Automated Complexity Analysis of Rewrite Systems

Florian Frohn

RWTH Aachen University, Germany

December 11, 2018

Automated Complexity Analysis of Rewrite Systems

Florian Frohn

RWTH Aachen University, Germany

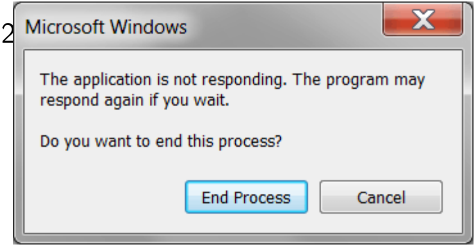
December 11, 2018

Automated Complexity Analysis of Rewrite Systems

Florian Frohn

RWTH Aachen University, Germany

December 11, 2



What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
    $x = x - 1$
    $z = z + y$
end
return z

- x loop iterations

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

- x loop iterations
- 5 operations per iteration

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

- x loop iterations
- 5 operations per iteration
- complexity $5 \cdot x + 2$

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
    $x = x - 1$
    $z = z + y$
end
return z

- x loop iterations
- 5 operations per iteration
- complexity $5 \cdot x + 2$
- upper bound: $7 \cdot x + 4$, ...

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

- x loop iterations
- 5 operations per iteration
- complexity $5 \cdot x + 2$
- upper bound: $7 \cdot x + 4$, ...
- lower bound: $x + 1$, ...

What is Complexity Analysis?

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

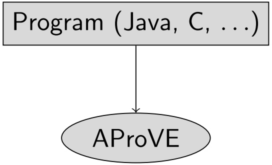
- x loop iterations
- 5 operations per iteration
- complexity $5 \cdot x + 2$
- upper bound: $7 \cdot x + 4$, ...
- lower bound: $x + 1$, ...

Goal: compute such bounds automatically

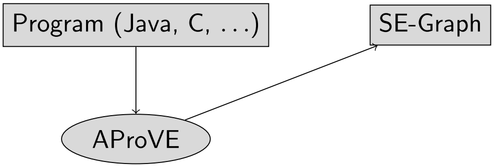
RWTH AACHEN UNIVERSITY

Complexity Analysis with AProVE

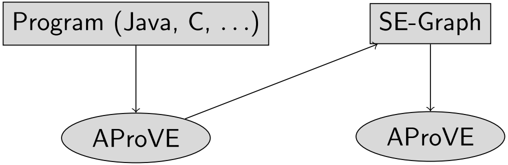
Complexity Analysis with AProVE



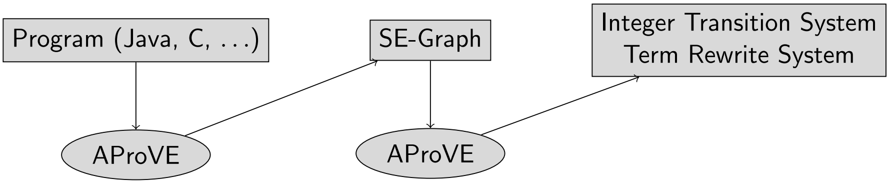
Complexity Analysis with AProVE



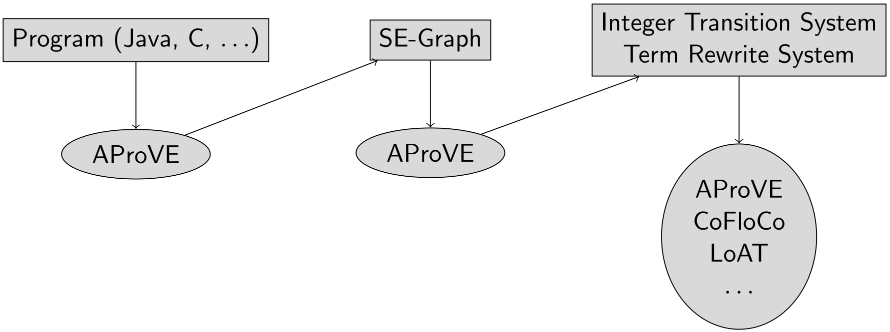
Complexity Analysis with AProVE



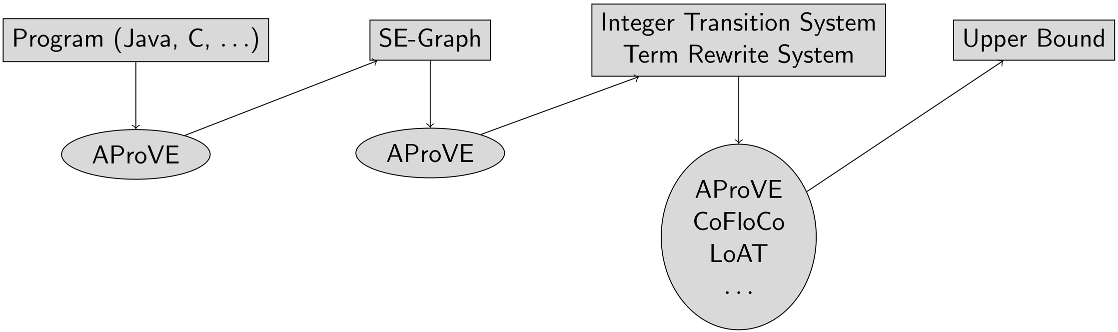
Complexity Analysis with AProVE



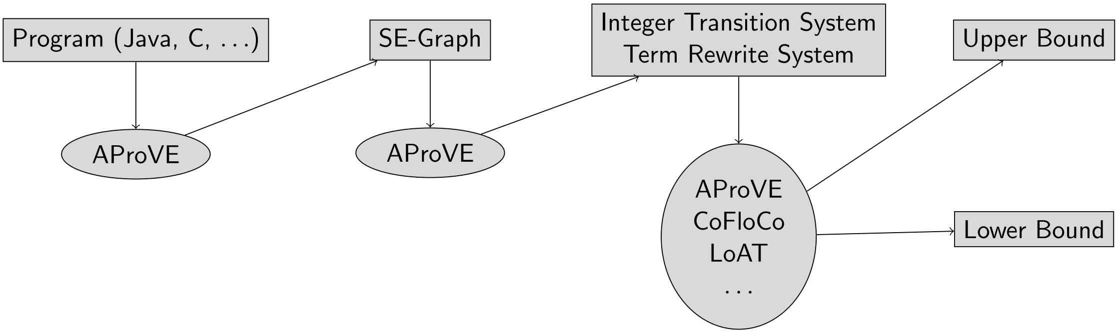
Complexity Analysis with AProVE



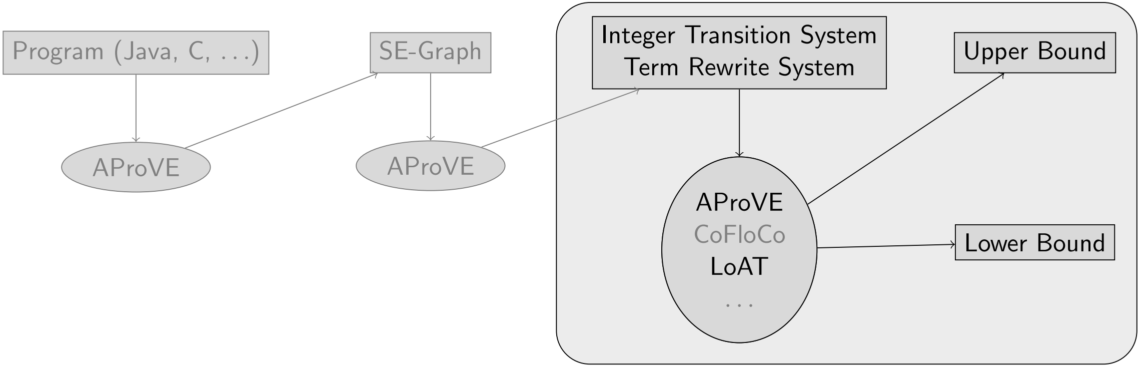
Complexity Analysis with AProVE



Complexity Analysis with AProVE



Complexity Analysis with AProVE



Rule-Based Representation of Programs

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \mathtt{init}(x,y) & \xrightarrow{1} & \mathtt{while}(x,y,0) & [x \geq 0 \wedge y \geq 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x > 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{1} & \mathtt{return}(z) & [x = 0] \end{array}$$

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\texttt{init}(x, y) \xrightarrow{1} \texttt{while}(x, y, 0) \quad [x \geq 0 \land y \geq 0]$$

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x - 1, y, z + y) \quad [x > 0]$$

$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$\mathtt{init}(x, y)$	$\xrightarrow{1}$	$\mathtt{while}(x, y, 0)$	$[x \geq 0 \land y \geq 0]$
$\mathtt{while}(x, y, z)$	$\xrightarrow{5}$	$\mathtt{while}(x - 1, y, z + y)$	$[x > 0]$
$\mathtt{while}(x, y, z)$	$\xrightarrow{1}$	$\mathtt{return}(z)$	$[x = 0]$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\texttt{init}(x, y) \xrightarrow{1} \texttt{while}(x, y, 0) \quad [x \geq 0 \land y \geq 0]$$

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$

$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\texttt{init}(x, y) \xrightarrow{1} \texttt{while}(x, y, 0) \quad [x \geq 0 \land y \geq 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$
\begin{array}{llll}
\texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\
\texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}
$$

Example (Term Rewrite System (TRS))

$$
\begin{array}{lll}
\texttt{length}(\mathbf{cons}(x,x')) & \to & \mathsf{s}(\texttt{length}(x')) \\
\texttt{length}(\mathbf{nil}) & \to & \mathbf{0}
\end{array}
$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \mathtt{init}(x,y) & \xrightarrow{1} & \mathtt{while}(x,y,0) & [x \geq 0 \wedge y \geq 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x > 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{1} & \mathtt{return}(z) & [x = 0] \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

$0 \curvearrowright \mathbf{0}$

Example (Term Rewrite System (TRS))

$$\begin{array}{lll} \mathtt{length}(\mathbf{cons}(x,x')) & \to & \mathtt{s}(\mathtt{length}(x')) \\ \mathtt{length}(\mathbf{nil}) & \to & \mathbf{0} \end{array}$$

RWTH AACHEN UNIVERSITY

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

Example (Term Rewrite System (TRS))

$$\begin{array}{lll} \texttt{length}(\mathbf{cons}(x,x')) & \to & \mathrm{s}(\texttt{length}(x')) \\ \texttt{length}(\mathbf{nil}) & \to & \mathbf{0} \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

$0 \curvearrowright \mathbf{0}$
$1 \curvearrowright \mathrm{s}(\mathbf{0})$

RWTH AACHEN UNIVERSITY

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

Example (Term Rewrite System (TRS))

$$\begin{array}{lll} \texttt{length}(\mathbf{cons}(x,x')) & \to & \mathsf{s}(\texttt{length}(x')) \\ \texttt{length}(\mathbf{nil}) & \to & \mathbf{0} \end{array}$$

input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

$$\begin{array}{lll} 0 & \curvearrowright & \mathbf{0} \\ 1 & \curvearrowright & \mathsf{s}(\mathbf{0}) \\ 2 & \curvearrowright & \mathsf{s}(\mathsf{s}(\mathbf{0})) \end{array}$$

RWTH AACHEN UNIVERSITY

Rule-Based Representation of Programs

Example (Integer Transition System (ITS))

$$\begin{array}{llll} \mathtt{init}(x,y) & \xrightarrow{1} & \mathtt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x > 0] \\ \mathtt{while}(x,y,z) & \xrightarrow{1} & \mathtt{return}(z) & [x = 0] \end{array}$$

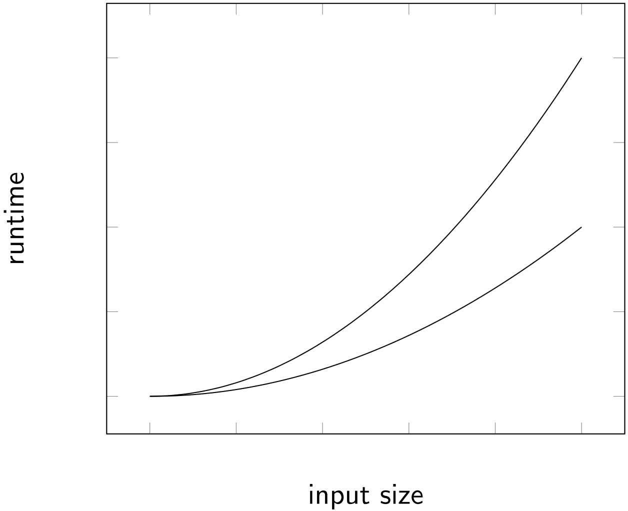
Example (Term Rewrite System (TRS))

$$\begin{array}{lll} \mathtt{length}(\mathbf{cons}(x,x')) & \to & \mathrm{s}(\mathtt{length}(x')) \\ \mathtt{length}(\mathbf{nil}) & \to & \mathbf{0} \end{array}$$

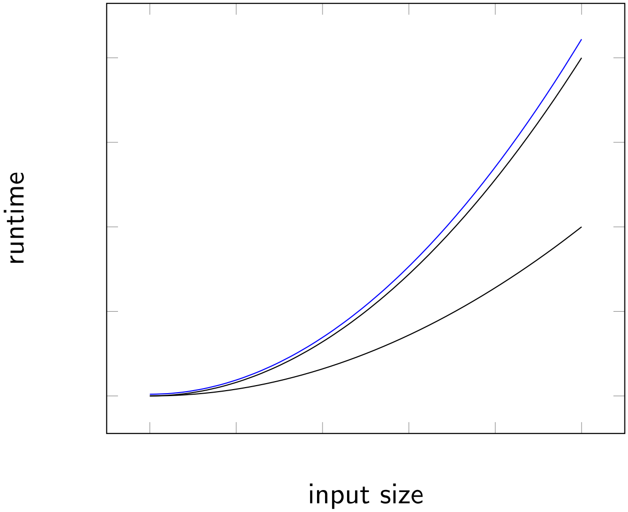
input: $x, y \in \mathbb{N}$
$z = 0$
while $x > 0$ **do**
  $x = x - 1$
  $z = z + y$
end
return z

$$\begin{array}{lll} 0 & \curvearrowright & \mathbf{0} \\ 1 & \curvearrowright & \mathrm{s}(\mathbf{0}) \\ 2 & \curvearrowright & \mathrm{s}(\mathrm{s}(\mathbf{0})) \\ & \dots & \end{array}$$

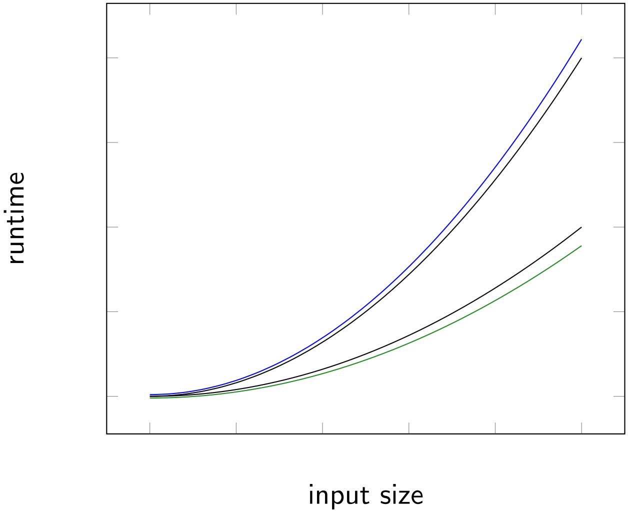
Upper, Lower, Worst Case, Best Case, ...



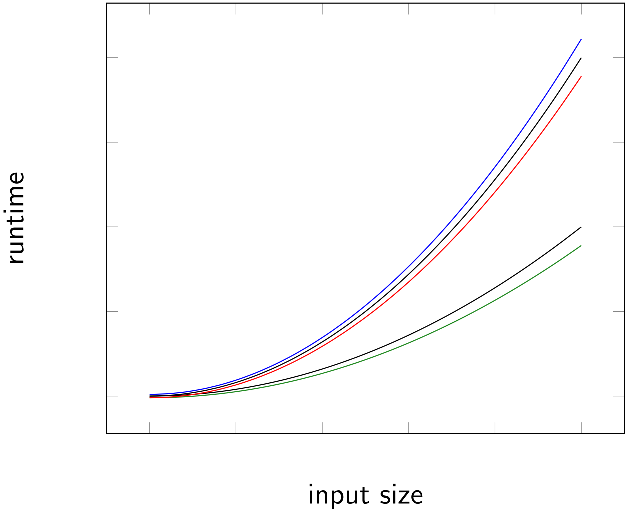
Upper, Lower, Worst Case, Best Case, ...



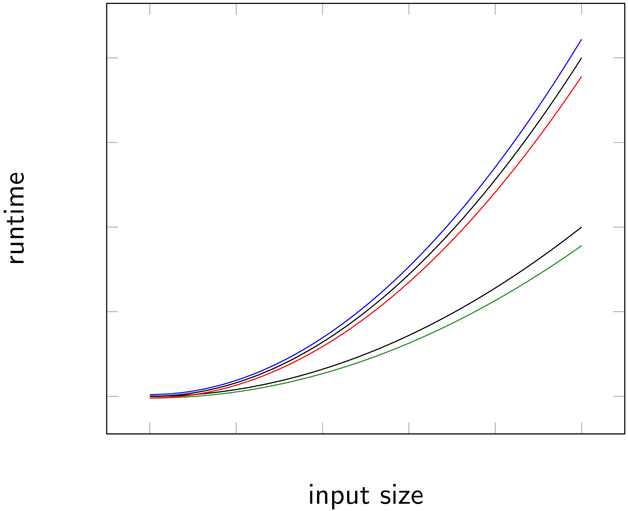
Upper, Lower, Worst Case, Best Case, ...

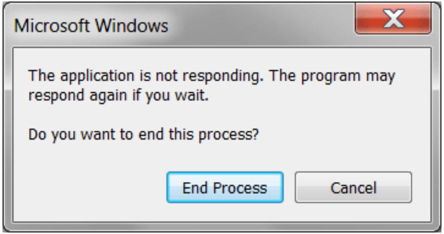


Upper, Lower, Worst Case, Best Case, ...



Upper, Lower, Worst Case, Best Case, ...





Outline

1. Introduction
2. **Lower Bounds for ITSs**
3. Lower Bounds for TRSs
4. Constant Upper Bounds for TRSs

Evaluating ITSs

Example (ITS)

$$\begin{array}{lcll} \texttt{while}(x, y, z) & \xrightarrow{5} & \texttt{while}(x-1, y, z+y) & [x > 0] \\ \texttt{while}(x, y, z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

Evaluating ITSs

Example (ITS)

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

Example (Evaluation)

$\texttt{while}(2, 2, 0)$

Evaluating ITSs

Example (ITS)

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

Example (Evaluation)

$\texttt{while}(2, 2, 0)$	$[x/2, y/2, z/0]$

Evaluating ITSs

Example (ITS)

$$\begin{array}{llll} \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\texttt{while}(2,2,0) \quad \Big| \quad [x/2, y/2, z/0] \models x > 0$$

Evaluating ITSs

Example (ITS)

$$\begin{array}{llll} \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\begin{array}{rl|l} & \texttt{while}(2,2,0) & [x/2,y/2,z/0] \models x>0 \\ \xrightarrow{5} & \texttt{while}(1,2,2) & \end{array}$$

Evaluating ITSs

Example (ITS)

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

Example (Evaluation)

	$\texttt{while}(2, 2, 0)$	$[x/2, y/2, z/0] \models x > 0$
$\xrightarrow{5}$	$\texttt{while}(1, 2, 2)$	$[x/1, y/2, z/2]$

Evaluating ITSs

Example (ITS)

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x-1, y, z+y) \quad [x > 0]$$
$$\texttt{while}(x, y, z) \xrightarrow{1} \texttt{return}(z) \quad [x = 0]$$

Example (Evaluation)

	$\texttt{while}(2, 2, 0)$	$[x/2, y/2, z/0] \models x > 0$
$\xrightarrow{5}$	$\texttt{while}(1, 2, 2)$	$[x/1, y/2, z/2] \models x > 0$

RWTH AACHEN UNIVERSITY

Evaluating ITSs

Example (ITS)

$$\begin{array}{lcll} \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\begin{array}{ll|l} & \texttt{while}(2,2,0) & [x/2,y/2,z/0] \models x>0 \\ \xrightarrow{5} & \texttt{while}(1,2,2) & [x/1,y/2,z/2] \models x>0 \\ \xrightarrow{5} & \texttt{while}(0,2,4) & \end{array}$$

RWTH AACHEN UNIVERSITY

Evaluating ITSs

Example (ITS)

$$\begin{array}{lcll} \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\begin{array}{ll|l} & \texttt{while}(2,2,0) & [x/2,y/2,z/0] \models x>0 \\ \xrightarrow{5} & \texttt{while}(1,2,2) & [x/1,y/2,z/2] \models x>0 \\ \xrightarrow{5} & \texttt{while}(0,2,4) & [x/0,y/2,z/4] \end{array}$$

RWTH AACHEN UNIVERSITY

Evaluating ITSs

Example (ITS)

$$
\begin{array}{llll}
\texttt{while}(x, y, z) & \xrightarrow{5} & \texttt{while}(x-1, y, z+y) & [x > 0] \\
\texttt{while}(x, y, z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}
$$

Example (Evaluation)

$$
\begin{array}{ll|l}
 & \texttt{while}(2, 2, 0) & [x/2, y/2, z/0] \models x > 0 \\
\xrightarrow{5} & \texttt{while}(1, 2, 2) & [x/1, y/2, z/2] \models x > 0 \\
\xrightarrow{5} & \texttt{while}(0, 2, 4) & [x/0, y/2, z/4] \not\models x > 0
\end{array}
$$

RWTH AACHEN UNIVERSITY

Evaluating ITSs

Example (ITS)

$$\begin{array}{llll} \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\begin{array}{ll|l} & \texttt{while}(2,2,0) & [x/2,y/2,z/0] \models x>0 \\ \xrightarrow{5} & \texttt{while}(1,2,2) & [x/1,y/2,z/2] \models x>0 \\ \xrightarrow{5} & \texttt{while}(0,2,4) & [x/0,y/2,z/4] \models x=0 \end{array}$$

RWTH AACHEN UNIVERSITY

Evaluating ITSs

Example (ITS)

$$\begin{array}{lll} \texttt{while}(x,y,z) & \xrightarrow{5} \texttt{while}(x-1,y,z+y) & [x>0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} \texttt{return}(z) & [x=0] \end{array}$$

Example (Evaluation)

$$\begin{array}{ll|l} & \texttt{while}(2,2,0) & [x/2,y/2,z/0] \models x>0 \\ \xrightarrow{5} & \texttt{while}(1,2,2) & [x/1,y/2,z/2] \models x>0 \\ \xrightarrow{5} & \texttt{while}(0,2,4) & [x/0,y/2,z/4] \models x=0 \\ \xrightarrow{1} & \texttt{return}(4) & \end{array}$$

Loop Acceleration

Example (Loop Acceleration)

$$\texttt{while}(x, y, z) \xrightarrow{5} \texttt{while}(x - 1, y, z + y) \qquad [x > 0]$$

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{llll} & \mathtt{while}(x,y,z) & \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \mathtt{while}(x,y,z) & \xrightarrow{5\cdot n} & \mathtt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{llll} & \mathtt{while}(x,y,z) \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \mathtt{while}(x,y,z) \xrightarrow{5\cdot n} & \mathtt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{lllll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{rlll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{llll} & \mathtt{while}(x,y,z) \xrightarrow{5} & \mathtt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \mathtt{while}(x,y,z) \xrightarrow{5\cdot n} & \mathtt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations

Example (Recurrence Equations)

$$\begin{array}{lll} x^{(n)} & = & x^{(n-1)} - 1 \\ y^{(n)} & = & y^{(n-1)} \\ z^{(n)} & = & z^{(n-1)} + y^{(n-1)} \end{array}$$

RWTH AACHEN UNIVERSITY

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{llll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations

Example (Recurrence Equations)

$$\begin{array}{lcl} x^{(n)} & = & x^{(n-1)} - 1 \\ y^{(n)} & = & y^{(n-1)} \\ z^{(n)} & = & z^{(n-1)} + y^{(n-1)} \end{array}$$

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{lllll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations
- x: maximal number of loop iterations

Example (Recurrence Equations)

$$\begin{array}{lll} x^{(n)} & = & x^{(n-1)} - 1 \\ y^{(n)} & = & y^{(n-1)} \\ z^{(n)} & = & z^{(n-1)} + y^{(n-1)} \end{array}$$

RWTH AACHEN UNIVERSITY

Loop Acceleration

Example (Loop Acceleration)

$$
\begin{array}{llll}
 & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\
\curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x]
\end{array}
$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations
- x: maximal number of loop iterations
  - $\frac{x}{2}, x-1, \ldots$ would be fine, too

Example (Recurrence Equations)

$$
\begin{aligned}
x^{(n)} &= x^{(n-1)} - 1 \\
y^{(n)} &= y^{(n-1)} \\
z^{(n)} &= z^{(n-1)} + y^{(n-1)}
\end{aligned}
$$

RWTH AACHEN UNIVERSITY

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{lllll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land {\color{red}0<n\leq x}] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations
- x: maximal number of loop iterations
  - $\frac{x}{2}, x-1, \ldots$ would be fine, too
  - search for *metering function*

Example (Recurrence Equations)

$$\begin{aligned} x^{(n)} &= x^{(n-1)} - 1 \\ y^{(n)} &= y^{(n-1)} \\ z^{(n)} &= z^{(n-1)} + y^{(n-1)} \end{aligned}$$

RWTHAACHEN UNIVERSITY

Loop Acceleration

Example (Loop Acceleration)

$$\begin{array}{rlcll} & \texttt{while}(x,y,z) & \xrightarrow{5} & \texttt{while}(x-1,y,z+y) & [x>0] \\ \curvearrowright & \texttt{while}(x,y,z) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,z+y\cdot n) & [x>0 \land 0<n\leq x] \end{array}$$

- $5 \cdot n$: cost of n iterations
- $x - n$: value of x after n iterations
- $z + y \cdot n$: value of z after n iterations

- x: maximal number of loop iterations
  - $\frac{x}{2}, x-1, \ldots$ would be fine, too
  - search for *metering function*

Example (Recurrence Equations)

$$\begin{aligned} x^{(n)} &= x^{(n-1)} - 1 \\ y^{(n)} &= y^{(n-1)} \\ z^{(n)} &= z^{(n-1)} + y^{(n-1)} \end{aligned}$$

RWTH AACHEN UNIVERSITY

Chaining

Example

$$
\begin{aligned}
\mathtt{init}(x, y) &\xrightarrow{1} \mathtt{while}(x, y, 0) && [x \geq 0 \land y \geq 0] \\
\mathtt{while}(x, y, z) &\xrightarrow{5 \cdot n} \mathtt{while}(x - n, y, z + y \cdot n) && [x > 0 \land 0 < n \leq x] \\
\mathtt{while}(x, y, z) &\xrightarrow{1} \mathtt{return}(z) && [x = 0]
\end{aligned}
$$

Chaining

Example

$$
\begin{array}{llll}
\texttt{init}(x, y) & \xrightarrow{1} & \texttt{while}(x, y, 0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x, y, z) & \xrightarrow{5 \cdot n} & \texttt{while}(x - n, y, z + y \cdot n) & [x > 0 \land 0 < n \leq x] \\
\texttt{while}(x, y, z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}
$$

Chaining

Example

$$\begin{array}{llll}
\texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x,y,0) & \xrightarrow{5\cdot n} & \texttt{while}(x-n,y,0+y\cdot n) & [x > 0 \land 0 < n \leq x] \\
\texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}$$

Example

$$\begin{array}{llll}
\texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x,y,0) & \xrightarrow{5 \cdot n} & \texttt{while}(x-n,y,y \cdot n) & [x > 0 \land 0 < n \leq x] \\
\texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}$$

RWTH AACHEN UNIVERSITY

Chaining

Example

$$\begin{array}{llll}
\texttt{init}(x, y) & \xrightarrow{1} & \texttt{while}(x, y, 0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x, y, 0) & \xrightarrow{5 \cdot n} & \texttt{while}(x - n, y, y \cdot n) & [x > 0 \land 0 < n \leq x] \\
\texttt{while}(x, y, z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}$$

RWTH AACHEN UNIVERSITY

Chaining

Example

$$
\begin{array}{llll}
\texttt{init}(x, y) & \xrightarrow{1} & \texttt{while}(x, y, 0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x, y, 0) & \xrightarrow{5 \cdot x} & \texttt{while}(x - x, y, y \cdot x) & [x > 0 \land 0 < x \leq x] \\
\texttt{while}(x, y, z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0]
\end{array}
$$

Chaining

Example

$$
\begin{aligned}
\texttt{init}(x, y) &\xrightarrow{1} \texttt{while}(x, y, 0) && [x \geq 0 \land y \geq 0] \\
\texttt{while}(x, y, 0) &\xrightarrow{5 \cdot x} \texttt{while}(0, y, y \cdot x) && [x > 0] \\
\texttt{while}(x, y, z) &\xrightarrow{1} \texttt{return}(z) && [x = 0]
\end{aligned}
$$

Chaining

Example

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,0) & \xrightarrow{5 \cdot x} & \texttt{while}(0,y,y \cdot x) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \\ & & & \\ \curvearrowright \quad \texttt{init}(x,y) & \xrightarrow{5 \cdot x + 2} & \texttt{return}(y \cdot x) & [x > 0 \land y \geq 0] \end{array}$$

Chaining

Example

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,0) & \xrightarrow{5 \cdot x} & \texttt{while}(0,y,y \cdot x) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

$$\curvearrowright \quad \texttt{init}(x,y) \xrightarrow{5 \cdot x+2} \texttt{return}(y \cdot x) \quad [x > 0 \land y \geq 0]$$

Chaining

Example

$$\begin{array}{llll} & \mathtt{init}(x,y) & \xrightarrow{1} & \mathtt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ & \mathtt{while}(x,y,0) & \xrightarrow{5\cdot x} & \mathtt{while}(0,y,y\cdot x) & [x > 0] \\ & \mathtt{while}(x,y,z) & \xrightarrow{1} & \mathtt{return}(z) & [x = 0] \\ \curvearrowright & \mathtt{init}(x,y) & \xrightarrow{5\cdot x+2} & \mathtt{return}(y\cdot x) & [x > 0 \land y \geq 0] \end{array}$$

$\curvearrowright$ whole program compressed into a single rule

Chaining

Example

$$\begin{array}{llll} \texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\ \texttt{while}(x,y,0) & \xrightarrow{5 \cdot x} & \texttt{while}(0,y,y \cdot x) & [x > 0] \\ \texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \end{array}$$

$$\curvearrowright \quad \texttt{init}(x,y) \xrightarrow{5 \cdot x + 2} \texttt{return}(y \cdot x) \quad [x > 0 \land y \geq 0]$$

- $\curvearrowright$ whole program compressed into a single rule
- $\curvearrowright$ lower bound $5 \cdot x + 2$ for all $x > 0, y \geq 0$

Chaining

Example

$$
\begin{array}{llll}
\texttt{init}(x,y) & \xrightarrow{1} & \texttt{while}(x,y,0) & [x \geq 0 \land y \geq 0] \\
\texttt{while}(x,y,0) & \xrightarrow{5 \cdot x} & \texttt{while}(0,y,y \cdot x) & [x > 0] \\
\texttt{while}(x,y,z) & \xrightarrow{1} & \texttt{return}(z) & [x = 0] \\
 & & & \\
\curvearrowright \ \texttt{init}(x,y) & \xrightarrow{5 \cdot x + 2} & \texttt{return}(y \cdot x) & [x > 0 \land y \geq 0]
\end{array}
$$

$\curvearrowright$ whole program compressed into a single rule

$\curvearrowright$ lower bound $5 \cdot x + 2$ for all $x > 0, y \geq 0$

Works for arbitrary programs!

RWTHAACHEN UNIVERSITY

first technique to infer worst case lower bounds for ITSs

Lower Bounds for ITSs – Wrap-Up

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*
- many other features, e.g.:

Lower Bounds for ITSs – Wrap-Up

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*
- many other features, e.g.:
  - non-determinism

Lower Bounds for ITSs – Wrap-Up

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*
- many other features, e.g.:
  - non-determinism
  - asymptotic bounds

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*
- many other features, e.g.:
  - non-determinism
  - asymptotic bounds
  - recursive `void` functions
  - SMT encodings
  - ...

RWTH AACHEN UNIVERSITY

Lower Bounds for ITSs – Wrap-Up

first technique to infer worst case lower bounds for ITSs

- key idea: *loop acceleration*
- many other features, e.g.:
  - non-determinism
  - asymptotic bounds
  - recursive `void` functions
  - SMT encodings
  - ...

	LoAT							
Best Upper Bound: rc(n)	$\Omega(1)$	$\Omega(n)$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^4)$	EXP	∞
$\mathcal{O}(1)$	(132)	–	–	–	–	–	–
$\mathcal{O}(n)$	37	126	–	–	–	–	–
$\mathcal{O}(n^2)$	8	14	35	–	–	–	–
$\mathcal{O}(n^3)$	2	–	2	1	–	–	–
$\mathcal{O}(n^4)$	1	–	–	–	2	–	–
EXP	–	–	–	–	–	5	–
∞	53	31	1	–	–	–	176

RWTH AACHEN UNIVERSITY

Outline

Evaluating TRSs

Example (TRS)

$$\mathtt{inf}(y) \longrightarrow \mathbf{cons}(y, \mathtt{inf}(\mathrm{s}(y)))$$

Evaluating TRSs

Example (TRS)

$$\texttt{inf}(y) \to \mathbf{cons}(y, \texttt{inf}(\mathbf{s}(y)))$$

Example (Evaluation)

$$\texttt{inf}(\mathbf{0})$$

Evaluating TRSs

Example (TRS)

$$\mathtt{inf}(y) \longrightarrow \mathbf{cons}(y, \mathtt{inf}(\mathrm{s}(y)))$$

Example (Evaluation)

$\underline{\mathtt{inf}(\mathbf{0})}$ | $[y/\mathbf{0}]$

Evaluating TRSs

Example (TRS)

$$\mathtt{inf}(y) \to \mathbf{cons}(y, \mathtt{inf}(\mathrm{s}(y)))$$

Example (Evaluation)

$$\underline{\mathtt{inf}(\mathbf{0})} \qquad [y/\mathbf{0}]$$

$$\to \underline{\mathbf{cons}(\mathbf{0}, \mathtt{inf}(\mathrm{s}(\mathbf{0})))}$$

Example (TRS)

$$\mathtt{inf}(y) \to \mathbf{cons}(y, \mathtt{inf}(\mathbf{s}(y)))$$

Example (Evaluation)

	$\mathtt{inf}(\mathbf{0})$	$[y/\mathbf{0}]$
$\to$	$\mathbf{cons}(\mathbf{0}, \underline{\mathtt{inf}(\mathbf{s}(\mathbf{0}))})$	$[y/\mathbf{s}(\mathbf{0})]$

Example (TRS)

$$\texttt{inf}(y) \to \mathbf{cons}(y, \texttt{inf}(\mathsf{s}(y)))$$

Example (Evaluation)

$$
\begin{array}{ll|l}
 & \texttt{inf}(\mathbf{0}) & [y/\mathbf{0}] \\
\to & \mathbf{cons}(\mathbf{0}, \underline{\texttt{inf}(\mathsf{s}(\mathbf{0}))}) & [y/\mathsf{s}(\mathbf{0})] \\
\to & \mathbf{cons}(\mathbf{0}, \underline{\mathbf{cons}(\mathsf{s}(\mathbf{0}), \texttt{inf}(\mathsf{s}(\mathsf{s}(\mathbf{0}))))}) &
\end{array}
$$

RWTH AACHEN UNIVERSITY

Evaluating TRSs

Example (TRS)

$$\mathtt{inf}(y) \to \mathbf{cons}(y, \mathtt{inf}(\mathrm{s}(y)))$$

Example (Evaluation)

	$\mathtt{inf}(\mathbf{0})$	$[y/\mathbf{0}]$
$\to$	$\mathbf{cons}(\mathbf{0}, \mathtt{inf}(\mathrm{s}(\mathbf{0})))$	$[y/\mathrm{s}(\mathbf{0})]$
$\to$	$\mathbf{cons}(\mathbf{0}, \mathbf{cons}(\mathrm{s}(\mathbf{0}), \underline{\mathtt{inf}(\mathrm{s}(\mathrm{s}(\mathbf{0})))}))$	$[y/\mathrm{s}(\mathrm{s}(\mathbf{0}))]$

Evaluating TRSs

Example (TRS)

$$\mathtt{inf}(y) \to \mathbf{cons}(y, \mathtt{inf}(\mathrm{s}(y)))$$

Example (Evaluation)

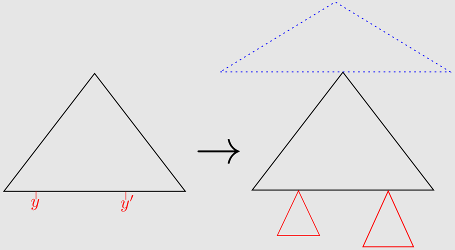
$$\begin{array}{ll|l} & \mathtt{inf}(\mathbf{0}) & [y/\mathbf{0}] \\ \to & \mathbf{cons}(\mathbf{0}, \mathtt{inf}(\mathrm{s}(\mathbf{0}))) & [y/\mathrm{s}(\mathbf{0})] \\ \to & \mathbf{cons}(\mathbf{0}, \mathbf{cons}(\mathrm{s}(\mathbf{0}), \underline{\mathtt{inf}(\mathrm{s}(\mathrm{s}(\mathbf{0})))})) & [y/\mathrm{s}(\mathrm{s}(\mathbf{0}))] \\ \to & \ldots & \end{array}$$

Evaluating TRSs

Example (TRS)

$\mathtt{inf}(y) \longrightarrow \mathbf{cons}(y, \mathtt{inf}(\mathbf{s}(y)))$

Loop

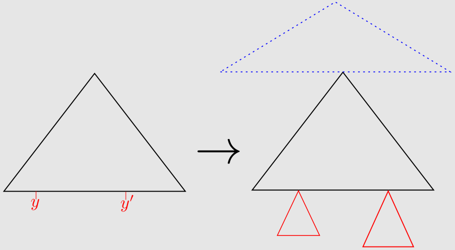


Evaluating TRSs

Example (TRS)

$\texttt{inf}(y) \to \mathbf{cons}(y, \texttt{inf}(\mathbf{s}(y)))$

Loop $\Longrightarrow$ Non-Termination

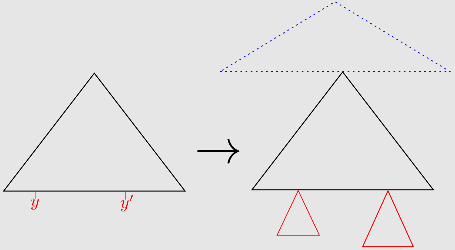


Evaluating TRSs

Example (TRS)

$\mathtt{inf}(y) \rightarrow \mathbf{cons}(y, \mathtt{inf}(\mathbf{s}(\mathbf{0})))$

Loop $\Longrightarrow$ Non-Termination



Decreasing Loops

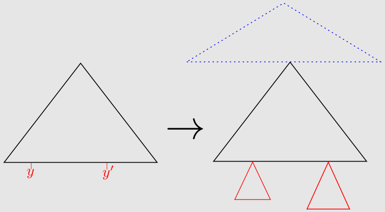
Example

$$\texttt{length}(\mathbf{cons}(x, x')) \to \mathbf{s}(\texttt{length}(x'))$$

Example

$\mathtt{length}(\mathbf{cons}(x, x')) \to \mathtt{s}(\mathtt{length}(x'))$

Loop

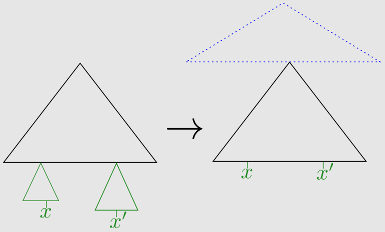


Decreasing Loops

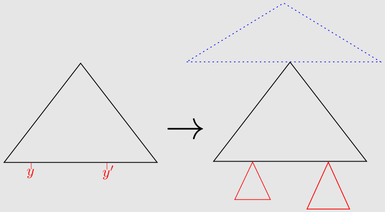
Example

$\mathtt{length}(\mathbf{cons}(x, x')) \to \mathbf{s}(\mathtt{length}(x'))$

Decreasing Loop



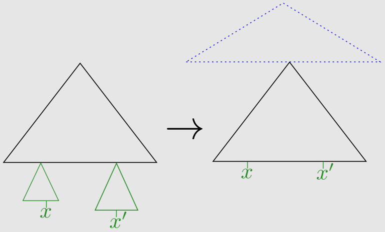
Loop



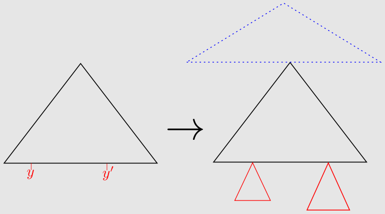
Example

$$\texttt{length}(\mathbf{cons}(x, x')) \to \mathbf{s}(\texttt{length}(x'))$$

Decreasing Loop $\Longrightarrow$ Linear Lower Bound



Loop

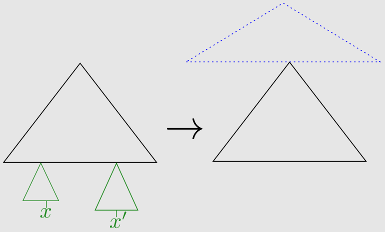


Decreasing Loops

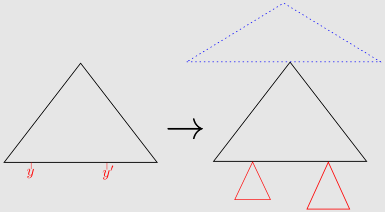
Example

$\mathtt{length}(\mathbf{cons}(x, x')) \to \mathbf{s}(\mathtt{length}(\mathbf{nil}))$

Decreasing Loop $\not\Rightarrow$ Linear Lower Bound



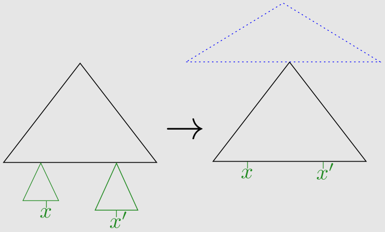
Loop



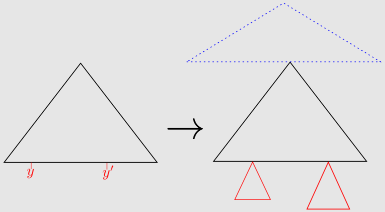
Example

$\mathtt{length}(\mathbf{cons}(x, x')) \to \mathbf{s}(\mathtt{length}(x'))$

Decreasing Loop $\Longrightarrow$ Linear Lower Bound



Loop



Decreasing Loops

Example

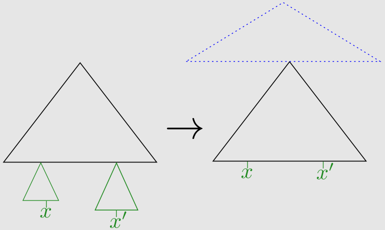
$$\begin{array}{rcl} \texttt{length}(\mathbf{cons}(x, x')) & \to & \texttt{s}(\texttt{length}(x')) \\ \curvearrowright \quad \texttt{length}(\mathbf{cons}(x, x'), y) & \to & \texttt{length}(x', \texttt{s}(y)) \end{array}$$

Example

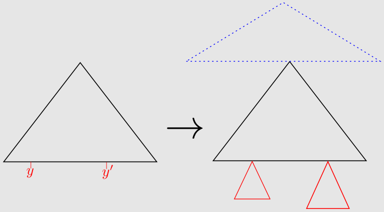
$$\mathtt{length}(\mathbf{cons}(x, x')) \to \mathtt{s}(\mathtt{length}(x'))$$

$$\curvearrowright \; \mathtt{length}(\mathbf{cons}(x, x'), y) \to \mathtt{length}(x', \mathbf{s}(y))$$

Decreasing Loops



Loops

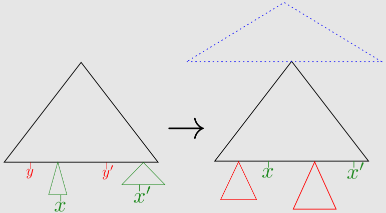


Decreasing Loops

Example

$$\begin{array}{rlcl} & \mathtt{length}(\mathbf{cons}(x, x')) & \to & \mathtt{s}(\mathtt{length}(x')) \\ \curvearrowright & \mathtt{length}(\mathbf{cons}(x, x'), y) & \to & \mathtt{length}(x', \mathbf{s}(y)) \end{array}$$

Decreasing Loops Revised



Decreasing Loops – Results

Theorem

If a TRS has a decreasing loop, then its complexity is at least linear.

Decreasing Loops – Results

Theorem

If a TRS has a decreasing loop, then its complexity is at least linear.

Theorem

If a TRS has several compatible decreasing loops, then its complexity is at least exponential.

Decreasing Loops – Results

Theorem

If a TRS has a decreasing loop, then its complexity is at least linear.

Theorem

If a TRS has several compatible decreasing loops, then its complexity is at least exponential.

Example (Compatible Decreasing Loops)

$$\mathtt{fib}(\mathtt{s}(\mathtt{s}(x))) \to \mathtt{plus}(\mathtt{fib}(\mathtt{s}(x)), \mathtt{fib}(x))$$

Decreasing Loops – Results

Theorem

If a TRS has a decreasing loop, then its complexity is at least linear.

Theorem

If a TRS has several compatible decreasing loops, then its complexity is at least exponential.

Theorem

Linear lower bounds are not semi-decidable.

Example (Compatible Decreasing Loops)

$$\mathtt{fib}(\mathtt{s}(\mathtt{s}(x))) \longrightarrow \mathtt{plus}(\mathtt{fib}(\mathtt{s}(x)), \mathtt{fib}(x))$$

Decreasing Loops – Results

Theorem

If a TRS has a decreasing loop, then its complexity is at least linear.

Theorem

If a TRS has several compatible decreasing loops, then its complexity is at least exponential.

Theorem

Linear lower bounds are not semi-decidable.

Lemma

Decreasing loops are incomplete.

Example (Compatible Decreasing Loops)

$\mathtt{fib}(\mathtt{s}(\mathtt{s}(x))) \to \mathtt{plus}(\mathtt{fib}(\mathtt{s}(x)), \mathtt{fib}(x))$

Lower Bounds for TRSs – Wrap-Up

first technique to infer worst case lower bounds for TRSs

first technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*

Lower Bounds for TRSs – Wrap-Up

first technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds

first technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds
  - quadratic, cubic, ...?

Lower Bounds for TRSs – Wrap-Up

first technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds
  - quadratic, cubic, ...?

  $\curvearrowright$ "induction technique"

Lower Bounds for TRSs – Wrap-Up

~~first~~ second technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds
  - quadratic, cubic, ...?
  - ↷ "induction technique"

Lower Bounds for TRSs – Wrap-Up

~~first~~ second technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds
  - quadratic, cubic, ...?
  - $\curvearrowright$ "induction technique"
- incomplete

Lower Bounds for TRSs – Wrap-Up

~~first~~ second technique to infer worst case lower bounds for TRSs

- key idea: generalize *loops*
- linear, exponential, infinite bounds
  - quadratic, cubic, ...?
  - ↷ "induction technique"
- incomplete

AProVE			
$\Omega(1)$	$\Omega(n)$	EXP	∞
66	600	143	90

Outline

1. Introduction
2. Lower Bounds for ITSs
3. Lower Bounds for TRSs
4. **Constant Upper Bounds for TRSs**

Narrowing

Example (from TPDB)

$$\begin{aligned} \mathbf{f}(\mathbf{a}) &\to \mathbf{g}(\mathbf{h}(\mathbf{a})) \\ \mathbf{h}(\mathbf{g}(x)) &\to \mathbf{g}(\mathbf{h}(\mathbf{f}(x))) \end{aligned}$$

Narrowing

Example (from TPDB)

$$\begin{aligned} \mathtt{f}(\mathtt{a}) &\to \mathtt{g}(\mathtt{h}(\mathtt{a})) \\ \mathtt{h}(\mathtt{g}(x)) &\to \mathtt{g}(\mathtt{h}(\mathtt{f}(x))) \end{aligned}$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Narrowing

Example (from TPDB)

$$\begin{aligned} \mathtt{f}(\mathbf{a}) &\to \mathtt{g}(\mathtt{h}(\mathbf{a})) \\ \mathtt{h}(\mathbf{g}(x)) &\to \mathbf{g}(\mathtt{h}(\mathtt{f}(x))) \end{aligned}$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

$\mathtt{h}(y)$

Narrowing

Example (from TPDB)

$$\begin{array}{lcl} \mathtt{f}(\mathbf{a}) & \rightarrow & \mathtt{g}(\mathtt{h}(\mathbf{a})) \\ \mathtt{h}(\mathbf{g}(x)) & \rightarrow & \mathtt{g}(\mathtt{h}(\mathtt{f}(x))) \end{array}$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

$\underline{\mathtt{h}(y)}$	$[y/\mathbf{g}(x)]$

Narrowing

Example (from TPDB)

$$\begin{aligned} \mathtt{f}(\mathtt{a}) &\to \mathtt{g}(\mathtt{h}(\mathtt{a})) \\ \mathtt{h}(\mathtt{g}(x)) &\to \mathtt{g}(\mathtt{h}(\mathtt{f}(x))) \end{aligned}$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

$\underline{\mathtt{h}(y)}$	$[y/\mathtt{g}(x)]$
$\rightsquigarrow \underline{\mathtt{g}(\mathtt{h}(\mathtt{f}(x)))}$	

Narrowing

Example (from TPDB)

$$\begin{aligned} \mathtt{f}(\mathbf{a}) &\to \mathtt{g}(\mathtt{h}(\mathbf{a})) \\ \mathtt{h}(\mathbf{g}(x)) &\to \mathtt{g}(\mathtt{h}(\mathtt{f}(x))) \end{aligned}$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

	$\mathtt{h}(y)$	$[y/\mathbf{g}(x)]$
$\rightsquigarrow$	$\mathbf{g}(\mathtt{h}(\underline{\mathtt{f}(x)}))$	$[x/\mathbf{a}]$

RWTH AACHEN UNIVERSITY

Narrowing

Example (from TPDB)

$$\mathtt{f}(\mathbf{a}) \rightarrow \mathtt{g}(\mathtt{h}(\mathbf{a}))$$
$$\mathtt{h}(\mathbf{g}(x)) \rightarrow \mathbf{g}(\mathtt{h}(\mathtt{f}(x)))$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

$\mathtt{h}(y)$	$[y/\mathbf{g}(x)]$
$\rightsquigarrow \mathbf{g}(\mathtt{h}(\underline{\mathtt{f}(x)}))$	$[x/\mathbf{a}]$
$\rightsquigarrow \mathbf{g}(\mathtt{h}(\underline{\mathbf{g}(\mathtt{h}(\mathbf{a}))}))$	

Narrowing

Example (from TPDB)

$$\mathtt{f}(\mathbf{a}) \to \mathbf{g}(\mathtt{h}(\mathbf{a}))$$
$$\mathtt{h}(\mathbf{g}(x)) \to \mathbf{g}(\mathtt{h}(\mathtt{f}(x)))$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

$\mathtt{h}(y)$	$[y/\mathbf{g}(x)]$
$\rightsquigarrow \mathbf{g}(\mathtt{h}(\mathtt{f}(x)))$	$[x/\mathbf{a}]$
$\rightsquigarrow \mathbf{g}(\underline{\mathtt{h}(\mathbf{g}(\mathtt{h}(\mathbf{a})))})$	$[x/\mathtt{h}(\mathbf{a})]$

Narrowing

Example (from TPDB)

$$\mathtt{f}(\mathbf{a}) \to \mathtt{g}(\mathtt{h}(\mathbf{a}))$$
$$\mathtt{h}(\mathbf{g}(x)) \to \mathbf{g}(\mathtt{h}(\mathtt{f}(x)))$$

Definition (Narrowing)

Just like rewriting, but with *unification* instead of matching.

Example (Evaluation)

	$\mathtt{h}(y)$	$[y/\mathbf{g}(x)]$
$\rightsquigarrow$	$\mathbf{g}(\mathtt{h}(\mathtt{f}(x)))$	$[x/\mathbf{a}]$
$\rightsquigarrow$	$\mathbf{g}(\underline{\mathtt{h}(\mathbf{g}(\mathtt{h}(\mathbf{a})))})$	$[x/\mathtt{h}(\mathbf{a})]$
$\rightsquigarrow$	$\mathbf{g}(\underline{\mathbf{g}(\mathtt{h}(\mathtt{f}(\mathtt{h}(\mathbf{a}))))})$	

Constant Upper Bounds

Theorem

The complexity of a TRS is constant iff narrowing terminates for all terms $\mathtt{f}(x_1, \ldots, x_n)$.

Constant Upper Bounds

Theorem

The complexity of a TRS is constant iff narrowing terminates for all terms $\mathtt{f}(x_1, \ldots, x_n)$.

Lemma

Constant complexity is semi-decidable.

Constant Upper Bounds – Wrap-Up

semi-decision procedure for constant bounds

Constant Upper Bounds – Wrap-Up

semi-decision procedure for constant bounds

key idea: use *narrowing*

Constant Upper Bounds – Wrap-Up

semi-decision procedure for constant bounds

key idea: use *narrowing*

$\curvearrowright$ AProVE can (dis-)prove constant complexity in almost all cases

semi-decision procedure for constant bounds

key idea: use *narrowing*

$\curvearrowright$ AProVE can (dis-)prove constant complexity in almost all cases

Evaluation (TPDB – 899 examples)

- succeeds for all 57 TRSs with constant complexity

Constant Upper Bounds – Wrap-Up

semi-decision procedure for constant bounds

key idea: use *narrowing*

$\curvearrowright$ AProVE can (dis-)prove constant complexity in almost all cases

Evaluation (TPDB – 899 examples)

- succeeds for all 57 TRSs with constant complexity
- solves 6 examples that couldn't be solved before

RWTH AACHEN UNIVERSITY

What's next?

- integers *and* data structures

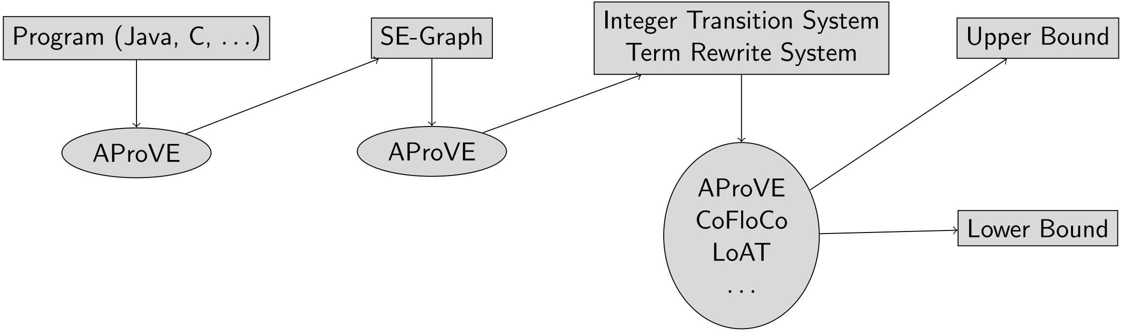
What's next?

- integers *and* data structures
- floats, arrays, . . .

What's next?

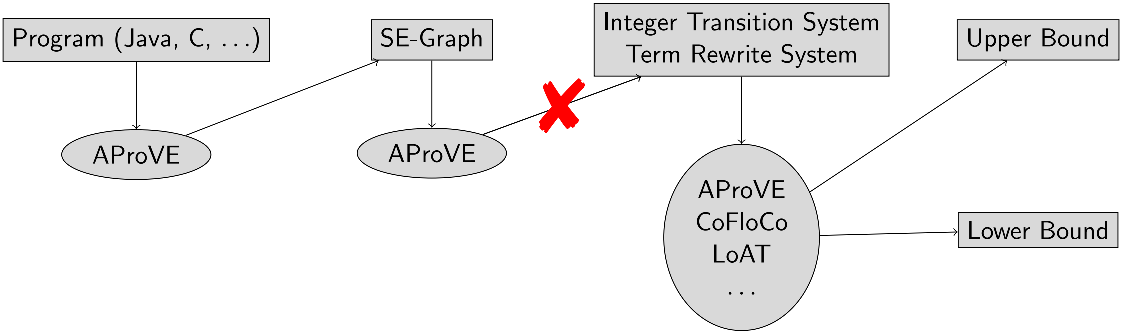
- integers *and* data structures
- floats, arrays, ...
- *underapproximating* program transformations

What's next?



- integers *and* data structures
- floats, arrays, ...
- *underapproximating* program transformations

What's next?



- integers *and* data structures
- floats, arrays, ...
- *underapproximating* program transformations

What else?

'15	Induction Technique for TRSs	RTA
'16	Lower Bounds for ITSs	IJCAR
'17	Decreasing Loops for TRSs	JAR
'17		LPAR
'17		FroCoS
'17	Upper Bounds for Java	iFM
'18	Constant Upper Bounds for TRSs	IPL

What else?

'15	Induction Technique for TRSs	RTA
'16	Lower Bounds for ITSs	IJCAR
'17	Decreasing Loops for TRSs	JAR
'17		LPAR
'17		FroCoS
'17	Upper Bounds for Java	iFM
'18	Constant Upper Bounds for TRSs	IPL

What else?

'15	Induction Technique for TRSs	RTA
'16	Lower Bounds for ITSs	IJCAR
'17	Decreasing Loops for TRSs	JAR
'17		LPAR
'17		FroCoS
'17	Upper Bounds for Java	iFM
'18	Constant Upper Bounds for TRSs	IPL

What else?

'15	Induction Technique for TRSs	RTA
'16	Lower Bounds for ITSs	IJCAR
'17	Decreasing Loops for TRSs	JAR
'17		LPAR
'17		FroCoS
'17	Upper Bounds for Java	iFM
'18	Constant Upper Bounds for TRSs	IPL

Termination and Complexity Analysis for C — IJCAR '14, SEFM '16, JAR '17, JLAMP '18

AProVE Tool Papers — IJCAR '14, JAR '17

What else?

'15	Induction Technique for TRSs	RTA
'16	Lower Bounds for ITSs	IJCAR
'17	Decreasing Loops for TRSs	JAR
'17		LPAR
'17		FroCoS
'17	Upper Bounds for Java	iFM
'18	Constant Upper Bounds for TRSs	IPL

Thank you!

Termination and Complexity Analysis for C — IJCAR '14, SEFM '16, JAR '17, JLAMP '18

AProVE Tool Papers — IJCAR '14, JAR '17

RWTH AACHEN UNIVERSITY

AProVE at TermComp

$$\frac{\text{points of AProVE}}{\text{points of best competitor}}$$

ITS	2016	2017	2018
w/ lower bounds	2.09	N/A	2.60
w/o lower bounds	0.66	N/A	0.92

TRS	2014	2015	2016	2017	2018
innermost	1.05	2.73	1.18	N/A	1.79
full	–	2.94	1.23	N/A	1.70

Recurrence Solving

Example (Recurrence Solving)

$$\begin{aligned} y^{(v+1)} &= 1 - y^{(v)} \\ x^{(v+1)} &= x^{(v)} + 1 - 3y^{(v)} \end{aligned}$$

$$\begin{aligned} y^{(v)} &= \frac{1}{2} - \frac{1}{2}(-1)^v + (-1)^v y \\ x^{(v)} &= \frac{3}{4} - \frac{3}{4}(-1)^v - \frac{3}{2}y - \frac{1}{2}v + \frac{3}{2}(-1)^v y + x \end{aligned}$$

RWTH AACHEN UNIVERSITY