\exists \lambda.\, \underbrace{3\lambda}\,1 = 5$

$\wedge\, 3\lambda 2 = 6$

$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$

Solution ✗	$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$	$\mathbf{2\lambda x + 1}$:	$1, 2x+1, 4x+1, \ldots$	$\exists\lambda\forall x.\, 2\lambda x + 1$ satisfies the specification
Solution ✗ ?	$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$	$\mathbf{3\lambda x}$:	$0, 3x, 6x, \ldots$	$\exists\lambda\forall x.\, 3\lambda x$ satisfies the specification

Example of an Unrealizable Problem

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

	$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$	$\boldsymbol{2\lambda x + 1}$ or $\boldsymbol{3\lambda x}$
Solution e	$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$	$\boldsymbol{2\lambda x + 1}$
Solution e	$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$	$\boldsymbol{3\lambda x}$

No Solution
Unrealizable

SyGuS with Examples

SyGuS problem

sy

$$f(1) = 5$$
$$x \neq 1 \rightarrow f(x) = 3x$$

$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$

$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$

$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$

SyGuS with examples problem

sy^E where $E := \{1,2\}$

$x = 1$

$x = 2$

$$\boxed{\begin{array}{c} f(1) = 5 \\ f(2) = 6 \end{array}}$$

$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$

$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$

$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$

Algorithm for Proving Unrealizability

High-level Idea

E:

$f(1) = 5$

$f(2) = 6$

Grammar

$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$

$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1$
$\mid 1$

$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2$
$\mid 0$

⇨

Equations

$V_{\text{Start}} = \cdots$

$V_{\text{Expr}_1} = \cdots$

$V_{\text{Expr}_2} = \cdots$

⇨

Solution

$\boldsymbol{2\lambda x + 1}$ or $\boldsymbol{3\lambda x}$

$\boldsymbol{2\lambda x + 1}$

$\boldsymbol{3\lambda x}$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

$$\mathrm{Expr}_1 \to x + x + \mathrm{Expr}_1$$
$$\mid 1$$

$\boldsymbol{2\lambda x + 1}$

Substitute x with input examples

$$\mathrm{Expr}_1 \to (1, 2) + (1, 2) + \mathrm{Expr}_1$$
$$\mid (1, 1)$$

Construct equation

$$V_{\mathrm{Expr}_1} = \{(1, 2)\} + \{(1, 2)\} + V_{\mathrm{Expr}_1} \cup \{(1, 1)\}$$

Evaluate algebraic operators

$$V_{\mathrm{Expr}_1} = \{(2, 4)\} + V_{\mathrm{Expr}_1} \cup \{(1, 1)\}$$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

$$\begin{aligned}\mathrm{Expr}_1 &\rightarrow x + x + \mathrm{Expr}_1 \\ &\mid 1\end{aligned}$$

$\mathbf{2\lambda x + 1}$

$$\begin{aligned}V_{\mathrm{Expr}_1} &= \{(2, 4)\} + V_{\mathrm{Expr}_1} \\ &\cup \{(1, 1)\}\end{aligned}$$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

$\text{Start} \to \text{Expr}_1 \mid \text{Expr}_2$

$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$

$2\lambda x + 1$ or **$3\lambda x$**

$\text{Expr}_1 \to x + x + \text{Expr}_1$
$\mid 1$

$V_{\text{Expr}_1} = \{(2, 4)\} + V_{\text{Expr}_1}$
$\cup \{(1, 1)\}$

$2\lambda x + 1$

$\text{Expr}_2 \to x + x + x + \text{Expr}_2$
$\mid 0$

$V_{\text{Expr}_2} = \{(3, 6)\} + V_{\text{Expr}_2}$
$\cup \{(0, 0)\}$

$3\lambda x$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$$
$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$
$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

Equation

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$
$$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$
$$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

Solution

$\mathbf{2\lambda x + 1}$ or $\mathbf{3\lambda x}$

$\mathbf{2\lambda x + 1}$

$\mathbf{3\lambda x}$

$$V_{\text{Start}} = \{(1,1) + \lambda(2,4)\} \cup \{(0,0) + \lambda(3,6)\}$$
$$V_{\text{Expr}_1} = \{(1,1) + \lambda(2,4)\}$$
$$V_{\text{Expr}_2} = \{(0,0) + \lambda(3,6)\}$$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\text{Start} \to \text{Expr}_1 \mid \text{Expr}_2$$

$$\text{Expr}_1 \to x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \to x + x + x + \text{Expr}_2 \mid 0$$

$\Rightarrow$

Equation

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$

$$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$

$$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

Solution

$\mathbf{2\lambda x + 1}$ or $\mathbf{3\lambda x}$

$\mathbf{2\lambda x + 1}$

$\mathbf{3\lambda x}$

$$V_{\text{Start}} = \{(1,1) + \lambda(2,4)\} \cup \{(0,0) + \lambda(3,6)\}$$

$$V_{\text{Expr}_1} = \{(1,1) + \lambda(2,4)\}$$

$$V_{\text{Expr}_2} = \{(0,0) + \lambda(3,6)\}$$

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}
\text{Start} &\to \text{Expr}_1 \mid \text{Expr}_2 \\
\text{Expr}_1 &\to x + x + \text{Expr}_1 \\
&\mid 1 \\
\text{Expr}_2 &\to x + x + x + \text{Expr}_2 \\
&\mid 0
\end{aligned}$$

$\Rightarrow$

Equation

$$\begin{aligned}
V_{\text{Start}} &= V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\
V_{\text{Expr}_1} &= \{(2,4)\} + V_{\text{Expr}_1} \\
&\quad \cup \{(1,1)\} \\
V_{\text{Expr}_2} &= \{(3,6)\} + V_{\text{Expr}_2} \\
&\quad \cup \{(0,0)\}
\end{aligned}$$

Solution

$$\begin{aligned}
V_{\text{Start}} &= \{(1,1) + \lambda(2,4)\} \\
&\quad \cup \{(0,0) + \lambda(3,6)\} \\
V_{\text{Expr}_1} &= \{(1,1) + \lambda(2,4)\} \\
V_{\text{Expr}_2} &= \{(0,0) + \lambda(3,6)\}
\end{aligned}$$

$$(5,6) \overset{?}{\in} V_{\text{Start}}$$

✓ $\Rightarrow$ Realizable for $x \in E$

✗ $\Rightarrow$ Unrealizable

High-level Idea

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

Equation

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$

$$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$

$$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

Solution

$$V_{\text{Start}} = \{(1,1) + \lambda(2,4)\} \cup \{(0,0) + \lambda(3,6)\}$$

$$V_{\text{Expr}_1} = \{(1,1) + \lambda(2,4)\}$$

$$V_{\text{Expr}_2} = \{(0,0) + \lambda(3,6)\}$$

How do I solve this?

Logical approach: Constrained Horn Clauses (CHC)

Iterative approach: Newton's method

High-level Idea

E:

$f(1) = 5$
$f(2) = 6$

Grammar

$$\begin{aligned}\text{Start} &\rightarrow \text{Expr}_1 \mid \text{Expr}_2 \\ \text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \\ &\mid 1 \\ \text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \\ &\mid 0\end{aligned}$$

Equation

$$\begin{aligned}V_{\text{Start}} &= V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\ V_{\text{Expr}_1} &= \{(2,4)\} + V_{\text{Expr}_1} \\ &\cup \{(1,1)\} \\ V_{\text{Expr}_2} &= \{(3,6)\} + V_{\text{Expr}_2} \\ &\cup \{(0,0)\}\end{aligned}$$

Solution

$$\begin{aligned}V_{\text{Start}} &= \{(1,1) + \lambda(2,4)\} \\ &\cup \{(0,0) + \lambda(3,6)\} \\ V_{\text{Expr}_1} &= \{(1,1) + \lambda(2,4)\} \\ V_{\text{Expr}_2} &= \{(0,0) + \lambda(3,6)\}\end{aligned}$$

How do I solve this?

Logical approach: Constrained Horn Clauses (CHC)

Iterative approach: Newton's method

Solving Equations with Horn Clauses

E:

$$f(1) = 5$$
$$f(2) = 6$$

Equation

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$

$$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$

$$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

$$\forall x.\, x \in V_{\text{Expr}_1} \vee x \in V_{\text{Expr}_2} \rightarrow x \in V_{\text{Start}}$$

$$\forall x.\, x \in V_{\text{Expr}_1} \rightarrow \big((2,4) + x\big) \in V_{\text{Expr}_1} \wedge (1,1) \in V_{\text{Expr}_1}$$

$$\forall x.\, x \in V_{\text{Expr}_2} \rightarrow \big((3,6) + x\big) \in V_{\text{Expr}_2} \wedge (0,0) \in V_{\text{Expr}_2}$$

assert $(5,6) \in V_{\text{Start}}$

Solving Equations with Horn Clauses

E:

$$f(1) = 5$$
$$f(2) = 6$$

Equation

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$

$$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$

$$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

$$\forall x.\, x \in V_{\text{Expr}_1} \lor x \in V_{\text{Expr}_2} \rightarrow x \in V_{\text{Start}}$$

$$\forall x.\, x \in V_{\text{Expr}_1} \rightarrow \big((2,4) + x\big) \in V_{\text{Expr}_1} \land (1,1) \in V_{\text{Expr}_1}$$

$$\forall x.\, x \in V_{\text{Expr}_2} \rightarrow \big((3,6) + x\big) \in V_{\text{Expr}_2} \land (0,0) \in V_{\text{Expr}_2}$$

assert $(5,6) \in V_{\text{Start}}$

- Complete, but undecidable
- Solvable by off-the-shelf Constrained Horn Clauses (CHC) solver

High-level Idea

E:

$$f(0) = 0$$
$$f(1) = 5$$

$$\text{Start} \rightarrow \text{Expr}_1 \mid \text{Expr}_2$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

$$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$$

$$V_{\text{Expr}_1} = \{(0,2)\} + V_{\text{Expr}_1} \cup \{(1,1)\}$$

$$V_{\text{Expr}_2} = \{(0,3)\} + V_{\text{Expr}_2} \cup \{(0,0)\}$$

How do I solve this?

Logical approach: Constrained Horn Clauses (CHC)

Iterative approach: Newton's method

Solving Equations with Semi-linear Sets

Domain: integers

Operators: +

Grammar

$$\begin{aligned} \text{Start} &\rightarrow \text{Expr}_1 \mid \text{Expr}_2 \\ \text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \\ &\mid 1 \\ \text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \\ &\mid 0 \end{aligned}$$

Equation

$$\begin{aligned} V_{\text{Start}} &= V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\ V_{\text{Expr}_1} &= \{(2, 4)\} + V_{\text{Expr}_1} \\ &\cup \{(1, 1)\} \\ V_{\text{Expr}_2} &= \{(3, 6)\} + V_{\text{Expr}_2} \\ &\cup \{(0, 0)\} \end{aligned}$$

Value of V can be modeled as semi-linear sets

Semi-linear Sets

$$V_{\text{Start}} = \{(1,1) + \lambda(2,4)\} \cup \{(0,0) + \lambda(3,6)\}$$

Semi-linear set $\{(1,1),(3,5),(5,9),\cdots\} \cup \{(0,0),(3,6),(6,12),\cdots\}$

$$V_{\text{Expr}_1} = \{(1,1) + \lambda(2,4)\}$$

$\rightarrow$ Linear set $(1,1),(3,5),(5,9),\cdots\}$

$$V_{\text{Expr}_2} = \{(0,0) + \lambda(3,6)\}$$

Solving Equations with Semi-linear Sets

Domain: integers

Operators: +

Grammar

$$\begin{aligned}\text{Start} &\rightarrow \text{Expr}_1 \mid \text{Expr}_2 \\ \text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \\ &\mid 1 \\ \text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \\ &\mid 0\end{aligned}$$

Equation

$$\begin{aligned}V_{\text{Start}} &= V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\ V_{\text{Expr}_1} &= \{(2,4)\} + V_{\text{Expr}_1} \\ &\cup \{(1,1)\} \\ V_{\text{Expr}_2} &= \{(3,6)\} + V_{\text{Expr}_2} \\ &\cup \{(0,0)\}\end{aligned}$$

Value of V can be modeled as semi-linear sets

Fact: These equations can be solved iteratively in no more than n rounds with n number of equations [Esparza et al. 2010]

Conditional Linear Integer Arithmetic (CLIA)

Domain:

Operators:

Conditional Linear Integer Arithmetic (CLIA)

Domain: integers, Boolean

Operators: +, ∧, ∨, ¬, if then else, <, ==

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\text{Start} \rightarrow \text{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \mid \text{Expr}_1 \mid \text{Expr}_2$$

$$\text{BExpr} \rightarrow x = 1$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}\text{Start} &\rightarrow \mathsf{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\ &\mid \text{Expr}_1 \mid \text{Expr}_2\end{aligned}$$

$$\text{BExpr} \rightarrow x = 1$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

$$\begin{aligned}V_{\text{Start}} &= \mathsf{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\ &\cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}\end{aligned}$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}\text{Start} &\rightarrow \mathsf{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\ &\quad\mid \text{Expr}_1 \mid \text{Expr}_2\end{aligned}$$

$$\text{BExpr} \rightarrow x = 1$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

$$\begin{aligned}V_{\text{Start}} &= \mathsf{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\ &\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}\end{aligned}$$

$$V_{\text{BExpr}} = (1, 2) == (1, 1)$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}
\text{Start} &\rightarrow \mathsf{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\
&\quad \mid \text{Expr}_1 \mid \text{Expr}_2 \\
\text{BExpr} &\rightarrow x = 1 \\
\text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \mid 1 \\
\text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \mid 0
\end{aligned}$$

$$\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\
&\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\
V_{\text{BExpr}} &= (1, 2) == (1, 1) \\
V_{\text{Expr}_1} &= \{(2, 4)\} + V_{\text{Expr}_1} \\
&\quad \cup \{(1, 1)\} \\
V_{\text{Expr}_2} &= \{(3, 6)\} + V_{\text{Expr}_2} \\
&\quad \cup \{(0, 0)\}
\end{aligned}$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}\text{Start} &\rightarrow \mathsf{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\ &\quad \mid \text{Expr}_1 \mid \text{Expr}_2\end{aligned}$$

$$\text{BExpr} \rightarrow x = 1$$

$$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1 \mid 1$$

$$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2 \mid 0$$

$$\begin{aligned}V_{\text{Start}} &= \mathsf{ITE}(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}) \\ &\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}\end{aligned}$$

$$V_{\text{BExpr}} = (1, 2) == (1, 1)$$

$$\begin{aligned}V_{\text{Expr}_1} &= \{(2, 4)\} + V_{\text{Expr}_1} \\ &\quad \cup \{(1, 1)\}\end{aligned}$$

$$\begin{aligned}V_{\text{Expr}_2} &= \{(3, 6)\} + V_{\text{Expr}_2} \\ &\quad \cup \{(0, 0)\}\end{aligned}$$

$$\begin{aligned}V_{\text{Start}} &= \mathsf{ITE}(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}) \\ &\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}\end{aligned}$$

$$V_{\text{BExpr}} = \{(\text{T}, \text{F})\}$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}
\text{Start} &\rightarrow \text{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\
&\quad \mid \text{Expr}_1 \mid \text{Expr}_2 \\
\text{BExpr} &\rightarrow x = 1 \\
\text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \mid 1 \\
\text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \mid 0
\end{aligned}$$

$$\begin{aligned}
V_{\text{Start}} &= \text{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\
&\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\
V_{\text{BExpr}} &= (1, 2) == (1, 1) \\
V_{\text{Expr}_1} &= \{(2, 4)\} + V_{\text{Expr}_1} \\
&\quad \cup \{(1, 1)\} \\
V_{\text{Expr}_2} &= \{(3, 6)\} + V_{\text{Expr}_2} \\
&\quad \cup \{(0, 0)\}
\end{aligned}$$

$$\begin{aligned}
V_{\text{Start}} &= \text{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\
&\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}
\end{aligned}$$

$$\left.\begin{aligned}
V_{\text{BExpr}} &= \{(\text{T}, \text{F})\} \\
V_{\text{Expr}_1} &= \{(1, 1) + \lambda(2, 4)\} \\
V_{\text{Expr}_2} &= \{(0, 0) + \lambda(3, 6)\}
\end{aligned}\right\} \text{constants}$$

Semantics of If-Then-Else using semi-linear sets

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

$$\begin{aligned}
\text{Start} &\rightarrow \mathsf{ITE}(\text{BExpr}, \text{Start}, \text{Start}) \\
&\quad \mid \text{Expr}_1 \mid \text{Expr}_2 \\
\text{BExpr} &\rightarrow x = 1 \\
\text{Expr}_1 &\rightarrow x + x + \text{Expr}_1 \mid 1 \\
\text{Expr}_2 &\rightarrow x + x + x + \text{Expr}_2 \mid 0
\end{aligned}$$

$$\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\
&\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2} \\
V_{\text{BExpr}} &= (1, 2) == (1, 1) \\
V_{\text{Expr}_1} &= \{(2, 4)\} + V_{\text{Expr}_1} \\
&\quad \cup \{(1, 1)\} \\
V_{\text{Expr}_2} &= \{(3, 6)\} + V_{\text{Expr}_2} \\
&\quad \cup \{(0, 0)\}
\end{aligned}$$

$$\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}\left(V_{\text{BExpr}}, V_{\text{Start}}, V_{\text{Start}}\right) \\
&\quad \cup V_{\text{Expr}_1} \cup V_{\text{Expr}_2}
\end{aligned}$$

$$\left.\begin{aligned}
V_{\text{BExpr}} &= \{(T, F)\} \\
V_{\text{Expr}_1} &= \{(1, 1) + \lambda(2, 4)\} \\
V_{\text{Expr}_2} &= \{(0, 0) + \lambda(3, 6)\}
\end{aligned}\right\} \text{constants}$$

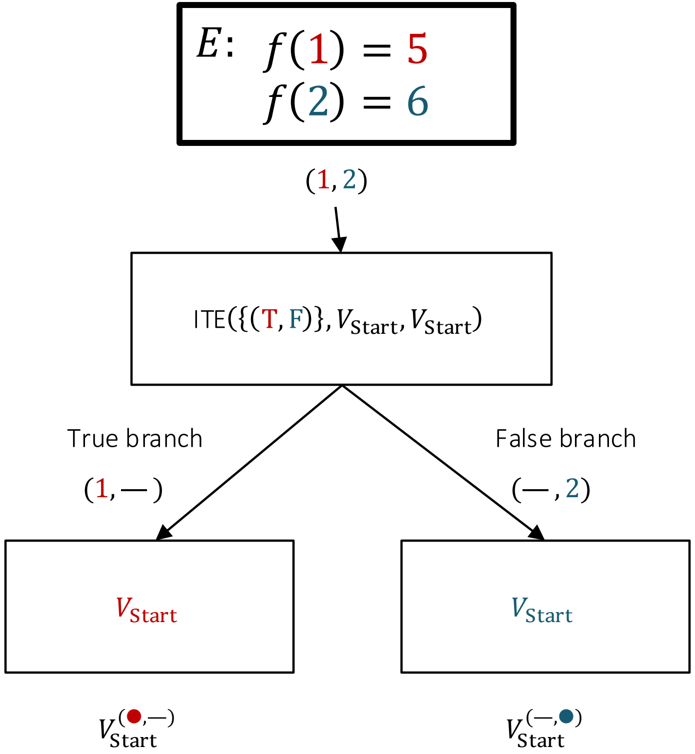
$$\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}(\{(T, F)\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\quad \cup \{(1, 1) + \lambda(2, 4)\} \\
&\quad \cup \{(0, 0) + \lambda(3, 6)\}
\end{aligned}$$

Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}(\{(\mathrm{T}, \mathrm{F})\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1, 1) + \lambda(2, 4)\} \\
&\cup \{(0, 0) + \lambda(3, 6)\}
\end{aligned}
$$

Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \text{ITE}(\{(\text{T},\text{F})\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1,1) + \lambda(2,4)\} \\
&\cup \{(0,0) + \lambda(3,6)\}
\end{aligned}
$$



Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \text{ITE}(\{(T, F)\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1, 1) + \lambda(2, 4)\} \\
&\cup \{(0, 0) + \lambda(3, 6)\}
\end{aligned}
$$

$$\Longrightarrow$$

$$
\begin{aligned}
V_{\text{Start}}^{(\bullet, -)} &= V_{\text{Start}}^{(\bullet, -)} \\
&\cup \{(1, -) + \lambda(0, -)\} \\
&\cup \{(0, -) + \lambda(0, -)\}
\end{aligned}
$$

$$
\begin{aligned}
V_{\text{Start}}^{(-, \bullet)} &= V_{\text{Start}}^{(-, \bullet)} \\
&\cup \{(-, 1) + \lambda(-, 2)\} \\
&\cup \{(-, 0) + \lambda(-, 3)\}
\end{aligned}
$$

Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}(\{(\mathrm{T},\mathrm{F})\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1,1) + \lambda(2,4)\} \\
&\cup \{(0,0) + \lambda(3,6)\}
\end{aligned}
$$

$$\Rightarrow$$

$$
\begin{aligned}
V_{\text{Start}}^{(\bullet,-)} &= V_{\text{Start}}^{(\bullet,-)} \\
&\cup \{1 + \lambda 2\} \\
&\cup \{0 + \lambda 3\}
\end{aligned}
$$

$$
\begin{aligned}
V_{\text{Start}}^{(-,\bullet)} &= V_{\text{Start}}^{(-,\bullet)} \\
&\cup \{1 + \lambda 4\} \\
&\cup \{0 + \lambda 6\}
\end{aligned}
$$

Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \mathsf{ITE}(\{(T, F)\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1,1) + \lambda(2,4)\} \\
&\cup \{(0,0) + \lambda(3,6)\}
\end{aligned}
$$

$$\Rightarrow$$

$$
\begin{aligned}
V_{\text{Start}}^{(\bullet,-)} &= V_{\text{Start}}^{(\bullet,-)} \\
&\cup \{(1,0) + \lambda(0,0)\} \\
&\cup \{(0,0) + \lambda(0,0)\}
\end{aligned}
$$

$$
\begin{aligned}
V_{\text{Start}}^{(-,\bullet)} &= V_{\text{Start}}^{(-,\bullet)} \\
&\cup \{(0,1) + \lambda(0,2)\} \\
&\cup \{(0,0) + \lambda(0,3)\}
\end{aligned}
$$

$$\mathsf{ITE}(\{(T, F)\}, V_{\text{Start}}, V_{\text{Start}}) = V_{\text{Start}}^{(\bullet,-)} + V_{\text{Start}}^{(-,\bullet)}$$

Semantics of If-Then-Else using semi-linear sets

$$
\begin{aligned}
V_{\text{Start}} &= \text{ITE}(\{(T, F)\}, V_{\text{Start}}, V_{\text{Start}}) \\
&\cup \{(1,1) + \lambda(2,4)\} \\
&\cup \{(0,0) + \lambda(3,6)\}
\end{aligned}
$$

$$\Rightarrow$$

$$
V_{\text{Start}} = \quad
\begin{aligned}
V_{\text{Start}}^{({\color{red}\bullet},{\color{blue}\bullet})} &= V_{\text{Start}}^{({\color{red}\bullet},-)} + V_{\text{Start}}^{(-,{\color{blue}\bullet})} \\
&\cup \{(1,1) + \lambda(0,2)\} \\
&\cup \{(0,0) + \lambda(0,3)\}
\end{aligned}
$$

$$
\begin{aligned}
V_{\text{Start}}^{({\color{red}\bullet},-)} &= V_{\text{Start}}^{({\color{red}\bullet},-)} \\
&\cup \{(1,0) + \lambda(0,0)\} \\
&\cup \{(0,0) + \lambda(0,0)\}
\end{aligned}
$$

$$
\begin{aligned}
V_{\text{Start}}^{(-,{\color{blue}\bullet})} &= V_{\text{Start}}^{(-,{\color{blue}\bullet})} \\
&\cup \{(0,1) + \lambda(0,2)\} \\
&\cup \{(0,0) + \lambda(0,3)\}
\end{aligned}
$$

Decidable Fragments of SyGuS with Examples

E:

$$f(1) = 5$$
$$f(2) = 6$$

Grammar

Start $\rightarrow$ $\text{Expr}_1 \mid \text{Expr}_2$

$\text{Expr}_1 \rightarrow x + x + \text{Expr}_1$
$\mid 1$

$\text{Expr}_2 \rightarrow x + x + x + \text{Expr}_2$
$\mid 0$

$\Rightarrow$

Equation

$V_{\text{Start}} = V_{\text{Expr}_1} \cup V_{\text{Expr}_2}$

$V_{\text{Expr}_1} = \{(2,4)\} + V_{\text{Expr}_1}$
$\cup \{(1,1)\}$

$V_{\text{Expr}_2} = \{(3,6)\} + V_{\text{Expr}_2}$
$\cup \{(0,0)\}$

Solution

$V_{\text{Start}} = \{(1,1) + \lambda(0,2)\}$
$\cup \{(0,0) + \lambda(0,3)\}$

$V_{\text{Expr}_1} = \{(1,1) + \lambda(0,2)\}$

$V_{\text{Expr}_2} = \{(0,0) + \lambda(0,3)\}$

Theorem

Given a CLIA SyGuS problem sy and a finite set of examples E, it is decidable whether the SyGuS problem sy^E is (un)realizable

Nope: contributions

First to be able to prove unrealizability for infinite program spaces

CEGIS for unrealizability

- Also shows that in practice a few examples are often enough

Reduction of unrealizability to unreachability

- Can use existing tools

Sound for both synthesis and unrealizability

- If Nope terminates and gives an answer, this answer is correct

Can be used to optimize syntactic objectives

Nope: limitations

Incomplete

- Why might Nope run forever?
- Can Nope run forever if the problem is realizable?

In practice only works for < half of the benchmarks

- Scales poorly with the size of grammar, number of examples

Limited to SyGuS

Nope: questions

Behavioral constraints? Structural constraints? Search strategy?

- First-order specs (like in SyGuS)
- Regular tree grammars (like in SyGuS)
- Enumerative for synthesis + reduction to reachability for unrealizability

Nope: questions

Example of an unrealizable program over bitvectors

```
f(x)=1
G := (Start := 0)
```

Sure, but this search space is finite ☹

```
f(x,y)=Or(x,y)
G := (Start := x | y | Not(Start) | And(Start,Start) )
```

This is realizable!

```
f(1010)=0001
G := (Start := x | Not(Start))
```

Correct!

Nope: questions

Why does Nope use examples in its encoding instead of directly encoding the full specification (i.e., proving unrealizability for all inputs at once)? What would one need to change?

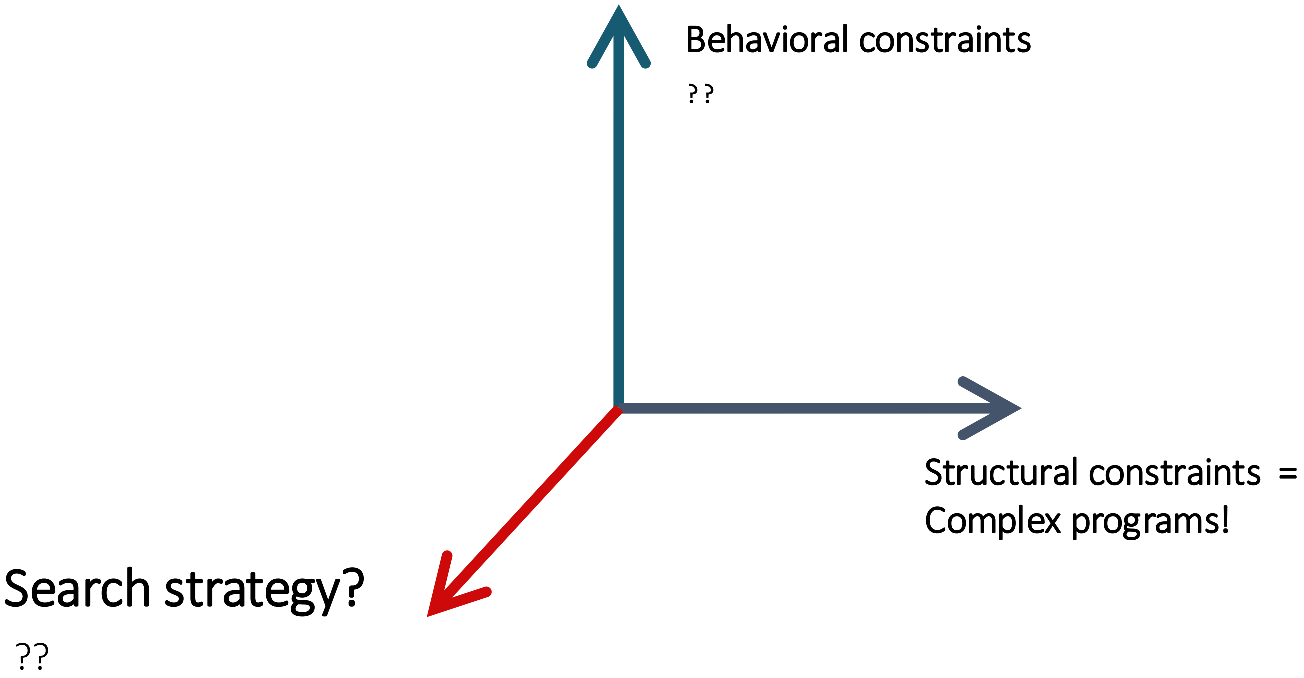
- We can't have quantifiers in the verification problem!

Nope: questions

Can the same encoding be used to synthesize programs instead of proving unrealizability?

- Yes, in fact a “failing test” is effectively a program encoding
- Similar idea to constraint-based search!

Next


Behavioral constraints
??
Structural constraints =
Complex programs!
Search strategy?
??