Data-Driven Runtime Complexity Analysis

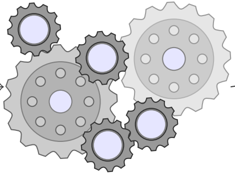
15th International Symposium on Frontiers of Combining Systems (FroCoS 2025)

Samuel Frontull Manuel Meitinger Georg Moser

https://tcs-informatik.uibk.ac.at

Resource Analysis

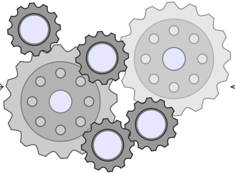
- Resource Analysis is a fundamental problem
- it is used to understand the efficiency of algorithms
- it is used to identify bugs

Source Code →  → Complexity

Resource Analysis

- Resource Analysis is a fundamental, undecidable problem
- it is used to understand the efficiency of algorithms
- it is used to identify bugs
- (Fully automated) Static analysis is in general not complete:

Source Code →



→ Complexity

or

→ Unknown

What To Do When Automation Fails?

Statistical Approach

run program on various inputs, measure the cost, infer complexity empirically

Our Work[1]

We have implemented this idea for term rewrite systems and compared our approach against known results.

[1] Samuel Frontull, Manuel Meitinger, and Georg Moser. "Data-Driven Runtime Complexity Analysis". In: *15th International Symposium on Frontiers of Combining Systems (FroCoS 2025)*. Springer Nature Switzerland, 2025, pp. 228–246.

Runtime Complexity of a Term Rewrite System (TRS)

Multiplication and Addition as TRS

$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$

$\mathcal{R} :=$

$$
\begin{aligned}
1:&\quad x \cdot \mathsf{s}(y) \to (x \cdot y) + x & 2:&\quad x \cdot 0 \to 0 \\
3:&\quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) & 4:&\quad 0 + x \to x \\
5:&\quad x + \mathsf{s}(y) \to \mathsf{s}(x + y) & 6:&\quad x + 0 \to x
\end{aligned}
$$

Multiplication and Addition as TRS

$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$

$\mathcal{R} :=$

$$\begin{array}{llcl@{\qquad}llcl}
1: & x \cdot \mathsf{s}(y) & \to & (x \cdot y) + x & 2: & x \cdot 0 & \to & 0 \\
3: & \mathsf{s}(x) + y & \to & \mathsf{s}(x + y) & 4: & 0 + x & \to & x \\
5: & x + \mathsf{s}(y) & \to & \mathsf{s}(x + y) & 6: & x + 0 & \to & x
\end{array}$$

Example

$t = \mathsf{s}(\mathsf{s}(0)) \cdot \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{aligned} 1:&\quad x \cdot \mathsf{s}(y) &\to (x \cdot y) + x &\qquad 2:&\quad x \cdot 0 &\to 0 \\ 3:&\quad \mathsf{s}(x) + y &\to \mathsf{s}(x + y) &\qquad 4:&\quad 0 + x &\to x \\ 5:&\quad x + \mathsf{s}(y) &\to \mathsf{s}(x + y) &\qquad 6:&\quad x + 0 &\to x \end{aligned}$$

Example

$$t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0)$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{llll} 1: & x \cdot \mathsf{s}(y) \to (x \cdot y) + x & 2: & x \cdot 0 \to 0 \\ 3: & \mathsf{s}(x) + y \to \mathsf{s}(x + y) & 4: & 0 + x \to x \\ 5: & x + \mathsf{s}(y) \to \mathsf{s}(x + y) & 6: & x + 0 \to x \end{array}$$

Example

$$t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) \to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0)$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{lrll lrll}
1: & x \cdot \mathsf{s}(y) & \to & (x \cdot y) + x & 2: & x \cdot 0 & \to & 0 \\
3: & \mathsf{s}(x) + y & \to & \mathsf{s}(x + y) & 4: & 0 + x & \to & x \\
5: & x + \mathsf{s}(y) & \to & \mathsf{s}(x + y) & 6: & x + 0 & \to & x
\end{array}$$

Example

$$t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) \to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{lrll lrll}
1: & x \cdot \mathsf{s}(y) & \to & (x \cdot y) + x & 2: & x \cdot 0 & \to & 0 \\
3: & \mathsf{s}(x) + y & \to & \mathsf{s}(x + y) & 4: & 0 + x & \to & x \\
5: & x + \mathsf{s}(y) & \to & \mathsf{s}(x + y) & 6: & x + 0 & \to & x
\end{array}$$

Example

$$\begin{aligned}
t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)
\end{aligned}$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{lrlclrl} 1: & x \cdot \mathsf{s}(y) & \to (x \cdot y) + x & \quad & 2: & x \cdot 0 & \to 0 \\ 3: & \mathsf{s}(x) + y & \to \mathsf{s}(x+y) & & 4: & 0 + x & \to x \\ 5: & x + \mathsf{s}(y) & \to \mathsf{s}(x+y) & & 6: & x + 0 & \to x \end{array}$$

Example

$$\begin{aligned} t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\ &\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \end{aligned}$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{lrll}
1: & x \cdot \mathsf{s}(y) & \to (x \cdot y) + x & \quad 2: & x \cdot 0 & \to 0 \\
3: & \mathsf{s}(x) + y & \to \mathsf{s}(x + y) & \quad 4: & 0 + x & \to x \\
5: & x + \mathsf{s}(y) & \to \mathsf{s}(x + y) & \quad 6: & x + 0 & \to x
\end{array}$$

Example

$$\begin{aligned}
t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (\mathsf{s}^2(0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)
\end{aligned}$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{aligned}
1:&\quad x \cdot \mathsf{s}(y) &\to (x \cdot y) + x &\qquad 2:&\quad x \cdot 0 &\to 0\\
3:&\quad \mathsf{s}(x) + y &\to \mathsf{s}(x + y) &\qquad 4:&\quad 0 + x &\to x\\
5:&\quad x + \mathsf{s}(y) &\to \mathsf{s}(x + y) &\qquad 6:&\quad x + 0 &\to x
\end{aligned}$$

Example

$$\begin{aligned}
t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)\\
&\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)\\
&\to (\mathsf{s}^2(0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to^* \mathsf{s}^6(0)
\end{aligned}$$

Multiplication and Addition as TRS

$$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$$

$$\mathcal{R} :=$$

$$\begin{array}{llll@{\qquad}llll}
1: & x \cdot \mathsf{s}(y) & \to & (x \cdot y) + x & 2: & x \cdot 0 & \to & 0 \\
3: & \mathsf{s}(x) + y & \to & \mathsf{s}(x + y) & 4: & 0 + x & \to & x \\
5: & x + \mathsf{s}(y) & \to & \mathsf{s}(x + y) & 6: & x + 0 & \to & x
\end{array}$$

Example

$$\begin{aligned}
t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (\mathsf{s}^2(0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to^* \mathsf{s}^6(0)
\end{aligned}$$

normal form

Multiplication and Addition as TRS

$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$

$\mathcal{R} :=$

1:	$x \cdot \mathsf{s}(y)$	$\to$	$(x \cdot y) + x$	2:	$x \cdot 0$	$\to$	0
3:	$\mathsf{s}(x) + y$	$\to$	$\mathsf{s}(x + y)$	4:	$0 + x$	$\to$	x
5:	$x + \mathsf{s}(y)$	$\to$	$\mathsf{s}(x + y)$	6:	$x + 0$	$\to$	x

Example

basic term

$$t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) \to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)$$

normal form

$$\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)$$

$$\to (\mathsf{s}^2(0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to^* \mathsf{s}^6(0)$$

Multiplication and Addition TRS

defined symbols

constructors

$\mathcal{F} := \{0/0, \mathsf{s}/1, +/2, \cdot/2\}$ $\quad \mathcal{F}_{\mathcal{D}} := \{\cdot, +\}$ $\quad \mathcal{F}_{\mathcal{C}} := \{0, \mathsf{s}\}$

$\mathcal{R} :=$

$$
\begin{array}{llll}
1: & x \cdot \mathsf{s}(y) \to (x \cdot y) + x & 2: & x \cdot 0 \to 0 \\
3: & \mathsf{s}(x) + y \to \mathsf{s}(x + y) & 4: & 0 + x \to x \\
5: & x + \mathsf{s}(y) \to \mathsf{s}(x + y) & 6: & x + 0 \to x
\end{array}
$$

Example

basic term

normal form

$$
\begin{aligned}
t = \mathsf{s}^2(0) \cdot \mathsf{s}^3(0) &\to (\mathsf{s}^2(0) \cdot \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((\mathsf{s}^2(0) \cdot \mathsf{s}(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (((\mathsf{s}^2(0) \cdot 0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to ((0 + \mathsf{s}^2(0)) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \\
&\to (\mathsf{s}^2(0) + \mathsf{s}^2(0)) + \mathsf{s}^2(0) \to^* \mathsf{s}^6(0)
\end{aligned}
$$

Runtime Complexity of a TRS

$$\mathsf{s}^2(0)\cdot\mathsf{s}^3(0) \to \ldots \to \mathsf{s}^6(0) \text{ in } 10 \text{ steps}$$

Runtime Complexity of a TRS

$$\mathsf{s}^2(0)\cdot\mathsf{s}^3(0) \to \ldots \to \mathsf{s}^6(0) \text{ in } 10 \text{ steps}$$

Derivation Height

Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that t is in normal form. The *derivation height* of a term s wrt. a well-founded, finitely-branching relation $\to$ is defined as $\mathsf{dh}(s, \to) := \max\{n \mid \exists t. s \to^n t\}$

$$\mathsf{dh}(\mathsf{s}^2(0)\cdot\mathsf{s}^3(0), \to) = \max\{10, \ldots\}$$

Runtime Complexity of a TRS

$$\mathsf{s}^2(0) \cdot \mathsf{s}^3(0) \to \ldots \to \mathsf{s}^6(0) \text{ in } 10 \text{ steps}$$

Derivation Height

Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that t is in normal form. The *derivation height* of a term s wrt. a well-founded, finitely-branching relation $\to$ is defined as $\mathsf{dh}(s, \to) := \max\{n \mid \exists t. s \to^n t\}$

$$\mathsf{dh}(\mathsf{s}^2(0) \cdot \mathsf{s}^3(0), \to) = \max\{10, \ldots\}$$

Runtime Complexity

$\mathsf{rc}_{\mathcal{R}}(n) := \max\{\mathsf{dh}(s, \to_{\mathcal{R}}) \mid s \text{ is basic and } |s| \leqslant n\}$

Runtime Complexity of a TRS

$$\mathsf{s}^2(0)\cdot\mathsf{s}^3(0) \to \ldots \to \mathsf{s}^6(0) \text{ in } 10 \text{ steps}$$

Derivation Height

Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that t is in normal form. The *derivation height* of a term s wrt. a well-founded, finitely-branching relation $\to$ is defined as $\mathrm{dh}(s, \to) := \max\{n \mid \exists t. s \to^n t\}$

$$\mathrm{dh}(\mathsf{s}^2(0)\cdot\mathsf{s}^3(0), \to) = \max\{10, \ldots\}$$

Runtime Complexity

$\mathrm{rc}_{\mathcal{R}}(n) := \max\{\mathrm{dh}(s, \to_{\mathcal{R}}) \mid s \text{ is basic and } |s| \leqslant n\}$

$$|\mathsf{s}^2(0)\cdot\mathsf{s}^3(0)| = 8,\ \mathrm{rc}_{\mathcal{R}}(8) = ?$$

universität innsbruck

Worst-Case Input Computation

Simple and Complex Rules

Definition

A rewrite rule $\ell \to r$ is *simple* if r does not contain any defined symbol or $\mathrm{arity}(\ell) = 0$. Otherwise, it is *complex*.

$$
\begin{array}{llll}
1: & x \cdot \mathsf{s}(y) \to (x \cdot y) + x & 2: & x \cdot 0 \to 0 \\
3: & \mathsf{s}(x) + y \to \mathsf{s}(x + y) & 4: & 0 + x \to x \\
5: & x + \mathsf{s}(y) \to \mathsf{s}(x + y) & 6: & x + 0 \to x
\end{array}
$$

Simple and Complex Rules

Definition

A rewrite rule $\ell \to r$ is *simple* if r does not contain any defined symbol or $\text{arity}(\ell) = 0$. Otherwise, it is *complex*.

Complex Rules:			**Simple Rules:**		
$1:$	$x \cdot \mathsf{s}(y)$	$\to^1 (x \cdot y) + x$	$2:$	$x \cdot 0$	$\to^1 0$
$3:$	$\mathsf{s}(x) + y$	$\to^1 \mathsf{s}(x + y)$	$4:$	$0 + x$	$\to^1 x$
$5:$	$x + \mathsf{s}(y)$	$\to^1 \mathsf{s}(x + y)$	$6:$	$x + 0$	$\to^1 x$

We write $s \to^n t$ if exactly n steps are performed to rewrite s to t.

Narrowing

Narrowing

$s \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} t$ if $\mathrm{mgu}(s|_p, \ell) = \mu$ and $t = s[r\mu]_p$ and $\ell \to r \in \mathcal{R}$ and $p \in \mathit{FPos}(s)$.

Combination of Simple and Complex Rules

Let $s : \ell_s \to^a r_s$ be a simple rule and $c : \ell_c \to^b r_c$ a complex rule[2]. If $r_c \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} \ell_s$, then $\ell_c\mu \to^{a+b} r_s\mu$. We denote this combination by $c \circ s$.

$c_1 : \quad x \cdot \mathsf{s}(y) \to^1 (x \cdot y) + x$

$s_6 : \quad x + 0 \to^1 x$

[2] we rename variables such that $\mathcal{V}ar(s) \cap \mathcal{V}ar(c) = \emptyset$

Narrowing

Narrowing

$s \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} t$ if $\mathrm{mgu}(s|_p, \ell) = \mu$ and $t = s[r\mu]_p$ and $\ell \to r \in \mathcal{R}$ and $p \in FPos(s)$.

Combination of Simple and Complex Rules

Let $s : \ell_s \to^a r_s$ be a simple rule and $c : \ell_c \to^b r_c$ a complex rule[2]. If $r_c \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} \ell_s$, then $\ell_c\mu \to^{a+b} r_s\mu$. We denote this combination by $c \circ s$.

$c_1 : \quad x_1 \cdot \mathsf{s}(y_1) \to^1 (x_1 \cdot y_1) + x_1 \qquad\qquad s_6 : \quad x_6 + 0 \to^1 x_6$

[2] we rename variables such that $\mathcal{V}ar(s) \cap \mathcal{V}ar(c) = \emptyset$

Narrowing

Narrowing

$s \rightsquigarrow_{\mathcal{R},\mu \restriction V(s)} t$ if $\mathrm{mgu}(s|_p, \ell) = \mu$ and $t = s[r\mu]_p$ and $\ell \to r \in \mathcal{R}$ and $p \in \mathit{FPos}(s)$.

Combination of Simple and Complex Rules

Let $s : \ell_s \to^a r_s$ be a simple rule and $c : \ell_c \to^b r_c$ a complex rule[2]. If $r_c \rightsquigarrow_{\mathcal{R},\mu \restriction V(s)} \ell_s$, then $\ell_c\mu \to^{a+b} r_s\mu$. We denote this combination by $c \circ s$.

$c_1 : \quad x_1 \cdot \mathsf{s}(y_1) \to^1 (x_1 \cdot y_1) + x_1 \qquad s_6 : \quad x_6 + 0 \to^1 x_6$

$r_{c_1} \rightsquigarrow_{\mathcal{R},\mu \restriction V(s)} \ell_{s_6}$ for $\mu = \{x_6 \mapsto x_1 \cdot y_1, x_1 \mapsto 0\}$

[2] we rename variables such that $\mathcal{V}ar(s) \cap \mathcal{V}ar(c) = \emptyset$

Narrowing

Narrowing

$s \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} t$ if $\mathrm{mgu}(s|_p, \ell) = \mu$ and $t = s[r\mu]_p$ and $\ell \to r \in \mathcal{R}$ and $p \in FPos(s)$.

Combination of Simple and Complex Rules

Let $s : \ell_s \to^a r_s$ be a simple rule and $c : \ell_c \to^b r_c$ a complex rule[2]. If $r_c \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} \ell_s$, then $\ell_c\mu \to^{a+b} r_s\mu$. We denote this combination by $c \circ s$.

$$c_1 : \quad x_1 \cdot \mathsf{s}(y_1) \to^1 (x_1 \cdot y_1) + x_1 \qquad\qquad s_6 : \quad x_6 + 0 \to^1 x_6$$

$$r_{c_1} \rightsquigarrow_{\mathcal{R},\mu\restriction V(s)} \ell_{s_6} \text{ for } \mu = \{x_6 \mapsto x_1 \cdot y_1, x_1 \mapsto 0\} \qquad c_1 \circ s_6 = 0 \cdot \mathsf{s}(y_1) \to^2 0 \cdot y_1$$

[2] we rename variables such that $\mathcal{V}ar(s) \cap \mathcal{V}ar(c) = \emptyset$

Backward Narrowing

$$
\begin{aligned}
x \cdot \mathsf{s}(y) &\to^1 (x \cdot y) + x \\
x \cdot 0 &\to^1 0 \\
\mathsf{s}(x) + y &\to^1 \mathsf{s}(x + y) \\
0 + x &\to^1 x \\
x + \mathsf{s}(y) &\to^1 \mathsf{s}(x + y) \\
x + 0 &\to^1 x
\end{aligned}
$$

Backward Narrowing

$$
\begin{aligned}
x \cdot \mathsf{s}(y) &\to^1 (x \cdot y) + x \\
x \cdot 0 &\to^1 0 \\
\mathsf{s}(x) + y &\to^1 \mathsf{s}(x + y) \\
0 + x &\to^1 x \\
x + \mathsf{s}(y) &\to^1 \mathsf{s}(x + y) \\
x + 0 &\to^1 x
\end{aligned}
$$

$$
\begin{aligned}
0 \cdot \mathsf{s}(y_1) &\to^2 0 \cdot y_1 \\
x_5 + \mathsf{s}(0) &\to^2 \mathsf{s}(x_5)
\end{aligned}
$$

Backward Narrowing

$$x \cdot \mathsf{s}(y) \to^1 (x \cdot y) + x$$
$$x \cdot 0 \to^1 0$$
$$\mathsf{s}(x) + y \to^1 \mathsf{s}(x + y)$$
$$0 + x \to^1 x$$
$$x + \mathsf{s}(y) \to^1 \mathsf{s}(x + y)$$
$$x + 0 \to^1 x$$

$$0 \cdot \mathsf{s}(y_1) \to^2 0 \cdot y_1$$
$$x_5 + \mathsf{s}(0) \to^2 \mathsf{s}(x_5)$$

$$0 \cdot \mathsf{s}(0) \to^3 0$$
$$\mathsf{s}(@\mathsf{var}_7) + 0 \to^2 \mathsf{s}(@\mathsf{var}_7)$$
$$0 + \mathsf{s}(@\mathsf{var}_6) \to^2 \mathsf{s}(@\mathsf{var}_6)$$
$$\mathsf{s}(0) + @\mathsf{var}_8 \to^2 \mathsf{s}(@\mathsf{var}_8)$$
$$\mathsf{s}(0) \cdot \mathsf{s}(@\mathsf{var}_2) \to^3 \mathsf{s}(\mathsf{s}(0) \cdot @\mathsf{var}_2)$$
$$\dots$$
$$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{19} \mathsf{s}^{12}(0) + \mathsf{s}^4(0)$$
$$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{24} \mathsf{s}^{16}(0)$$
$$\mathsf{s}^{16}(0) + \mathsf{s}^{10}(@\mathsf{var}_{74}) \to^{27} \mathsf{s}^{26}(@\mathsf{var}_{74})$$
$$\dots$$

Backward Narrowing

$$x \cdot \mathsf{s}(y) \to^1 (x \cdot y) + x$$
$$x \cdot 0 \to^1 0$$
$$\mathsf{s}(x) + y \to^1 \mathsf{s}(x + y)$$
$$0 + x \to^1 x$$
$$x + \mathsf{s}(y) \to^1 \mathsf{s}(x + y)$$
$$x + 0 \to^1 x$$

$$0 \cdot \mathsf{s}(y_1) \to^2 0 \cdot y_1$$
$$x_5 + \mathsf{s}(0) \to^2 \mathsf{s}(x_5)$$

$$0 \cdot \mathsf{s}(0) \to^3 0$$
$$\mathsf{s}(@\mathrm{var}_7) + 0 \to^2 \mathsf{s}(@\mathrm{var}_7)$$
$$0 + \mathsf{s}(@\mathrm{var}_6) \to^2 \mathsf{s}(@\mathrm{var}_6)$$
$$\mathsf{s}(0) + @\mathrm{var}_8 \to^2 \mathsf{s}(@\mathrm{var}_8)$$
$$\mathsf{s}(0) \cdot \mathsf{s}(@\mathrm{var}_2) \to^3 \mathsf{s}(\mathsf{s}(0) \cdot @\mathrm{var}_2)$$
$$\cdots$$
$$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{19} \mathsf{s}^{12}(0) + \mathsf{s}^4(0)$$
$$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{24} \mathsf{s}^{16}(0)$$
$$\mathsf{s}^{16}(0) + \mathsf{s}^{10}(@\mathrm{var}_{74}) \to^{27} \mathsf{s}^{26}(@\mathrm{var}_{74})$$
$$\cdots$$

Method designed for term rewrite systems that correspond to functional programs.

universität innsbruck

Estimating the Asymptotic Complexity

Measurements

Rule-induced Measurement

We define the rule-induced measurement as a function

$$\mathrm{meas}(\ell \rightarrow^{n} r) = (\mathrm{root}(\ell), |\ell|, n).$$

Measurements

Rule-induced Measurement

We define the rule-induced measurement as a function

$$\mathrm{meas}(\ell \to^{n} r) = (\mathrm{root}(\ell), |\ell|, n).$$

rule	meas
$x \cdot \mathrm{s}(y) \to^{1} (x \cdot y) + x$	

Measurements

Rule-induced Measurement

We define the rule-induced measurement as a function

$$\mathsf{meas}(\ell \to^n r) = (\mathsf{root}(\ell), |\ell|, n).$$

rule	meas
$x \cdot \mathsf{s}(y) \to^1 (x \cdot y) + x$	$(\cdot, 4, 1)$

Measurements

Rule-induced Measurement

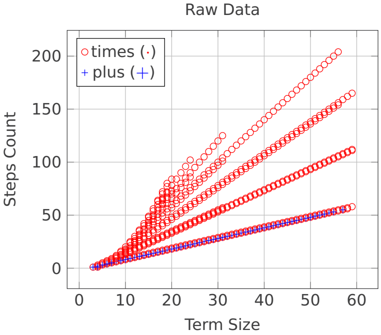
We define the rule-induced measurement as a function

$$\mathrm{meas}(\ell \to^{n} r) = (\mathrm{root}(\ell), |\ell|, n).$$

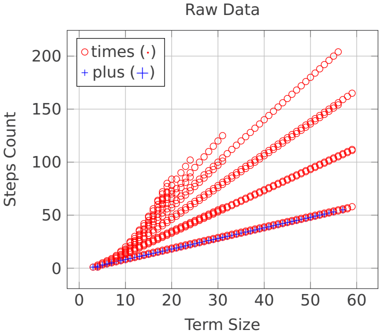
rule	meas
$x \cdot \mathsf{s}(y) \to^{1} (x \cdot y) + x$	$(\cdot, 4, 1)$
$x \cdot 0 \to^{1} 0$	$(\cdot, 3, 1)$
$0 \cdot \mathsf{s}(y_1) \to^{2} 0 \cdot y_1$	$(\cdot, 4, 2)$
$0 \cdot \mathsf{s}(0) \to^{3} 0$	$(\cdot, 4, 3)$
$\mathsf{s}(0) + @\mathsf{var}_8 \to^{2} \mathsf{s}(@\mathsf{var}_8)$	$(+, 4, 2)$

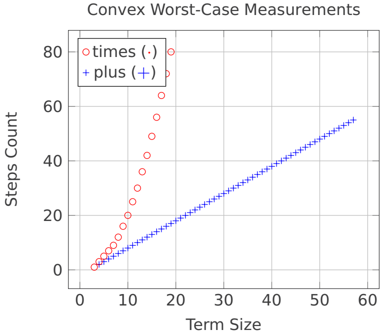
rule	meas
$\mathsf{s}(@\mathsf{var}_7) + 0 \to^{2} \mathsf{s}(@\mathsf{var}_7)$	$(+, 4, 2)$
$x_5 + \mathsf{s}(0) \to^{2} \mathsf{s}(x_5)$	$(+, 4, 2)$
$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{19} \mathsf{s}^{12}(0) + \mathsf{s}^4(0)$	$(\cdot, 11, 19)$
$\mathsf{s}^4(0) \cdot \mathsf{s}^4(0) \to^{24} \mathsf{s}^{16}(0)$	$(\cdot, 11, 24)$
$\mathsf{s}^{16}(0) + \mathsf{s}^{10}(@\mathsf{var}_{74}) \to^{27} \mathsf{s}^{26}(@\mathsf{var}_{74})$	$(+, 29, 27)$

Convex Worst-Case Measurements



Convex Worst-Case Measurements

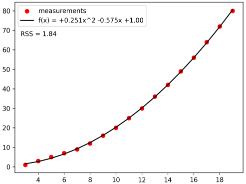


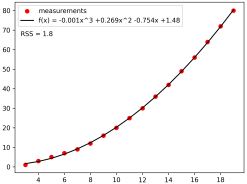


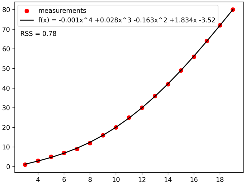
From Measurements to Complexity

Problem: Any set of points can be fitted arbitrarily well

Residual sum of squares is not a reliable indicator of asymptotic complexity.





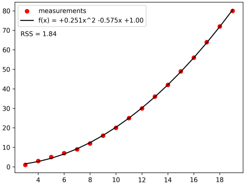


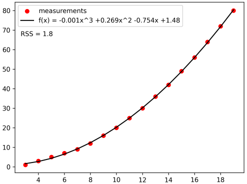
[3] C. C. McGeoch and P. R. Cohen. “How to Find Big-Oh in Your Data Set (and How Not To)”. In: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*. Ed. by David Madigan and Padhraic Smyth. Vol. R1. Proceedings of Machine Learning Research. PMLR, 1997, pp. 347–354. URL: `https://proceedings.mlr.press/r1/mcgeoch97a.html`.

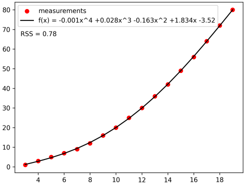
From Measurements to Complexity

Problem: Any set of points can be fitted arbitrarily well

Residual sum of squares is not a reliable indicator of asymptotic complexity.







Solution: Use the *difference rule heuristic*[3].

[3] C. C. McGeoch and P. R. Cohen. “How to Find Big-Oh in Your Data Set (and How Not To)”. In: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*. Ed. by David Madigan and Padhraic Smyth. Vol. R1. Proceedings of Machine Learning Research. PMLR, 1997, pp. 347–354. URL: `https://proceedings.mlr.press/r1/mcgeoch97a.html`.

From Measurements to Complexity

Lagrange Interpolation Theorem[4]

Given $n+1$ points $(x_0, y_0), \ldots, (x_n, y_n)$ with pairwise distinct x_i, there exists a *unique* polynomial P_n of degree at most n such that $P_n(x_i) = y_i$ for all $i \in \{0, 1, \ldots, n\}$.

The Newton form of a polynomial of degree n is expressed as:

$$P_n(x) = \sum_{k=0}^{n} a_k \cdot \prod_{j=0}^{k-1} (x - x_j)$$

where $\prod_{j=0}^{-1}(x - x_j) = 1$.

The (unique) coefficients a_k can be computed using the divided-difference scheme.

[4] Josef Stoer et al. *Introduction to numerical analysis*. Vol. 1993. Springer, 1980.

Divided-Differences

	$k=0$	$k=1$	$k=2$	$\dots$	$k=n$
x_0	y_0	y_0^1	y_0^2	$\dots$	y_0^n
x_1	y_1	y_1^1	y_1^2		
x_2	y_2	y_2^1			
$\vdots$	$\vdots$				
x_n	y_n				

where $y_j^{k+1} = \frac{y_{j+1}^k - y_j^k}{x_{j+1} - x_j}$ $(y_i^0 = y_i)$ with $k > 0$ and $j \leq n - k$.

Interpolating Polynomial

$$P_n(x) = y_0 + y_0^1(x - x_0) + \cdots + y_0^n(x - x_0)(x - x_1)\dots(x - x_{n-1})$$

Divided-Differences

Theorem

If $P(x)$ is a polynomial of degree n with $P(x_i) = y_i$, then $y_0^k = 0$ for $k > n$.

Proof.

Because of the unique solvability $P_k(x) = P(x)$ for $k \geq n$. The coefficients y_0^k in P_n must therefore vanish for $k > n$. ∎

Divided-Differences

Theorem

If $P(x)$ is a polynomial of degree n with $P(x_i) = y_i$, then $y_0^k = 0$ for $k > n$.

Proof.

Because of the unique solvability $P_k(x) = P(x)$ for $k \geq n$. The coefficients y_0^k in P_n must therefore vanish for $k > n$. ∎

x_i	$k = 0$	1	2	3	4
0	0	1	1	0	0
1	1	3	1	0	
2	4	5	1		
3	9	7			
4	16				

$\Longrightarrow$ We can exploit this method to estimate the asymptotic complexity.

Divided-Differences

Theorem

If $P(x)$ is a polynomial of degree n with $P(x_i) = y_i$, then $y_0^k = 0$ for $k > n$.

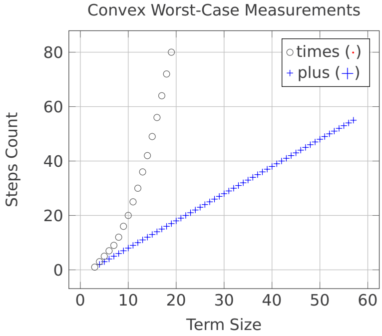
Proof.

Because of the unique solvability $P_k(x) = P(x)$ for $k \geq n$. The coefficients y_0^k in P_n must therefore vanish for $k > n$. ∎

x_i	$k = 0$	1	2	3	4
0	0	1	1	0	0
1	1	3	1	0	
2	4	5	1		
3	9	7			
4	16				

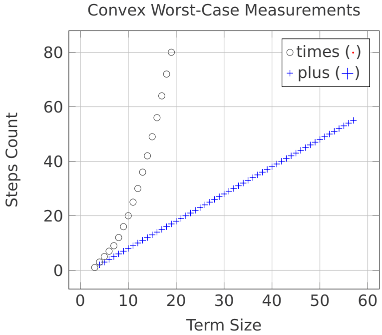
$\Longrightarrow$ We can exploit this method to estimate the asymptotic complexity.

$\mathcal{R}_2$ $+$



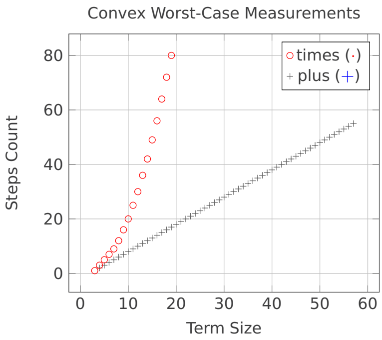
$\cdot$	$k=0$	1	2	3	4
3	1	1	0	0	0
4	2	1	0	0	0
5	3	1	0	0	0
6	4	1	0	0	0
7	5	1	0	0	0
8	6	1	0	0	0
9	7	1	0	0	0
10	8	1	0	0	0
11	9	1	0	0	0
12	10	1	0	0	0
13	11	1	0	0	0
...	...	...	...	...	...

$\mathcal{R}_2$ $+$



$\Longrightarrow$ **linear (n)**

$\cdot$	$k = 0$	1	2	3	4
3	1	1	0	0	0
4	2	1	0	0	0
5	3	1	0	0	0
6	4	1	0	0	0
7	5	1	0	0	0
8	6	1	0	0	0
9	7	1	0	0	0
10	8	1	0	0	0
11	9	1	0	0	0
12	10	1	0	0	0
13	11	1	0	0	0
...	...	...	...	...	...



$\cdot$	$k = 0$	1	2	3	4	5	6	7
3	1	2	0	0	0	0.01	0	0
4	3	2	0	0	0.04	-0.02	0	0
5	5	2	0	0.17	-0.04	0	0	0
6	7	2	0.50	0	-0.04	0.03	0.01	0
7	9	3	0.50	-0.17	0.08	-0.03	0.01	0
8	12	4	0	0.17	-0.08	0.03	0.01	0
9	16	4	0.50	-0.17	0.08	-0.03	0.01	0
10	20	5	0	0.17	-0.08	0.03	0.01	0
11	25	5	0.50	-0.17	0.08	-0.03	0.01	0
12	30	6	0	0.17	-0.08	0.03	0.01	0
13	36	6	0.50	-0.17	0.08	-0.03	0	0

$\Longrightarrow n^3$

Results

Termination Problem Database (TPDB)

The Termination Problems Data Base

collects termination problems that are being used in termination competitions

`https://termination-portal.org/wiki/TPDB`

Termination and Complexity Competition (termCOMP)[5]

several categories for termination and complexity from the areas of term rewriting

`https://termcomp.github.io/Y2024/`

[5] `https://termination-portal.org/wiki/Termination_Competition`

Termination Problem Database (TPDB)

The Termination Problems Data Base

collects termination problems that are being used in termination competitions

`https://termination-portal.org/wiki/TPDB`

Termination and Complexity Competition (termCOMP)[5]

several categories for termination and complexity from the areas of term rewriting

`https://termcomp.github.io/Y2024/`

Our Method vs. VBS

compare the results to validate our approach (timeout 3 seconds)

[5] `https://termination-portal.org/wiki/Termination_Competition`

Experimental Validation

fp + term.	**#**	**unsound (%)**	**compl. (%)**	**no val.**	**unk.**	**new (%)**	**acc. (%)**
innermost	169	7 (4.1)	138 (81.7)	24	0	27 (16.0)	74 (74.7)
full	111	7 (6.3)	80 (72.1)	24	0	30 (27.0)	39 (75.0)

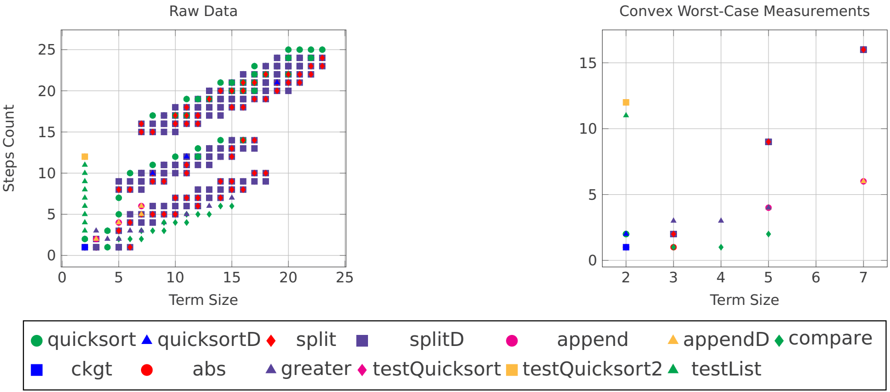
- Accuracy is defined wrt. to known results on the (innermost) runtime complexity of TRSs.
- *fp + term.* are TRSs that encode (uniformly) terminating functional programs (left-linear, strongly non-overlapping constructor TRSs).

Experimental Validation

fp + term.	#	unsound (%)	compl. (%)	no val.	unk.	new (%)	acc. (%)
innermost	169	7 (4.1)	138 (81.7)	24	0	27 (16.0)	74 (74.7)
full	111	7 (6.3)	80 (72.1)	24	0	30 (27.0)	39 (75.0)
TPDB							
innermost	663	81 (12.2)	486 (73.3)	75	21	224 (33.8)	138 (51.3)
full	959	203 (21.2)	568 (59.2)	113	75	306 (31.9)	204 (46.8)

- Accuracy is defined wrt. to known results on the (innermost) runtime complexity of TRSs.
- *fp + term.* are TRSs that encode (uniformly) terminating functional programs (left-linear, strongly non-overlapping constructor TRSs).

Limitation



We may not be able to find (enough) worst-case inputs

timeout or memory constraints

Summary

Conclusion and Next Steps

Contributions

- we have developed a (fast) method to synthesise worst-case inputs for term rewrite systems that model functional programs
- we have shown how to estimate the asymptotic complexity on the inferred measurements
- we have validated our approach on the TPDB

Future Work

- integration of machine learning techniques for rule selection
- infer polynomial bounds
- guarantees on the input size

Thank You for Your Attention!

Web-Tool

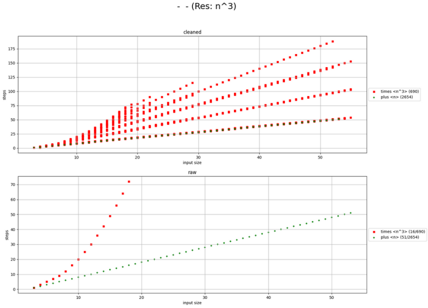
AG01/#3.16.xml.trs (n^2, n^3)

```
(VAR x y )
(STRATEGY INNERMOST)
(RULES
times(x,0) -> 0
times(x,s(y)) -> plus(times(x,y),x)
plus(x,0) -> x
plus(0,x) -> x
plus(x,s(y)) -> s(plus(x,y))
plus(s(x),y) -> s(plus(x,y))

)
```

Analyse

Version	worstcase4.ts
Timeout [in ms]	10000



WCA LOG

RESULT: **n^3**

Ground Truth: **(n^2,n^3)**

https://www.uibk.ac.at/en/theoretical-computer-science/research/ddrca/

Worst-Case Input Computation

Algorithm 1 Pseudo-code of work list algorithm.

```
Require: initialRules
  foundRules ← ∅
  pendingRules ← initialRules
  delayedRecursiveRules ← ∅
  while pendingRules ≠ ∅ or delayedRecursiveRules ≠ ∅ do
    if pendingRules = ∅ then
      steps ← 0
      for rule in delayedRecursiveRules do
        steps ← max(steps, getStepsToLastRecursion(rule))
      end for
      for rule in delayedRecursiveRules do
        if getStepsToLastRecursion(rule) = steps then
          pendingRules ← pendingRules ∪ rule
          delayedRecursiveRules ← delayedRecursiveRules \ rule
        end if
      end for
    end if
    for rule in pendingRules do
      foundRules ← foundRules ∪ rule
      pendingRules ← pendingRules \ rule
      for derivedRule in getDerivableRules(rule, foundRules) do
        if isRecursive(derivedRule) then
          delayedRecursiveRules ← delayedRecursiveRules ∪ derivedRule
        else
          pendingRules ← pendingRules ∪ derivedRule
        end if
      end for
    end for
  end while
```

Divided-Differences – Example

	$k=0$	$k=1$	$k=2$
0	1	2	$-5/6$
1	3	$-1/2$	
3	2		

Interpolating Polynomial

$$P_n(x) = y_0 + y_0^1(x - x_0) + \cdots + y_0^n(x - x_0)(x - x_1)\dots(x - x_{n-1})$$
$$P_2(x) = 1 + 2(x - 0) - 5/6(x - 0)(x - 1)$$
$$P_2(x) = 1 + \frac{17}{6}x - \frac{5}{6}x^2$$

Divided-Differences – Example

	$k=0$	$k=1$	$k=2$
0	1	2	$-5/6$
1	3	$-1/2$	
3	2		

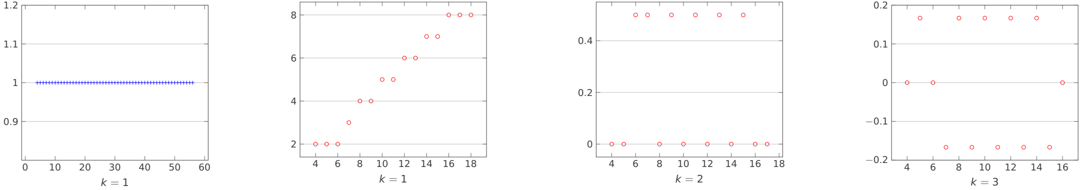
Interpolating Polynomial

$$P_n(x) = y_0 + y_0^1(x - x_0) + \cdots + y_0^n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$
$$P_2(x) = 1 + 2(x - 0) - 5/6(x - 0)(x - 1)$$
$$P_2(x) = 1 + \frac{17}{6}x - \frac{5}{6}x^2$$

The Difference Rule[6]

The Difference Rule

extends Newton’s divided difference method for polynomial interpolation to be defined when Y contains random noise. The method iterates numerical differentiation until the data appears non-increasing.



[6] C. C. McGeoch and P. R. Cohen. “How to Find Big-Oh in Your Data Set (and How Not To)”. In: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*. Ed. by David Madigan and Padhraic Smyth. Vol. R1. Proceedings of Machine Learning Research. PMLR, 1997, pp. 347–354. URL: `https://proceedings.mlr.press/r1/mcgeoch97a.html`.