Comparing Calculi of Explicit Substitutions with Eta-reduction

Mauricio Ayala Rincón, Flávio Leonardo Cavalcanti de Moura
Departamento de Matemática
Universidade de Brasília
Brasília - Brasil.

Fairouz Kamareddine
Mathematical and Computer Science
Heriot-Watt University
Edinburgh, Scotland.
Summary

1. $\lambda$-calculus and Explicit Substitutions Calculi.

2. The *Eta* rule for the Suspension Calculus.

3. Adequacy: Comparing $\lambda\sigma$, $\lambda_{se}$ and $\lambda_{SUSP}$.

4. Conclusions and Future Work.
1. \(\lambda\)-calculus and Explicit Substitutions Calculi

1.1. \(\lambda\)-calculus.

Application: \((M \; N)\) 

Abstraction: \(\lambda x. M\)

\((\beta)\) \(\frac{}{(\lambda x. M) \; N} \rightarrow M \{N/x\}\)

\((\eta)\) \(\frac{}{\lambda x. (M \; x) \rightarrow M \text{ if } x \not\in \text{FV}(M)}\)

Terms

\[ M, N ::= n | x | \lambda M | (M \; N) \]
1.2. Explicit Substitutions Calculi

Variations of the $\lambda$-calculus that manipulate explicitly the substitution operation.

Desireable properties:

(a) Simulation of the $\beta$-reduction.
(b) Termination or Strong Normalization (SN).
(c) Confluence (CR): ground terms and open terms.
(d) Preservation of Termination (PSN).
2.1. $\lambda\sigma$-calculus.

The first calculus of explicit substitutions.

**Terms**

$$ M, N ::= 1 \mid \lambda \mid \lambda M \mid (M \ N) \mid M[S] $$

**Substitutions**

$$ S, T ::= \text{id} \mid \uparrow \mid M.S \mid S \circ T $$
2.2. $\lambda s_e$-calculus


- Extension of the $\lambda s$ (Kamareddine and Ríos, 1995).
- Remains closer to the syntactical structure of the $\lambda$-calculus.

**Terms**

$$M, N ::= n | \mathcal{X} | \lambda M | (M N) | M\sigma^i N | \varphi^i_k M$$

where $k \geq 0$ and $i \geq 1$. 
2.3. **Suspension Calculus**  
Nadathur and Wilson, 1998.

**Motivation:** Implementational questions related with \(\lambda\)-Prolog that uses typed \(\lambda\)-terms as data structure.

Suspended terms
\[
M, N ::= n | \lambda M | (M N) | \langle M, i, j, e_1 \rangle
\]

Environments
\[
e_1, e_2 ::= nil | et :: e_1 | \{e_1, i, j, e_2\}
\]

Environment Terms
\[
et ::= @i | (M, i) | \langle et, i, j, e_1 \rangle
\]
<table>
<thead>
<tr>
<th>Properties</th>
<th>$\lambda\sigma$</th>
<th>$\lambda_{se}$</th>
<th>Susp. Calc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation of $\beta$-reduction</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Termination of substitution</td>
<td>yes</td>
<td>?</td>
<td>yes for WF</td>
</tr>
<tr>
<td>Confluence</td>
<td>$M_{\nu}$</td>
<td>yes</td>
<td>yes for WF</td>
</tr>
<tr>
<td>PSN</td>
<td>No</td>
<td>No</td>
<td>?</td>
</tr>
</tbody>
</table>

$M_{\nu}$ : Confluence on semi-open expressions, i.e., only with meta-variables of terms.

WF : Well-formed terms.
2. The $\textit{Eta}$ rule for the Suspension Calculus. 
($\lambda_{\text{Susp}}$)

\[
(Eta_{\text{SUSP}}) \ (\lambda \ (t_1 \ 1)) \rightarrow t_2, \quad \text{if} \quad t_1 =_{rm} \ [t_2, 0, 1, nil]
\]

Figura 1: The $\textit{Eta}$ rule for the $\lambda_{\text{Susp}}$

\begin{prop}[Soundness of the $\textit{Eta}$ rule]
Every application of the $\textit{Eta}$ rule of $\lambda_{\text{Susp}}$ to the redex $\lambda(t_1 \ 1)$ gives effectively the term $t_2$ obtained from $t_1$ by decrementing all its de Bruijn free indices by one.
\end{prop}
Lemma 4.3 [susp plus \( Eta \) is SN]
\( \text{SUSP} \cup \{ Eta \} \) is terminating.

Lemma 4.5 [Local-confluence of susp plus \( Eta \)]
\( \text{SUSP} \cup \{ Eta \} \) is locally-confluent.

Theorem 4.6 [Confluence of susp plus \( Eta \)]
\( \text{SUSP} \cup \{ Eta \} \) is confluent.
3. Comparing the Adequacy of the Calculi

**Definition 5.1 (Adequacy)** \( \lambda \xi_1 \prec \lambda \xi_2 \) if,

- \( \forall a \rightarrow_\beta b \)
  \( \forall a \rightarrow_\lambda^{\xi_2} b \iff \exists a \rightarrow_\lambda^{\xi_1} b \) such that \( m \leq n \);

- \( \exists a \rightarrow_\beta b \)
  \( \exists a \rightarrow_\lambda^{\xi_1} b \) such that \( \forall a \rightarrow_\lambda^{\xi_2} b \) we have \( m < n \).

If neither \( \lambda \xi_1 \prec \lambda \xi_2 \) nor \( \lambda \xi_2 \prec \lambda \xi_1 \), then we say that \( \lambda \xi_1 \) and \( \lambda \xi_2 \) are *non comparable*. 
**Proposition 5.2** The $\lambda\sigma$- and the $\lambda{s}_e$-calculi are non-comparable.

**Proposition 5.6** The $\lambda\sigma$- and $\lambda_{susp}$-calculi are non-comparable.
Proposition 5.11 Every $\textit{SUSP}$-derivation of

$$[[A, k, k - 1, \mathtt{\@} k - 2 :: \ldots :: \mathtt{\@} 0 :: (B, l) :: \text{nil}]]$$

where $A, B \in \Lambda$ and $k \geq 0$ to its $\textit{SUSP}$-nf has length greater than or equal to $Q_k(A, B)$.

Proposition 5.12 Let $A, B \in \Lambda$ and $k \geq 1$. Every $\textit{se}$-derivation of $A\sigma^kB$ to its $\textit{se}$-nf has length less than or equal to $Q_k(A, B)$. 
For the second condition in the definition of adequacy, consider $\beta$-conversion: $(\lambda 2) \ 1 \rightarrow \beta 1$

$\lambda_{SUSP}: (\lambda 2) \ 1 \rightarrow \ [2, 1, 0, (1, 0) :: nil] \rightarrow \ [1, 0, 0, nil] \rightarrow 1$

$\lambda_{S_e}: (\lambda 2) \ 1 \rightarrow \ 2\sigma^1 1 \rightarrow 1$

or a more complex example: $(\lambda 3 3)(\lambda 2 3) \rightarrow \beta \lambda^4 5$ which have a unique simulation in $\lambda_{S_e}$ with 7 steps against 11 steps in a unique simulation in the $\lambda_{SUSP}$.

**Theorem 5.13** $[\lambda_{S_e} \prec \lambda_{SUSP}]$ The $\lambda_{S_e}$- is more adequate than the $\lambda_{SUSP}$-calculus.
**Definition 5.14 [Efficiency]** $\lambda x_1 \ll \lambda x_2$ if,

$$\forall a \rightarrow_\beta b,$$

$$\forall a \rightarrow^m_{\lambda x_1} b \text{ and } \forall a \rightarrow^n_{\lambda x_2} b \implies m \leq n;$$

**Proposition 5.15 [\(\lambda x_e \ll \lambda x_{\text{susp}}\)]** The $\lambda x_e$-calculus is more efficient than the $\lambda x_{\text{susp}}$-calculus.
Compare the simulations of $\beta$-reduction from the term $(\lambda(\lambda^n i))_j$, where $n \geq 0$:

$$(\lambda(\lambda^n i))_j \rightarrow$$

$$(\lambda(\lambda^n i))_j \rightarrow$$
Compare the simulations of $\beta$-reduction from the term
\[(\lambda(\lambda^n i)) \_ j, \text{ where } n \geq 0:\]
\[(\lambda(\lambda^n i)) \_ j \rightarrow \]
\[(\lambda^n i)\sigma^1 j \rightarrow^n\]

\[(\lambda(\lambda^n i)) \_ j \rightarrow \]
\|[\_ \lambda^n i, 1, 0, (j, 0) :: nil] \rightarrow^n\]
Compare the simulations of $\beta$-reduction from the term $(\lambda(\lambda^ni))\ j$, where $n \geq 0$:

$$(\lambda(\lambda^ni))\ j \rightarrow$$

$$(\lambda^n i)^{\sigma^1 j} \rightarrow^n$$

$$\lambda^n (i^{\sigma^{n+1} j}) =: t_1$$

$$(\lambda(\lambda^ni))\ j \rightarrow$$

$$[\lambda^n i, 1, 0, (j, 0) :: nil] \rightarrow^n$$

$$\lambda^n [i, n + 1, n, @n - 1 :: \ldots :: @0 :: (j, 0) :: nil] =: t_2$$
After that the $\lambda s_{e}$ completes the simulation in one or two steps by checking arithmetic inequations:

$$t_1 \rightarrow \begin{cases} 
\lambda^n \underline{i}, & \text{if } i < n + 1 \\
\lambda^n \underline{i - 1}, & \text{if } i > n + 1 \\
\lambda^n (\varphi_0^{n+1} j) \rightarrow \lambda^n \underline{j + n}, & \text{if } i = n + 1
\end{cases}$$

But in the $\lambda_{susp}$ we have to destruct the environment list, environment by environment:

$$t_2 \begin{cases} 
\rightarrow^{i-1} \lambda^n \underline{[1, n - i + 2, n, @n - i : \ldots : @0 :: (j, 0) :: nil]} \rightarrow \lambda^n \underline{i}, & \text{if } i < n + 1 \\
\rightarrow^{n+1} \lambda^n \underline{[i - n - 1, 0, n, nil]} \rightarrow \lambda^n \underline{i - 1}, & \text{if } i > n + 1 \\
\rightarrow^{i-1} \lambda^n \underline{[1, 1, n, (j, 0) :: nil]} \rightarrow \lambda^n \underline{[j, 0, n, nil]} \rightarrow \lambda^n \underline{j + n}, & \text{if } i = n + 1
\end{cases}$$

- Enlarged Suspension Calculus with an adequate Eta-rule (Soundness, Termination and Confluence).

- $\lambda\sigma$ and $\lambda_{Se}$ are non comparable.

- $\lambda\sigma$ and $\lambda_{SUSP}$ are non comparable.

- $\lambda_{Se}$ is more efficient than $\lambda_{SUSP}$. 
5. Further Work.


⇒ An implementation of the 3 Explicit Substitution Calculi with Eta-reduction (Ocaml)

Group of Theory of Computation
Universidade de Brasília
http://www.mat.unb.br/~ayala/TCgroup

The Useful Logics, Types, Rewriting and Applications (ULTRA)
http://www.cee.hw.ac.uk/ultra
6. Future Work.

- Is $\lambda_{\text{SUSP}}$ PSN?
- Is $s_e$-calculus terminating?
Main references


