A Combinatorial Framwork for Complexity

Martin Avanzini Georg Moser

Institute of Computer Science
University of Innsbruck, Austria

August 30, 2013

Combinatorial Framwork for Complexity

Complexity Problem

abstract

- **complexity problem** $\mathcal{P}$ is tuple $\langle \to, \mathcal{T} \rangle$
  1. $\to$ is binary relation on terms

     $$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}{}^*_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}{}^*_{\mathcal{W}}$$

     - $\mathcal{S}, \mathcal{W}$ are TRSs
     - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \to_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  2. $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathrm{cp}_{\mathcal{P}}(n) := \max\{\mathrm{dh}(t, \to) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical derivational complexity problem**

$$\langle \mathcal{R}/\varnothing, \varnothing, \texttt{all terms} \rangle$$

Complexity Problem

concrete

- **complexity problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

  ❶ $\rightarrow$ is binary relation on terms

$$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}}$$

  - $\mathcal{S}$,$\mathcal{W}$ are TRSs
  - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \rightarrow_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  ❷ $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathsf{cp}_{\mathcal{P}}(n) := \max\{\mathsf{dh}(t, \rightarrow) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical derivational complexity problem**

$$\langle \mathcal{R}/\varnothing, \varnothing, \texttt{all terms} \rangle$$

Complexity Problem

concrete

- **complexity problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

  ❶ $\rightarrow$ is binary relation on terms

$$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}}$$

  - $\mathcal{S}$,$\mathcal{W}$ are TRSs
  - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \rightarrow_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  ❷ $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathsf{cp}_{\mathcal{P}}(n) := \max\{\mathsf{dh}(t, \rightarrow) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical derivational complexity problem**

$$\langle \mathcal{R}/\varnothing, \varnothing, \texttt{all terms} \rangle$$

Complexity Problem

concrete

- **complexity problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$
  1. $\to$ is binary relation on terms

$$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}}$$

     - $\mathcal{S}$,$\mathcal{W}$ are TRSs
     - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \to_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  2. $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathsf{cp}_{\mathcal{P}}(n) := \max\{\mathsf{dh}(t, \to) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical derivational complexity problem**

$$\langle \mathcal{R}/\varnothing, \varnothing, \texttt{all terms} \rangle$$

Complexity Problem

concrete

- **complexity problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

  ❶ $\rightarrow$ is binary relation on terms

  $$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}}$$

  - $\mathcal{S}$,$\mathcal{W}$ are TRSs
  - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \rightarrow_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  ❷ $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

  $$\mathsf{cp}_{\mathcal{P}}(n) := \max\{\mathsf{dh}(t, \rightarrow) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical runtime complexity problem**

  $$\langle \mathcal{R}/\varnothing, \varnothing, \texttt{basic terms} \rangle$$

Complexity Problem

concrete

- **complexity problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$
  1. $\to$ is binary relation on terms

$$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}^*_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}^*_{\mathcal{W}}$$

    - $\mathcal{S}$,$\mathcal{W}$ are TRSs
    - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \to_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  2. $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathrm{cp}_{\mathcal{P}}(n) := \max\{\mathrm{dh}(t, \to) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- **canonical innermost runtime complexity problem**

$$\langle \mathcal{R}/\varnothing, \mathcal{R}, \texttt{basic terms} \rangle$$

Complexity Judgements, Processors and Proofs

complexity judgement is statement $\vdash \mathcal{P} : f$

- $\mathcal{P}$ is a complexity problem
- $f : \mathbb{N} \to \mathbb{N}$ is bounding function
- valid if $\mathsf{cp}_{\mathcal{P}}(n) \in \mathsf{O}(f(n))$

complexity processor is inference rule

$$\frac{\vdash \mathcal{P}_1 : f_1 \quad \cdots \quad \vdash \mathcal{P}_n : f_n}{\vdash \mathcal{P} : f}$$

- sound if validity of judgements preserved

complexity proof of $\vdash \mathcal{P} : f$ is a deduction using sound processors only.

Complexity Judgements, Processors and Proofs

complexity judgement is statement $\vdash \mathcal{P} \colon f$

- $\mathcal{P}$ is a complexity problem
- $f \colon \mathbb{N} \to \mathbb{N}$ is bounding function
- valid if $\mathsf{cp}_{\mathcal{P}}(n) \in \mathsf{O}(f(n))$

complexity processor is inference rule

$$\frac{\vdash \mathcal{P}_1 \colon f_1 \quad \cdots \quad \vdash \mathcal{P}_n \colon f_n}{\vdash \mathcal{P} \colon f}$$

- sound if validity of judgements preserved

complexity proof of $\vdash \mathcal{P} \colon f$ is a deduction using sound processors only.

Complexity Judgements, Processors and Proofs

complexity judgement is statement $\vdash \mathcal{P} : f$

- $\mathcal{P}$ is a complexity problem
- $f : \mathbb{N} \to \mathbb{N}$ is bounding function
- valid if $\mathsf{cp}_{\mathcal{P}}(n) \in \mathsf{O}(f(n))$

complexity processor is inference rule

$$\frac{\vdash \mathcal{P}_1 : f_1 \quad \cdots \quad \vdash \mathcal{P}_n : f_n}{\vdash \mathcal{P} : f}$$

- sound if validity of judgements preserved

complexity proof of $\vdash \mathcal{P} : f$ is a deduction using sound processors only.

Orders for Complexity

collapsible orders

- order $\succ$ on terms called collapsible if for mapping $\mathsf{G} : \mathtt{terms} \to \mathbb{N}$

$$s \succ t \text{ implies } \mathsf{G}(s) >_{\mathbb{N}} \mathsf{G}(t) \qquad \text{for all steps } s \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} t$$

- order $\succ$ induces complexity bound $f : \mathbb{N} \to \mathbb{N}$ on terms $\mathcal{T}$

$$\mathsf{G}(t) \leqslant f(\mathsf{size}(t)) \qquad \text{for all } t \in \mathcal{T}$$

assume:

$$\begin{tikzcd}
t_1 \arrow[r, "\mathcal{Q}"]_{\mathcal{S}/\mathcal{W}} \arrow[d, dotted] & t_2 \arrow[r, "\mathcal{Q}"]_{\mathcal{S}/\mathcal{W}} \arrow[d, dotted] & t_3 \arrow[r, "\mathcal{Q}"]_{\mathcal{S}/\mathcal{W}} \arrow[d, dotted] & \cdots \arrow[r, "\mathcal{Q}"]_{\mathcal{S}/\mathcal{W}} & t_\ell \arrow[d, dotted] \\
t_1 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & t_2 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & t_3 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & \cdots \arrow[r, phantom, "\succ"] & t_\ell \arrow[d, dotted] \\
\mathsf{G}(t_1) \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_2) \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_3) \arrow[r, phantom, ">_{\mathbb{N}}"] & \cdots \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_\ell)
\end{tikzcd}$$

Orders for Complexity

collapsible orders

- order $\succ$ on terms called collapsible if for mapping $\mathsf{G} : \mathtt{terms} \to \mathbb{N}$

$$s \succ t \text{ implies } \mathsf{G}(s) >_{\mathbb{N}} \mathsf{G}(t) \qquad \text{for all steps } s \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} t$$

- order $\succ$ induces complexity bound $f : \mathbb{N} \to \mathbb{N}$ on terms $\mathcal{T}$

$$\mathsf{G}(t) \leqslant f(\mathsf{size}(t)) \qquad \text{for all } t \in \mathcal{T}$$

assume:

$$\begin{array}{ccccccccc}
t_1 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_2 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_3 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & \cdots & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_\ell \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow \\
t_1 & \succ & t_2 & \succ & t_3 & \succ & \cdots & \succ & t_\ell \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow \\
\mathsf{G}(t_1) & >_{\mathbb{N}} & \mathsf{G}(t_2) & >_{\mathbb{N}} & \mathsf{G}(t_3) & >_{\mathbb{N}} & \cdots & >_{\mathbb{N}} & \mathsf{G}(t_\ell)
\end{array}$$

Orders for Complexity

collapsible orders

- order $\succ$ on terms called collapsible if for mapping $\mathsf{G} : \mathtt{terms} \to \mathbb{N}$

$$s \succ t \text{ implies } \mathsf{G}(s) >_{\mathbb{N}} \mathsf{G}(t) \qquad\qquad \text{for all steps } s \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} t$$

- order $\succ$ induces complexity bound $f : \mathbb{N} \to \mathbb{N}$ on terms $\mathcal{T}$

$$\mathsf{G}(t) \leqslant f(\mathsf{size}(t)) \qquad\qquad \text{for all } t \in \mathcal{T}$$

assume:

$$\begin{tikzcd}
t_1 \arrow[r, phantom, "\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}}"] \arrow[d, dotted] & t_2 \arrow[r, phantom, "\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}}"] \arrow[d, dotted] & t_3 \arrow[r, phantom, "\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}}"] \arrow[d, dotted] & \cdots \arrow[r, phantom, "\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}}"] & t_\ell \arrow[d, dotted] \\
t_1 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & t_2 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & t_3 \arrow[r, phantom, "\succ"] \arrow[d, dotted] & \cdots \arrow[r, phantom, "\succ"] & t_\ell \arrow[d, dotted] \\
\mathsf{G}(t_1) \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_2) \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_3) \arrow[r, phantom, ">_{\mathbb{N}}"] & \cdots \arrow[r, phantom, ">_{\mathbb{N}}"] & \mathsf{G}(t_\ell)
\end{tikzcd}$$

Orders for Complexity

collapsible orders

- order $\succ$ on terms called collapsible if for mapping $\mathsf{G} : \mathtt{terms} \to \mathbb{N}$

$$s \succ t \text{ implies } \mathsf{G}(s) >_{\mathbb{N}} \mathsf{G}(t) \qquad \text{for all steps } s \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} t$$

- order $\succ$ induces complexity bound $f : \mathbb{N} \to \mathbb{N}$ on terms $\mathcal{T}$

$$\mathsf{G}(t) \leqslant f(\mathsf{size}(t)) \qquad \text{for all } t \in \mathcal{T}$$

assume:

$$\begin{array}{ccccccccc}
t_1 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_2 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_3 & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & \cdots & \xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} & t_\ell \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow \\
t_1 & \succ & t_2 & \succ & t_3 & \succ & \cdots & \succ & t_\ell \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow \\
\mathsf{G}(t_1) & >_{\mathbb{N}} & \mathsf{G}(t_2) & >_{\mathbb{N}} & \mathsf{G}(t_3) & >_{\mathbb{N}} & \cdots & >_{\mathbb{N}} & \mathsf{G}(t_\ell)
\end{array}$$

Orders for Complexity

complexity pairs

$$\frac{\mathcal{S} \subseteq \succ \quad \mathcal{W} \subseteq \succcurlyeq}{\vdash \langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f}$$

1. order $\succ$ induces complexity $f : \mathbb{N} \to \mathbb{N}$ on $\mathcal{T}$
2. $\succcurlyeq$ preorder with $\succcurlyeq \cdot \succ \cdot \succcurlyeq \subseteq \succ$
3. $(\succcurlyeq, \succ)$ are stable under substitutions and monotone

   $\sim$ order $\succcurlyeq$ "*monotone under $\mathcal{W}$-reducible argument positions*"

   $\sim$ order $\succ$ "*monotone under $\mathcal{S}$-reducible argument positions*"

Harald Zankl and Martin Korp
Modular Complexity Analysis via Relative Complexity.
Proc. 21st RTA, pages 385–400, 2010

Orders for Complexity

$\mathcal{P}$-monotone complexity pairs

$$\frac{\mathcal{S} \subseteq \succ \quad \mathcal{W} \subseteq \succcurlyeq}{\vdash \langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f}$$

1. order $\succ$ induces complexity $f : \mathbb{N} \to \mathbb{N}$ on $\mathcal{T}$
2. $\succcurlyeq$ preorder with $\succcurlyeq \cdot \succ \cdot \succcurlyeq \subseteq \succ$
3. $(\succcurlyeq, \succ)$ are stable under substitutions and $\mathcal{P}$-monotone
   - $\sim$ order $\succcurlyeq$ "*monotone under $\mathcal{W}$-reducible argument positions*"
   - $\sim$ order $\succ$ "*monotone under $\mathcal{S}$-reducible argument positions*"

Orders For Complexity

practice

+ very powerful in theory
– difficult to synthesise
  1. disaster for systems with many rules
  2. implementations usually do not go beyond induced complexity n^3

⇒ decomposition into manageable pieces
  1. simplify rules
  2. reduce complexity

Experimental Evaluation

testsuite

straight forward translations of resource aware ML programs

machine

super nice workstation with 12 Intel® Core™ i7-3930K (3.20GHz)

Experimental Evaluation

Input	#rules	direct	decompose	DG decompose	secs
appendAll	12	$O(n^2)$	$O(n^2)$	$O(n)$	2.9
bfs	57	?	?	$O(n)$	27.9
bft mmult	59	?	?	$O(n^3)$	55.3
bitonic	78	?	?	$O(n^4)$	143.0
bitvectors	148	?	?	$O(n^2)$	35.5
clevermmult	39	?	?	$O(n^2)$	220.0
duplicates	37	?	$O(n^2)$	$O(n^2)$	3.5
dyade	31	?	?	$O(n^2)$	1.0
eratosthenes	74	?	$O(n^3)$	$O(n^2)$	15.6
flatten	31	?	?	$O(n^2)$	25.0
insertionsort	36	?	$O(n^3)$	$O(n^2)$	6.7
listsort	56	?	?	$O(n^2)$	22.8
lcs	87	?	?	$O(n^2)$	24.5
matrix	74	?	?	$O(n^3)$	72.4
mergesort	35	?	?	$O(n^3)$	29.0
minsort	26	?	$O(n^3)$	$O(n^2)$	2.4
queue	35	?	?	$O(n^5)$	148.4
quicksort	46	?	?	$O(n^2)$	38.4
rationalPotential	14	$O(n)$	$O(n)$	$O(n)$	0.27
splitandsort	70	?	?	$O(n^3)$	61.4
subtrees	8	?	$O(n^2)$	$O(n^2)$	3.6
tuples	33	?	?	?	–

Orders For Complexity

practice

- \+ very powerful in theory
- – very very very difficult to synthesise efficiently
  1. disaster for systems with many rules
  2. implementations usually do not go beyond induced complexity n^3

⇒ decomposition into manageable pieces

1. simplify rules
2. reduce complexity

Additive Decomposition

$$\frac{\begin{array}{cc} \mathcal{S}_1 \subseteq \succ \quad \mathcal{S}_2 \cup \mathcal{W} \subseteq \succsim & \\ \vdash \langle \mathcal{S}_1 / \mathcal{S}_2 \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_1 & \vdash \langle \mathcal{S}_2 / \mathcal{S}_1 \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_2 \end{array}}{\vdash \langle \mathcal{S}_1 \cup \mathcal{S}_2 / \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_1 + f_2}$$

Harald Zankl and Martin Korp
Modular Complexity Analysis via Relative Complexity.
Proc. 21st RTA, pages 385–400, 2010

Additive Decomposition

$$\frac{\dfrac{\mathcal{S}_1 \subseteq \succ \quad \mathcal{S}_2 \cup \mathcal{W} \subseteq \succsim}{\vdash \langle \mathcal{S}_1 / \mathcal{S}_2 \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_1} \qquad \vdash \langle \mathcal{S}_2 / \mathcal{S}_1 \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_2}{\vdash \langle \mathcal{S}_1 \cup \mathcal{S}_2 / \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f_1 + f_2}$$

Harald Zankl and Martin Korp
Modular Complexity Analysis via Relative Complexity.
Proc. 21st RTA, pages 385–400, 2010

Experimental Evaluation

Input	#rules	direct	decompose	DG decompose	secs
appendAll	12	$O(n^2)$	$O(n^2)$	$O(n)$	2.9
bfs	57	?	?	$O(n)$	27.9
bft mmult	59	?	?	$O(n^3)$	55.3
bitonic	78	?	?	$O(n^4)$	143.0
bitvectors	148	?	?	$O(n^2)$	35.5
clevermmult	39	?	?	$O(n^2)$	220.0
duplicates	37	?	$O(n^2)$	$O(n^2)$	3.5
dyade	31	?	?	$O(n^2)$	1.0
eratosthenes	74	?	$O(n^3)$	$O(n^2)$	15.8
flatten	31	?	?	$O(n^2)$	25.0
insertionsort	36	?	$O(n^3)$	$O(n^2)$	6.7
listsort	56	?	?	$O(n^2)$	22.8
lcs	87	?	?	$O(n^2)$	24.5
matrix	74	?	?	$O(n^3)$	72.4
mergesort	35	?	?	$O(n^3)$	29.0
minsort	26	?	$O(n^3)$	$O(n^2)$	2.4
queue	35	?	?	$O(n^5)$	148.4
quicksort	46	?	?	$O(n^2)$	38.4
rationalPotential	14	$O(n)$	$O(n)$	$O(n)$	0.27
splitandsort	70	?	?	$O(n^3)$	61.4
subtrees	8	?	$O(n^2)$	$O(n^2)$	3.6
tuples	33	?	?	?	–

Towards Modularity

dependency graph decomposition

Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$① : 0 \times y \rightarrow 0 \qquad ② : \mathsf{s}(x) \times y \rightarrow y + (x \times y)$$

$$③ : 0 + y \rightarrow y \qquad ④ : \mathsf{s}(x) + y \rightarrow \mathsf{s}(x + y)$$

$\mathsf{dh}(\mathbf{m} \times \mathbf{n}, \rightarrow_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$

1. #steps $\times$ $\qquad$ $\mathrm{O}(m)$
2. #steps $+$ $\qquad$ $\mathrm{O}(m \cdot n)$

   2.1 #calls linear in #steps $\times$

   #recursions linear in first argument

3. total: $\qquad$ $\mathrm{O}(m \cdot n)$

Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$① : 0 \times y \to 0 \qquad ② : \mathsf{s}(x) \times y \to y + (x \times y)$$
$$③ : 0 + y \to y \qquad ④ : \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

$$\mathrm{dh}(\mathbf{m} \times \mathbf{n}, \to_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

❶ #steps $\times$ $\mathrm{O}(m)$

❷ #steps $+$ $\mathrm{O}(m \cdot n)$

2.1 #calls linear in #steps $\times$

2.2 #recursions linear in first argument

❸ total: $\mathrm{O}(m \cdot n)$

Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$
\begin{array}{ll}
① : 0 \times y \rightarrow 0 & ② : \mathsf{s}(x) \times y \rightarrow y + (x \times y) \\
③ : 0 + y \rightarrow y & ④ : \mathsf{s}(x) + y \rightarrow \mathsf{s}(x + y)
\end{array}
$$

$$\mathsf{dh}(\mathbf{m} \times \mathbf{n}, \rightarrow_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

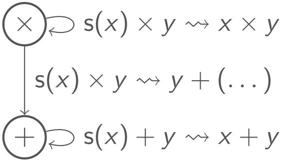
❶ #steps $\times$ — $\mathrm{O}(m)$

❷ #steps $+$ — $\mathrm{O}(m \cdot n)$

2.1 #calls linear in #steps $\times$

2.2 #recursions linear in first argument

❸ total: — $\mathrm{O}(m \cdot n)$



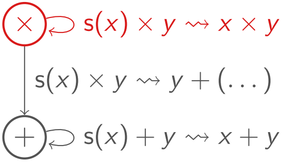
Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$
\begin{array}{ll}
① : 0 \times y \to 0 & ② : \mathsf{s}(x) \times y \to y + (x \times y) \\
③ : 0 + y \to y & ④ : \mathsf{s}(x) + y \to \mathsf{s}(x + y)
\end{array}
$$

$$\mathrm{dh}(\mathbf{m} \times \mathbf{n}, \to_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

1. #steps $\times$ — $\mathrm{O}(m)$
2. #steps $+$ — $\mathrm{O}(m \cdot n)$
   - 2.1 #calls linear in #steps $\times$ — $\mathrm{O}(m)$
   - 2.2 #recursions linear in first argument
3. total: — $\mathrm{O}(m \cdot n)$



Motivation

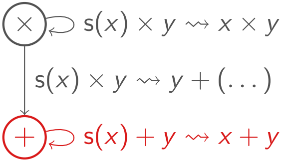
Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$\text{①} : 0 \times y \to 0 \qquad \text{②} : \mathsf{s}(x) \times y \to y + (x \times y)$$

$$\text{③} : 0 + y \to y \qquad \text{④} : \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

$$\mathsf{dh}(\mathbf{m} \times \mathbf{n}, \to_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

❶	#steps $\times$	$\mathrm{O}(m)$
❷	#steps $+$	$\mathrm{O}(m \cdot n)$
	2.1 #calls linear in #steps $\times$	$\mathrm{O}(m)$
	2.2 #recursions linear in first argument	
❸	total:	$\mathrm{O}(m \cdot n)$



Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$
\begin{array}{ll}
① : 0 \times y \to 0 & ② : \mathsf{s}(x) \times y \to y + (x \times y) \\
③ : 0 + y \to y & ④ : \mathsf{s}(x) + y \to \mathsf{s}(x + y)
\end{array}
$$

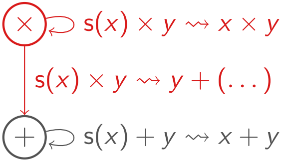
$$\mathrm{dh}(\mathbf{m} \times \mathbf{n}, \to_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

❶ #steps $\times$ $\qquad \mathrm{O}(m)$

❷ #steps $+$ $\qquad \mathrm{O}(m \cdot n)$

2.1	#calls linear in #steps $\times$	$\mathrm{O}(m)$
2.2	#recursions linear in n	$\mathrm{O}(n)$

❸ total: $\qquad \mathrm{O}(m \cdot n)$



Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$① : 0 \times y \to 0 \qquad ② : \mathsf{s}(x) \times y \to y + (x \times y)$$

$$③ : 0 + y \to y \qquad ④ : \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

$$\mathsf{dh}(\mathbf{m} \times \mathbf{n}, \to_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

❶ #steps $\times$ — $\mathrm{O}(m)$

❷ #steps $+$ — $\mathrm{O}(m \cdot n)$

- 2.1 #calls linear in #steps $\times$ — $\mathrm{O}(m)$
- 2.2 #recursions linear in n — $\mathrm{O}(n)$

❸ total: — $\mathrm{O}(m \cdot n)$

$$\begin{tikzcd}
\times \arrow[loop right, "\mathsf{s}(x) \times y \rightsquigarrow x \times y"] \arrow[d, "\mathsf{s}(x) \times y \rightsquigarrow y + (\dots)"] \\
+ \arrow[loop right, "\mathsf{s}(x) + y \rightsquigarrow x + y"]
\end{tikzcd}$$

Motivation

Let $\mathcal{D} = \{+, \times\}$ and $\mathcal{C} = \{0, \mathsf{s}\}$ and consider following TRS $\mathcal{R}_\times$

$$① : 0 \times y \rightarrow 0 \qquad ② : \mathsf{s}(x) \times y \rightarrow y + (x \times y)$$

$$③ : 0 + y \rightarrow y \qquad ④ : \mathsf{s}(x) + y \rightarrow \mathsf{s}(x + y)$$

howto formalise?

$$\mathsf{dh}(\mathbf{m} \times \mathbf{n}, \rightarrow_{\mathcal{R}_\times}) \in \mathrm{O}(m \cdot n)$$

1. #steps $\times$ — $\mathrm{O}(m)$
2. #steps $+$ — $\mathrm{O}(m \cdot n)$
   - 2.1 #calls linear in #steps $\times$ — $\mathrm{O}(m)$
   - 2.2 #recursions linear in n — $\mathrm{O}(n)$
3. total: — $\mathrm{O}(m \cdot n)$

$$\begin{array}{ll} \times & \mathsf{s}(x) \times y \rightsquigarrow x \times y \\ \downarrow & \mathsf{s}(x) \times y \rightsquigarrow y + (\ldots) \\ + & \mathsf{s}(x) + y \rightsquigarrow x + y \end{array}$$

Dependency Pair Complexity Problems

- dependency pair for termination is rule $l^\sharp \to r^\sharp$
  - $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - $f(t_1, \ldots, t_n)^\sharp = f^\sharp(t_1, \ldots, t_n)$ and $x^\sharp = x$ otherwise

- DP complexity problem is complexity problem

$$\langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$$

  - $\mathcal{S}^\sharp, \mathcal{W}^\sharp$ are two sets of dependency pairs
  - $\mathcal{S}, \mathcal{W}$ and $\mathcal{Q}$ are TRSs as before
  - $\mathcal{T}^\sharp$ some marked and basic terms

Dependency Pair Complexity Problems

- dependency pair for complexity is rule $l^\sharp \to \mathsf{c}_n(r_1^\sharp, \ldots, r_n^\sharp)$
  - $l, r_1, \ldots, r_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - $f(t_1, \ldots, t_n)^\sharp = f^\sharp(t_1, \ldots, t_n)$ and $x^\sharp = x$ otherwise
- DP complexity problem is complexity problem

$$\langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$$

  - $\mathcal{S}^\sharp, \mathcal{W}^\sharp$ are two sets of dependency pairs
  - $\mathcal{S}, \mathcal{W}$ and $\mathcal{Q}$ are TRSs as before
  - $\mathcal{T}^\sharp$ some marked and basic terms

Dependency Pair Complexity Problems

- dependency pair for complexity is rule $l^\sharp \to \mathsf{c}_n(r_1^\sharp, \dots, r_n^\sharp)$
  - $l, r_1, \dots, r_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - $f(t_1, \dots, t_n)^\sharp = f^\sharp(t_1, \dots, t_n)$ and $x^\sharp = x$ otherwise

- DP complexity problem is complexity problem

$$\langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$$

  - $\mathcal{S}^\sharp, \mathcal{W}^\sharp$ are two sets of dependency pairs
  - $\mathcal{S}, \mathcal{W}$ and $\mathcal{Q}$ are TRSs as before
  - $\mathcal{T}^\sharp$ some marked and basic terms

Dependency Pair Complexity Problems

- sound (complete) processors that translate to DP problems exist

Nao Hirokawa and Georg Moser
Automated Complexity Analysis Based on the Dependency Pair Method
Proc. 4th IJCAR, pages 364–379, 2008.

Lars Noschinski, Fabian Emmes, and Jürgen Giesl
A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems
Proc. 23th CADE, pages 422–438, 2011.

Dependency Pair Co

- sound (complete) pro

$$\frac{\vdash \langle \{\times, +\}/\mathcal{R}_\times, \mathcal{R}_\times, \texttt{basic terms}^\sharp\rangle \colon f}{\vdots}$$

$$\overline{\vdash \langle \mathcal{R}_\times/\varnothing, \varnothing, \texttt{basic terms}\rangle \colon f}$$

$\times\colon\ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y)$

$+\colon\ \mathsf{s}(x) +^\sharp y \to x +^\sharp y$

Nao Hirokawa and Ge
Automated Complexit
Method
Proc. 4th IJCAR, page

Lars Noschinski, Fabian Emmes, and Jürgen Giesl
A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems
Proc. 23th CADE, pages 422–438, 2011.

Dependency Graph

dependency graph of $\langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$ is graphs such that

1. nodes are dependency pairs $\mathcal{S}^\sharp \cup \mathcal{W}^\sharp$
2. there is edge

$$(s^\sharp \to \mathsf{c}_n(t_1^\sharp, \dots, t_n^\sharp)) \;\to\; (u^\sharp \to \mathsf{c}_m(v_1^\sharp, \dots, v_m^\sharp))$$

if there exists substitutions σ, τ such that $t_i^\sharp \sigma \xrightarrow{\mathcal{Q}}{}^*_{\mathcal{S} \cup \mathcal{W}} u^\sharp \tau$

Dependency Graph

dependency graph of $\langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$ is graphs such that

1. nodes are dependency pairs $\mathcal{S}^\sharp \cup \mathcal{W}^\sharp$
2. there is edge

$$(s^\sharp \to \mathsf{c}_n(t_1^\sharp, \ldots, t_n^\sharp)) \ \to \ (u^\sharp \to \mathsf{c}_m(v_1^\sharp, \ldots, v_m^\sharp))$$

if there exists substitutions σ, τ such that $t_i^\sharp \sigma \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{S} \cup \mathcal{W}} u^\sharp \tau$

Dependency Graph

Example

$\mathbf{3} \times^\sharp \mathbf{2}$

$$\begin{tikzcd} \times \arrow[loop right] \arrow[d] \\ + \arrow[loop right] \end{tikzcd}$$

$$\mathcal{S}^\sharp_\uparrow := \left\{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \right\}$$

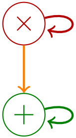
$$\mathcal{S}^\sharp_\downarrow := \left\{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \right\}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$$\mathbf{3} \times^\sharp \mathbf{2}$$



$$\mathcal{S}^\sharp_\uparrow := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \,\}$$

$$\mathcal{S}^\sharp_\downarrow := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \,\}$$

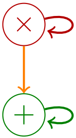
$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$\mathbf{3} \times^\sharp \mathbf{2}$

$\mathbf{2} +^\sharp (\underline{\mathbf{2} \times \mathbf{2}})$ $\qquad$ $\mathbf{2} \times^\sharp \mathbf{2}$



$$\mathcal{S}^\sharp_\uparrow := \left\{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \right\}$$

$$\mathcal{S}^\sharp_\downarrow := \left\{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \right\}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

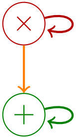
Dependency Graph

Example

$$\mathbf{3} \times^\sharp \mathbf{2}$$

$$\mathbf{2} +^\sharp (\underline{\mathbf{2} \times \mathbf{2}}) \qquad \mathbf{2} \times^\sharp \mathbf{2}$$

$$\mathbf{2} +^\sharp (\mathbf{2} + (\underline{\mathbf{1} \times \mathbf{2}}))$$



$$\mathcal{S}^\sharp_\uparrow := \{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

$$\mathcal{S}^\sharp_\downarrow := \{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$$3 \times^\sharp 2$$

$$2 +^\sharp (\underline{2 \times 2}) \qquad\qquad 2 \times^\sharp 2$$

$$2 +^\sharp (2 + (\underline{1 \times 2}))$$

$$\vdots$$

$$2 +^\sharp 4$$

$$\begin{tikzcd} \times \arrow[loop right] \arrow[d] \\ + \arrow[loop right] \end{tikzcd}$$

$$\mathcal{S}^\sharp_\uparrow := \{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

$$\mathcal{S}^\sharp_\downarrow := \{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$$\mathbf{3} \times^\sharp \mathbf{2}$$

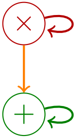
$$\mathbf{2} +^\sharp (\underline{\mathbf{2} \times \mathbf{2}}) \qquad\qquad \mathbf{2} \times^\sharp \mathbf{2}$$

$$\mathbf{2} +^\sharp (\mathbf{2} + (\underline{\mathbf{1} \times \mathbf{2}}))$$

$$\vdots$$

$$\mathbf{2} +^\sharp \mathbf{4}$$

$$\mathbf{1} +^\sharp \mathbf{4}$$



$$\mathcal{S}^\sharp_\uparrow := \{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

$$\mathcal{S}^\sharp_\downarrow := \{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$$3 \times^\sharp 2$$

$$2 +^\sharp (\underline{2 \times 2}) \qquad 2 \times^\sharp 2$$

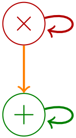
$$2 +^\sharp (2 + (\underline{1 \times 2}))$$

$$\vdots$$

$$2 +^\sharp 4$$

$$1 +^\sharp 4$$

$$0 +^\sharp 4$$



$$\mathcal{S}_\uparrow^\sharp := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \,\}$$

$$\mathcal{S}_\downarrow^\sharp := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \,\}$$

$$\mathsf{sep}(\mathcal{S}_\uparrow^\sharp) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$$\mathbf{3} \times^\sharp \mathbf{2}$$

$$\mathbf{2} +^\sharp (\underline{\mathbf{2} \times \mathbf{2}}) \qquad \mathbf{2} \times^\sharp \mathbf{2}$$

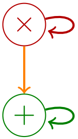
$$\mathbf{2} +^\sharp (\mathbf{2} + (\underline{\mathbf{1} \times \mathbf{2}})) \qquad \mathbf{2} +^\sharp (\underline{\mathbf{1} \times \mathbf{2}}) \qquad \mathbf{1} \times^\sharp \mathbf{2}$$

$$\vdots$$

$$\mathbf{2} +^\sharp \mathbf{4}$$

$$\mathbf{1} +^\sharp \mathbf{4}$$

$$\mathbf{0} +^\sharp \mathbf{4}$$



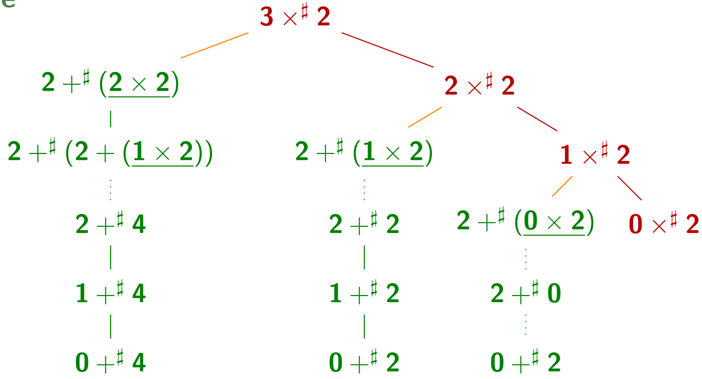
$$\mathcal{S}^\sharp_\uparrow := \{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

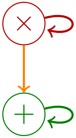
$$\mathcal{S}^\sharp_\downarrow := \{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example





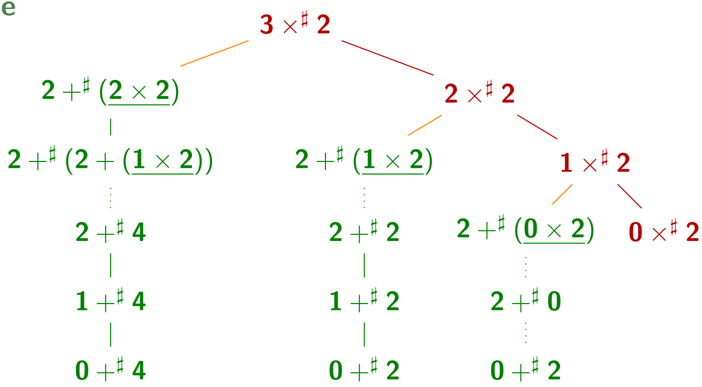
$$\mathcal{S}^\sharp_\uparrow := \left\{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \right\}$$

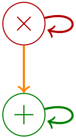
$$\mathcal{S}^\sharp_\downarrow := \left\{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \right\}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example





$$\mathcal{S}^\sharp_\uparrow \quad := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

$$\mathcal{S}^\sharp_\downarrow \quad := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example

$3 \times^\sharp 2$

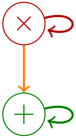
$2 +^\sharp (\underline{2 \times 2})$ $\quad$ $2 \times^\sharp 2$

$2 +^\sharp (2 + (\underline{1 \times 2}))$ $\quad$ $2 +^\sharp (\underline{1 \times 2})$ $\quad$ $1 \times^\sharp 2$

$2 +^\sharp 4$ $\quad$ $2 +^\sharp 2$ $\quad$ $2 +^\sharp (\underline{0 \times 2})$ $\quad$ $0 \times^\sharp 2$

$1 +^\sharp 4$ $\quad$ $1 +^\sharp 2$ $\quad$ $2 +^\sharp 0$

$0 +^\sharp 4$ $\quad$ $0 +^\sharp 2$ $\quad$ $0 +^\sharp 2$



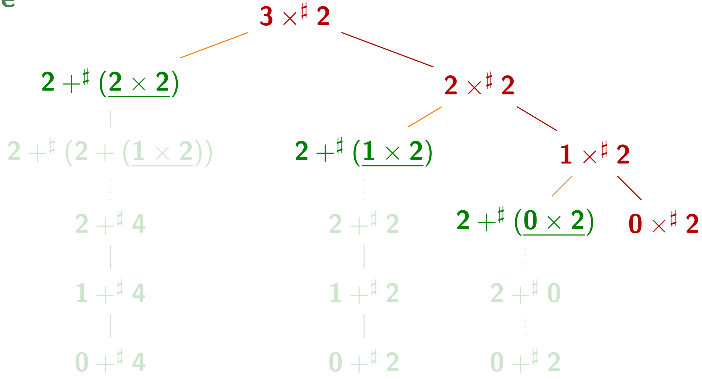
$$\mathcal{S}^\sharp_\uparrow := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

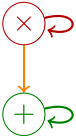
$$\mathcal{S}^\sharp_\downarrow := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example





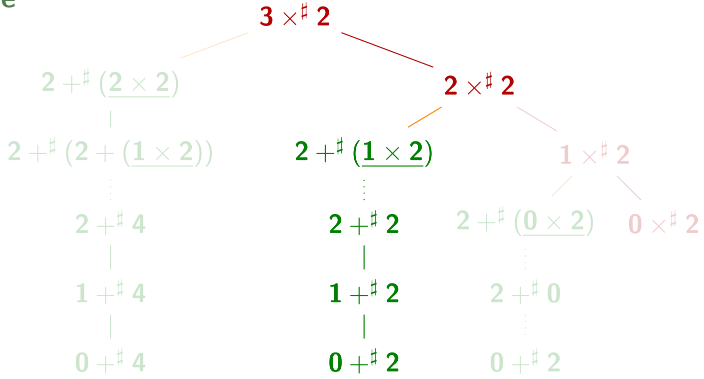
$$\mathcal{S}^\sharp_\uparrow := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

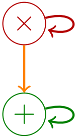
$$\mathcal{S}^\sharp_\downarrow := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example





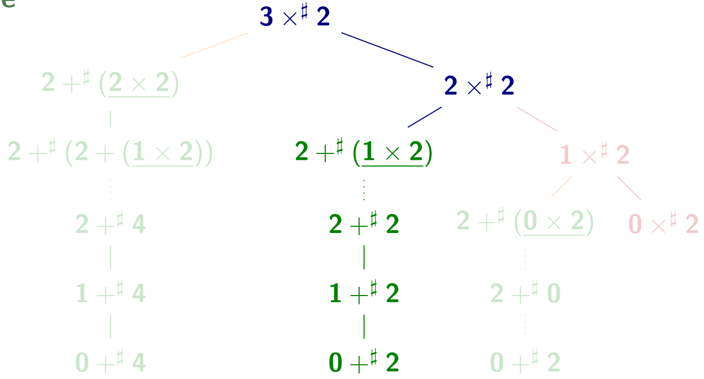
$$\mathcal{S}^\sharp_\uparrow := \{\, \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

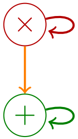
$$\mathcal{S}^\sharp_\downarrow := \{\, \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}^\sharp_\uparrow) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph

Example





$$\mathcal{S}_\uparrow^\sharp := \{ \mathsf{s}(x) \times^\sharp y \to \mathsf{c}_2(y +^\sharp (x \times y), x \times^\sharp y) \}$$

$$\mathcal{S}_\downarrow^\sharp := \{ \mathsf{s}(x) +^\sharp y \to x +^\sharp y \}$$

$$\mathsf{sep}(\mathcal{S}_\uparrow^\sharp) := \left\{ \begin{array}{l} \mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y) \\ \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y \end{array} \right\}$$

Dependency Graph Decomposition

consider $\mathcal{P} = \langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$ with dependency graph $\mathcal{G}$.

- partition $\mathcal{S}^\sharp = \mathcal{S}^\sharp_\downarrow \cup \mathcal{S}^\sharp_\uparrow$ and $\mathcal{W}^\sharp = \mathcal{W}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\uparrow$
- suppose $\mathcal{S}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\downarrow$ closed under successors in $\mathcal{G}$
- define $\mathsf{sep}(l^\sharp \to \mathsf{c}(r_1^\sharp, \ldots, r_m^\sharp)) := \{l^\sharp \to r_i^\sharp \mid i = 1, \ldots, m\}$
- predecessors of $\mathcal{S}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\downarrow$ in $\mathcal{G}$ contained in $\mathcal{S}^\sharp_\uparrow$

$$\frac{\vdash \langle \mathcal{S}^\sharp_\uparrow \cup \mathcal{S} / \mathcal{W}^\sharp_\uparrow \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f \qquad \vdash \langle \mathcal{S}^\sharp_\downarrow \cup \mathcal{S} / \mathcal{W}^\sharp_\downarrow \cup \mathsf{sep}(\mathcal{S}^\sharp_\uparrow \cup \mathcal{W}^\sharp_\uparrow) \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon g}{\vdash \langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon \lambda n. f(n) * g(n)}$$

Dependency Graph Decomposition

consider $\mathcal{P} = \langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T}^\sharp \rangle$ with dependency graph $\mathcal{G}$.

- partition $\mathcal{S}^\sharp = \mathcal{S}^\sharp_\downarrow \cup \mathcal{S}^\sharp_\uparrow$ and $\mathcal{W}^\sharp = \mathcal{W}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\uparrow$
- suppose $\mathcal{S}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\downarrow$ closed under successors in $\mathcal{G}$
- define $\mathsf{sep}(l^\sharp \to \mathsf{c}(r^\sharp_1, \dots, r^\sharp_m)) := \{l^\sharp \to r^\sharp_i \mid i = 1, \dots, m\}$
- predecessors of $\mathcal{S}^\sharp_\downarrow \cup \mathcal{W}^\sharp_\downarrow$ in $\mathcal{G}$ contained in $\mathcal{S}^\sharp_\uparrow$

$$\frac{\vdash \langle \mathcal{S}^\sharp_\uparrow \cup \mathcal{S} / \mathcal{W}^\sharp_\uparrow \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon f \qquad \vdash \langle \mathcal{S}^\sharp_\downarrow \cup \mathcal{S} / \mathcal{W}^\sharp_\downarrow \cup \mathsf{sep}(\mathcal{S}^\sharp_\uparrow \cup \mathcal{W}^\sharp_\uparrow) \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon g}{\vdash \langle \mathcal{S}^\sharp \cup \mathcal{S} / \mathcal{W}^\sharp \cup \mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle \colon \lambda n. f(n) * g(n)}$$

Experimental Evaluation

Input	#rules	direct	decompose	DG decompose	secs
appendAll	12	$O(n^2)$	$O(n^2)$	$O(n)$	2.9
bfs	57	?	?	$O(n)$	27.9
bft mmult	59	?	?	$O(n^3)$	55.3
bitonic	78	?	?	$O(n^4)$	143.0
bitvectors	148	?	?	$O(n^2)$	35.5
clevermmult	39	?	?	$O(n^2)$	220.0
duplicates	37	?	$O(n^2)$	$O(n^2)$	3.5
dyade	31	?	?	$O(n^2)$	1.0
eratosthenes	74	?	$O(n^3)$	$O(n^2)$	15.8
flatten	31	?	?	$O(n^2)$	25.0
insertionsort	36	?	$O(n^3)$	$O(n^2)$	6.7
listsort	56	?	?	$O(n^2)$	22.8
lcs	87	?	?	$O(n^2)$	24.5
matrix	74	?	?	$O(n^3)$	72.4
mergesort	35	?	?	$O(n^3)$	29.0
minsort	26	?	$O(n^3)$	$O(n^2)$	2.4
queue	35	?	?	$O(n^5)$	148.4
quicksort	46	?	?	$O(n^2)$	38.4
rationalPotential	14	$O(n)$	$O(n)$	$O(n)$	0.27
splitandsort	70	?	?	$O(n^3)$	61.4
subtrees	8	?	$O(n^2)$	$O(n^2)$	3.6
tuples	33	?	?	?	–

Experimental Evaluation

Input	#rules	direct	decompose	DG decompose	secs
appendAll	12	$O(n^2)$	$O(n^2)$	$O(n)$	2.9
bfs	57	?	?	$O(n)$	27.9
bft mmult	59	?	?	$O(n^3)$	55.3
bitonic	78	?	?	$O(n^4)$	143.0
bitvectors	148	?	?	$O(n^2)$	35.5
clevermmult	39	?	?	$O(n^2)$	220.0
duplicates	37	?	$O(n^2)$	$O(n^2)$	3.5
dyade	31	?	?	$O(n^2)$	1.0
eratosthenes	74	?	$O(n^3)$	$O(n^2)$	15.8
flatten	31	?	?	$O(n^2)$	25.0
insertionsort	36	?	$O(n^3)$	$O(n^2)$	6.7
listsort	56	?	?	$O(n^2)$	22.8
lcs	87	?	?	$O(n^2)$	24.5
matrix	74	?	?	$O(n^3)$	72.4
mergesort	35	?	?	$O(n^3)$	29.0
minsort	26	?	$O(n^3)$	$O(n^2)$	2.4
queue	35	?	?	$O(n^5)$	148.4
quicksort	46	?	?	$O(n^2)$	38.4
rationalPotential	14	$O(n)$	$O(n)$	$O(n)$	0.27
splitandsort	70	?	?	$O(n^3)$	61.4
subtrees	8	?	$O(n^2)$	$O(n^2)$	3.6
tuples	33	?	?	?	–