Lifting Böhm Trees from Terms to Term Graphs

Stefan Blom

Computational Logic
Institute of Computer Science
University of Innsbruck

21. 03. 2006

Outline

1. Böhm Semantics
2. Term Graph Rewriting
3. Lifting
4. Conclusion

Semantics of Term Rewriting Systems

- If a TRS is confluent and terminating then we may use normal forms.
- If a TRS is not terminating, it can still produce results. For example with the rule

  $$\mathrm{inf}(X) \to X : \mathrm{inf}(X)$$

  we have the infinite sequence

  $$\mathrm{inf}(1) \to 1 : \mathrm{inf}(1) \to 1 : 1 : \mathrm{inf}(1) \to \cdots$$

  which somehow yields

  $$1 : 1 : 1 : \cdots$$

  This result can be obtained with two theories
  - Infinitary Rewriting.
  - Böhm Semantics.

Infinite Terms

- Let $\mathcal{T}$ denote the set of terms.
- The order $\leq_\Omega$ is the transitive, reflexive, compatible closure of
$$\Omega \leq_\Omega t \in \mathcal{T}$$
- An ideal is a set I, which is
  - Downward closed: $\forall x, y : x \leq_\Omega y \wedge y \in I \Rightarrow x \in I$.
  - Directed: $\forall x, y \in I : \exists z \in I : x, y \leq_\Omega z$.
- The ideal completion of a partial order is the set of ideals order by inclusion.
- The set of infinite terms $\mathcal{T}^\infty$ is the ideal completion of $(\mathcal{T}, \leq_\Omega)$.

Direct approximation

- A function $\omega : \mathcal{T} \to \mathcal{T}^\infty$ is monotonic w.r.t. $R \subseteq \mathcal{T} \times \mathcal{T}$ if

$$\forall t_1, t_2 \in \mathcal{T} : \; t_1 \; R \; t_2 \implies \omega(t_1) \subseteq \omega(t_2)$$

- A direct approximation is a function $\omega : \mathcal{T} \to \mathcal{T}^\infty$, which is monotonic w.r.t. $\to$.
- For example

$$\begin{array}{rcl} \omega(x) & = & x \\ \omega(n) & = & n \\ \omega(\Omega) & = & \Omega \\ \omega(\mathrm{inf}(t)) & = & \Omega \\ \omega(t_1 : t_2) & = & \omega(t_1) : \omega(t_2) \end{array}$$

Böhm Semantics

- Define the *Böhm Semantics* $\mathrm{BT}_\omega : \mathcal{T} \to 2^{\mathcal{T}}$ as follows:

$$\mathrm{BT}_\omega(t) = \bigcup \{\omega(t') \mid t \twoheadrightarrow t'\}$$

- The Böhm Semantics is unique (UN_ω) if

$$\forall t_1, t_2 \in \mathcal{T} : \; t_1 = t_2 \implies \mathrm{BT}_\omega(t_1) = \mathrm{BT}_\omega(t_2)$$

  where $=$ denotes conversion.
- If the Böhm Semantics is unique then it is an infinite term:

$$\mathrm{UN}_\omega \implies \forall t \in \mathcal{T} : \; \mathrm{BT}_\omega(t) \in \mathcal{T}^\infty$$

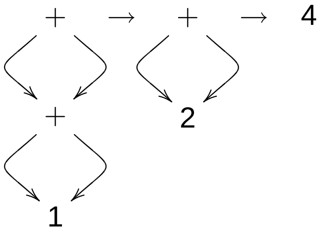
- A Böhm Tree is a unique Böhm Semantics.
- Confluence implies uniqueness:

$$\mathrm{CR} \implies \mathrm{UN}_\omega$$

Implementation of Term Rewriting Systems

$$(1+1)+(1+1) \to 2+(1+1) \to 2+2 \to 4$$

Needs 7 nodes, 6 edges and 3 steps.



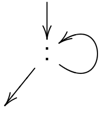
Needs 3 nodes, 4 edges and 2 steps.

$$\langle y+y \mid y=x+x, x=1\rangle \to \langle y+y \mid y=2\rangle \to 4$$

Cyclic Graphs

- Finite representation of infinite structures.

$1 : 1 : 1 : \cdots$ as



or $\langle x \mid x = 1 : x \rangle$

- Recursive functions (using lambda calculus).

$$\langle \mathrm{fib}\,10 \mid \mathrm{fib} = \lambda n.\mathrm{fib}\,(n-1) + \mathrm{fib}\,(n-2) \triangleleft n > 1 \triangleright 1 \rangle$$

where

$$x \triangleleft \mathrm{T} \triangleright y \to x \quad x \triangleleft \mathrm{F} \triangleright y \to y$$

Rewriting on Term Graphs

To implement term rewriting take a TRS and add

$$\begin{array}{lcl}
\langle C[x] \mid x = M, D\rangle & \xrightarrow[\text{es}]{} & \langle C[M] \mid x = M, D\rangle \\
\langle N \mid x = M, y = C[x], D\rangle & \xrightarrow[\text{is}]{} & \langle N \mid x = M, y = C[M], D\rangle \\
\langle N \mid x = C[x], D\rangle & \xrightarrow[\text{cs}]{} & \langle N \mid x = C[C[x]], D\rangle
\end{array}$$

$$\begin{array}{lcl}
\langle y + y \mid y = x + x, x = 1\rangle & \xrightarrow[\text{is}]{} & \langle y + y \mid y = 1 + x, x = 1\rangle \\
& \xrightarrow[\text{is}]{} & \langle y + y \mid y = 1 + 1, x = 1\rangle \\
& \xrightarrow[\text{R}]{} & \langle y + y \mid y = 2, x = 1\rangle \\
& \xrightarrow[\text{es}]{} & \langle 2 + y \mid y = 2, x = 1\rangle \\
& \xrightarrow[\text{es}]{} & \langle 2 + 2 \mid y = 2, x = 1\rangle \\
& \xrightarrow[\text{R}]{} & \langle 4 \mid y = 2, x = 1\rangle
\end{array}$$

Unwinding

- The set of cyclic terms $\mathcal{T}^\circ$ is given by

$$M ::= \langle M \mid x_1 = M_1, \cdots, x_n = M_n \rangle \mid f(M_1, \cdots, M_1) \mid x$$

- The unwinding $\mathrm{Unw}(M)$ of a cyclic term M is given as

$$\mathrm{Unw}(M) = \mathrm{BT}^{\overrightarrow{\mathrm{es}}}_{\mathrm{ext}}(M), \text{ where}$$

$$\begin{array}{lcl}
\langle C[x] \mid x = M, D \rangle & \xrightarrow[\mathrm{es}]{} & \langle C[M] \mid x = M, D \rangle \\
\mathrm{ext}(x) & = & x \\
\mathrm{ext}(f(M_1, \cdots, M_1)) & = & f(\mathrm{ext}(M_1), \cdots, \mathrm{ext}(M_1)) \\
\mathrm{ext}(\langle M \mid x_1 = M_1, \cdots, x_n = M_n \rangle) & = & \mathrm{ext}(M[x_1 := \Omega, \cdots, x_n :=
\end{array}$$

- The unwinding is unique because $\xrightarrow[\mathrm{es}]{}$ is confluent.

Problem

- Confluence is lost:

$$\begin{tikzcd}
\langle x \mid x = F(x) \rangle \arrow[r, "\mathrm{es}"] \arrow[d, "\mathrm{cs}"'] & \langle F(x) \mid x = F(x) \rangle \arrow[d, "\mathrm{cs}"'] \\
\langle x \mid x = F(F(x)) \rangle & \langle F(x) \mid x = F(F(x)) \rangle
\end{tikzcd}$$

- Adding rules solves the problem for orthogonal TRSs.
- Doesn’t work for the lambda calculus.

Problem

- Böhm Semantics proofs are easy for confluent rewrite systems.
- Graph rewriting systems are often non-confluent.
- Böhm Semantics proofs are tedious for confluent rewrite systems.

Solution

- Graph rewriting systems often consist of
  - Rules derived from a confluent TRS.
    For example
    $$\mathrm{seq}\, f\, x \to (f\, x) : (\mathrm{seq}\, f\, (f\, x))$$
    gets optimized into
    $$\mathrm{seq}\, f\, x \to \langle y : \mathrm{seq}\, g\, y \mid y = g\, x, g = f \rangle$$
  - Unwinding preserving rules.
- So we lift the the Böhm Tree for the TRS to the GRS, using infinite terms as a reference point.

Lifting a rewrite relation form terms to infinite terms

- A set $S' \subseteq S \in \mathcal{T}^\infty$ is a spanning set if

$$\forall s \in S : \; \exists s' \in S' : \; s \leq_\Omega s'$$

- We define $S[\to\rangle T$ if
  - There exists a spanning set for S, such that every element rewrites to an element in T.
  - There exists a spanning set for T, such that every element can be obtained by rewriting an element from S.
- which is equivalent to

$$\begin{array}{rl} & \forall s \in S : \exists s' \in S, t' \in T : s \leq_\Omega s' \wedge s' \to t' \\ \wedge & \forall t \in T : \exists s' \in S, t' \in T : s' \to t' \wedge t \leq_\Omega t' \end{array}$$

Example

$$A(A(X)) \rightarrow B(X)$$

$$\begin{array}{rcl}
\Omega & \twoheadrightarrow & \Omega \\
A(A(\Omega)) & \twoheadrightarrow & B(\Omega) \\
A(A(A(A(\Omega)))) & \twoheadrightarrow & B(B(\Omega)) \\
\vdots & & \vdots \\
A(A(A(\cdots))) & [\twoheadrightarrow\rangle & B(B(B(\cdots)))
\end{array}$$

Monotonicity of the Böhm Tree.

- A rewrite relation $\rightarrow$ is monotonic if

$$\forall s, s', t : \; s' \geq_\Omega s \rightarrow t \implies \exists t' : \; s' \rightarrow t' \geq_\Omega t$$

- If $\rightarrow$ is monotonic then a Böhm Tree BT_ω will be monotonic as well:

$$\forall s, t : \; s \leq_\Omega t \implies \mathrm{BT}_\omega(s) \subseteq \mathrm{BT}_\omega(t)$$

- For the remainder assume that $\rightarrow$ is monotonic and that BT_ω is a Böhm tree.

Lifting Böhm Trees from terms to infinite terms

- The direct extension is

$$\mathrm{BT}^{\infty}_{\omega}(S) = \cup\{\mathrm{BT}_{\omega}(s) \mid s \in S\}$$

- The derived extension uses the direct approximation

$$\omega^{\infty}(S) = \cup\{\omega(s) \mid s \in S\}$$

- The extensions are equivalent:

$$\mathrm{BT}^{\infty}_{\omega}(S) = \mathrm{BT}^{[\twoheadrightarrow\rangle}_{\omega^{\infty}}(S)$$

Cyclic extension

A cyclic extension $\xrightarrow[R^\circ]{} \subset \mathcal{T}^\circ \times \mathcal{T}^\circ$ of $\xrightarrow[R]{} \subset \mathcal{T} \times \mathcal{T}$ is

- Sound if

$$\forall M, N \in \mathcal{T}^\circ :\ M \xrightarrow[R^\circ]{} N \implies \mathrm{Unw}(M)[\underset{R}{\twoheadrightarrow}\rangle \mathrm{Unw}(N)$$

- Complete up to $\preceq$ if

$$\begin{array}{l} \forall M \in \mathcal{T}^\circ, s \in \mathrm{Unw}(M) :\ s \underset{R}{\twoheadrightarrow} t \implies \\ \quad \exists N \in \mathcal{T}^\circ, t' \in \mathrm{Unw}(N) :\ M \underset{R^\circ}{\twoheadrightarrow} N \wedge t \preceq t' \end{array}$$

- A typical choice for $\preceq$ is $\underset{R}{\twoheadrightarrow}$.
- For the remainder assume that $\xrightarrow[R^\circ]{}$ is a sound and complete up to $\underset{R}{\twoheadrightarrow}$ extension of $\xrightarrow[R]{}$.

Lifting Böhm Trees from terms to term graphs (1/2)

- The direct extension is

$$\mathrm{BT}^{\circ}_{\omega}(M) = \mathrm{BT}^{\infty}_{\omega}(\mathrm{Unw}(M))$$

- The derived extension uses the direct approximation

$$\omega^{\circ}(M) = \cup\{\omega(s) \mid s \in \mathrm{Unw}(M)\}$$

- The extensions are equivalent:

$$\mathrm{BT}^{\circ}_{\omega}(M) = \mathrm{BT}^{\overset{\twoheadrightarrow}{R^{\circ}}}_{\omega^{\circ}}(M)$$

Lifting Böhm Trees from terms to term graphs (2/2)

- Given a subset $\xrightarrow[R^s]{}$ of $\xrightarrow[R^\circ]{}$ and a direct approximation ω_s, such that

  $$\mathrm{BT}_{\omega_s}(M) = \mathrm{Unw}(M)$$

  define

  $$\omega_s^\circ(M) = \omega^\infty(\omega_s(M))$$

- If ω_s° is a direct approximation then

  $$\mathrm{BT}^{\underset{R^\circ}{\twoheadrightarrow}}_{\omega^\circ}(M) = \mathrm{BT}^{\underset{R^\circ}{\twoheadrightarrow}}_{\omega_s^\circ}(M)$$

Summary

- We have shown how to lift Böhm Trees from terms to graphs.
- We have not shown how to define Böhm Semantics.

Future work

Lifting Böhm Trees from dags to cyclic graphs.

- This is needed for modeling programming languages with side-effects.
- This is not yet included is current work:
  - Finite dags are CRS terms.

$$\text{let } x = M \text{ in } N \approx \mathrm{let}(M, \mathrm{in}([x](N)))$$

  - Infinite dags are not infinite CRS terms.

Example

$$\text{let } x_1 = M_1 \text{ in let } x_2 = M_2 \text{ in } \cdots f([x_1, x_2, \cdots], [x_1, x_2, \cdots])$$

is not a CRS term because we would need infinitely many abstractions above the f symbol.

- We need $f(\Omega) \leq \text{let } x = 1 \text{ in } f(x)$
- And also $d(f(\Omega), \text{let } x = 1 \text{ in } f(x)) = \frac{1}{2}$