Complexity Analysis of Unfolding Graph Rewriting: The first step

Naohi Eguchi

Institute of Computer Science
University of Innsbruck
Austria

27 November 2013, Seminar 1

Introduction 1/2

Example (Recursion over free algebras)

General form of primitive recursion:

$$\begin{aligned} f(0, \vec{y}) &= g(\vec{y}) \\ f(c(x_1, \ldots, x_k), \vec{y}) &= h(\vec{x}, \vec{y}, f(x_1, \vec{y}), \ldots, f(x_k, \vec{y})) \end{aligned}$$

Fact

For any general ramified function f there exists $F \in \mathrm{FP}$ such that

$$F(\lceil \mathrm{graph}(c) \rceil) = \lceil \mathrm{graph}(f(c)) \rceil$$

for any value c over an underlying vocabulary.

General Ramified Recurrence is Sound for Polynomial Time
U. Dal Lago, S. Martini and M. Zorzi. Proc. DICE 2010.

Introduction 2/2

Example (Set-theoretic recursion)

Primitive recursion on (hereditarily) finite sets:

$$
\begin{aligned}
f(\emptyset, \vec{y}) &= g(\vec{y}) \\
f(\{x_1, \ldots, x_k\}, \vec{y}) &= h(\{\vec{x}\}, \vec{y}, \{f(x_1, \vec{y}), \ldots, f(x_k, \vec{y})\})
\end{aligned}
$$

Fact

For any $f \in \mathrm{PCSF}$ *there exists* $F \in \mathrm{FP}$ *such that*

$$F(\ulcorner G \urcorner) = \ulcorner \mathrm{graph}(f(\mathrm{set}(G))) \urcorner$$

for any directed acyclic graph G.

Predicatively Computable Functions on Sets
T. Arai. Submitted.

Outline

Fact (Dal Lago-Martini-Zorzi, 2010)

For any general ramified function f there exists $F \in \mathrm{FP}$ such that

$$F(\lceil \mathrm{graph}(c) \rceil) = \lceil \mathrm{graph}(f(c)) \rceil$$

for any value c over an underlying vocabulary.

- Unfolding graph rewriting is employed to show the fact.
- (As far as the speaker knows) no complexity analysis of unfolding rewriting is known.
- This talk: primitive recursive complexity analysis of unfolding graph rewriting.

Motivating example 1/2

Example (A rewrite system $\mathcal{R}$)

$$\begin{array}{llcllcl}(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) \to & h(f(x),f(x)) \\ (2) & f(0) & \to & 0 & (4) & f(c(x,y)) \to & h(f(x),f(y))\end{array}$$

The following (innermost) rewriting is possible:

$$\begin{array}{rcll} f(c(s^n(0),0)) & \to_{\mathcal{R}} & h(f(s^n(0)),f(0)) & \text{by (4)} \\ & \to_{\mathcal{R}} & h(f(s^n(0)),0) & \text{by (2)} \\ & \to_{\mathcal{R}} & h(h(f(s^{n-1}(0)),f(s^{n-1}(0)))) & \text{by (3)} \\ & \cdots & (2^{n-1} \text{ times application of (3)}) & \\ & \to_{\mathcal{R}} & h(h^n(0),0) & \text{by (3)} \\ & \cdots & (2^n \text{ times application of (1)}) & \\ & \to_{\mathcal{R}} & c(c^n(0),0) & \text{by (1)} \end{array}$$

where $h^0(0,0) = 0$ and $h^{k+1}(0) = h(h^k(0),h^k(0))$.

Motivating example 2/2

Example (A rewrite system $\mathcal{R}$)

$$
\begin{array}{llll}
(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) & \to & h(f(x),f(x)) \\
(2) & f(0) & \to & 0 & (4) & f(c(x,y)) & \to & h(f(x),f(y))
\end{array}
$$

Observe: the length of the rewriting sequence is $2 + 2^n + 2^n$:

$$f(c(s^n(0),0)) \to_{\mathcal{R}} \cdots \to_{\mathcal{R}} c(c^n(0),0)$$

Derivational complexity (or even the innermost runtime complexity) of $\mathcal{R}$ is optimally exponential.

Representing by an infinite rewrite system

An infinite rewrite system $\mathcal{R}^*$ unfolding $\mathcal{R}$:

$$
\begin{array}{rcll}
h(x,y) & \to & c(x,y) & (1) \\
f(0) & \to & 0 & (2) \\
f(s(0)) & \to & h(0,0) & (3.1) \\
f(s(s(0))) & \to & h(h(0,0)h(0,0)) & (3.2) \\
& \cdots & & \\
f(c(0,0)) & \to & h(0,0)) & (4.1) \\
f(c(s(0),0)) & \to & h(h(0,0),0) & (4.2) \\
f(c(c(0,0),0)) & \to & h(h(0,0),0) & (4.3) \\
& \cdots & &
\end{array}
$$

Observe:

- $\mathcal{R}^*$ collects all the possible instances of the rules (3) & (4).
- Derivational complexity of $\mathcal{R}^*$ is still exponential.
- Because terms of the form $h(\cdots,\cdots)$ are duplicated.

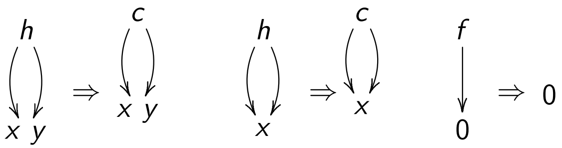
Unfolding graph rewriting 1/4

Example (A rewrite system $\mathcal{R}$)

$$
\begin{array}{llcl llcl}
(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) & \to & h(f(x),f(x)) \\
(2) & f(0) & \to & 0 & (4) & f(c(x,y)) & \to & h(f(x),f(y))
\end{array}
$$

The unfolding graph rewrite system $\mathcal{G}_{\mathcal{R}}$ contains:

(1.1) (1.2) (2)

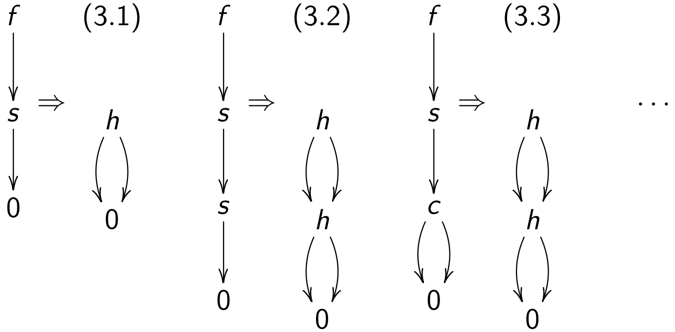


Unfolding graph rewriting 2/4

Example (A rewrite system $\mathcal{R}$)

$$\begin{array}{llcl@{\qquad}llcl}
(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) & \to & h(f(x),f(x)) \\
(2) & f(0) & \to & 0 & (4) & f(c(x,y)) & \to & h(f(x),f(y))
\end{array}$$

The unfolding graph rewrite system $\mathcal{G}_{\mathcal{R}}$ also contains:

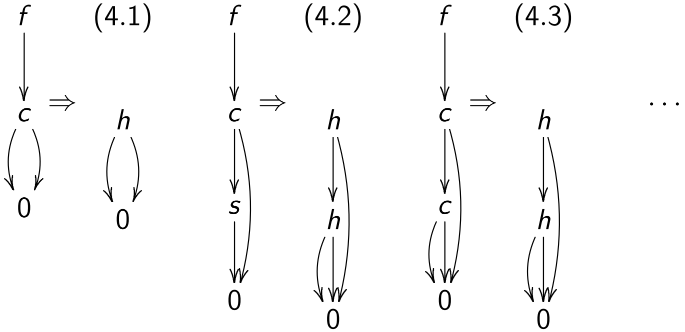


Unfolding graph rewriting 3/4

Example (A rewrite system $\mathcal{R}$)

$$
\begin{array}{llcl llcl}
(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) & \to & h(f(x),f(x)) \\
(2) & f(0) & \to & 0 & (4) & f(c(x,y)) & \to & h(f(x),f(y))
\end{array}
$$

The unfolding graph rewrite system $\mathcal{G}_{\mathcal{R}}$ finally contains:



Unfolding graph rewriting 4/4

Example (A rewrite system $\mathcal{R}$)

$$\begin{array}{llcllrcl}
(1) & h(x,y) & \to & c(x,y) & (3) & f(s(x)) & \to & h(f(x),f(x)) \\
(2) & f(0) & \to & 0 & (4) & f(c(x,y)) & \to & h(f(x),f(y))
\end{array}$$

The unfolding graph rewrite system $\mathcal{G}_{\mathcal{R}}$ is:

- an infinite graph rewrite system unfolding the rules (3) & (4). (More precisely, the rule (1) should be also unfolded)
- every term graph in the rules is maximally shared..

Observe:

The maximal length of (innermost) rewriting sequences in $\mathcal{G}_{\mathcal{R}}$ can be bounded by $O(n)$. (n: the size of a starting graph)

Term graphs 1/2

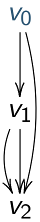
Definition (Term graphs)

Formally a term graph is a directed acyclic graph $G = (V, E)$ with a unique root root_G equipped with

- an attachment function $\mathrm{att}_G : V \to V^*$ and
- a labeling function $\mathrm{lab}_G : V \to \mathcal{F} \cup \mathcal{V}$.

$\mathrm{root}_G =$

$G =$

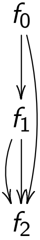


$\mathrm{att}_G(v_0) = \langle v_1, v_2 \rangle$

$\mathrm{att}_G(v_1) = \langle v_2, v_2 \rangle$

$\mathrm{att}_G(v_2) = \langle \rangle$

$H =$

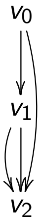


G is written as H if $\mathrm{lab}_G(v_j) = f_j$ for each $j = 0, 1, 2$.

Term graphs 2/2

Definition (Initial sub-graphs)

Let $G = (V, E)$ and $v \in V$. Then an initial sub-graph $G \upharpoonright v$ is the maximal sub-graph of G whose root is v.

$G =$ 

$G \upharpoonright v_0 = G$

$G \upharpoonright v_1 =$ 

$G \upharpoonright v_2 = v_2$

Complexity analysis 1/3: Reduction orders

Structure of unfolding rewrite systems is rather simple.
Hence reduction orders might help.

Definition

Suppose G, H are term graphs such that $\mathsf{lab}_G : V_G \to \mathcal{F} \cup \mathcal{V}$ and $\mathsf{lab}_H : V_H \to \mathcal{F} \cup \mathcal{V}$.
Let $>_{\mathcal{F}}$: a precedence on $\mathcal{F}$.
Suppose $\mathsf{att}_G(\mathsf{root}_G) = \langle u_1, \ldots, u_k \rangle$.
Then $G > H$ if one of the following two cases hold:

1. $G \upharpoonright u_i \geqslant H$ for some $i \in \{1, \ldots, k\}$.
2. The following three conditions are fulfilled.
    - $\mathsf{lab}_G(\mathsf{root}_G) >_{\mathcal{F}} \mathsf{lab}_H(\mathsf{root}_H)$.
    - $\mathsf{att}_H(\mathsf{root}_H) = \langle v_1, \ldots, v_l \rangle$ where $G > H \upharpoonright v_j$ for all $j \in \{1, \ldots, l\}$.
    - $lab_G(\mathsf{root}_G)$ does not appear in the image $\{f \in \mathcal{F} \mid \exists v \in V_H(\mathsf{lab}_H(v) = f)\}$ of lab_H.

Complexity analysis 2/3: Interpretation

Definition (Bounding primitive recursive functions)

Let d: a natural number such that $2 \leq d$.

$$\begin{array}{rcl} F_0(n) & = & d(1+n) \\ F_{m+1}(n) & = & F_m^{d(1+n)}(n) \end{array}$$

Recall: rank $\mathrm{rk} : \mathcal{F} \to \mathbb{N}$ is defined as $f >_{\mathcal{F}} g \Leftrightarrow \mathrm{rk}(f) > \mathrm{rk}(g)$.

Definition (Interpretation of term graphs)

Suppose $>_{\mathcal{F}}$: a precedence on $\mathcal{F}$.
Let G: (closed) term graph such that $\mathrm{att}_G(\mathrm{root}_G) = \langle v_1, \dots, v_k \rangle$.
Let $\mathrm{lab}_G(\mathrm{root}_G) = f \in \mathcal{F}$.

$$\mathcal{I}(G) := F_{\mathrm{rk}(f)}^{|G|}\left(\sum_{j=1}^{k} \mathcal{I}(G \upharpoonright v_j)\right) \qquad (|G| : \text{ the size of } G)$$

Complexity analysis 3/3: Interpretation theorem

Lemma (Main lemma)

Let G, H: term graphs over a signature $\mathcal{F}$ such that $|H| \leq d(1 + |G|)$.
Let σ: a substitution.
Suppose $\max\{\text{arity}(f) \mid f \in \mathcal{F}\} \leq d$.
If $G > H$, then, for the interpretation $\mathcal{I}$ induced by d, $\mathcal{I}(G\sigma) > \mathcal{I}(H\sigma)$.

Theorem (Interpretation theorem)

Let $\mathcal{G}$: a (possibly infinite) graph rewrite system over $\mathcal{F}$ such that $L > R$ for any $L \Rightarrow R \in \mathcal{G}$.
Suppose $|R| \leq d(1 + |L|)$ for any $L \Rightarrow R \in \mathcal{G}$.
Suppose $\max\{\text{arity}(f) \mid f \in \mathcal{F}\} \leq d$.
If $G \Rightarrow_{\mathcal{G}} H$, then $\mathcal{I}(G) > \mathcal{I}(H)$ for the interpretation induced by d.

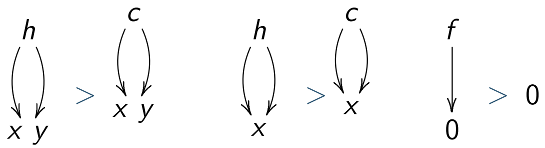
Application 1/5

Example (A rewrite system $\mathcal{R}$)

$$
\begin{array}{llll llll}
(1) & h(x,y) & \rightarrow & c(x,y) & (3) & f(s(x)) & \rightarrow & h(f(x),f(x)) \\
(2) & f(0) & \rightarrow & 0 & (4) & f(c(x,y)) & \rightarrow & h(f(x),f(y))
\end{array}
$$

Define a precedence by $f >_{\mathcal{F}} h >_{\mathcal{F}} c, s, 0$.

(1.1) (1.2) (2)

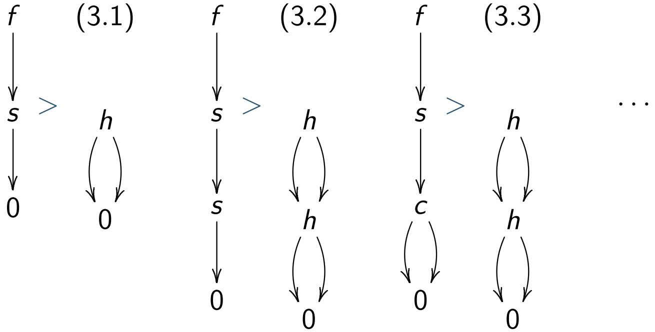


Application 2/5

Example (A rewrite system $\mathcal{R}$)

$$
\begin{array}{llcl llcl}
(1) & h(x,y) & \rightarrow & c(x,y) & (3) & f(s(x)) & \rightarrow & h(f(x),f(x)) \\
(2) & f(0) & \rightarrow & 0 & (4) & f(c(x,y)) & \rightarrow & h(f(x),f(y))
\end{array}
$$

Define a precedence by $f >_{\mathcal{F}} h >_{\mathcal{F}} c, s, 0$.

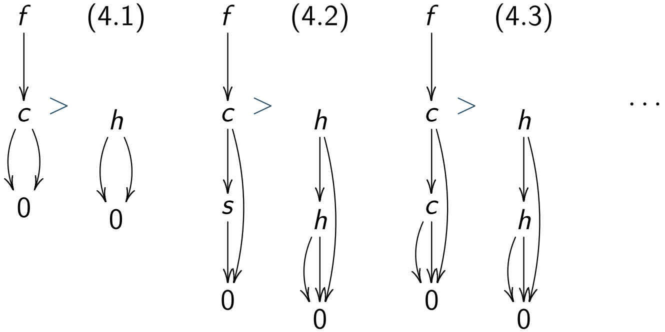


Application 3/5

Example (A rewrite system $\mathcal{R}$)

$$\begin{array}{llll} (1) & h(x,y) \to c(x,y) & (3) & f(s(x)) \to h(f(x), f(x)) \\ (2) & f(0) \to 0 & (4) & f(c(x,y)) \to h(f(x), f(y)) \end{array}$$

Define a precedence by $f >_{\mathcal{F}} h >_{\mathcal{F}} c, s, 0$.



Application 4/5

Theorem (Interpretation theorem)

Let $\mathcal{G}$: a (possibly infinite) graph rewrite system over $\mathcal{F}$ such that $L > R$ *for any* $L \Rightarrow R \in \mathcal{G}$.
Suppose $|R| \leq d(1 + |L|)$ *for any* $L \Rightarrow R \in \mathcal{G}$.
Suppose $\max\{\text{arity}(f) \mid f \in \mathcal{F}\} \leq d$.
If $G \Rightarrow_{\mathcal{G}} H$*, then* $\mathcal{I}(G) > \mathcal{I}(H)$ *for the interpretation induced by* d.

Observe:

1. $|R| \leq |L|$ for any $L \Rightarrow R \in \mathcal{G}_{\mathcal{R}}$.
2. $\max\{\text{arity}(g) \mid g \in \mathcal{F}\} \leq 2$.

By Interpretation theorem, the length of any graph rewriting sequence in $\mathcal{G}_{\mathcal{R}}$ starting with a closed G is bounded by $\mathcal{I}(G)$.

Application 5/5

Example:

$$G = \mathrm{graph}(f(c(s^n(0),0))) = f \longrightarrow c \longrightarrow s \longrightarrow \cdots s \longrightarrow 0$$

See:

1. $|G| = n + 3$.
2. $\mathrm{rk}(f) = 2$, $\mathrm{rk}(c) = \mathrm{rk}(s) = \mathrm{rk}(0) = 0$.

Can be shown: $\mathcal{I}(G) \leq F_3^2(|G|)$.
Unfortunately:

- The upper bound F_3^2 is not tight.
- Because $2^n \leq F_1(n)$, $\left.2^{\cdot^{\cdot^{2}}}\right\} n \leq F_2(n)$, ...

Conclusion

Summary:

- Graph representation is more appropriate than term representation to discuss about computational complexity of some functions over specific structures (set-theoretic functions, functions over free algebras, etc.).
- Duplication can be avoided by unfolding graph rewriting.
- This work: primitive recursive complexity analysis of unfolding graph rewriting.

Future work:

- Polynomial complexity analysis of unfolding graph rewriting.

References

Predicatively Computable Functions on Sets
Toshiyasu Arai
Submitted. Available at arXiv: 1204.558.

General Ramified Recurrence is Sound for Polynomial Time
Ugo Dal Lago, Simone Martini and Margherita Zorzi
Proc. DICE 2010.

Thank you for your attention!