Constant Runtime Complexity of Term Rewriting is Semi-Decidable

Jürgen Giesl

LuFG Informatik 2, RWTH Aachen University, Germany

joint work with Florian Frohn

Runtime Complexity

Example TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{d}) \to \mathsf{c}(\mathsf{g}(\mathsf{d})) \qquad \mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(\mathsf{f}(x)))$$

Rewrite Sequence

$$\underline{\mathsf{g}(\mathsf{c}(\mathsf{d}))} \to_{\mathcal{R}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(\mathsf{d})})) \to_{\mathcal{R}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \to_{\mathcal{R}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

Defined Symbols:	roots of lhs	f, g
Constructors:	other function symbols	c, d
Basic Term:	defined symbol only at the root	$\mathsf{f}(x)$, $\mathsf{g}(\mathsf{c}(x))$, $\mathsf{g}(\mathsf{c}(\mathsf{d}))$

Runtime Complexity $\mathsf{rc}_{\mathcal{R}}(n)$

length of longest $\to_{\mathcal{R}}$-sequence starting with basic term t where $|t| \leq n$

Example: $\mathsf{rc}_{\mathcal{R}}(3) \geq 3$ since $|\,\mathsf{g}(\mathsf{c}(\mathsf{d}))\,| = 3$

Runtime Complexity

Contribution: Constant runtime complexity is semi-decidable

	$\mathcal{R}$ has constant runtime complexity
$\Leftrightarrow$	$\mathsf{rc}_{\mathcal{R}}(n) \in \mathcal{O}(1)$
$\Leftrightarrow$	$\exists m \in \mathbb{N}.$ all $\to_{\mathcal{R}}$-evaluations of basic terms take at most m steps

Motivation

Complexity Analysis: bounds on program's resource usage

Constant Bounds: detect bugs

Runtime Complexity $\mathsf{rc}_{\mathcal{R}}(n)$

length of longest $\to_{\mathcal{R}}$-sequence starting with basic term t where $|t| \leq n$

Constructor-Based Narrowing

Example TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{d}) \to \mathsf{c}(\mathsf{g}(\mathsf{d})) \qquad \mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(\mathsf{f}(x)))$$

Constructor-Based Narrowing Sequence

$$\underline{\mathsf{g}(x)} \overset{\{x/\mathsf{c}(x')\}}{\underset{\varepsilon}{\leadsto}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(x')})) \overset{\{x'/\mathsf{d}\}}{\underset{1.1}{\leadsto}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\underset{1}{\leadsto}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

Narrowing sequence $t_0 \overset{\sigma_1}{\leadsto} \cdots \overset{\sigma_n}{\leadsto} t_n$ is constructor based if $t_0\, \sigma_1 \cdots \sigma_n$ is basic.

Constructor-based $\mathcal{R}$-narrowing terminates:

$\mathsf{g}(x),\ \mathsf{g}(\mathsf{c}(x)),\ \mathsf{g}(\mathsf{c}(\mathsf{d}))$	3 steps
$\mathsf{g}(\mathsf{c}(t))$ for other terms t	1 step
$\mathsf{g}(t)$ for other terms t	0 steps
$\mathsf{f}(x),\ \mathsf{f}(\mathsf{d})$	1 step
$\mathsf{f}(t)$ for other terms t	0 steps

Goal: Show termination of cb narrowing by inspecting finitely many start terms

Main Theorem

$\mathcal{R}$ has constant runtime complexity iff constructor-based $\mathcal{R}$-narrowing terminates

Constructor-Based Narrowing

Example TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{d}) \to \mathsf{c}(\mathsf{g}(\mathsf{d})) \qquad\qquad \mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(\mathsf{f}(x)))$$

Constructor-Based Narrowing Sequence

$$\underline{\mathsf{g}(x)} \overset{\{x/\mathsf{c}(x')\}}{\underset{\varepsilon}{\rightsquigarrow}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(x')})) \overset{\{x'/\mathsf{d}\}}{\underset{1.1}{\rightsquigarrow}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\underset{1}{\rightsquigarrow}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

Narrowing sequence $t_0 \overset{\sigma_1}{\rightsquigarrow} \cdots \overset{\sigma_n}{\rightsquigarrow} t_n$ is constructor based if $t_0\,\sigma_1 \cdots \sigma_n$ is basic.

For $\mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(x))$, cb narrowing would not terminate:

$$\underline{\mathsf{g}(x)} \overset{\{x/\mathsf{c}(x')\}}{\rightsquigarrow} \mathsf{c}(\underline{\mathsf{g}(x')}) \overset{\{x'/\mathsf{c}(x'')\}}{\rightsquigarrow} \mathsf{c}(\mathsf{c}(\underline{\mathsf{g}(x'')})) \rightsquigarrow \cdots$$

Goal: Show termination of cb narrowing by inspecting finitely many start terms

Main Theorem

$\mathcal{R}$ has constant runtime complexity iff constructor-based $\mathcal{R}$-narrowing terminates

Constructor-Based Narrowing

Example TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{d}) \to \mathsf{c}(\mathsf{g}(\mathsf{d})) \qquad\qquad \mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(\mathsf{f}(x)))$$

Constructor-Based Narrowing Sequence

$$\underline{\mathsf{g}(x)} \overset{\{x/\mathsf{c}(x')\}}{\underset{\varepsilon}{\rightsquigarrow}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(x')})) \overset{\{x'/\mathsf{d}\}}{\underset{1.1}{\rightsquigarrow}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\underset{1}{\rightsquigarrow}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

$$\underline{\mathsf{g}(\mathsf{c}(x'))} \overset{\varnothing}{\underset{\varepsilon}{\rightsquigarrow}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(x')})) \overset{\{x'/\mathsf{d}\}}{\underset{1.1}{\rightsquigarrow}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\underset{1}{\rightsquigarrow}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

$$\underline{\mathsf{g}(\mathsf{c}(\mathsf{d}))} \overset{\varnothing}{\underset{\varepsilon}{\rightsquigarrow}} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(\mathsf{d})})) \overset{\varnothing}{\underset{1.1}{\rightsquigarrow}} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\underset{1}{\rightsquigarrow}} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

Def: $s_0 \overset{\sigma_1}{\underset{\pi_1}{\rightsquigarrow}} \cdots \overset{\sigma_n}{\underset{\pi_n}{\rightsquigarrow}} s_n$ is **more general** than $t_0 \overset{\theta_1}{\underset{\pi_1}{\rightsquigarrow}} \cdots \overset{\theta_n}{\underset{\pi_n}{\rightsquigarrow}} t_n$ if there is a substitution η with

$$\begin{array}{lllll} s_0 & \sigma_1\sigma_2\cdots\sigma_n & \eta & = & t_0 \quad \theta_1\theta_2\cdots\theta_n \\ s_1 & \sigma_2\cdots\sigma_n & \eta & = & t_1 \quad \theta_2\cdots\theta_n \\ & & & \cdots & \\ s_n & & \eta & = & t_n \end{array}$$

Goal: Show termination of cb narrowing by inspecting finitely many start terms

Narrowing Lemma

For every $f(\ldots) \rightsquigarrow^n t$ there is a more general sequence $f(x_1,\ldots,x_k) \rightsquigarrow^n s$.

Main Theorem

Main Theorem

$\mathcal{R}$ has constant runtime complexity iff constructor-based $\mathcal{R}$-narrowing terminates

Proof of "$\Leftarrow$"

Assume that $\mathcal{R}$ does not have constant runtime complexity

$\Rightarrow \quad f(\vec{q_1}) \to^{n_1} t_1, \quad f(\vec{q_2}) \to^{n_2} t_2, \ \ldots$ with $n_1 < n_2 < \cdots$

$\Rightarrow \quad f(\vec{q_1}) \overset{\varnothing}{\leadsto}{}^{n_1} t_1, \quad f(\vec{q_2}) \overset{\varnothing}{\leadsto}{}^{n_2} t_2, \ \ldots$

$\Rightarrow \quad f(x_1, \ldots, x_k) \overset{\sigma_1}{\leadsto}{}^{n_1} s_1, \quad f(x_1, \ldots, x_k) \overset{\sigma_2}{\leadsto}{}^{n_2} s_2, \ \ldots$ by Narrowing Lemma

$\Rightarrow$ cb narrowing tree with root $f(x_1, \ldots, x_k)$

- has infinitely many nodes
- is finitely branching (as $\mathcal{R}$ is finite)
- has infinite path (by König's Lemma), i.e., infinite cb narrowing sequence $\lightning$

Narrowing Lemma

For every $f(\ldots) \leadsto^n t$ there is a more general sequence $f(x_1, \ldots, x_k) \leadsto^n s$.

Main Theorem

Main Theorem

$\mathcal{R}$ has constant runtime complexity iff constructor-based $\mathcal{R}$-narrowing terminates

Proof of "$\Rightarrow$"

Assume that there is an infinite sequence $t_0 \overset{\sigma_1}{\leadsto} t_1 \overset{\sigma_2}{\leadsto} \cdots$

$\Rightarrow$ $t_0\, \sigma_1 \cdots \sigma_{m+1} \rightarrow^{m+1} t_{m+1}$ for all $m \in \mathbb{N}$

$\Rightarrow$ $\forall m.$ there is an $\rightarrow_{\mathcal{R}}$-evaluation of a basic term with more than m steps

$\Leftrightarrow$ $\mathcal{R}$ does not have constant runtime complexity $\lightning$

Narrowing Lemma

For every $f(\ldots) \leadsto^n t$ there is a more general sequence $f(x_1, \ldots, x_k) \leadsto^n s$.

Semi-Decision Procedure for Constant Runtime Complexity

Example TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{d}) \to \mathsf{c}(\mathsf{g}(\mathsf{d})) \qquad\qquad \mathsf{g}(\mathsf{c}(x)) \to \mathsf{c}(\mathsf{g}(\mathsf{f}(x)))$$

CB Narrowing Trees

$$\underline{\mathsf{f}(x)} \overset{\{x/\mathsf{d}\}}{\rightsquigarrow} \mathsf{c}(\mathsf{g}(\mathsf{d}))$$

$$\underline{\mathsf{g}(x)} \overset{\{x/\mathsf{c}(x')\}}{\rightsquigarrow} \mathsf{c}(\mathsf{g}(\underline{\mathsf{f}(x')})) \overset{\{x'/\mathsf{d}\}}{\rightsquigarrow} \mathsf{c}(\underline{\mathsf{g}(\mathsf{c}(\mathsf{g}(\mathsf{d})))}) \overset{\varnothing}{\rightsquigarrow} \mathsf{c}(\mathsf{c}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{d})))))$$

Semi-Decision Procedure

- For all defined symbols f, build cb narrowing tree for $f(x_1, \ldots, x_k)$.
- If constructing the trees terminates, then return "*constant runtime*".

Undecidability of Constant Runtime Complexity

Theorem

Constant runtime complexity of TRSs is semi-decidable, but not decidable.

Proof

$$\begin{aligned}
& \text{Turing machine } \mathcal{M} \text{ is immortal} \\
\Leftrightarrow\quad & \text{rewriting } \textit{infinite} \text{ basic terms with } \mathcal{R}_\mathcal{M} \text{ does not terminate} \\
\Leftrightarrow\quad & \text{narrowing basic terms } \mathsf{f}(x_1, \ldots, x_k) \text{ with } \mathcal{R}_\mathcal{M} \text{ does not terminate} \\
\Leftrightarrow\quad & \mathcal{R}_\mathcal{M} \text{ does not have constant runtime complexity}
\end{aligned}$$

(Im)mortality undecidable $\Rightarrow$ constant runtime complexity undecidable $\square$

Experiments

Theorem

Constant runtime complexity of TRSs is semi-decidable, but not decidable.

- Implementation and integration of semi-decision procedure in AProVE
- Full rewriting (959 examples from the TPDB, 60 s per example)
  - 57 TRSs with constant runtime
  - 57 TRSs detected by semi-decision procedure, 1.8 s avg. on successes
  - 51 TRSs detected by TcT or AProVE without semi-decision procedure
- Innermost rewriting (1022 examples from the TPDB, 60 s per example)
  - 59 TRSs with constant runtime
  - 58 TRSs detected by semi-decision procedure, 1.4 s avg. on successes
  - 1 TRS with *relative* rules not detected by semi-decision procedure
  - 55 TRSs detected by TcT or AProVE without semi-decision procedure

Constant Runtime Complexity of TRSs is Semi-Decidable

$\mathcal{R}$ has constant runtime complexity

$\Leftrightarrow$ $\exists m \in \mathbb{N}.$ all $\to_{\mathcal{R}}$-evaluations of basic terms take at most m steps

Main Theorem

$\mathcal{R}$ has constant runtime complexity iff
constructor-based $\mathcal{R}$-narrowing of all terms $f(x_1, \ldots, x_k)$ terminates

Semi-Decision Procedure (implemented in AProVE)

- For all defined symbols f, build cb narrowing tree for $f(x_1, \ldots, x_k)$.
- If constructing the trees terminates, then return "*constant runtime*".

Theorem

Constant runtime complexity of TRSs is semi-decidable, but not decidable.