Termgraph Rewriting

Maria Schett

Supervisors: Martin Avanzini, Georg Moser

November 8, 2011
Outline

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

Functional Program ~ Rewrite System

\[
\text{data Nat} = 0 \mid \text{Suc Nat} \\
\text{add } 0 \ y = y \\
\text{add } (\text{Suc } x) \ y = \text{Suc } (\text{add } x \ y) \\
\text{mul } 0 \ y = 0 \\
\text{mul } (\text{Suc } x) \ y = \text{add } (\text{mul } x \ y) \ y
\]

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

\[
\begin{align*}
\text{add } 0 \ y & \rightarrow y \\
\text{add } (\text{Suc } x) \ y & \rightarrow \text{Suc } (\text{add } x \ y) \\
\text{mul } 0 \ y & \rightarrow 0 \\
\text{mul } (\text{Suc } x) \ y & \rightarrow \text{add } (\text{mul } x \ y) \ y
\end{align*}
\]

Computation ~ Rewrite Sequence

\[
\begin{align*}
\text{mul } (\text{add } (0, 0), \text{add } (0, 0)) & \rightarrow \\
\text{mul } (0, \text{add } (0, 0)) & \rightarrow \\
0
\end{align*}
\]
Termgraphs

\[ s = \text{mul}(\text{add}(0, 0), \text{add}(0, 0)) \]
Termgraph Rewriting

Graph Rewrite System

\[
\begin{align*}
\text{add} & \quad \Rightarrow \quad y \\
0 & \quad y & \quad \Rightarrow & \quad 0 \\
\text{mul} & \quad \Rightarrow \quad 0 \\
0 & \quad y
\end{align*}
\]
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

\[
\begin{align*}
\text{dup}(x) & \rightarrow f(x, x) \\
\text{b} & \rightarrow \text{a} \\
f(a, b) & \rightarrow \text{a}
\end{align*}
\]

\[
\begin{align*}
\text{dup} & \quad \Rightarrow \quad f \\
\text{x} & \quad \Rightarrow \quad (x) \\
f(a, b) & \rightarrow a = t
\end{align*}
\]

\[
S = \underbrace{\text{dup}(b)} \overset{o}{\rightarrow} f(b, b) \overset{o}{\rightarrow} f(a, b) \overset{o}{\rightarrow} \text{a} = t
\]

\[
S' = \begin{array}{c}
\text{dup} \\
\text{b}
\end{array} \overset{o}{\Rightarrow} \begin{array}{c}
f \\
\text{b}
\end{array} \overset{o}{\Rightarrow} \begin{array}{c}
f \\
\text{a}
\end{array} = T
\]

Unique path to the redex is required.
Adequacy for Outermost Termgraph Rewriting

No Morphism Found

\[ \text{comp}(F) \rightarrow T \]
\[ \text{eq}(x, x) \rightarrow T \]

\[ s = \text{eq}(\text{comp}(F), T) \Rightarrow \text{eq}(T, T) \Rightarrow T = t \]

\[ S = \text{comp} \quad \text{eq} \quad T \Rightarrow \text{eq} \quad T \quad \text{eq} \quad T \quad \text{eq} \quad x \]
To fully simulate outermost term rewriting with termgraph rewriting sharing ($\geq$) and unsharing ($\leq$) are required.
Demonstration

ASCII Representation of Termgraphs

\[ f(\text{\textgreater}X_2,\text{\textgreater}X_2) \]

where \( X_2 = b \)

\[ f(b,b) \]
Evaluation of the Implementation

![Graph and Table](image-url)
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