A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems

Lars Noschinski, Fabian Emmes, Jürgen Giesl
(by: Cynthia Kop)

September 10, 2024

Term Rewriting Systems

definition

Terms: strings built from variables and function symbols.
Rules: generate a relation between terms.

Term Rewriting Systems

definition

Terms: strings built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols: $\texttt{Plus} : 2, \texttt{Times} : 2, \texttt{S} : 1, \texttt{0} : 0$.
Rules:

$$\begin{aligned} \texttt{Plus}(x, \texttt{0}) &\to x \\ \texttt{Plus}(x, \texttt{S}(y)) &\to \texttt{S}(\texttt{Plus}(x, y)) \\ \texttt{Times}(x, \texttt{0})) &\to \texttt{0} \\ \texttt{Times}(x, \texttt{S}(y)) &\to \texttt{Plus}(x, \texttt{Times}(x, y)) \end{aligned}$$

Term Rewriting Systems

example

$$\begin{aligned} \texttt{Plus}(x, \texttt{0}) &\rightarrow x \\ \texttt{Plus}(x, \texttt{S}(y)) &\rightarrow \texttt{S}(\texttt{Plus}(x, y)) \end{aligned}$$

Example reduction:

$\underline{\texttt{Plus}}(\texttt{0}, \texttt{S}(\texttt{0}))$

Term Rewriting Systems

example

$$\begin{array}{rcll} \texttt{Plus}(x, \texttt{0}) & \to & x & \\ \texttt{Plus}(x, \texttt{S}(y)) & \to & \texttt{S}(\texttt{Plus}(x, y)) & \Longleftarrow \end{array}$$

Example reduction:

$\underline{\texttt{Plus}}(\texttt{0}, \texttt{S}(\texttt{0}))$

Term Rewriting Systems

example

$$\begin{aligned} \texttt{Plus}(x, \texttt{0}) &\rightarrow x \\ \texttt{Plus}(x, \texttt{S}(y)) &\rightarrow \texttt{S}(\texttt{Plus}(x, y)) \end{aligned}$$

Example reduction:

$\underline{\texttt{Plus}}(\texttt{0}, \texttt{S}(\texttt{0})) \rightarrow \texttt{S}(\underline{\texttt{Plus}}(\texttt{0}, \texttt{0}))$

Term Rewriting Systems

example

$$\begin{aligned} \texttt{Plus}(x, \texttt{0}) &\rightarrow x \quad \Longleftarrow \\ \texttt{Plus}(x, \texttt{S}(y)) &\rightarrow \texttt{S}(\texttt{Plus}(x, y)) \end{aligned}$$

Example reduction:

$\underline{\texttt{Plus}}(\texttt{0}, \texttt{S}(\texttt{0})) \rightarrow \texttt{S}(\underline{\texttt{Plus}}(\texttt{0}, \texttt{0}))$

Term Rewriting Systems

example

$$\begin{aligned} \texttt{Plus}(x, \texttt{0}) &\to x \\ \texttt{Plus}(x, \texttt{S}(y)) &\to \texttt{S}(\texttt{Plus}(x, y)) \end{aligned}$$

Example reduction:

$\underline{\texttt{Plus}}(\texttt{0}, \texttt{S}(\texttt{0})) \to \texttt{S}(\underline{\texttt{Plus}}(\texttt{0}, \texttt{0})) \to \texttt{S}(\texttt{0})$

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Starting point: $\mathtt{function}(\mathtt{data}_1, \ldots, \mathtt{data}_n)$

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Starting point: $\mathtt{function}(\mathtt{data}_1, \ldots, \mathtt{data}_n)$

Example: $\mathtt{Plus}(s, t)$:

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Starting point: $\mathtt{function}(\mathtt{data}_1, \ldots, \mathtt{data}_n)$

Example: $\mathtt{Plus}(s, t)$: $\mathcal{O}(n)$

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Starting point: $\mathtt{function}(\mathtt{data}_1, \ldots, \mathtt{data}_n)$

Example: $\mathtt{Plus}(s, t)$: $\mathcal{O}(n)$

Example: $\mathtt{Times}(s, t)$:

Runtime complexity

Main Question

How long does a computation take (using innermost reduction)?

Starting point: $\mathtt{function}(\mathtt{data}_1, \ldots, \mathtt{data}_n)$

Example: $\mathtt{Plus}(s, t)$: $\mathcal{O}(n)$

Example: $\mathtt{Times}(s, t)$: $\mathcal{O}(n^2)$

Investigating termination and complexity

Idea: consider function calls

Investigating termination and complexity

Idea: consider function calls

$$\begin{array}{rcl}
\mathtt{Plus}(x, \mathtt{0}) & \to & x \\
\mathtt{Plus}(x, \mathtt{S}(y)) & \to & \mathtt{S}(\underline{\mathtt{Plus}}(x, y)) \\
\mathtt{Times}(x, \mathtt{0})) & \to & \mathtt{0} \\
\mathtt{Times}(x, \mathtt{S}(y)) & \to & \underline{\mathtt{Plus}}(x, \underline{\mathtt{Times}}(x, y))
\end{array}$$

Investigating termination and complexity

Idea: consider function calls

$$\begin{array}{rcl}
\mathtt{Plus}(x, \mathtt{0}) & \to & x \\
\mathtt{Plus}(x, \mathtt{S}(y)) & \to & \mathtt{S}(\underline{\mathtt{Plus}}(x, y)) \\
\mathtt{Times}(x, \mathtt{0})) & \to & \mathtt{0} \\
\mathtt{Times}(x, \mathtt{S}(y)) & \to & \underline{\mathtt{Plus}}(x, \underline{\mathtt{Times}}(x, y))
\end{array}$$

List all the calls for each rule:

$$\begin{array}{rcl}
\mathtt{Plus}^\sharp(x, \mathtt{0}) & \to & \langle\rangle \\
\mathtt{Plus}^\sharp(x, \mathtt{S}(y)) & \to & \langle \mathtt{Plus}^\sharp(x, y) \rangle \\
\mathtt{Times}^\sharp(x, \mathtt{0})) & \to & \langle\rangle \\
\mathtt{Times}^\sharp(x, \mathtt{S}(y)) & \to & \langle\ \mathtt{Plus}^\sharp(x, \mathtt{Times}(x, y))\ ; \\
 & & \ \mathtt{Times}^\sharp(x, y)\ \rangle
\end{array}$$

Various methods!

- usable rules
- reduction pairs
- leaf removal
- knowledge propagation
- narrowing