Alternating Runtime and Size Complexity Analysis of Integer Programs

M. Brockschmidt[1] F. Emmes[2] S. Falke[3] C. Fuhs[4] J. Giesl[2]

[1]Microsoft Research, Cambridge

[2]RWTH Aachen University

[3]Karlsruhe Institute of Technology

[4]University College London

TACAS'14

Why care about complexity, anyway?

- **Mobile devices**: Bound energy usage
- **Security**: Denial of service attacks
- **Specifications**

Why care about complexity, anyway?

- **Mobile devices**: Bound energy usage
- **Security**: Denial of service attacks
- **Specifications**
- It's fun!

How is it different from termination?

Example (Linear runtime)

while $\texttt{i} > 0$ **do**
  $\texttt{i} = \texttt{i} - 1$

done
while $\texttt{x} > 0$ **do**
  $\texttt{x} = \texttt{x} - 1$
done

How is it different from termination?

Example (Linear runtime)

while $\mathtt{i} > 0$ **do**
  $\mathtt{i} = \mathtt{i} - 1$

done
while $\mathtt{x} > 0$ **do**
  $\mathtt{x} = \mathtt{x} - 1$
done

Example (Quadratic runtime)

while $\mathtt{i} > 0$ **do**
  $\mathtt{i} = \mathtt{i} - 1$
  $\mathtt{x} = \mathtt{x} + \mathtt{i}$
done
while $\mathtt{x} > 0$ **do**
  $\mathtt{x} = \mathtt{x} - 1$
done

How is it different from termination?

Example (Linear runtime)

while $\texttt{i} > 0$ **do**
  $\texttt{i} = \texttt{i} - 1$

done
while $\texttt{x} > 0$ **do**
  $\texttt{x} = \texttt{x} - 1$
done

Example (Quadratic runtime)

while $\texttt{i} > 0$ **do**
  $\texttt{i} = \texttt{i} - 1$
  $\texttt{x} = \texttt{x} + \texttt{i}$
done
while $\texttt{x} > 0$ **do**
  $\texttt{x} = \texttt{x} - 1$
done

Same termination argument, different complexity

How is it different from termination?

Example (Linear runtime)

while $\mathtt{i} > 0$ **do**
  $\mathtt{i} = \mathtt{i} - 1$

done
while $\mathtt{x} > 0$ **do**
  $\mathtt{x} = \mathtt{x} - 1$
done

Example (Quadratic runtime)

while $\mathtt{i} > 0$ **do**
  $\mathtt{i} = \mathtt{i} - 1$
  $\mathtt{x} = \mathtt{x} + \mathtt{i}$
done
while $\mathtt{x} > 0$ **do**
  $\mathtt{x} = \mathtt{x} - 1$
done

Same termination argument, different complexity

$\Leftarrow$ Difference in size of $\mathtt{x}$

But surely you are not the first?

- Using tools from termination proving:
  `ABC`, `AProVE`, `COSTA/PUBS`, `Loopus`, `Rank`, `TcT`, ...

But surely you are not the first?

- Using tools from termination proving:
  `ABC`, `AProVE`, `COSTA/PUBS`, `Loopus`, `Rank`, `TcT`, ...
- Using invariant generation:
  `SPEED`

But surely you are not the first?

- Using tools from termination proving:
  `ABC`, `AProVE`, `COSTA/PUBS`, `Loopus`, `Rank`, `TcT`, ...
- Using invariant generation:
  `SPEED`
- Using type-based amortized analysis:
  `RAML`, ...

Show me an example!

Example (List program)

Input: List x

ℓ_0: List y = null
ℓ_1: **while** x ≠ null **do**
    y = **new** List(x.val, y)
    x = x.next
  done
  List z = y
ℓ_2: **while** z ≠ null **do**
    List u = z.next
ℓ_3:   **while** u ≠ null **do**
      z.val += u.val
      u = u.next
    done
    z = z.next
  done

Show me an example!

Example (List program)

Input: `List x`

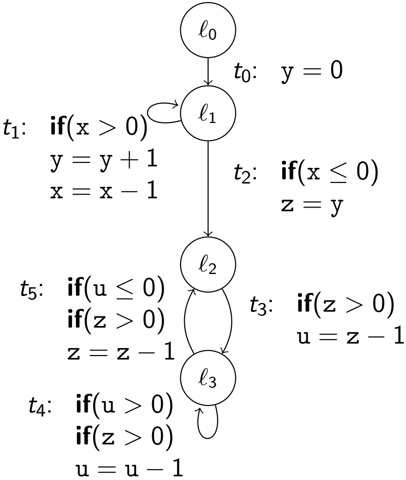
```
ℓ0: List y = null
ℓ1: while x ≠ null do
        y = new List(x.val, y)
        x = x.next
    done
    List z = y
ℓ2: while z ≠ null do
        List u = z.next
ℓ3:     while u ≠ null do
            z.val += u.val
            u = u.next
        done
        z = z.next
    done
```

Example (Integer abstraction)

Input: `int x`

```
ℓ0: int y = 0
ℓ1: while x ≠ 0 do
        y = y + 1
        x = x − 1
    done
    int z = y
ℓ2: while z ≠ 0 do
        int u = z − 1
ℓ3:     while u ≠ 0 do
            skip
            u = u − 1
        done
        z = z − 1
    done
```

Show me an example!



Example (Integer abstraction)

Input: `int x`

ℓ_0: `int` $\mathtt{y} = 0$

ℓ_1: **while** $\mathtt{x} \neq 0$ **do**

    $\mathtt{y} = \mathtt{y} + 1$

    $\mathtt{x} = \mathtt{x} - 1$

  done

  `int` $\mathtt{z} = \mathtt{y}$

ℓ_2: **while** $\mathtt{z} \neq 0$ **do**

    `int` $\mathtt{u} = \mathtt{z} - 1$

ℓ_3:   **while** $\mathtt{u} \neq 0$ **do**

      skip

      $\mathtt{u} = \mathtt{u} - 1$

    done

    $\mathtt{z} = \mathtt{z} - 1$

  done

How does the problem look like?

- **Programs** as Integer Transition Systems:
  - Locations $\mathcal{L}$: ℓ_0 start
  - Variables $\mathcal{V}$
  - Transitions $\mathcal{T}$: Formula over pre- $(x, \ldots)$, postvariables $(x', \ldots)$

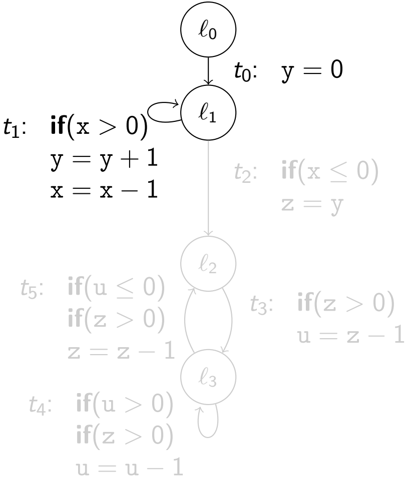
How does the problem and solution look like?

- **Programs** as Integer Transition Systems:
  - Locations $\mathcal{L}$: ℓ_0 start
  - Variables $\mathcal{V}$
  - Transitions $\mathcal{T}$: Formula over pre- $(x, \ldots)$, postvariables $(x', \ldots)$
- **Runtime complexity**:
  - $\mathcal{R}(t)$ bound on number of uses of $t \in \mathcal{T}$ in execution
  - $\mathcal{R}(t)$ monotonic function in $\mathcal{V}$, e.g. $|x|^2 + |y| + 1$
  - $\mathcal{R}(t)$ expresses bound in *input values*

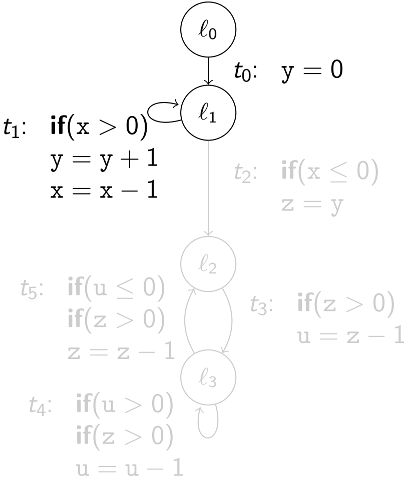
How does the problem and solution look like?

- **Programs** as Integer Transition Systems:
  - Locations $\mathcal{L}$: ℓ_0 start
  - Variables $\mathcal{V}$
  - Transitions $\mathcal{T}$: Formula over pre- $(x, \dots)$, postvariables $(x', \dots)$
- **Runtime complexity**:
  - $\mathcal{R}(t)$ bound on number of uses of $t \in \mathcal{T}$ in execution
  - $\mathcal{R}(t)$ monotonic function in $\mathcal{V}$, e.g. $|x|^2 + |y| + 1$
  - $\mathcal{R}(t)$ expresses bound in *input values*
- **Size complexity**:
  - $\mathcal{S}(t, v')$ bound on size of $v \in \mathcal{V}$ after using $t \in \mathcal{T}$
  - $\mathcal{S}(t, v')$ monotonic function in $\mathcal{V}$
  - $\mathcal{S}(t, v')$ expresses bound in *input values*

Runtime bounds I



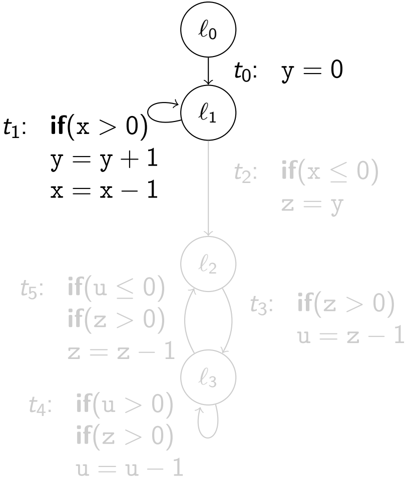
Runtime bounds I (PRFs)



Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Runtime bounds I (PRFs)



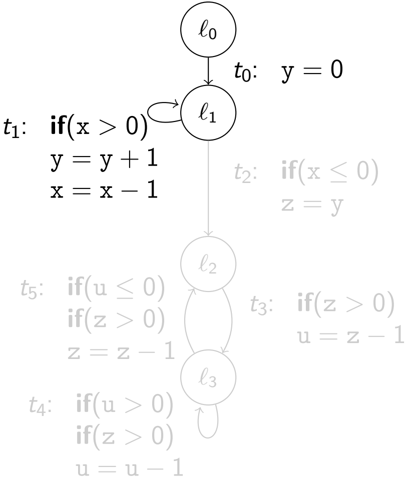
Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Example (PRF I)

$$\mathcal{P}_1(\ell) = x \quad \text{for all } \ell \in \mathcal{L}$$

Runtime bounds I (PRFs)



Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

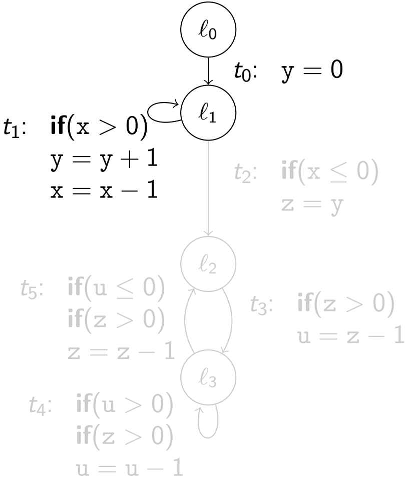
1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Example (PRF I)

$$\mathcal{P}_1(\ell) = x \quad \text{for all } \ell \in \mathcal{L}$$

no increase on any transition
t_1 **decreases**, **bounded**

Runtime bounds I (PRFs for complexity)

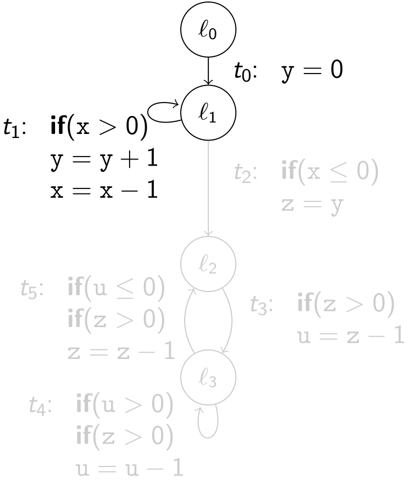


Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Key idea:
decreasing t used at most $\mathcal{P}(\ell_0)$

Runtime bounds I (PRFs for complexity)

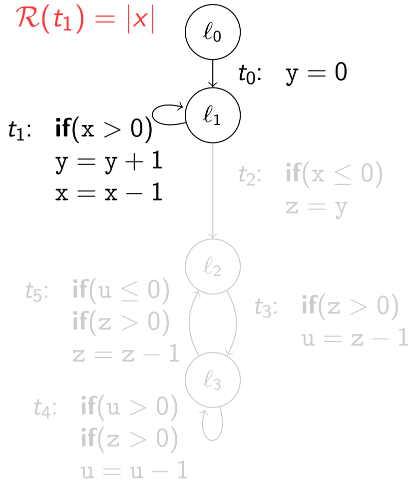


Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Key idea:
decreasing t used at most $\mathcal{P}(\ell_0)$
$\hookrightarrow \mathcal{R}(t) \leq [\mathcal{P}(\ell_0)]$
$[\,-\,] \equiv$ "make monotonic"

Runtime bounds I (PRFs for complexity)



Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

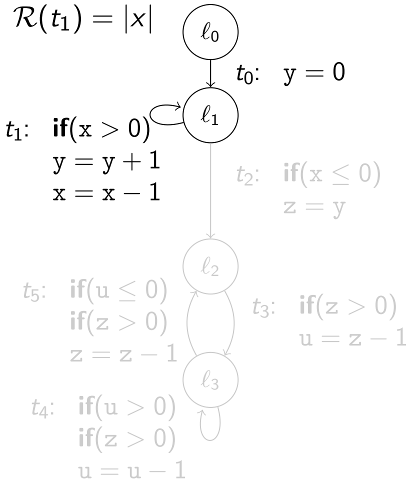
1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Key idea:
decreasing t used at most $\mathcal{P}(\ell_0)$
$\hookrightarrow \mathcal{R}(t) \leq [\mathcal{P}(\ell_0)]$
$[\,-\,] \equiv$ "make monotonic"

Runtime bounds I (PRFs for complexity)

$\mathcal{R}(t_0) = 1$

$\mathcal{R}(t_1) = |x|$



Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

Example (PRF II)

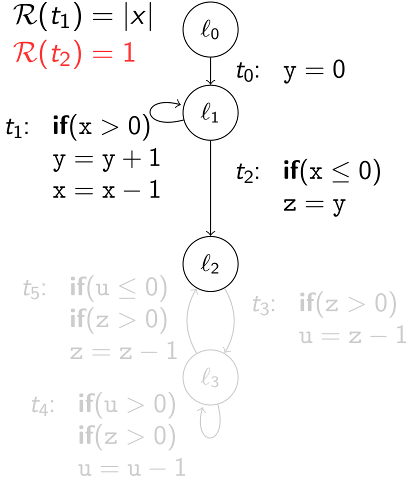
$$\mathcal{P}_2(\ell_0) = 1$$

$$\mathcal{P}_2(\ell) = 0 \quad \text{for all } \ell \in \mathcal{L} \setminus \{\ell_0\}$$

no increase on any transition
t_0 **decreases**, **bounded**

Runtime bounds I (PRFs for complexity)

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Polynomial rank function (PRF):
$\mathcal{P} : \mathcal{L} \to \mathbb{Z}[\mathcal{V}]$ with

1. **no increase**
   No transition increases
2. **decrease**
   At least one decreases
3. **bounded**
   Bounded from below by 1

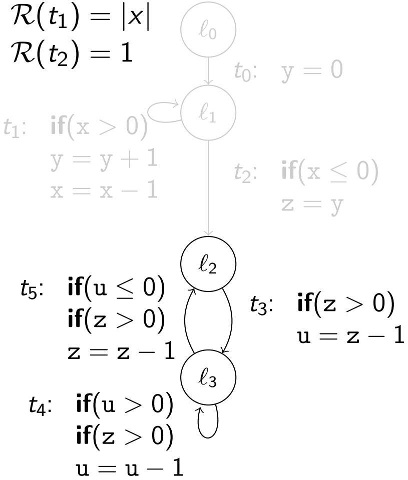
Example (PRF III)

$$\mathcal{P}_3(\ell) = 1 \quad \text{for all } \ell \in \{\ell_0, \ell_1\}$$
$$\mathcal{P}_3(\ell) = 0 \quad \text{for all } \ell \in \{\ell_2, \ell_3\}$$

no increase on any transition
t_2 **decreases**, **bounded**

Size bounds

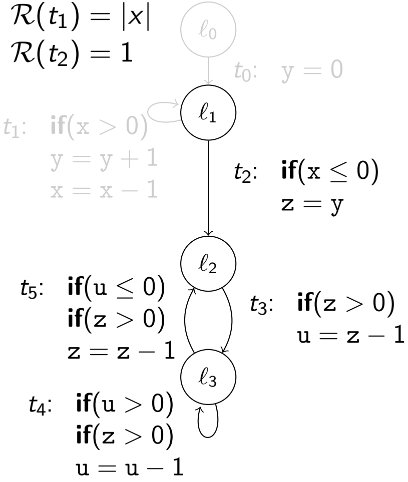
$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Second loop depends on z

Size bounds

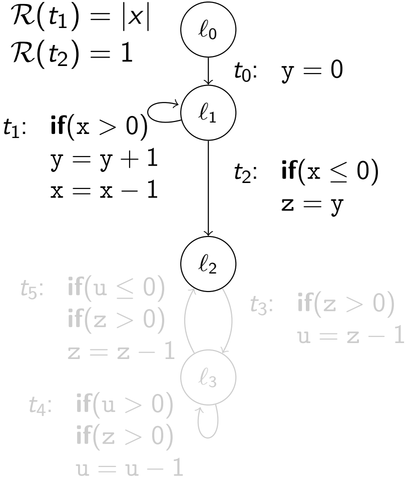
$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Second loop depends on z
$\hookrightarrow$ Compute $\mathcal{S}(t_2, z')$

Size bounds

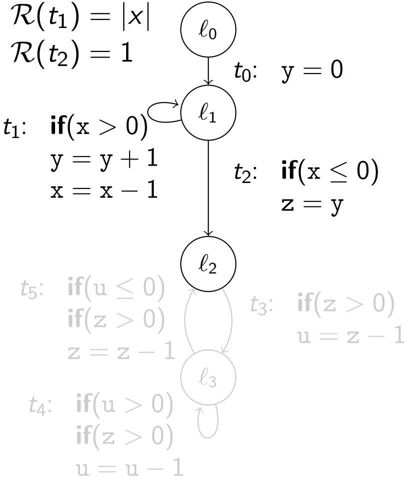
$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Second loop depends on z
$\hookrightarrow$ Compute $\mathcal{S}(t_2, z')$
... which depends on y after t_0, t_1

Size bounds: Local

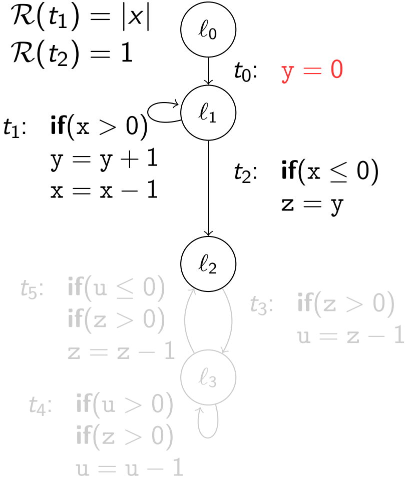
$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Result Variable Graph:

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



$0 \geq |t_0, \mathrm{y}'|$

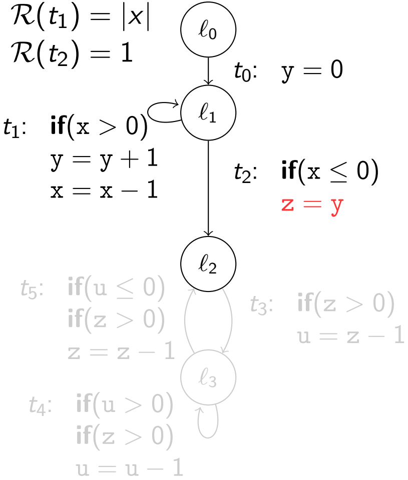
Result Variable Graph:

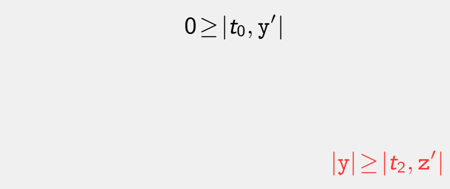
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$





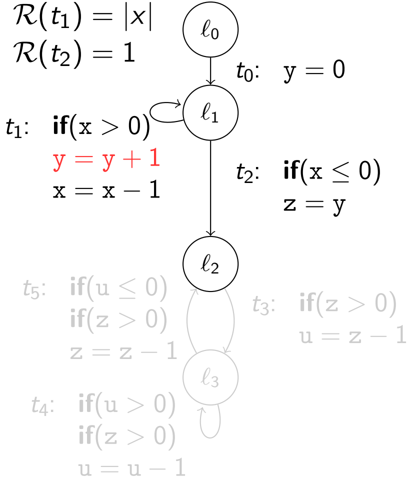
Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



$$0 \geq |t_0, \mathrm{y}'|$$

$$|\mathrm{y}| + 1 \geq |t_1, \mathrm{y}'|$$

$$|\mathrm{y}| \geq |t_2, \mathrm{z}'|$$

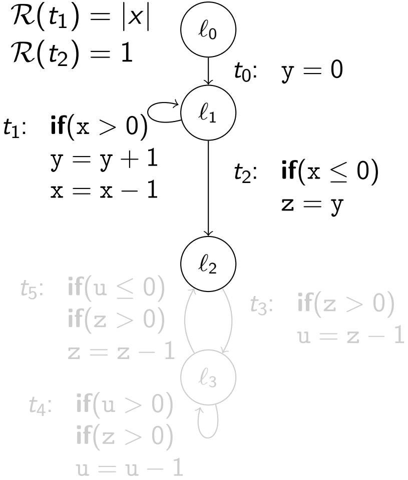
Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



$$0 \geq |t_0, \mathrm{y}'|$$

$$|\mathrm{y}| + 1 \geq |t_1, \mathrm{y}'|$$

$$|\mathrm{y}| \geq |t_2, \mathrm{z}'|$$

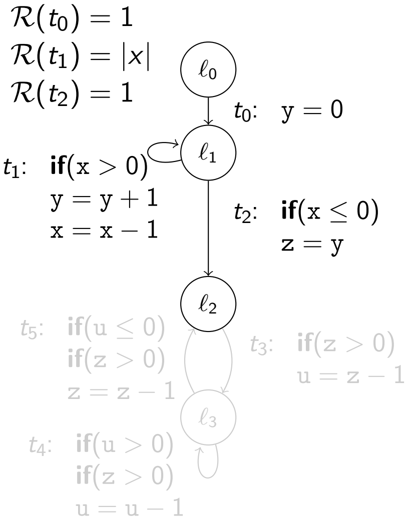
Result Variable Graph:

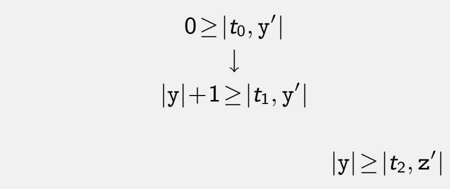
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

  $$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Size bounds: Local





Result Variable Graph:

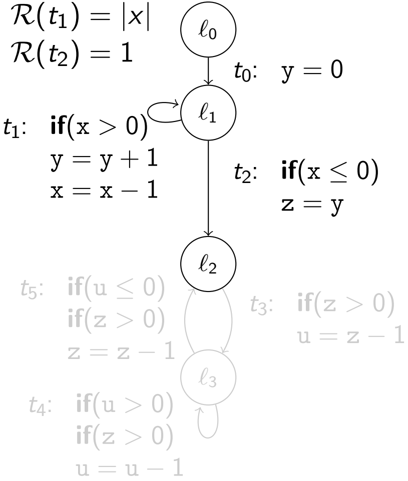
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

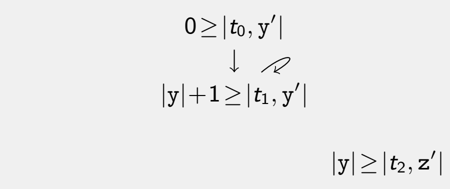
$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$





Result Variable Graph:

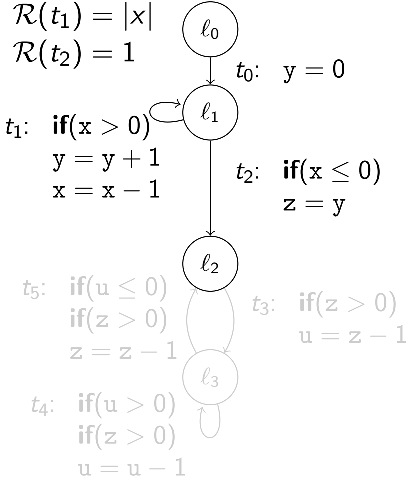
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

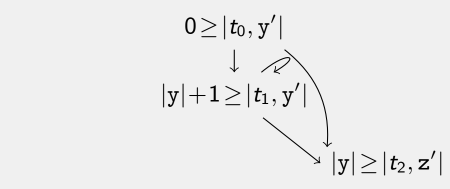
$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$





Result Variable Graph:

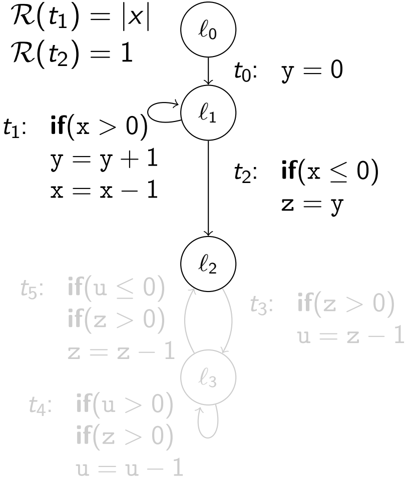
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

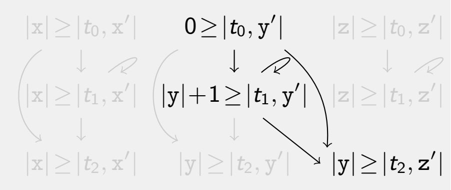
$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Size bounds: Local

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$





Result Variable Graph:

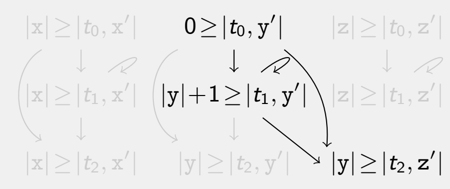
- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Size bounds: Global

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$



Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

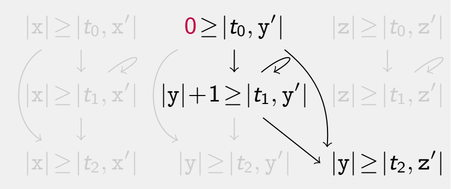
- Edges:
  Flow of information

Computing $\mathcal{S}(t, v')$:

Size bounds: Global

$\mathcal{R}(t_0) = 1$
$\mathcal{R}(t_1) = |x|$
$\mathcal{R}(t_2) = 1$

$\mathcal{S}(t_0, y') = 0$



Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Computing $\mathcal{S}(t, v')$:

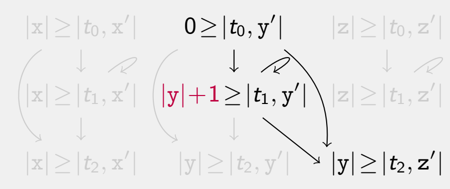
- No cycles: $\mathcal{S}_l$

Size bounds: Global

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$



Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

- Edges:
  Flow of information

Computing $\mathcal{S}(t, v')$:

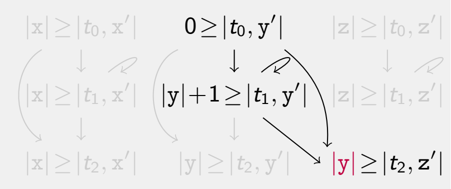
- No cycles: $\mathcal{S}_l$
- Cycles: Combine $\mathcal{R}$, $\mathcal{S}_l$
  - if $\mathcal{S}_l \approx v + c$, $c \in \mathbb{Z}$:
    $\mathcal{S}(t, v') = \mathcal{S}(\tilde{t}, v') + \mathcal{R}(t) \cdot c$
    $\tilde{t}$ predecessor of t

Size bounds: Global

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$



Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

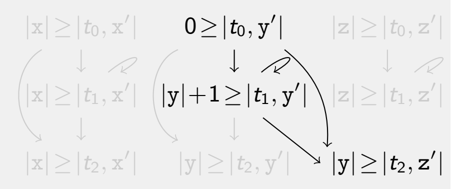
- Edges:
  Flow of information

Computing $\mathcal{S}(t, v')$:

- No cycles: $\mathcal{S}_l$ (+ propagation)
- Cycles: Combine $\mathcal{R}$, $\mathcal{S}_l$
  - if $\mathcal{S}_l \approx v + c$, $c \in \mathbb{Z}$:
    $\mathcal{S}(t, v') = \mathcal{S}(\tilde{t}, v') + \mathcal{R}(t) \cdot c$
    $\tilde{t}$ predecessor of t

Size bounds: Global

$$\mathcal{R}(t_0) = 1 \qquad \mathcal{S}(t_0, y') = 0$$
$$\mathcal{R}(t_1) = |x| \qquad \mathcal{S}(t_1, y') = |x|$$
$$\mathcal{R}(t_2) = 1 \qquad \mathcal{S}(t_2, z') = |x|$$



Result Variable Graph:

- Nodes $|t, v'|$, labels $S_l(t, v')$
  Change of v in one use of t:

$$t \implies S_l(t, v)(\mathcal{V}) \geq v'$$

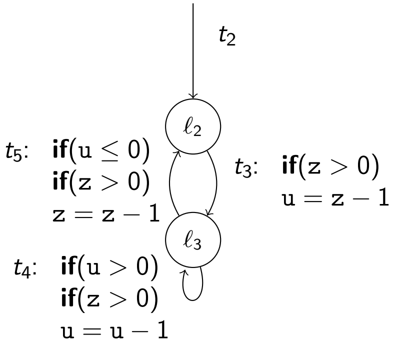
- Edges:
  Flow of information

Computing $\mathcal{S}(t, v')$:

- No cycles: $\mathcal{S}_l$ (+ propagation)
- Cycles: Combine $\mathcal{R}$, $\mathcal{S}_l$
  - if $\mathcal{S}_l \approx v + c$, $c \in \mathbb{Z}$:
    $\mathcal{S}(t, v') = \mathcal{S}(\tilde{t}, v') + \mathcal{R}(t) \cdot c$
    $\tilde{t}$ predecessor of t
  - More complex: See paper

Runtime bounds II: Modularity

$$\mathcal{R}(t_0) = 1 \qquad \mathcal{S}(t_0, y') = 0$$
$$\mathcal{R}(t_1) = |x| \qquad \mathcal{S}(t_1, y') = |x|$$
$$\mathcal{R}(t_2) = 1 \qquad \mathcal{S}(t_2, z') = |x|$$

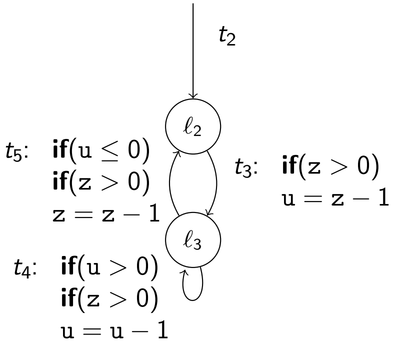


Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1 \qquad \mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x| \qquad \mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1 \qquad \mathcal{S}(t_2, z') = |x|$



Example (PRF IV)

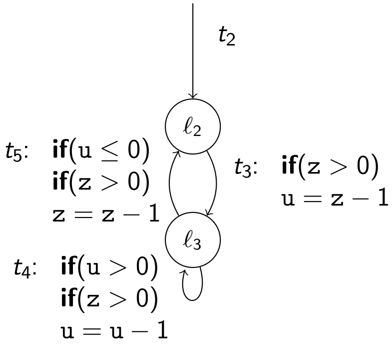
Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad\mathcal{S}(t_2, z') = |x|$



Example (PRF IV)

Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

$$\mathcal{P}_4(\ell_2) = \mathcal{P}_4(\ell_3) = z$$

no increase on transitions $\mathcal{T}_1$

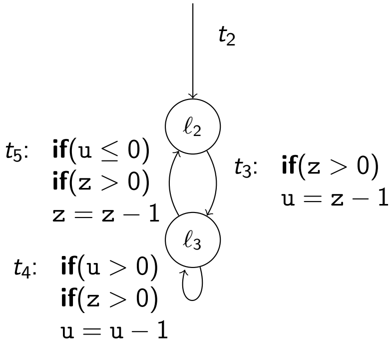
t_5 **decreases**, **bounded**

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$



Example (PRF IV)

Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

$$\mathcal{P}_4(\ell_2) = \mathcal{P}_4(\ell_3) = z$$

no increase on transitions $\mathcal{T}_1$

t_5 **decreases**, **bounded**

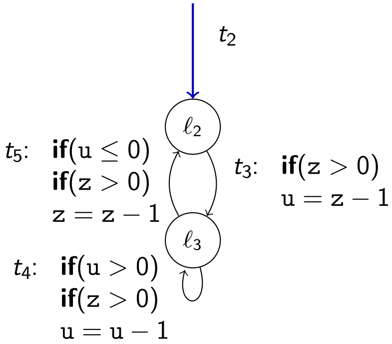
$\hookrightarrow$ **When** $\mathcal{T}_1$ reached, then z **steps**:

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$



Example (PRF IV)

Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

$$\mathcal{P}_4(\ell_2) = \mathcal{P}_4(\ell_3) = z$$

no increase on transitions $\mathcal{T}_1$
t_5 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_1$ reached, then z **steps**:

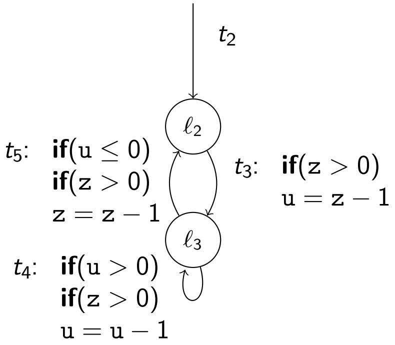
$\mathcal{T}_1$ reached $\mathcal{R}(t_2) = 1$ time

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad \mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad \mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad \mathcal{S}(t_2, z') = |x|$



Example (PRF IV)

Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

$$\mathcal{P}_4(\ell_2) = \mathcal{P}_4(\ell_3) = z$$

no increase on transitions $\mathcal{T}_1$
t_5 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_1$ reached, then z **steps**:

$\mathcal{T}_1$ reached $\mathcal{R}(t_2) = 1$ time

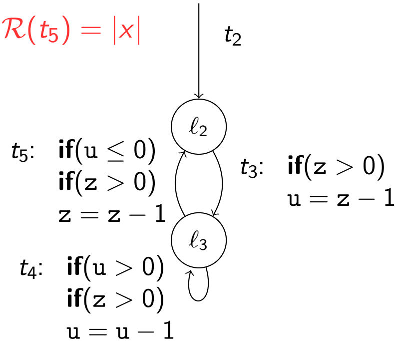
z has size $\mathcal{S}(t_2, y') = |x|$

$\mathcal{R}(t_0) = 1$ $\qquad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\qquad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\qquad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_5) = |x|$



Example (PRF IV)

Consider only $\mathcal{T}_1 = \{t_3, t_4, t_5\}$

$$\mathcal{P}_4(\ell_2) = \mathcal{P}_4(\ell_3) = z$$

no increase on transitions $\mathcal{T}_1$

t_5 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_1$ reached, then z **steps**:

$\mathcal{T}_1$ reached $\mathcal{R}(t_2) = 1$ time

z has size $\mathcal{S}(t_2, y') = |x|$

$$\hookrightarrow \mathcal{R}(t_5) = \mathcal{R}(t_2) \cdot \mathcal{S}(t_2, y') = 1 \cdot |x|$$

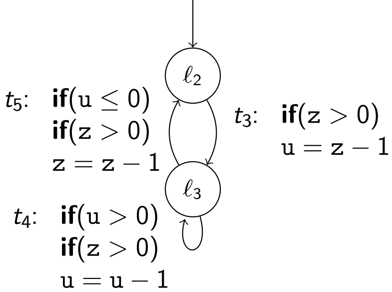
Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_5) = |x|$



Example (PRF V)

Consider only $\mathcal{T}_2 = \{t_3, t_4\}$

$$\mathcal{P}_4(\ell_2) = 1 \quad \mathcal{P}_4(\ell_3) = 0$$

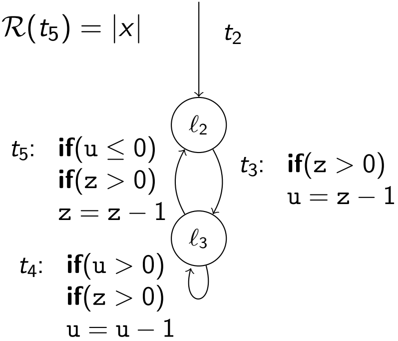
no increase on transitions $\mathcal{T}_2$

t_3 **decreases**, **bounded**

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$
$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$
$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_5) = |x|$



Example (PRF V)

Consider only $\mathcal{T}_2 = \{t_3, t_4\}$

$$\mathcal{P}_4(\ell_2) = 1 \quad \mathcal{P}_4(\ell_3) = 0$$

no increase on transitions $\mathcal{T}_2$
t_3 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_2$ reached, then 1 **step**:

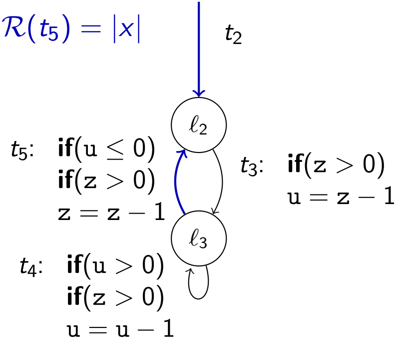
Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\qquad \mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\qquad \mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\qquad \mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_5) = |x|$



Example (PRF V)

Consider only $\mathcal{T}_2 = \{t_3, t_4\}$

$$\mathcal{P}_4(\ell_2) = 1 \quad \mathcal{P}_4(\ell_3) = 0$$

no increase on transitions $\mathcal{T}_2$
t_3 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_2$ reached, then 1 **step**:

$\mathcal{T}_2$ reached
$\mathcal{R}(t_2) = 1$ time and
$\mathcal{R}(t_5) = x$ times

Runtime bounds II: Modularity

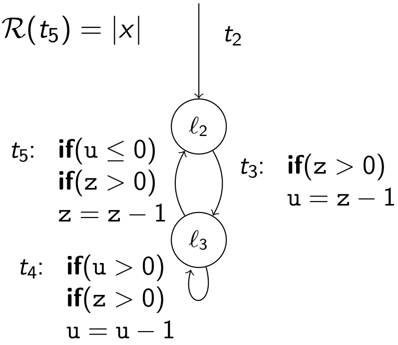
$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_3) = |x| + 1$

$\mathcal{R}(t_5) = |x|$



Example (PRF V)

Consider only $\mathcal{T}_2 = \{t_3, t_4\}$

$$\mathcal{P}_4(\ell_2) = 1 \quad \mathcal{P}_4(\ell_3) = 0$$

no increase on transitions $\mathcal{T}_2$
t_3 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_2$ reached, then 1 **step**:

$\mathcal{T}_2$ reached
$\mathcal{R}(t_2) = 1$ time and
$\mathcal{R}(t_5) = x$ times

$$\hookrightarrow \mathcal{R}(t_3) = \mathcal{R}(t_2) \cdot 1 + \mathcal{R}(t_5) \cdot 1$$
$$= 1 \cdot 1 + |x| \cdot 1$$

Runtime bounds II: Modularity

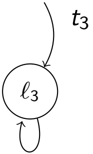
$\mathcal{R}(t_0) = 1 \qquad \mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x| \qquad \mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1 \qquad \mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_3) = |x| + 1$

$\mathcal{R}(t_5) = |x|$



t_4: **if**($\mathtt{u} > 0$)
if($\mathtt{z} > 0$)
$\mathtt{u} = \mathtt{u} - 1$

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

Runtime bounds II: Modularity

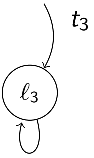
$\mathcal{R}(t_0) = 1$ $\qquad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\qquad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\qquad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_3) = |x| + 1$

$\mathcal{R}(t_5) = |x|$



t_4: **if**$(\mathrm{u} > 0)$
if$(\mathrm{z} > 0)$
$\mathrm{u} = \mathrm{u} - 1$

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

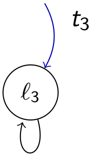
no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_3$ reached, then u **steps**:

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\qquad \mathcal{S}(t_0, y') = 0$
$\mathcal{R}(t_1) = |x|$ $\qquad \mathcal{S}(t_1, y') = |x|$
$\mathcal{R}(t_2) = 1$ $\qquad \mathcal{S}(t_2, z') = |x|$
$\mathcal{R}(t_3) = |x| + 1$

$\mathcal{R}(t_5) = |x|$



t_4:
```
if(u > 0)
if(z > 0)
u = u - 1
```

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_3$ reached, then u **steps**:

$\mathcal{T}_3$ reached $\mathcal{R}(t_3) = |x| + 1$ times

Runtime bounds II: Modularity

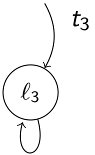
$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_3) = |x| + 1$

$\mathcal{R}(t_5) = |x|$



t_4: $\textbf{if}(\mathtt{u} > 0)$
$\textbf{if}(\mathtt{z} > 0)$
$\mathtt{u} = \mathtt{u} - 1$

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_3$ reached, then u **steps**:

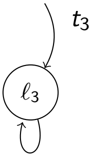
$\mathcal{T}_3$ reached $\mathcal{R}(t_3) = |x| + 1$ times

u has size $\mathcal{S}(t_3, u')$

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad\mathcal{S}(t_0, y') = 0$
$\mathcal{R}(t_1) = |x|$ $\quad\mathcal{S}(t_1, y') = |x|$
$\mathcal{R}(t_2) = 1$ $\quad\mathcal{S}(t_2, z') = |x|$
$\mathcal{R}(t_3) = |x| + 1$ $\quad\mathcal{S}(t_3, u') = |x|$

$\mathcal{R}(t_5) = |x|$



t_4: **if**$(\mathrm{u} > 0)$
if$(\mathrm{z} > 0)$
$\mathrm{u} = \mathrm{u} - 1$

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_3$ reached, then u **steps**:

$\mathcal{T}_3$ reached $\mathcal{R}(t_3) = |x| + 1$ times
u has size $\mathcal{S}(t_3, u') = |x|$

Runtime bounds II: Modularity

$\mathcal{R}(t_0) = 1$ $\quad$ $\mathcal{S}(t_0, y') = 0$

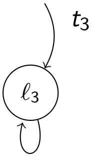
$\mathcal{R}(t_1) = |x|$ $\quad$ $\mathcal{S}(t_1, y') = |x|$

$\mathcal{R}(t_2) = 1$ $\quad$ $\mathcal{S}(t_2, z') = |x|$

$\mathcal{R}(t_3) = |x| + 1$ $\quad$ $\mathcal{S}(t_3, u') = |x|$

$\mathcal{R}(t_4) = |x|^2 + |x|$

$\mathcal{R}(t_5) = |x|$



t_4: $\textbf{if}(\mathtt{u} > 0)$
$\textbf{if}(\mathtt{z} > 0)$
$\mathtt{u} = \mathtt{u} - 1$

Example (PRF VI)

Consider only $\mathcal{T}_3 = \{t_4\}$

$$\mathcal{P}_5(\ell_3) = u$$

no increase on transitions $\mathcal{T}_3$
t_4 **decreases**, **bounded**

$\hookrightarrow$ **When** $\mathcal{T}_3$ reached, then u **steps**:

$\mathcal{T}_3$ reached $\mathcal{R}(t_3) = |x| + 1$ times

u has size $\mathcal{S}(t_3, u') = |x|$

$$\hookrightarrow \mathcal{R}(t_4) = \mathcal{R}(t_3) \cdot \mathcal{S}(t_3, u') = (|x| + 1) \cdot |x|$$

TimeBounds: Procedure

TimeBounds($\mathcal{R}, \mathcal{S}$)

Input: Runtime bounds $\mathcal{R}$, Size bounds $\mathcal{S}$

$\mathcal{T}' \leftarrow \{t \in \mathcal{T} \mid \mathcal{R}(t) \text{ unbounded}\}$

$\mathcal{P} \leftarrow \text{synthPRF}(\mathcal{T}')$

$\mathcal{L}_{\downarrow} \leftarrow \text{entryLocations}(\mathcal{T}')$

$\mathcal{T}_{\ell} \leftarrow \text{leadingTo}(\ell, \mathcal{T} \setminus \mathcal{T}')$

$\mathcal{R}' \leftarrow \mathcal{R}$

for all $t \in \mathcal{T}'$ decreasing under $\mathcal{P}$ **do**

$\mathcal{R}'(t) \leftarrow \sum_{\ell \in \mathcal{L}_{\downarrow}, \tilde{t} \in \mathcal{T}_{\ell}} \mathcal{R}(\tilde{t}) \cdot [\mathcal{P}(\ell)](\mathcal{S}(\tilde{t}, v_1'), \ldots, \mathcal{S}(\tilde{t}, v_n'))$

end for

Output: $\mathcal{R}'$

SizeBounds: Procedure

SizeBoundsTriv($\mathcal{R}, \mathcal{S}, C$)

Input: Runtime bounds $\mathcal{R}$, Size bounds $\mathcal{S}$, $C = \{|t, v'|\}$

$$\mathcal{T}_t \leftarrow \text{leadingTo}(t, \mathcal{T})$$
$$\mathcal{S}' \leftarrow \mathcal{S}$$
$$\mathcal{S}'(t, v') \leftarrow \max\{\mathcal{S}_l(t, v')(\mathcal{S}(\tilde{t}, v'_1), \ldots, \mathcal{S}(\tilde{t}, v'_n)) \mid \tilde{t} \in \mathcal{T}_t\}$$

Output: $\mathcal{S}'$

SizeBounds: Procedure

SizeBoundsTriv($\mathcal{R}, \mathcal{S}, C$)

Input: Runtime bounds $\mathcal{R}$, Size bounds $\mathcal{S}$, $C = \{|t, v'|\}$

$$\mathcal{T}_t \leftarrow \text{leadingTo}(t, \mathcal{T})$$
$$\mathcal{S}' \leftarrow \mathcal{S}$$
$$\mathcal{S}'(t, v') \leftarrow \max\{\mathcal{S}_l(t, v')(\mathcal{S}(\tilde{t}, v'_1), \ldots, \mathcal{S}(\tilde{t}, v'_n)) \mid \tilde{t} \in \mathcal{T}_t\}$$

Output: $\mathcal{S}'$

SizeBoundsNonTriv($\mathcal{R}, \mathcal{S}, C$)

Case C non-trivial SCC: See paper

AlternatingCompl: Overall procedure

AlternatingCompl($\mathcal{T}, \mathcal{V}$)

Input: Program of transitions $\mathcal{T}$, variables $\mathcal{V}$

    $\mathcal{R} \leftarrow$ unboundedTimeCompl($\mathcal{T}$)
    $\mathcal{S} \leftarrow$ unboundedSizeCompl($\mathcal{T}, \mathcal{V}$)
    while $\mathcal{R}, \mathcal{S}$ have unbounded elements **do**
        $\mathcal{R} \leftarrow$ TimeBounds($\mathcal{R}, \mathcal{S}$)
        for all C SCC of RVG($\mathcal{T}, \mathcal{V}$) **do**
            $\mathcal{S} \leftarrow$ SizeBounds($\mathcal{R}, \mathcal{S}, C$)
        end for
    end while

Output: $\mathcal{R}$, $\mathcal{S}$

Did you ever test that?

Prototype: `KoAT`. 682 examples, taken from

- prior evaluations (of `ABC`, `Loopus`, `PUBS/COSTA`, `Rank`, `SPEED`)
- termination benchmarks (of `T2`, `AProVE`)
- examples from the paper

Did you ever test that?

Prototype: KoAT. 682 examples, taken from

- prior evaluations (of ABC, Loopus, PUBS/COSTA, Rank, SPEED)
- termination benchmarks (of T2, AProVE)
- examples from the paper

	1	log n	n	$n \log n$	n^2	n^3	$n^{>3}$	EXP	Time
KoAT	121	0	145	0	59	3	3	0	1.1 s
PUBS	116	5	131	5	22	7	0	6	0.8 s
Rank	56	0	19	0	8	1	0	0	0.5 s

(Timeout 60s)

Did you ever test that?

Prototype: KoAT. 682 examples, taken from

- prior evaluations (of ABC, Loopus, PUBS/COSTA, Rank, SPEED)
- termination benchmarks (of T2, AProVE)
- examples from the paper

	1	$\log n$	n	$n \log n$	n^2	n^3	$n^{>3}$	EXP	Time
KoAT	121	0	145	0	59	3	3	0	1.1 s
PUBS	116	5	131	5	22	7	0	6	0.8 s
Rank	56	0	19	0	8	1	0	0	0.5 s

(Timeout 60s)

Experiments, Implementation available:

http://aprove.informatik.rwth-aachen.de/eval/IntegerComplexity

Did you ever test that?

Prototype: KoAT. 682 examples, taken from

- prior evaluations (of ABC, Loopus, PUBS/COSTA, Rank, SPEED)
- termination benchmarks (of T2, AProVE)
- examples from the paper

	1	log n	n	$n \log n$	n^2	n^3	$n^{>3}$	EXP	Time
KoAT	121	0	145	0	59	3	3	0	1.1 s
PUBS	116	5	131	5	22	7	0	6	0.8 s
Rank	56	0	19	0	8	1	0	0	0.5 s

(Timeout 60s)

Experiments, Implementation available:

http://aprove.informatik.rwth-aachen.de/eval/IntegerComplexity

Source:

https://github.com/s-falke/kittel-koat

So, is that everything?

Extensions implemented/on the way:

- Recursion
- Exponential data (e.g., **while** $\mathtt{x} > 0$ **do** $\mathtt{y} = 2 \cdot \mathtt{y}; \mathtt{x} - -;$ **done**)
- Methods handled independently, composing results at call sites
- Other cost measures (e.g., network calls, energy usage, . . .)

So, is that everything?

Extensions implemented/on the way:

- Recursion
- Exponential data (e.g., **while** $\mathtt{x} > 0$ **do** $\mathtt{y} = 2 \cdot \mathtt{y}$; $\mathtt{x} - -$; **done**)
- Methods handled independently, composing results at call sites
- Other cost measures (e.g., network calls, energy usage, ...)

Future plans:

- Exponential calls (e.g., $f(n) = f(n-1) + f(n-2)$)
- Recurrences for more precise results (where applicable)
- Heap
- Programming language frontends (via `KITTeL`, `AProVE`, ...)
- *Lower* bounds

What should I take home?

Key insights:

- Data size influences runtime
- Runtime influences data size
- *Other influences minor*

What should I take home?

Key insights:

- Data size influences runtime
- Runtime influences data size
- *Other influences minor*

Solution:

- Alternating size/runtime analysis
- Modularity by using *only* these results