Lower Runtime Bounds for Integer Programs

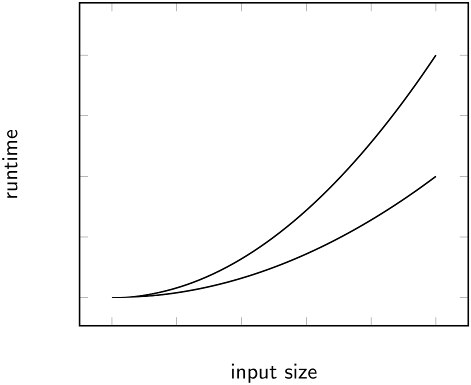
Florian Frohn[1] Matthias Naaf[1] Jera Hensel[1]
Marc Brockschmidt[2] Jürgen Giesl[1]

[1]RWTH Aachen University, Germany

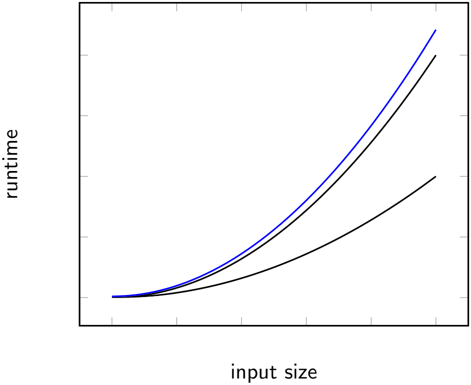
[2]Microsoft Research, Cambridge, UK

September 6, 2016

Lower Bounds?

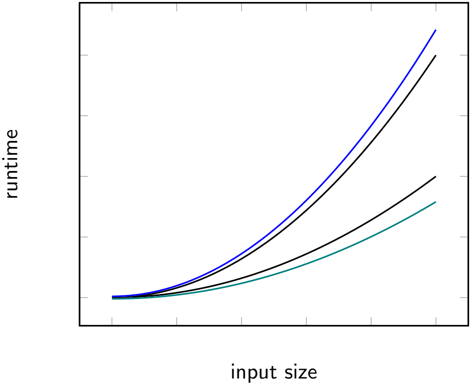


Lower Bounds?



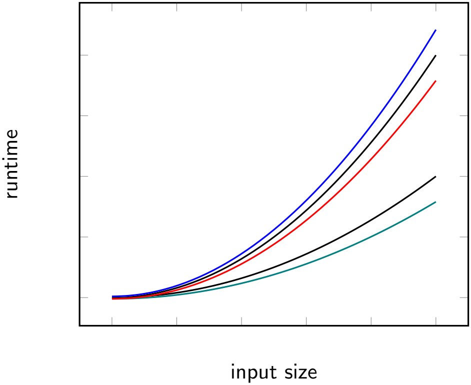
- worst case upper bounds

Lower Bounds?



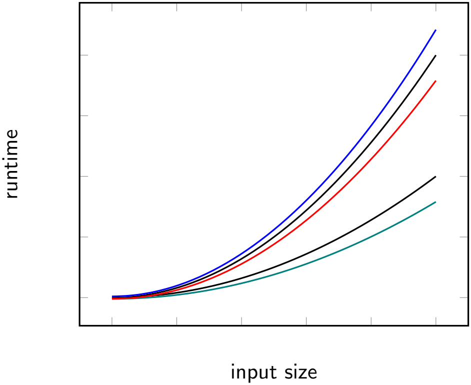
- worst case upper bounds
- best case lower bounds

Lower Bounds?



- worst case upper bounds
- best case lower bounds
- worst case lower bounds

Lower Bounds?

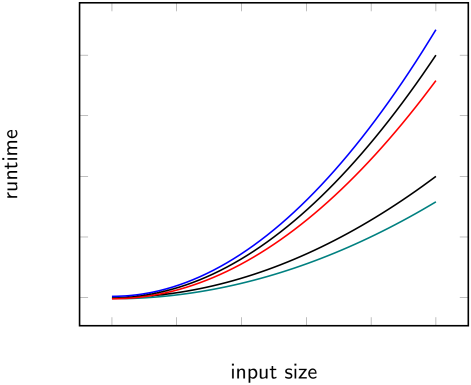


- worst case upper bounds
- best case lower bounds
- worst case lower bounds

Why?

- *tight* bounds

Lower Bounds?



- worst case upper bounds
- best case lower bounds
- worst case lower bounds

Why?

- *tight* bounds
- identify attacks

What kind of programs?

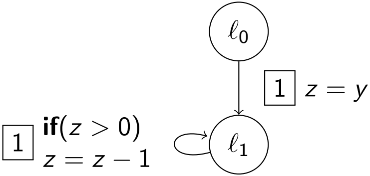
non-recursive integer programs:

```
$z = y$
while ($z > 0$) do
    $z = z - 1$
done
```

What kind of programs?

non-recursive integer programs:

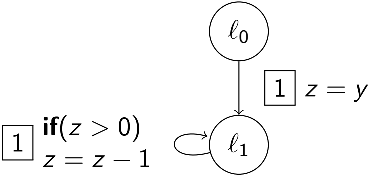
```
$z = y$
while $(z > 0)$ do
    $z = z - 1$
done
```



What kind of programs?

non-recursive integer programs:

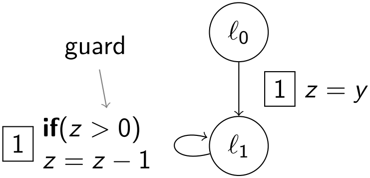
```
$\ell_0$ :  $z = y$
$\ell_1$ :  while $(z > 0)$ do
                $z = z - 1$
            done
```



What kind of programs?

non-recursive integer programs:

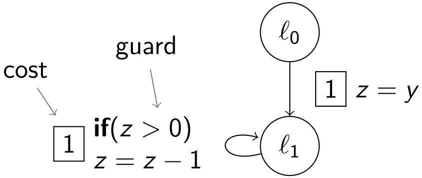
```
$\ell_0$ :  $z = y$
$\ell_1$ :  while $(z > 0)$ do
                $z = z - 1$
            done
```



What kind of programs?

non-recursive integer programs:

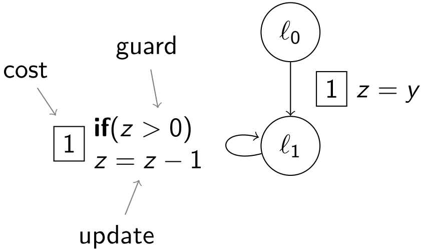
```
$\ell_0$ :  $z = y$
$\ell_1$ :  while $(z > 0)$ do
                $z = z - 1$
            done
```



What kind of programs?

non-recursive integer programs:

```
$\ell_0$ :  $z = y$
$\ell_1$ :  while $(z > 0)$ do
                $z = z - 1$
            done
```



What kind of programs?

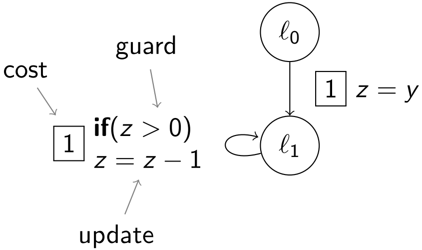
non-recursive integer programs:

ℓ_0 : $z = y$

ℓ_1 : while $(z > 0)$ do

$z = z - 1$

done

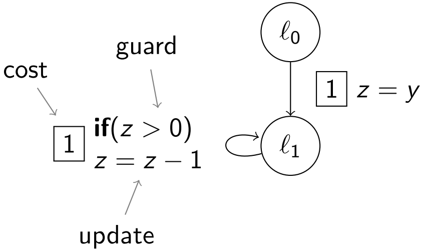


- arithmetic: polynomial (nonlinear)

What kind of programs?

non-recursive integer programs:

```
$\ell_0$ :  $z = y$
$\ell_1$ :  while $(z > 0)$ do
                $z = z - 1$
            done
```



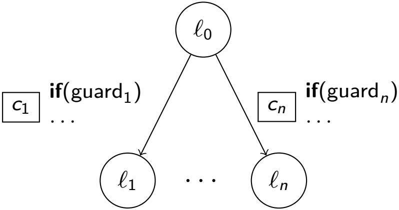
- arithmetic: polynomial (nonlinear)
- cost: can be specified in the input, nonnegative

The Technique

- **step 1**: underapproximating program simplification

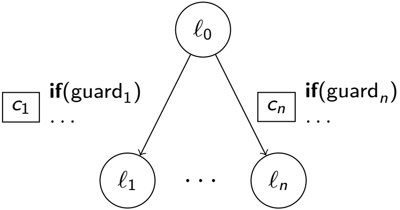
The Technique

- **step 1**: underapproximating program simplification



The Technique

- **step 1**: underapproximating program simplification



- **step 2**: infer asymptotic lower bound

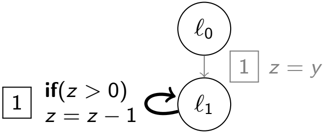
Program Simplification

Acceleration and *Chaining*

Program Simplification

Acceleration and *Chaining*

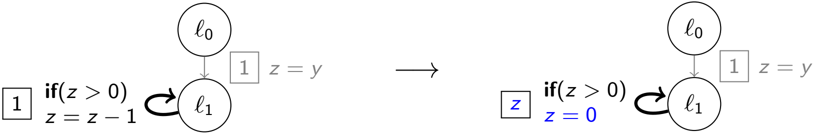
- **accelerate simple loops**



Program Simplification

Acceleration and *Chaining*

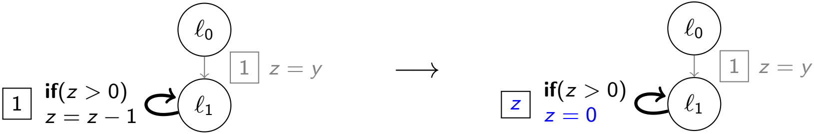
- **accelerate simple loops**



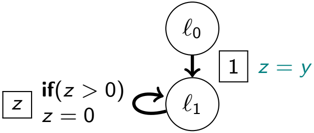
Program Simplification

Acceleration and *Chaining*

- **accelerate simple loops**



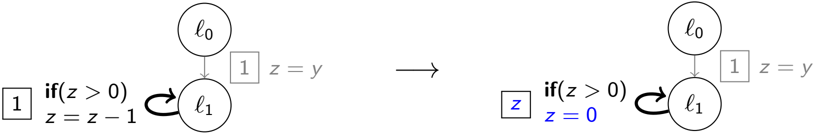
- **chain subsequent transitions**



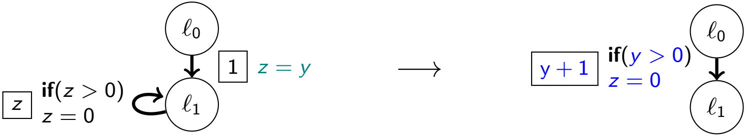
Program Simplification

Acceleration and *Chaining*

- **accelerate simple loops**



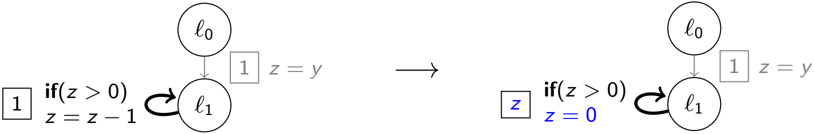
- **chain subsequent transitions**



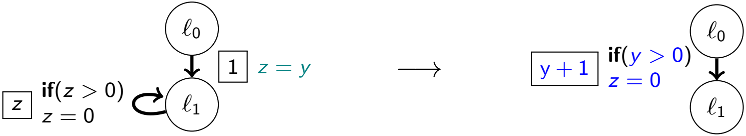
Program Simplification

Acceleration and *Chaining*

- **accelerate simple loops**

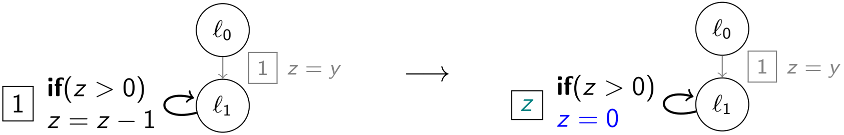


- **chain subsequent transitions**

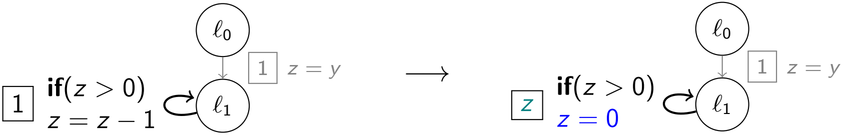


- **iterate**

Loop Acceleration: Challenges

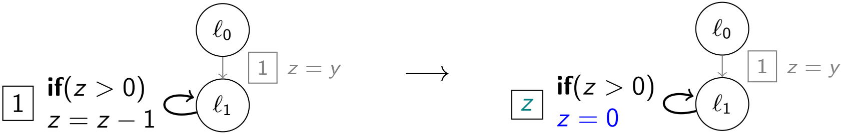


Loop Acceleration: Challenges



- What's the **result**?

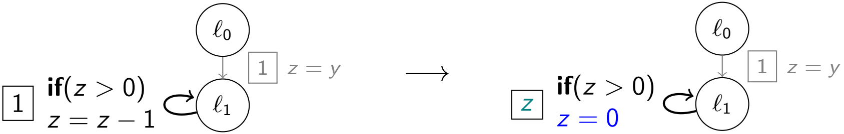
Loop Acceleration: Challenges



- What's the **result**?

- What does it **cost**?

Loop Acceleration: Challenges



$\boxed{1}$ **if**$(z > 0)$ $z = z - 1$ ℓ_0 ℓ_1 $\boxed{1}$ $z = y$

$\longrightarrow$

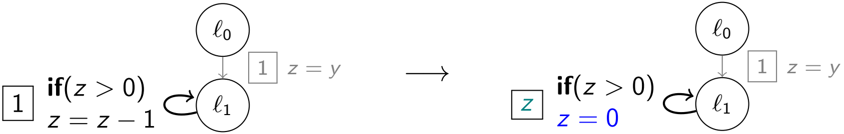
$\boxed{z}$ **if**$(z > 0)$ $z = 0$ ℓ_0 ℓ_1 $\boxed{1}$ $z = y$

- What's the **result**?

- What does it **cost**?

- How many **repetitions**?

Loop Acceleration: Challenges

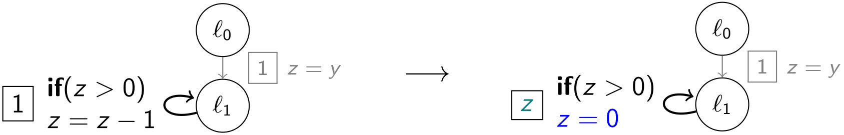


- What's the **result**?
    - build recurrence equations for *iterated update*

- What does it **cost**?

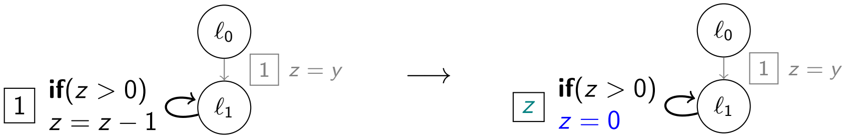
- How many **repetitions**?

Loop Acceleration: Challenges



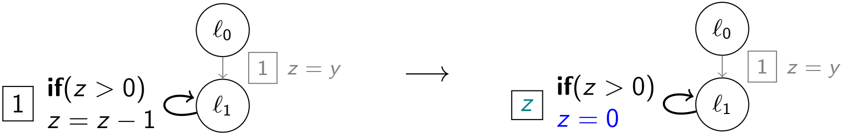
- What's the **result**?
    - build recurrence equations for *iterated update*
    - solve using existing tools
- What does it **cost**?
- How many **repetitions**?

Loop Acceleration: Challenges



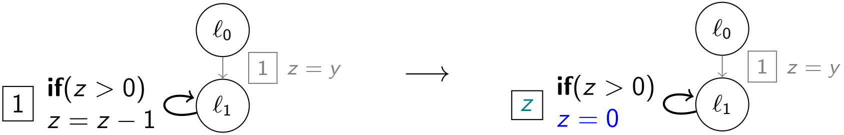
- What's the **result**?
  - build recurrence equations for *iterated update*
  - solve using existing tools
  - $z^{(1)} = z - 1$ and $z^{(n+1)} = z^{(n)} - 1 \quad \curvearrowright \quad z^{(n)} = z - n$
- What does it **cost**?
- How many **repetitions**?

Loop Acceleration: Challenges



- What's the **result**?
  - build recurrence equations for *iterated update*
  - solve using existing tools
  - $z^{(1)} = z - 1$ and $z^{(n+1)} = z^{(n)} - 1$ $\curvearrowright$ $z^{(n)} = z - n$
- What does it **cost**?
  - similar to iterated update $\curvearrowright$ $n \cdot 1$
- How many **repetitions**?

Loop Acceleration: Challenges



- What's the **result**?
  - build recurrence equations for *iterated update*
  - solve using existing tools
  - $z^{(1)} = z - 1$ and $z^{(n+1)} = z^{(n)} - 1 \quad \curvearrowright \quad z^{(n)} = z - n$
- What does it **cost**?
  - similar to iterated update $\quad \curvearrowright \quad n \cdot 1$
- How many **repetitions**?
  - use *metering functions* $\quad \curvearrowright \quad n = z$

Metering Functions

```
while (guard) do
    update
done
```

- variation of *ranking functions*

Metering Functions

```
while (guard) do
    update
done
```

- variation of *ranking functions*
- ranking function: "$\geq$ max. number of iterations"

Metering Functions

```
while (guard) do
    update
done
```

- variation of *ranking functions*
- ranking function: "$\geq$ max. number of iterations"
- metering function: "$\leq$ max. number of iterations"

Metering Functions

```
while (guard) do
    update
done
```

- variation of *ranking functions*
- ranking function: "$\geq$ max. number of iterations"
- metering function: "$\leq$ max. number of iterations"
- b is a *metering* *function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Metering Functions

while (*guard*) do
    update
done

- variation of *ranking functions*
- ranking function: "$\geq$ max. number of iterations"
- metering function: "$\leq$ max. number of iterations"
- b is a *metering* *function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

$\Rightarrow$ transition can be applied at least b times

Metering Functions

```
while (guard) do
    update
done
```

- variation of *ranking functions*
- ranking function: "$\geq$ max. number of iterations"
- metering function: "$\leq$ max. number of iterations"
- b is a *metering (resp. ranking) function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and guard} \Rightarrow \text{update}(b) \geq b - 1$$

$$\text{guard} \Rightarrow b > 0 \text{ and guard} \Rightarrow \text{update}(b) \leq b - 1$$

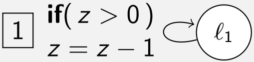
$\Rightarrow$ transition can be applied at least b times

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



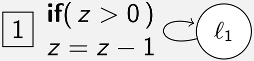
$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



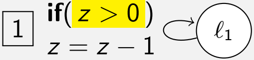
$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

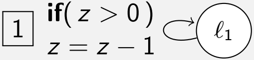
$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)

$z = z - 1$

ℓ_1

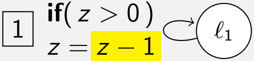
$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

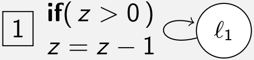
$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

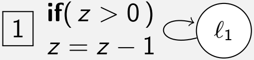
- $b(z) = z, z - 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

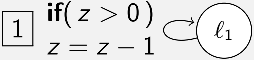
- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

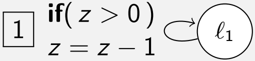
- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $b(z) = z, z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

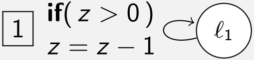
- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

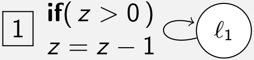
- $b(z) = z,\ z - 1, \ldots$
- $b(z) = 0,\ -1, \ldots$
- $b(z) = z,\ z + 1, \ldots$
- $b(z) = 2 \cdot z,\ 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

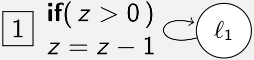
- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

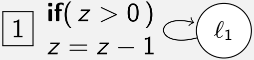
- $b(z) = z,\ z - 1, \ldots$
- $b(z) = 0,\ -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
- $b(z) = z,\ z + 1, \ldots$
- $b(z) = 2 \cdot z,\ 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

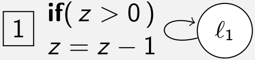
- $b(z) = z,\ z - 1, \ldots$
- $b(z) = 0,\ -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$
- $b(z) = z,\ z + 1, \ldots$
- $b(z) = 2 \cdot z,\ 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

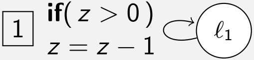
- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

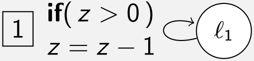
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

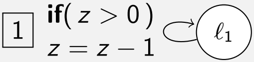
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**($z > 0$)
$z = z - 1$

$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z,\ z - 1, \ldots$
- $b(z) = 0,\ -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

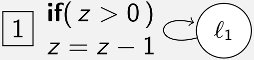
- $b(z) = z,\ z + 1, \ldots$
- $b(z) = 2 \cdot z,\ 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z-1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

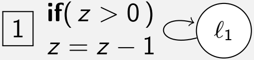
- $b(z) = z, z+1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$\boxed{1}$ **if**$(z > 0)$
$z = z - 1$

ℓ_1

$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

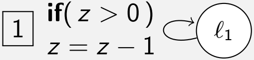
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z - 1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

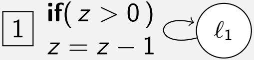
- $b(z) = z, z + 1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$ ↯

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z,\ z - 1, \ldots$
- $b(z) = 0,\ -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

- $b(z) = z,\ z + 1, \ldots$
- $b(z) = 2 \cdot z,\ 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$ ↯

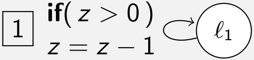
finding them:

Metering Functions

b is a *metering function* iff

$$\neg\text{guard} \Rightarrow b \leq 0 \text{ and } \text{guard} \Rightarrow \text{update}(b) \geq b - 1$$

Example



$$z \leq 0 \Rightarrow b(z) \leq 0 \text{ and } z > 0 \Rightarrow b(z-1) \geq b(z) - 1$$

- $b(z) = z, z-1, \ldots$
- $b(z) = 0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

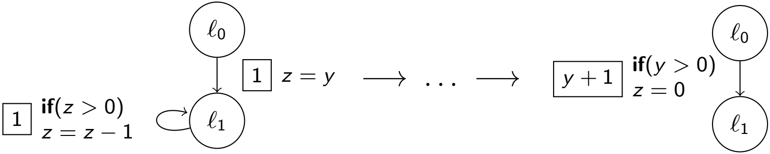
- $b(z) = z, z+1, \ldots$
- $b(z) = 2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ $\lightning$
  $z > 0 \Rightarrow 2 \cdot (z-1) \geq 2 \cdot z - 1$ $\lightning$

finding them: just like ranking functions

Program Simplification

Algorithm

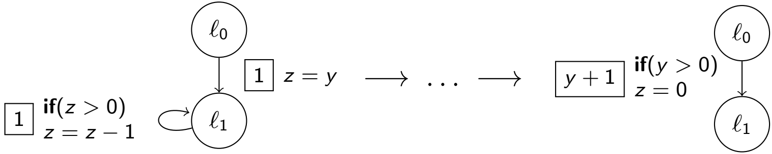
- while there is a path of length > 1



Program Simplification

Algorithm

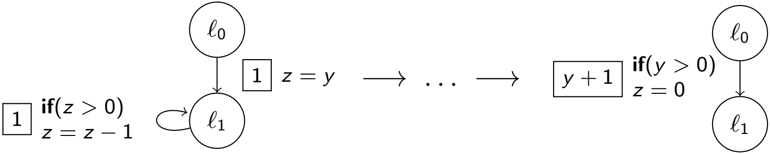
- while there is a path of length > 1
    - **accelerate** simple loops



Program Simplification

Algorithm

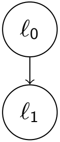
- while there is a path of length > 1
  - **accelerate** simple loops
  - **chain** subsequent transitions



Simplified Programs

$\boxed{x^2 - y}$ **if**($0 < x < 10 \land y + x < 0$)

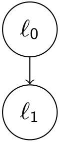
...



Simplified Programs

$\boxed{x^2 - y}$

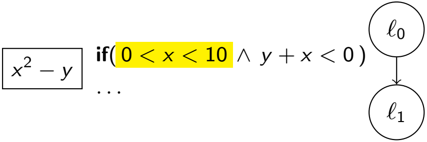
if($0 < x < 10 \land y + x < 0$)
. . .



- inferring lower bound still non-trivial

Example

Simplified Programs



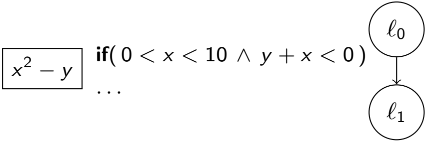
$\boxed{x^2 - y}$ **if**($0 < x < 10 \land y + x < 0$)

$\dots$

- inferring lower bound still non-trivial
- runtime depends on cost **and guard**

Example

Simplified Programs



$\boxed{x^2 - y}$ **if**($0 < x < 10 \wedge y + x < 0$)

…

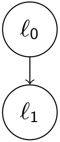
- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\mathbf{v}_n$ of valuations which satisfies the guard for large n

Example

Simplified Programs

$\boxed{x^2 - y}$ **if**($0 < x < 10 \wedge y + x < 0$)

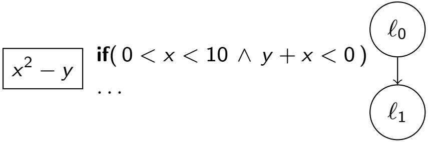
…



- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\mathbf{v}_n$ of valuations which satisfies the guard for large n
- apply $\mathbf{v}_n$ to cost to get asymptotic bound

Example

Simplified Programs



$\boxed{x^2 - y}$ **if**($0 < x < 10 \land y + x < 0$)

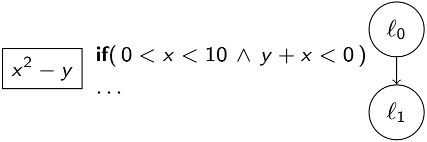
$\dots$

- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\mathbf{v}_n$ of valuations which satisfies the guard for large n
- apply $\mathbf{v}_n$ to cost to get asymptotic bound

Example

- $\mathbf{v}_n = \{x/1, y/-n\}$

Simplified Programs



$\boxed{x^2 - y}$ **if**$(\, 0 < x < 10 \wedge y + x < 0 \,)$

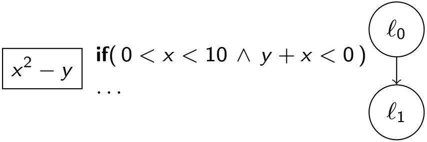
$\ldots$

- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\mathbf{v}_n$ of valuations which satisfies the guard for large n
- apply $\mathbf{v}_n$ to cost to get asymptotic bound

Example

- $\mathbf{v}_n = \{x/1, y/-n\}$ satisfies guard for $n \geq 2$

Simplified Programs



$\boxed{x^2 - y}$ **if**($0 < x < 10 \land y + x < 0$)

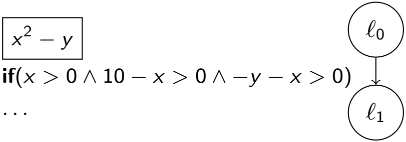
...

- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\mathbf{v}_n$ of valuations which satisfies the guard for large n
- apply $\mathbf{v}_n$ to cost to get asymptotic bound

Example

- $\mathbf{v}_n = \{x/1, y/-n\}$ satisfies guard for $n \geq 2$
- $\mathbf{v}_n(x^2 - y) = 1 + n \implies \Omega(n)$

Example



$\boxed{x^2 - y}$

if$(x > 0 \land 10 - x > 0 \land -y - x > 0)$

$\ldots$

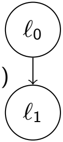
goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

Example

$\boxed{x^2 - y}$

if$(x > 0 \land 10 - x > 0 \land -y - x > 0)$

$\dots$



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

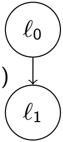
$$\{x^{+!}, (10 - x)^{+!}, (-y - x)^{+}\}$$

Example

$\boxed{x^2 - y}$

if$(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

$\ldots$

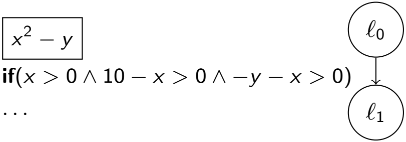


goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

$$\{x^{+!}, (10 - x)^{+!}, (-y - x)^{+}\}$$

- a^{+} : $\mathbf{v}_n(a)$ "increases with n" $\quad$ a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example



$\boxed{x^2 - y}$

$\textbf{if}(x > 0 \land 10 - x > 0 \land -y - x > 0)$

$\ldots$

goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe : $(a-b)^{+}$ if a^{+} and $b^{+!}$

$$\{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\}$$

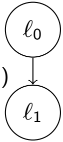
- a^{+} : $\mathbf{v}_n(a)$ "increases with n" a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

$\textbf{if}(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

$\ldots$



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe : $(a - b)^{+}$ if a^{+} and $b^{+!}$

$$\{x^{+!}, (10 - x)^{+!}, (-y - x)^{+}\} \quad \rightsquigarrow \quad \{x^{+!}, (10 - x)^{+!}, (-y)^{+}\}$$

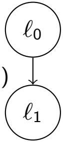
- a^{+} : $\mathbf{v}_n(a)$ “increases with n” a^{-} : $\mathbf{v}_n(a)$ “decreases with n”
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

if$(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

$\dots$



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe : $(a-b)^+$ if a^+ and $b^{+!}$
$(-a)^+$ if a^-

$$\{x^{+!}, (10-x)^{+!}, (-y-x)^+\} \quad \rightsquigarrow \quad \{x^{+!}, (10-x)^{+!}, (-y)^+\}$$

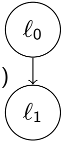
- $a^+ : \mathbf{v}_n(a)$ "increases with n" $\qquad a^- : \mathbf{v}_n(a)$ "decreases with n"
- $a^{+!} : \mathbf{v}_n(a)$ is pos. constant $\qquad a^{-!} : \mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

if$(x > 0 \land 10 - x > 0 \land -y - x > 0)$

$\dots$



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe : $(a - b)^{+}$ if a^{+} and $b^{+!}$
$(-a)^{+}$ if a^{-}

$$\{x^{+!}, (10 - x)^{+!}, (-y - x)^{+}\} \quad \rightsquigarrow \quad \{x^{+!}, (10 - x)^{+!}, (-y)^{+}\}$$

$$\rightsquigarrow \quad \{x^{+!}, (10 - x)^{+!}, y^{-}\}$$

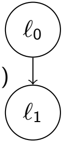
- a^{+} : $\mathbf{v}_n(a)$ "increases with n" a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

if$(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

. . .



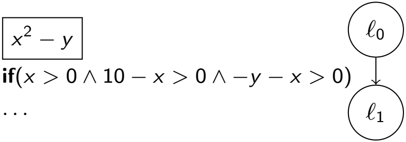
goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

$$\textbf{observe}: \begin{array}{lll} (a-b)^{+} & \text{if} & a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if} & a^{-} \\ (a-b)^{+!} & \text{if} & a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{lll} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} \quad \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \end{array}$$

- a^{+} : $\mathbf{v}_n(a)$ "increases with n" $\quad$ a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe :
$$\begin{array}{ll} (a-b)^{+} & \text{if } a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if } a^{-} \\ (a-b)^{+!} & \text{if } a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{llcl} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} & \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \rightsquigarrow & \{x^{+!}, 10^{+!}, x^{-!}, y^{-}\} \end{array}$$

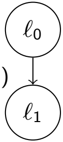
- a^{+} : $\mathbf{v}_n(a)$ "increases with n" $\qquad$ a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $\qquad$ $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

$\textbf{if}(x > 0 \land 10 - x > 0 \land -y - x > 0)$

$\ldots$



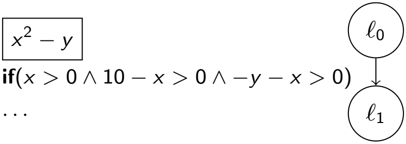
goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

$$\textbf{observe} : \begin{array}{lll} (a-b)^{+} & \text{if} & a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if} & a^{-} \\ (a-b)^{+!} & \text{if} & a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{lll} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} & \rightsquigarrow \quad \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \end{array}$$

- a^{+} : $\mathbf{v}_n(a)$ "increases with n" $\qquad$ a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $\qquad$ $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example



$\boxed{x^2 - y}$

if$(x > 0 \land 10 - x > 0 \land -y - x > 0)$

...

goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe : $$\begin{array}{lll} (a-b)^{+} & \text{if} & a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if} & a^{-} \\ (a-b)^{+!} & \text{if} & a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{llcl} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} & \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \overset{\{x/1\}}{\rightsquigarrow} & \{1^{+!}, 9^{+!}, y^{-}\} \end{array}$$

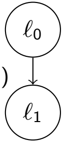
- a^{+} : $\mathbf{v}_n(a)$ "increases with n" a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

if$(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

$\dots$



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

observe :

$$\begin{array}{ll} (a-b)^{+} & \text{if } a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if } a^{-} \\ (a-b)^{+!} & \text{if } a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{llcl} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} & \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \overset{\{x/1\}}{\rightsquigarrow} & \{1^{+!}, 9^{+!}, y^{-}\} \rightsquigarrow^{2} \{y^{-}\} \end{array}$$

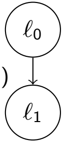
- a^{+} : $\mathbf{v}_n(a)$ "increases with n" a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Example

$\boxed{1^2 - (-n)}$

if$(x > 0 \wedge 10 - x > 0 \wedge -y - x > 0)$

...



goal: infer $\mathbf{v}_n = \{x/1, y/-n\}$

$$\textbf{observe}: \begin{array}{ll} (a-b)^{+} & \text{if } a^{+} \text{ and } b^{+!} \\ (-a)^{+} & \text{if } a^{-} \\ (a-b)^{+!} & \text{if } a^{+!} \text{ and } b^{-!} \end{array}$$

$$\begin{array}{lccl} & \{x^{+!}, (10-x)^{+!}, (-y-x)^{+}\} & \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, (-y)^{+}\} \\ \rightsquigarrow & \{x^{+!}, (10-x)^{+!}, y^{-}\} & \overset{\{x/1\}}{\rightsquigarrow} & \{1^{+!}, 9^{+!}, y^{-}\} \rightsquigarrow^{2} \{y^{-}\} \end{array}$$

- a^{+} : $\mathbf{v}_n(a)$ "increases with n" $\qquad$ a^{-} : $\mathbf{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\mathbf{v}_n(a)$ is pos. constant $\qquad$ $a^{-!}$: $\mathbf{v}_n(a)$ is neg. constant

Summary

- simplify program

Summary

- simplify program
- normalize guard to $a_1 > 0 \land \cdots \land a_k > 0$

Summary

- simplify program
- normalize guard to $a_1 > 0 \wedge \cdots \wedge a_k > 0$
- start with $\{a_1^{\bullet_1}, \ldots, a_k^{\bullet_k}\}$ where $\bullet_i \in \{+, +_!\}$

Summary

- simplify program
- normalize guard to $a_1 > 0 \wedge \cdots \wedge a_k > 0$
- start with $\{a_1^{\bullet_1}, \ldots, a_k^{\bullet_k}\}$ where $\bullet_i \in \{+, +_!\}$
- simplify with "$\overset{\sigma}{\rightsquigarrow}$"

Summary

- simplify program
- normalize guard to $a_1 > 0 \wedge \cdots \wedge a_k > 0$
- start with $\{a_1^{\bullet_1}, \ldots, a_k^{\bullet_k}\}$ where $\bullet_i \in \{+, +_!\}$
- simplify with "$\overset{\sigma}{\rightsquigarrow}$"
- just variables left $\curvearrowright$ $\mathbf{v}_n$

Summary

- simplify program
- normalize guard to $a_1 > 0 \wedge \cdots \wedge a_k > 0$
- start with $\{a_1^{\bullet_1}, \ldots, a_k^{\bullet_k}\}$ where $\bullet_i \in \{+, +_!\}$
- simplify with "$\overset{\sigma}{\rightsquigarrow}$"
- just variables left $\curvearrowright$ $\mathbf{v}_n$
- apply $\mathbf{v}_n$ to cost

Summary

- simplify program
- normalize guard to $a_1 > 0 \wedge \cdots \wedge a_k > 0$
- start with $\{a_1^{\bullet_1}, \dots, a_k^{\bullet_k}\}$ where $\bullet_i \in \{+, +_!\}$
- simplify with "$\overset{\sigma}{\rightsquigarrow}$"
- just variables left $\curvearrowright$ $\mathbf{v}_n$
- apply $\mathbf{v}_n$ to cost
- $\mathbf{v}_n$ reveals influence of variables on the runtime

Experiments

- `https://github.com/aprove-developers/LoAT`

Experiments

- `https://github.com/aprove-developers/LoAT`
- comparison with KoAT, RanK, Loopus, CoFloCo

Experiments

- `https://github.com/aprove-developers/LoAT`
- comparison with KoAT, RanK, Loopus, CoFloCo
- examples from KoAT-evaluation (timeout 60s, average 2.4s)

Experiments

- `https://github.com/aprove-developers/LoAT`
- comparison with KoAT, RanK, Loopus, CoFloCo
- examples from KoAT-evaluation (timeout 60s, average 2.4s)

runtime	$\Omega(1)$	$\Omega(n)$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^4)$	EXP	$\Omega(\omega)$
$\mathcal{O}(1)$	(132)	–	–	–	–	–	–
$\mathcal{O}(n)$	45	125	–	–	–	–	–
$\mathcal{O}(n^2)$	9	18	33	–	–	–	–
$\mathcal{O}(n^3)$	2	–	–	3	–	–	–
$\mathcal{O}(n^4)$	1	–	–	–	2	–	–
EXP	–	–	–	–	–	5	–
$\mathcal{O}(\omega)$	57	31	3	–	–	–	173

Experiments

- `https://github.com/aprove-developers/LoAT`
- comparison with KoAT, RanK, Loopus, CoFloCo
- examples from KoAT-evaluation (timeout 60s, average 2.4s)

runtime	$\Omega(1)$	$\Omega(n)$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^4)$	EXP	$\Omega(\omega)$
$\mathcal{O}(1)$	(132)	–	–	–	–	–	–
$\mathcal{O}(n)$	45	125	–	–	–	–	–
$\mathcal{O}(n^2)$	9	18	33	–	–	–	–
$\mathcal{O}(n^3)$	2	–	–	3	–	–	–
$\mathcal{O}(n^4)$	1	–	–	–	2	–	–
EXP	–	–	–	–	–	5	–
$\mathcal{O}(\omega)$	57	31	3	–	–	–	173

- non-trivial bounds: 78%, tight bounds: 67%

Conclusion

- underapproximating program simplification framework

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach
- polynomial, exponential, and infinite bounds

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach
- polynomial, exponential, and infinite bounds
- full paper presented at IJCAR 2016

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach
- polynomial, exponential, and infinite bounds
- full paper presented at IJCAR 2016
- part of DARPA project STAC to detect vulnerabilities
  (Space/Time Analysis for Cybersecurity)

Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach
- polynomial, exponential, and infinite bounds
- full paper presented at IJCAR 2016
- part of DARPA project STAC to detect vulnerabilities
  (Space/Time Analysis for Cybersecurity)

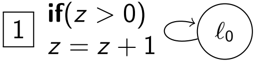
Conclusion

- underapproximating program simplification framework
- calculus to obtain asymptotic lower bounds
- modular approach
- polynomial, exponential, and infinite bounds
- full paper presented at IJCAR 2016
- part of DARPA project STAC to detect vulnerabilities
  (Space/Time Analysis for Cybersecurity)

Infinite Lower Bounds $\Omega(\omega)$

- **nontermination**

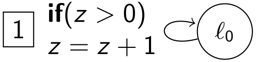
```
while ($z > 0$) do
    $z = z + 1$
done
```



Infinite Lower Bounds $\Omega(\omega)$

- **nontermination**

```
while $(z > 0)$ do
    $z = z + 1$
done
```



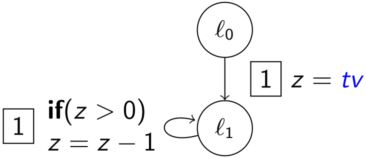
$\boxed{1}$ **if**$(z > 0)$

$z = z + 1$

ℓ_0

- **unbounded runtime**

```
$z = random$
while $(z > 0)$ do
    $z = z - 1$
done
```



ℓ_0

$\boxed{1}$ $z = tv$

$\boxed{1}$ **if**$(z > 0)$

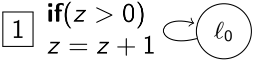
$z = z - 1$

ℓ_1

Infinite Lower Bounds $\Omega(\omega)$

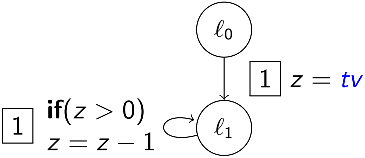
- **nontermination**

```
while $(z > 0)$ do
    $z = z + 1$
done
```



- **unbounded runtime**

```
$z = random$
while $(z > 0)$ do
    $z = z - 1$
done
```



⇒ e.g. to model nondeterminism, user input, . . .