Lower Runtime Bounds for Integer Programs

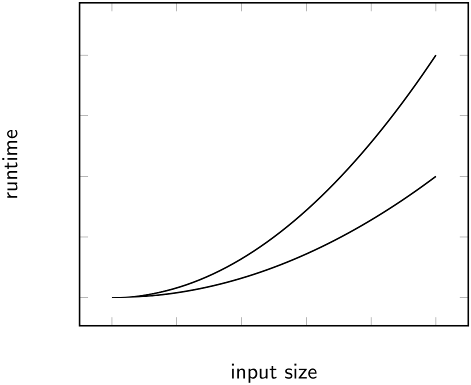
Florian Frohn[1] Matthias Naaf[1] Jera Hensel[1]
Marc Brockschmidt[2] Jürgen Giesl[1]

[1]RWTH Aachen University, Germany

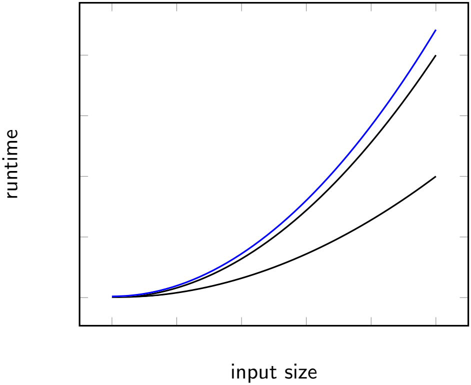
[2]Microsoft Research, Cambridge, UK

June 27, 2016

Lower Bounds?

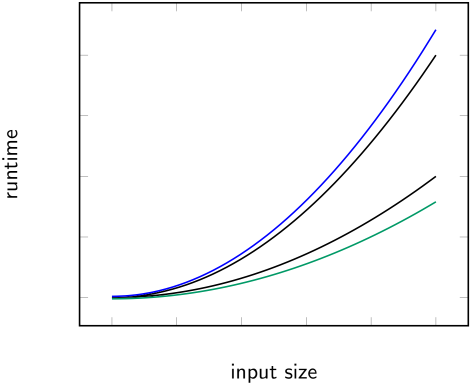


Lower Bounds?



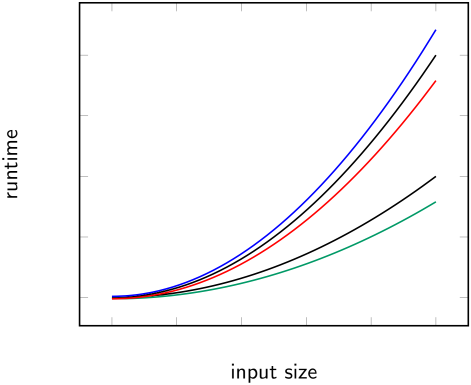
- worst case upper bounds

Lower Bounds?



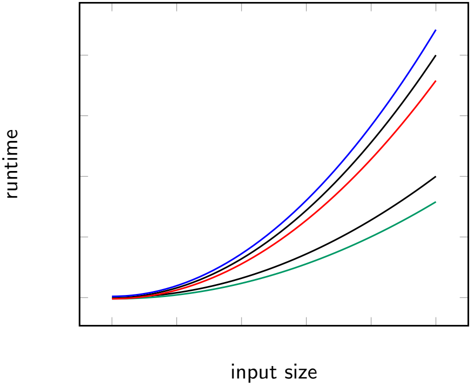
- worst case upper bounds
- best case lower bounds

Lower Bounds?



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

Lower Bounds?

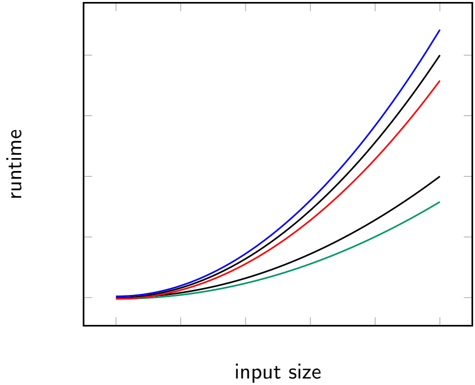


- worst case upper bounds
- best case lower bounds
- worst case lower bounds

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?

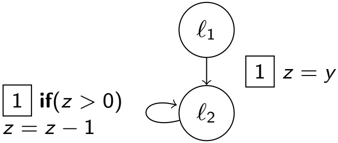
non-recursive integer programs:

```
z = y
while (z > 0)
  z = z - 1
```

What kind of programs?

non-recursive integer programs:

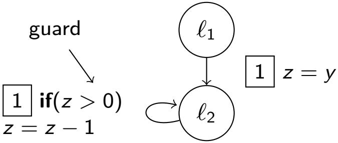
```
z = y
while (z > 0)
  z = z - 1
```



What kind of programs?

non-recursive integer programs:

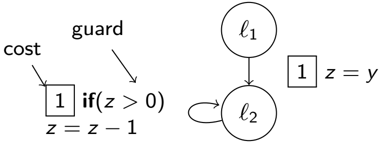
```
z = y
while (z > 0)
  z = z - 1
```



What kind of programs?

non-recursive integer programs:

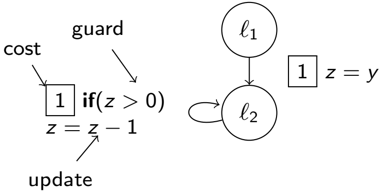
```
z = y
while (z > 0)
  z = z - 1
```



What kind of programs?

non-recursive integer programs:

```
z = y
while (z > 0)
  z = z - 1
```

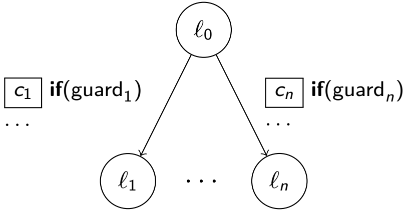


The Technique

- step 1: underapproximating program simplification

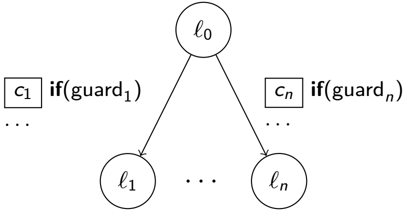
The Technique

- step 1: underapproximating program simplification



The Technique

- step 1: underapproximating program simplification



- step 2: infer asymptotic lower bound

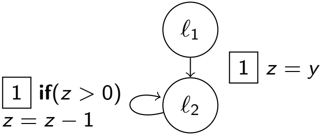
Program Simplification

Acceleration and *Chaining*

Program Simplification

Acceleration and *Chaining*

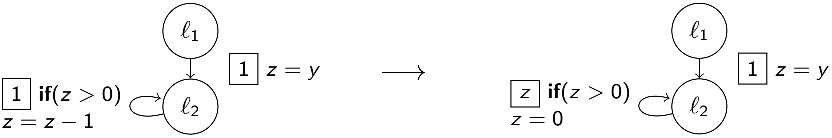
- *accelerate* simple loops



Program Simplification

Acceleration and *Chaining*

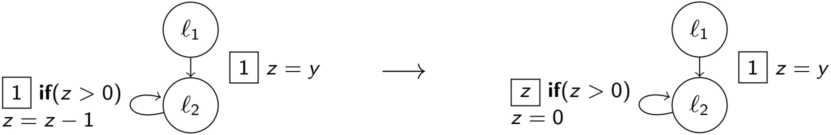
- *accelerate* simple loops



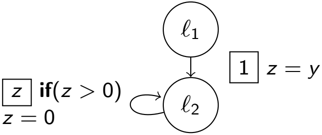
Program Simplification

Acceleration and *Chaining*

- *accelerate* simple loops



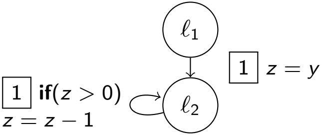
- *chain* subsequent transitions



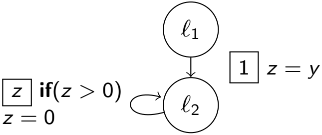
Program Simplification

Acceleration and *Chaining*

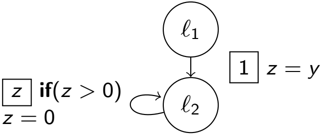
- *accelerate* simple loops



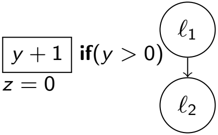
$\longrightarrow$



- *chain* subsequent transitions



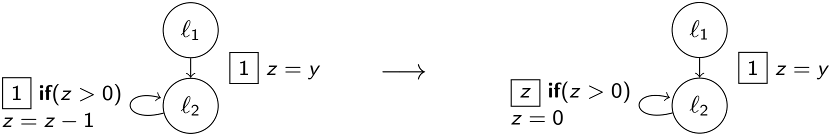
$\longrightarrow$



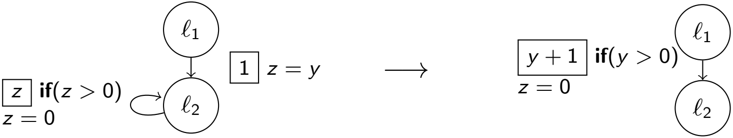
Program Simplification

Acceleration and *Chaining*

- *accelerate* simple loops

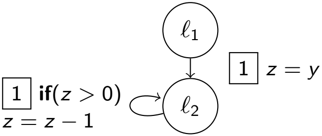


- *chain* subsequent transitions

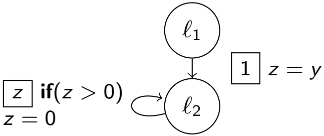


- iterate

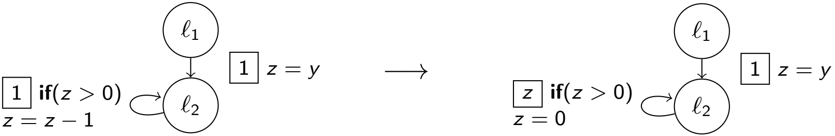
Loop Acceleration: Challenges



$\longrightarrow$

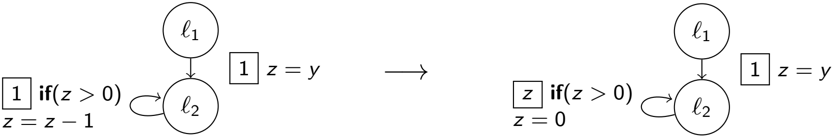


Loop Acceleration: Challenges



- What's the result?

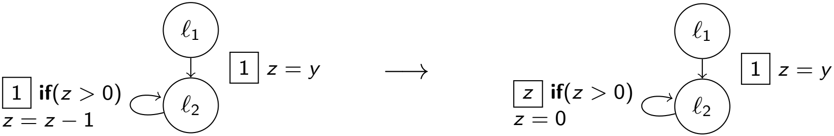
Loop Acceleration: Challenges



- What's the result?

- What does it cost?

Loop Acceleration: Challenges

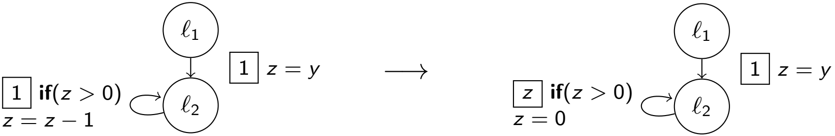


- What's the result?

- What does it cost?

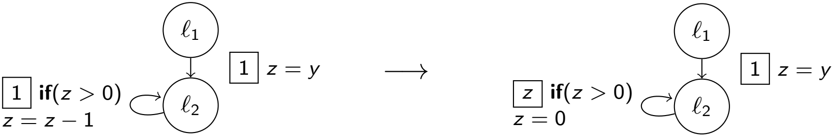
- How many repetitions?

Loop Acceleration: Challenges



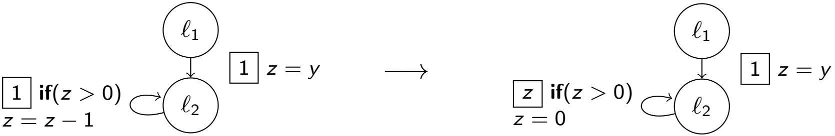
- What's the result?
  - build recurrence equations
- What does it cost?
- How many repetitions?

Loop Acceleration: Challenges



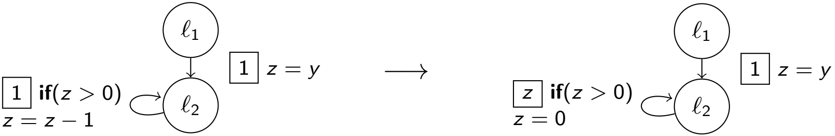
- What's the result?
  - build recurrence equations
  - solve using existing tools
- What does it cost?
- How many repetitions?

Loop Acceleration: Challenges



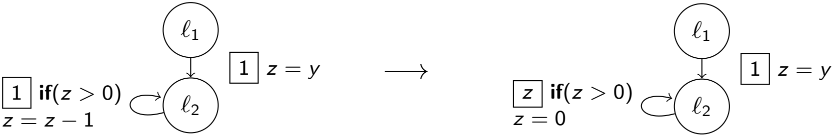
- What's the result?
  - build recurrence equations
  - solve using existing tools
  - $z^{(1)} = z - 1$ and $z^{(n+1)} = z^{(n)} - 1 \curvearrowright z^{(n)} = z - n$
- What does it cost?
- How many repetitions?

Loop Acceleration: Challenges



- What's the result?
  - build recurrence equations
  - 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
- How many repetitions?

Loop Acceleration: Challenges



- What's the result?
  - build recurrence equations
  - 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
- How many repetitions?
  - use *metering functions*

Metering Functions

- variation of *ranking functions*

Metering Functions

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

Metering Functions

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

Metering Functions

- 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

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

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

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

Metering Functions

- 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

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

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

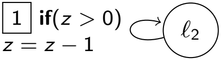
$\Rightarrow$ transition can be applied at least b times

Metering Functions

b is a *metering function* iff

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

Example

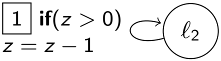


Metering Functions

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



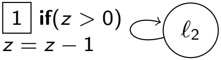
- $z, z - 1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



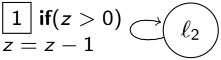
- $z, z - 1, \ldots$
- $0, -1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



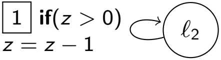
- $z, z-1, \ldots$
- $0, -1, \ldots$
- $z, z+1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



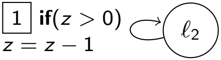
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



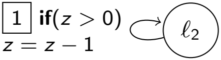
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



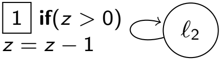
- $z, z - 1, \ldots$
- $z, z + 1, \ldots$
- $0, -1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓

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



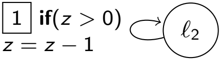
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



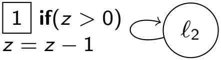
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$

Metering Functions

b is a *metering function* iff

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

Example



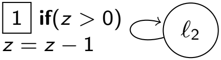
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$

Metering Functions

b is a *metering function* iff

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

Example



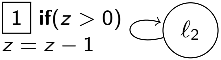
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯

Metering Functions

b is a *metering function* iff

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

Example



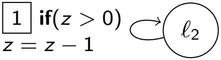
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$

Metering Functions

b is a *metering function* iff

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

Example



- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

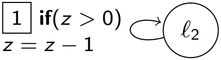
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$ ↯

Metering Functions

b is a *metering function* iff

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

Example



- $z, z-1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓

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

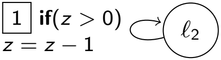
finding them:

Metering Functions

b is a *metering function* iff

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

Example



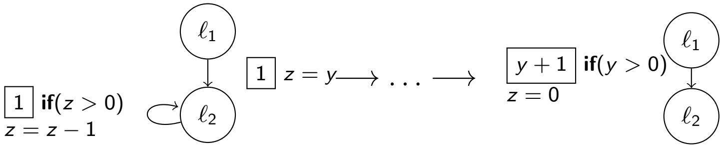
- $z, z - 1, \ldots$
- $0, -1, \ldots$
- $z \leq 0 \Rightarrow z \leq 0$ ✓
  $z > 0 \Rightarrow z - 1 \geq z - 1$ ✓
- $z, z + 1, \ldots$
- $2 \cdot z, 3 \cdot z + 1, \ldots$
- $z \leq 0 \Rightarrow z + 1 \leq 0$ ↯
  $z > 0 \Rightarrow 2 \cdot (z - 1) \geq 2 \cdot z - 1$ ↯

finding them: just like ranking functions

Program Simplification

Algorithm

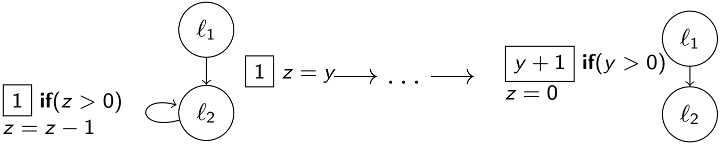
- while there is a path of length > 1



Program Simplification

Algorithm

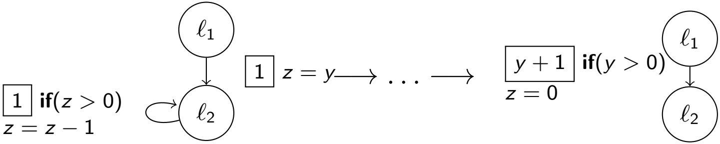
- while there is a path of length > 1
  - accelerate simple loops



Program Simplification

Algorithm

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

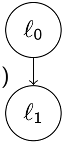


Simplified Programs

$\boxed{x^2 - y}$

$\textbf{if}(0 < x < 10 \land y + x < 0)$

$\dots$

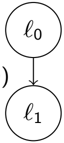


Simplified Programs

$\boxed{x^2 - y}$

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

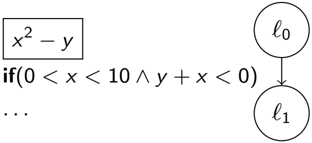
$\ldots$



- inferring lower bound still non-trivial

Example

Simplified Programs



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

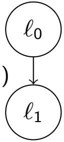
Example

Simplified Programs

$\boxed{x^2 - y}$

if$(0 < x < 10 \wedge y + x < 0)$

$\ldots$



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

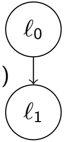
Example

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 $\boldsymbol{v}_n$ of valuations which satisfies the guard for large n
- apply $\boldsymbol{v}_n$ to cost to get asymptotic bound

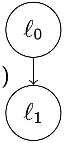
Example

Simplified Programs

$\boxed{x^2 - y}$

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

$\ldots$



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

Example

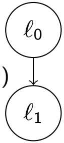
- $\boldsymbol{v}_n = \{x/1, y/-n\}$

Simplified Programs

$\boxed{x^2 - y}$

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

$\dots$

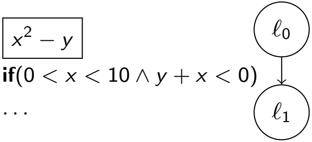


- inferring lower bound still non-trivial
- runtime depends on cost **and guard**
- search family $\boldsymbol{v}_n$ of valuations which satisfies the guard for large n
- apply $\boldsymbol{v}_n$ to cost to get asymptotic bound

Example

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

Simplified Programs



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

Example

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

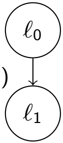
Example

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

$\boxed{x^2 - y}$

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

$\dots$



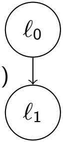
Example

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

$\boxed{x^2 - y}$

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

...



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

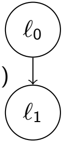
Example

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

$\boxed{x^2 - y}$

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

...



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

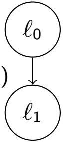
- $a^+ : \boldsymbol{v}_n(a)$ "increases with n" $\quad a^- : \boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!} : \boldsymbol{v}_n(a)$ is pos. constant $\quad a^{-!} : \boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\dots$



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

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

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

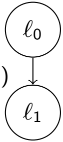
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\quad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

...



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

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

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

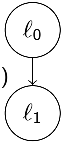
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\qquad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\qquad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\ldots$



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

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

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

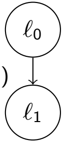
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\quad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\dots$



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

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

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

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

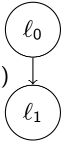
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\quad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\dots$



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

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

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

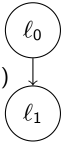
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\quad a^-$: $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\dots$



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

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

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

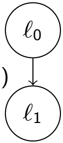
- a^{+} : $\boldsymbol{v}_n(a)$ "increases with n" $\quad$ a^{-} : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\dots$



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

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

$(-a)^+$ if a^-

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

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

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

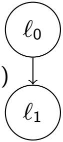
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\qquad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\qquad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

$\ldots$



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

observe :

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

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

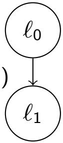
- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" $\quad$ a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $\quad$ $a^{-!}$: $\boldsymbol{v}_n(a)$ is neg. constant

Example

$\boxed{x^2 - y}$

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

...



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

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

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

- a^+ : $\boldsymbol{v}_n(a)$ "increases with n" a^- : $\boldsymbol{v}_n(a)$ "decreases with n"
- $a^{+!}$: $\boldsymbol{v}_n(a)$ is pos. constant $a^{-!}$: $\boldsymbol{v}_n(a)$ 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 $a_1 > 0 \land \cdots \land a_k > 0$
- 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 $\curvearrowright$ $\boldsymbol{v}_n$

Summary

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

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

Experiments

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

Experiments

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

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

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