

Termgraph Rewriting

Maria Schett

Supervisors: Martin Avanzini, Georg Moser

November 8, 2011

Outline

- 1 Rewriting
- 2 Adequacy
- 3 Demonstration
- 4 Evaluation

Rewriting

Functional Program \sim Rewrite System

data Nat = 0 | Suc Nat

$\mathcal{F} = \{\text{add}, \text{mul}, 0, \text{Suc}\}$

add 0 y = y

$\text{add}(0, y) \rightarrow y$

add (Suc x) y = Suc (add x y)

$\text{add}(\text{Suc}(x), y) \rightarrow \text{Suc}(\text{add}(x, y))$

mul 0 y = 0

$\text{mul}(0, y) \rightarrow 0$

mul (Suc x) y = add (mul x y) y

$\text{mul}(\text{Suc}(x), y) \rightarrow \text{add}((\text{mul}(x, y), y))$

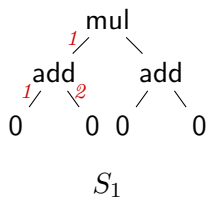
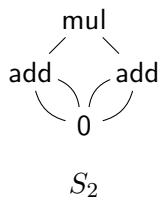
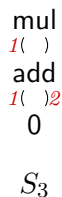
Computation \sim Rewrite Sequence

$\text{mul}(\underline{\text{add}(0, 0)}, \text{add}(0, 0)) \rightarrow$

$\underline{\text{mul}(0, \text{add}(0, 0))} \rightarrow$

0

Termgraphs

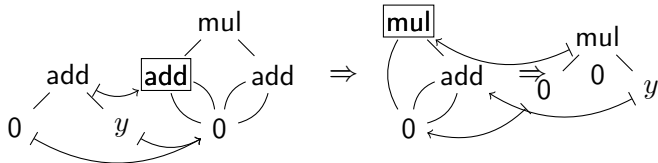
$$s = \text{mul}(\text{add}(0, 0), \text{add}(0, 0))$$

 \cong

 \cong


Termgraph Rewriting

Graph Rewrite System

$$\begin{array}{c} \text{add} \\ / \quad \backslash \\ 0 \quad y \end{array} \Rightarrow y$$

$$\begin{array}{c} \text{mul} \\ / \quad \backslash \\ 0 \quad y \end{array} \Rightarrow 0$$



Adequacy

Question

Can every outermost term rewrite step be simulated by an outermost termgraph rewrite step?

Definition

Termgraph rewriting is **adequate** to simulate term rewriting, if every normal form obtained by term rewriting can be obtained by termgraph rewriting.

Adequacy for Outermost Termgraph Rewriting

Accidental Parallel Rewriting

$$\text{dup}(x) \rightarrow f(x, x)$$

$$b \rightarrow a$$

$$f(a, b) \rightarrow a$$

$$\begin{array}{c} \text{dup} \\ | \\ x \end{array} \Rightarrow \begin{array}{c} f \\ (\\ x \end{array}$$

$$s = \underline{\text{dup}(b)} \xrightarrow{\circ} f(\underline{b}, b) \xrightarrow{\circ} \underline{f(a, b)} \xrightarrow{\circ} a = t$$

$$S = \begin{array}{c} \boxed{\text{dup}} \\ | \\ b \end{array} \xRightarrow{\circ} \begin{array}{c} f \\ (\\ \boxed{b} \end{array} \xRightarrow{\circ} \begin{array}{c} f \\ (\\ a \end{array} = T$$

Unique path to the redex is required.

Adequacy for Outermost Termgraph Rewriting

No Morphism Found

$$\text{comp}(F) \rightarrow T$$

$$\text{eq}(x, x) \rightarrow T$$

$$\begin{array}{c} \text{comp} \\ | \\ F \\ \text{eq} \\ () \\ x \end{array} \Rightarrow T$$

$$s = \text{eq}(\underline{\text{comp}(F)}, T) \xrightarrow{\circ} \underline{\text{eq}(T, T)} \xrightarrow{\circ} T = t$$



Observation

To fully simulate outermost term rewriting with termgraph rewriting sharing (\succcurlyeq) and unsharing (\preccurlyeq) are required.

Demonstration

ASCII Representation of Termgraphs

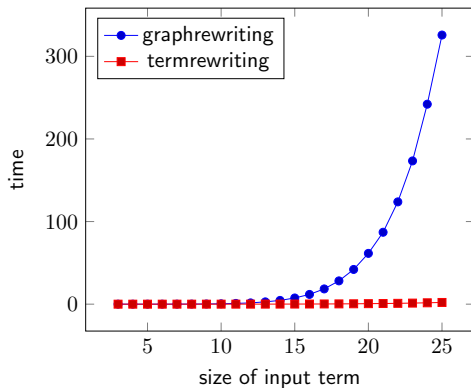
$f(>X2, >X2)$
where $X2 = b$

f
()
b

$f(b, b)$

f
/ \
b b

Evaluation of the Implementation



Summary

- ▶ term rewriting & termgraph rewriting
- ▶ adequacy and problems introduced by sharing
- ▶ investigation of outermost termgraph rewriting
- ▶ demonstration & evaluation of the implementation in Haskell

Thesis

- ▶ easy accessible reader on termgraph rewriting
- ▶ more on adequacy
- ▶ detailed investigation of the implementation