Is the s_e calculus strongly normalising?

Fairouz Kamareddine
Joint work with
Alejandro Ríos
`http://www.macs.hw.ac.uk/~fairouz/talks/talks2001/termination01.ps`

May 2001

The $\lambda\sigma$-calculus

Terms	$\Lambda\sigma^t ::= 1 \mid \Lambda\sigma^t\Lambda\sigma^t \mid \lambda\Lambda\sigma^t \mid \Lambda\sigma^t[\Lambda\sigma^s]$
Substitutions	$\Lambda\sigma^s ::= id \mid \uparrow \mid \Lambda\sigma^t \cdot \Lambda\sigma^s \mid \Lambda\sigma^s \circ \Lambda\sigma^s$

(Beta)	$(\lambda a)\, b$	$\longrightarrow$	$a\,[b \cdot id]$
(VarId)	$\mathtt{1}\,[id]$	$\longrightarrow$	$\mathtt{1}$
(VarCons)	$\mathtt{1}\,[a \cdot s]$	$\longrightarrow$	a
(App)	$(a\, b)[s]$	$\longrightarrow$	$(a\,[s])\,(b\,[s])$
(Abs)	$(\lambda a)[s]$	$\longrightarrow$	$\lambda(a\,[\mathtt{1} \cdot (s \circ \uparrow)])$
(Clos)	$(a\,[s])[t]$	$\longrightarrow$	$a\,[s \circ t]$
(IdL)	$id \circ s$	$\longrightarrow$	s
(ShiftId)	$\uparrow \circ id$	$\longrightarrow$	$\uparrow$
(ShiftCons)	$\uparrow \circ (a \cdot s)$	$\longrightarrow$	s
(Map)	$(a \cdot s) \circ t$	$\longrightarrow$	$a\,[t] \cdot (s \circ t)$
(Ass)	$(s_1 \circ s_2) \circ s_3$	$\longrightarrow$	$s_1 \circ (s_2 \circ s_3)$

We can code $\mathtt{n}$ by the term $\mathtt{1}[\uparrow^{n-1}]$.

The $\lambda\upsilon$-rules

$$\Lambda\upsilon^t ::= I\!N \mid \Lambda\upsilon^t\Lambda\upsilon^t \mid \lambda\Lambda\upsilon^t \mid \Lambda\upsilon^t[\Lambda\upsilon^s]$$
$$\Lambda\upsilon^s ::= \uparrow \mid \Uparrow(\Lambda\upsilon^s) \mid \Lambda\upsilon^t.$$

(Beta)	$(\lambda a)\, b$	$\longrightarrow$	$a\,[b/]$
(App)	$(a\, b)[s]$	$\longrightarrow$	$(a\,[s])\,(b\,[s])$
(Abs)	$(\lambda a)[s]$	$\longrightarrow$	$\lambda(a\,[\Uparrow(s)])$
(FVar)	$\mathtt{1}\,[a/]$	$\longrightarrow$	a
(RVar)	$\mathtt{n}+\mathtt{1}\,[a/]$	$\longrightarrow$	$\mathtt{n}$
(FVarLift)	$\mathtt{1}\,[\Uparrow(s)]$	$\longrightarrow$	$\mathtt{1}$
(RVarLift)	$\mathtt{n}+\mathtt{1}\,[\Uparrow(s)]$	$\longrightarrow$	$\mathtt{n}\,[s]\,[\uparrow]$
(VarShift)	$\mathtt{n}\,[\uparrow]$	$\longrightarrow$	$\mathtt{n}+\mathtt{1}$

The $\lambda\sigma_{\Uparrow}$-rules

$$\Lambda\sigma^{t}_{\Uparrow} ::= I\!N \mid \Lambda\sigma^{t}_{\Uparrow}\Lambda\sigma^{t}_{\Uparrow} \mid \lambda\Lambda\sigma^{t}_{\Uparrow} \mid \Lambda\sigma^{t}_{\Uparrow}[\Lambda\sigma^{s}_{\Uparrow}]$$

$$\Lambda\sigma^{s}_{\Uparrow} ::= id \mid \uparrow \mid \Uparrow(\Lambda\sigma^{s}_{\Uparrow}) \mid \Lambda\sigma^{t}_{\Uparrow} \cdot \Lambda\sigma^{s}_{\Uparrow} \mid \Lambda\sigma^{s}_{\Uparrow} \circ \Lambda\sigma^{s}_{\Uparrow}.$$

(Beta)	$(\lambda a)\, b$	$\longrightarrow$	$a\,[b \cdot id]$
(App)	$(a\, b)[s]$	$\longrightarrow$	$(a\,[s])\,(b\,[s])$
(Abs)	$(\lambda a)[s]$	$\longrightarrow$	$\lambda(a\,[\Uparrow(s)])$
(Clos)	$(a\,[s])[t]$	$\longrightarrow$	$a\,[s \circ t]$
(Varshift1)	$\mathtt{n}\,[\uparrow]$	$\longrightarrow$	$\mathtt{n}+\mathtt{1}$
(Varshift2)	$\mathtt{n}\,[\uparrow \circ s]$	$\longrightarrow$	$\mathtt{n}+\mathtt{1}\,[s]$
(FVarCons)	$\mathtt{1}\,[a \cdot s]$	$\longrightarrow$	a
(RVarCons)	$\mathtt{n}+\mathtt{1}\,[a \cdot s]$	$\longrightarrow$	$\mathtt{n}\,[s]$
(FVarLift1)	$\mathtt{1}\,[\Uparrow(s)]$	$\longrightarrow$	$\mathtt{1}$
(FVarLift2)	$\mathtt{1}\,[\Uparrow(s) \circ t]$	$\longrightarrow$	$\mathtt{1}\,[t]$

(RVarLift1)	$\mathtt{n}+\mathtt{1}\,[\Uparrow(s)]$	$\longrightarrow$	$\mathtt{n}[s \circ \uparrow]$
(RVarLift2)	$\mathtt{n}+\mathtt{1}\,[\Uparrow(s) \circ t]$	$\longrightarrow$	$\mathtt{n}[s \circ (\uparrow \circ t)]$
(Map)	$(a \cdot s) \circ t$	$\longrightarrow$	$a\,[t] \cdot (s \circ t)$
(Ass)	$(s \circ t) \circ u$	$\longrightarrow$	$s \circ (t \circ u)$
(ShiftCons)	$\uparrow \circ (a \cdot s)$	$\longrightarrow$	s
(ShiftLift1)	$\uparrow \circ \Uparrow(s)$	$\longrightarrow$	$s \circ \uparrow$
(ShiftLift2)	$\uparrow \circ (\Uparrow(s) \circ t)$	$\longrightarrow$	$s \circ (\uparrow \circ t)$
(Lift1)	$\Uparrow(s) \circ \Uparrow(t)$	$\longrightarrow$	$\Uparrow(s \circ t)$
(Lift2)	$\Uparrow(s) \circ (\Uparrow(t) \circ u)$	$\longrightarrow$	$\Uparrow(s \circ t) \circ u$
(LiftEnv)	$\Uparrow(s) \circ (a \cdot t)$	$\longrightarrow$	$a \cdot (s \circ t)$
(IdL)	$id \circ s$	$\longrightarrow$	s
(IdR)	$s \circ id$	$\longrightarrow$	s
(LiftId)	$\Uparrow(id)$	$\longrightarrow$	id
(Id)	$a\,[id]$	$\longrightarrow$	a

Lambda calculus with de Bruijn indices

- $\Lambda ::= \mathbb{N} \mid (\Lambda\Lambda) \mid (\lambda\Lambda)$ $\qquad\qquad$ $(\lambda A)\, B \to_\beta A\{\!\{1 \leftarrow B\}\!\}$

- *meta-updatings* $U_k^i : \Lambda \to \Lambda$ for $k \geq 0$ and $i \geq 1$:

  $U_k^i(AB) \equiv U_k^i(A)\, U_k^i(B)$ $\qquad\qquad$ $U_k^i(\lambda A) \equiv \lambda(U_{k+1}^i(A))$

  $$U_k^i(\mathtt{n}) \equiv \begin{cases} \mathtt{n} + \mathtt{i} - 1 & \text{if} \quad n > k \\ \mathtt{n} & \text{if} \quad n \leq k\,. \end{cases}$$

- *meta-substitutions* at level $i \geq 1$, of a term $B \in \Lambda$ in a term $A \in \Lambda$:

$$
\begin{aligned}
(A_1 A_2)\{\!\{\mathtt{i} \leftarrow B\}\!\} &\equiv (A_1\{\!\{\mathtt{i} \leftarrow B\}\!\})\,(A_2\{\!\{\mathtt{i} \leftarrow B\}\!\}) \\
(\lambda A)\{\!\{\mathtt{i} \leftarrow B\}\!\} &\equiv \lambda(A\{\!\{\mathtt{i}+\mathtt{1} \leftarrow B\}\!\}) \\
\mathtt{n}\{\!\{\mathtt{i} \leftarrow B\}\!\} &\equiv \begin{cases} \mathtt{n}-\mathtt{1} & \text{if} \quad n > i \\ U_0^i(B) & \text{if} \quad n = i \\ \mathtt{n} & \text{if} \quad n < i\,. \end{cases}
\end{aligned}
$$

- **Lemma 1.**

  - $$\begin{array}{lcll} U^i_k(A)\{\!\{\mathtt{n}\leftarrow B\}\!\} & \equiv & U^{i-1}_k(A) & \textit{if } k<n<k+i \\ U^i_k(A)\{\!\{\mathtt{n}\leftarrow B\}\!\} & \equiv & U^i_k(A\{\!\{\mathtt{n}-\mathtt{i}+\mathtt{1}\leftarrow B\}\!\}) & \textit{if } k+i<n \end{array}$$

  - $$\begin{array}{lcll} U^i_k(U^j_p(A)) & \equiv & U^{j+i-1}_p(A) & \textit{if } p\leq k<j+p \\ U^i_k(U^j_p(A)) & \equiv & U^j_p(U^i_{k+1-j}(A)) & \textit{if } j+p\leq k+1 \end{array}$$

  – Meta-substitution lemma *For* $1\leq i\leq n$ *we have:*
  $A\{\!\{\mathtt{i}\leftarrow B\}\!\}\{\!\{\mathtt{n}\leftarrow C\}\!\}\equiv A\{\!\{\mathtt{n}+\mathtt{1}\leftarrow C\}\!\}\{\!\{\mathtt{i}\leftarrow B\{\!\{\mathtt{n}-\mathtt{i}+\mathtt{1}\leftarrow C\}\!\}\}\!\}$.

  – Distribution lemma
  For $n\leq k+1$ *we have:* $U^i_k(A\{\!\{\mathtt{n}\leftarrow B\}\!\})\equiv U^i_{k+1}(A)\{\!\{\mathtt{n}\leftarrow U^i_{k-n+1}(B)\}\!\}$.

The λs-calculus

$$\Lambda s ::= I\!N \mid \Lambda s \Lambda s \mid \lambda \Lambda s \mid \Lambda s \, \sigma^j \Lambda s \mid \varphi_k^i \Lambda s \qquad where \quad j, \, i \geq 1 \, , \quad k \geq 0 \, .$$

σ-*generation*	$(\lambda a)\, b$	$\longrightarrow$	$a\, \sigma^1\, b$
σ-λ-*transition*	$(\lambda a)\, \sigma^j b$	$\longrightarrow$	$\lambda(a \sigma^{j+1} b)$
σ-*app-transition*	$(a_1\, a_2)\, \sigma^j b$	$\longrightarrow$	$(a_1\, \sigma^j b)\, (a_2\, \sigma^j b)$
σ-*destruction*	$\mathtt{n}\, \sigma^j b$	$\longrightarrow$	$\begin{cases} \mathtt{n-1} & \text{if } n > j \\ \varphi_0^j\, b & \text{if } n = j \\ \mathtt{n} & \text{if } n < j \end{cases}$
φ-λ-*transition*	$\varphi_k^i(\lambda a)$	$\longrightarrow$	$\lambda(\varphi_{k+1}^i\, a)$
φ-*app-transition*	$\varphi_k^i(a_1\, a_2)$	$\longrightarrow$	$(\varphi_k^i\, a_1)\, (\varphi_k^i\, a_2)$
φ-*destruction*	$\varphi_k^i\, \mathtt{n}$	$\longrightarrow$	$\begin{cases} \mathtt{n+i-1} & \text{if } n > k \\ \mathtt{n} & \text{if } n \leq k \end{cases}$

The extra rules of the λs_e-calculus

- $\Lambda s_{op} ::= \mathbf{V} \mid \mathbb{N} \mid \Lambda s_{op} \Lambda s_{op} \mid \lambda \Lambda s_{op} \mid \Lambda s_{op}\, \sigma^j \Lambda s_{op} \mid \varphi_k^i \Lambda s_{op}$

- *Loss of confluence*

$$(X\sigma^1 Y)\sigma^1 1 \leftarrow ((\lambda X)Y)\sigma^1 1 \rightarrow ((\lambda X)\sigma^1 1)(Y\sigma^1 1)$$

$(X\sigma^1 Y)\sigma^1 1$ and $((\lambda X)\sigma^1 1)(Y\sigma^1 1)$ *have no common reduct*

σ-σ-*transition*	$(a\,\sigma^i b)\,\sigma^j\,c$	$\longrightarrow$	$(a\,\sigma^{j+1}\,c)\;\sigma^i\,(b\,\sigma^{j-i+1}\,c)$	if	$i \leq j$
σ-φ-*transition 1*	$(\varphi^i_k\,a)\,\sigma^j\,b$	$\longrightarrow$	$\varphi^{i-1}_k\,a$	if	$k < j < k+i$
σ-φ-*transition 2*	$(\varphi^i_k\,a)\,\sigma^j\,b$	$\longrightarrow$	$\varphi^i_k(a\,\sigma^{j-i+1}\,b)$	if	$k+i \leq j$
φ-σ-*transition*	$\varphi^i_k(a\,\sigma^j\,b)$	$\longrightarrow$	$(\varphi^i_{k+1}\,a)\,\sigma^j\,(\varphi^i_{k+1-j}\,b)$	if	$j \leq k+1$
φ-φ-*transition 1*	$\varphi^i_k\,(\varphi^j_l\,a)$	$\longrightarrow$	$\varphi^j_l\,(\varphi^i_{k+1-j}\,a)$	if	$l+j \leq k$
φ-φ-*transition 2*	$\varphi^i_k\,(\varphi^j_l\,a)$	$\longrightarrow$	$\varphi^{j+i-1}_l\,a$	if	$l \leq k < l+j$

- For every $\xi \in \{\sigma, \sigma_{\Uparrow}, \upsilon, s\}$, ξ is SN and $\lambda\xi$ is confluent [1, 4, 9, 10, 6, 2] on closed terms.

- Only $\lambda\sigma_{\Uparrow}$ and the λs_e are confluent on open terms [3, 7]

- Only $\lambda\upsilon$ and λs have Preservation of Strong Normalisation (PSN) [6, 2]

- λs has an extension λs_e [7] which is confluent on open terms, but $\lambda\upsilon$ does not.

- Is s_e Strongly Normalising? We know s_e Weakly Normalising [7].

- We have fully proof checked the proof of SN of σ in ALF, we have investigated different termination techniques, but are still unable to show SN of s_e.

Techniques for showing the σ-calculus is SN

There are various proofs of this theorem in the literature:

1. The first is based on the strong normalisation of $SUBST$ [5], which is, within CCL, the set of rewriting rules that compute the substitutions.

2. The proof in [4] shows the termination of σ via a strict translation from σ to another calculus σ_0 (an economic variant of σ) and the termination of σ_0.

3. Zantema gives two proofs in [9, 10]. The first is based on a suitable generalisation of polynomial orders to show the termination of the calculus σ_0 (and hence the termination of σ). The second uses semantic labelling to show the termination of σ.

These techniques cannot be applied for s_e

- **Problem 1: Unable to use recursive path ordering** By taking a look at the s_e-rules, it becomes obvious that the unfriendly rules, with respect to SN, are σ-σ-$transition$ and to a lesser extent φ-σ-$transition$. These rules prevent us from finding an order on the set of operators in order to solve the normalisation problem with a recursive path ordering (rpo).

- **Problem 2: Unable to use Zantema's distribution elimination lemma.** The s_e-rules "look like" associative rules but unfortunately they are not; e.g. in σ-σ-$transition$ one could think the σ^j-operator distributes over the σ^i-operator, but it is not a "true" distribution: σ^j changes to σ^{j+1} when acting on the first term and to σ^{j-i+1} when acting on the second. This prevents use of Zantema's distribution elimination method [9] to show SN.

Can we apply modularity?

- Another technique to show SN is modularity where SN is proved for certain subcalculi os s_e which are shown to satisfy a commutation property.

- s_e can be divided into two subcalculi which are SN.

  $*\varphi = \{\sigma\text{-}\varphi\text{-tr.1}, \sigma\text{-}\varphi\text{-tr.2}, \varphi\text{-}\varphi\text{-tr.1}, \varphi\text{-}\varphi\text{-tr.2}\}$,
  $*\sigma = \{\sigma\text{-}\sigma\text{-tr.}, \varphi\text{-}\sigma\text{-tr.}\}$,
  $*\varphi^{-} = \{\sigma\text{-}\varphi\text{-tr.1}, \varphi\text{-}\varphi\text{-tr.2}\}$, $*\varphi^{--} = \{\sigma\text{-}\varphi\text{-tr.2}, \varphi\text{-}\varphi\text{-tr.1}\}$.

  Note that $s_e = (s + *\varphi) + *\sigma$.

- Unfortunately, the needed commutation results do not hold.

SN of $s + *\varphi$

- We prove that $s + *\varphi$ is SN by giving a weight that decreases by reduction.
- Let $P : \Lambda s_{op} \to I\!N$ and $W : \Lambda s_{op} \to I\!N$ be defined by:

$$\begin{array}{ll} P(X) = P(\mathtt{n}) = 2 & W(X) = W(\mathtt{n}) = 1 \\ P(a\,b) = P(a) + P(b) & W(a\,b) = W(a) + W(b) + 1 \\ P(\lambda a) = P(a) & W(\lambda a) = W(a) + 1 \\ P(a\,\sigma^j b) = j * P(a) * P(b) & W(a\,\sigma^j b) = 2 * W(a) * (W(b) + 1) \\ P(\varphi^i_k a) = (k+1) * (P(a) + 1) & W(\varphi^i_k a) = 2 * W(a) \end{array}$$

- $(W(a), P(a))$ decreases with the lexicographical order for each $s + *\varphi$-reduction.

The $\lambda\omega$- and $\lambda\omega_e$-calculi

- In order to establish SN of $*\sigma$, we will use an isomorphism established in [8] between λs_e and $\lambda\omega_e$, a calculus written in the $\lambda\sigma$-style.

$$\begin{array}{ll} \textbf{Terms} & \Lambda\omega^t ::= I\!N \mid \Lambda\omega^t\Lambda\omega^t \mid \lambda\Lambda\omega^t \mid \Lambda\omega^t[\Lambda\omega^s]_j \\ \textbf{Substitutions} & \Lambda\omega^s ::= \uparrow^i \mid \Lambda\omega^t/ \end{array}$$

σ-*generation*	$(\lambda a)\, b$	$\longrightarrow$	$a\,[b/]_1$
σ-*app-transition*	$(a\, b)[s]_j$	$\longrightarrow$	$(a\,[s]_j)\,(b\,[s]_j)$
σ-λ-*transition*	$(\lambda a)[s]_j$	$\longrightarrow$	$\lambda(a[s]_{j+1})$
σ-/-*destruction*	$\mathtt{n}[a/]_j$	$\longrightarrow$	$\begin{cases} \mathtt{n}-\mathtt{1} & \text{if } n > j \\ a[\uparrow^{j-1}]_1 & \text{if } n = j \\ \mathtt{n} & \text{if } n < j \end{cases}$
σ-$\uparrow$-*destruction*	$\mathtt{n}[\uparrow^i]_j$	$\longrightarrow$	$\begin{cases} \mathtt{n}+\mathtt{i} & \text{if } n \geq j \\ \mathtt{n} & \text{if } n < j \end{cases}$

Figure 1: The $\lambda\omega$-calculus

$$\begin{array}{ll} \textbf{Open Terms} & \Lambda\omega^t_{op} ::= \mathbf{V} \mid I\!N \mid \Lambda\omega^t_{op}\Lambda\omega^t_{op} \mid \lambda\Lambda\omega^t_{op} \mid \Lambda\omega^t_{op}[\Lambda\omega^s_{op}]_j \\ \textbf{Substitutions} & \Lambda\omega^s_{op} ::= \uparrow^i \mid \Lambda\omega^t_{op}/ \end{array}$$

$$\begin{array}{|lll|} \hline \sigma\text{-}/\text{-}transition & a\,[b/]_k[s]_j & \longrightarrow \quad a\,[s]_{j+1}[b[s]_{j-k+1}/]_k \quad \text{if} \quad k \leq j \\ /\text{-}\uparrow\text{-}transition & a\,[\uparrow^i]_k[b/]_j & \longrightarrow \begin{cases} a[b/]_{j-i}[\uparrow^i]_k & \text{if} \quad k+i \leq j \\ a[\uparrow^{i-1}]_k & \text{if} \quad k \leq j < k+i \end{cases} \\ \uparrow\text{-}\uparrow\text{-}transition & a\,[\uparrow^i]_k[\uparrow^l]_j & \longrightarrow \begin{cases} a[\uparrow^l]_{j-i}[\uparrow^i]_k & \text{if} \quad k+i < j \\ a[\uparrow^{i+l}]_k & \text{if} \quad k \leq j \leq k+i \end{cases} \\ \hline \end{array}$$

Figure 2: The new rules of the $\lambda\omega_e$-calculus

Properties of $\lambda\omega$ and $\lambda\omega_e$

1. The ω-calculus is SN and confluent on $\Lambda\omega^t$.
2. Let $a, b \in \Lambda$. If $a \twoheadrightarrow_{\lambda\omega} b$ then $a \twoheadrightarrow_\beta b$. If $a \to_\beta b$ then $a \twoheadrightarrow_{\lambda\omega} b$.
3. The $\lambda\omega$-calculus is confluent on $\Lambda\omega^t$.
4. Pure terms which are SN in the λ-calculus are also SN in the $\lambda\omega$-calculus.
5. The ω_e-calculus is weakly normalising and confluent.
6. The $\lambda\omega_e$-calculus is confluent on open terms.

7. Let $a, b \in \Lambda$. If $a \twoheadrightarrow_{\lambda\omega_e} b$ then $a \twoheadrightarrow_{\beta} b$. If $a \to_{\beta} b$ then $a \twoheadrightarrow_{\lambda\omega_e} b$.

SN for $*\sigma$

- To prove SN for $*\sigma$ we use the above isomorphism and the technique that Zantema used to prove SN for the calculus whose only rule is σ-σ-$transition$.

- Following this isomorphism, the schemes σ-σ-$tr.$ and φ-σ-$tr.$ of λs_e both translate into the same scheme of $\lambda\omega_e$, namely σ-/-$transition$.

- Hence, to show that $*\sigma$ is SN, it is enough to show that the calculus whose only rule is σ-/-$transition$, let us call it σ-/-calculus, is SN.

  1. The σ-/-calculus is weakly normalising.
  2. The σ-/-calculus is locally confluent.
  3. The σ-/-calculus is increasing.

Commutation does not hold

- $*\sigma$ does not commute over $s + *\varphi$.

  Let $k+i \leq j$, $h \leq j-i+1$ and $h > k+1$. Now take the following derivation: $(\varphi_k^i(a\,\sigma^h b))\,\sigma^j c \to_{*\varphi} \varphi_k^i((a\,\sigma^h b)\,\sigma^{j-i+1}c) \to_{\sigma-\sigma-tr} \varphi_k^i((a\,\sigma^{j-i+2}c)\,\sigma^h(b\,\sigma^{j-i-h+2}c))$
  It is easy to see that $(\varphi_k^i(a\,\sigma^h b))\,\sigma^j c$ does not contain any $*\sigma$-redex.

- $s + *\varphi$ does not commute over $*\sigma$.

  Let $i \leq j$ and take the derivation: $((\lambda a)\,\sigma^i b)\,\sigma^j c) \to_{\sigma-\sigma-tr} ((\lambda a)\,\sigma^{j+1}c)\,\sigma^i(b\,\sigma^{j-i+1}c) \to_s (\lambda(a\,\sigma^{j+2}c))\,\sigma^i(b\,\sigma^{j-i+1}c)$.

  Reducing the only s-redex in $((\lambda a)\,\sigma^i b)\,\sigma^j c)$ we get $(\lambda(a\,\sigma^{i+1}b))\,\sigma^j c$ which also has a unique s-redex. Reducing it we get $\lambda((a\,\sigma^{i+1}b)\,\sigma^{j+1}c)$ and now there

is only the σ-σ-$transition$ redex which gives us $\lambda((a\,\sigma^{j+2}c)\sigma^{i+1}(b\,\sigma^{j-i+1}c))$ which has no further redexes. Therefore, $(\lambda(a\,\sigma^{j+2}c))\,\sigma^{i}(b\,\sigma^{j-i+1}c)$ cannot be reached from $((\lambda a)\,\sigma^{i}b)\,\sigma^{i}c)$ with an s_e-derivation beginning with an s-step.

Conclusions

Does the s_e-calculus TERMINATE???

References

[1] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit Substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991.

[2] Z. Benaissa, D. Briaud, P. Lescanne, and J. Rouyer-Degli. $\lambda\upsilon$, a calculus of explicit substitutions which preserves strong normalisation. *Functional Programming*, 6(5), 1996.

[3] P-L Curien, T. Hardin, and J-J Lévy. Confluence Properties of weak and strong calculi of explicit substitutions. *Journal of the ACM*, 43, 362-397, 1996.

[4] P-L Curien, T. Hardin, and A. Ríos. Strong normalisation of substitutions. *Logic and Computation*, 6:799–817, 1996.

[5] T. Hardin and A. Laville. Proof of Termination of the Rewriting System SUBST on CCL. *Theoretical Computer Science*, 46:305–312, 1986.

[6] F. Kamareddine and A. Ríos. A λ-calculus à la de Bruijn with explicit substitutions. Proceedings of PLILP'95. *LNCS*, 982:45–62, 1995.

[7] F. Kamareddine and A. Ríos. Extending a λ-calculus with explicit substitution which preserves strong normalisation into a confluent calculus on open terms. *Journal of Functional Programming*, 7(4):395–420, 1997.

[8] F. Kamareddine and A. Ríos. Relating the $\lambda\sigma$- and λs-styles of explicit substitutions. *Logic and Computation*, 10(3):349–380, 2000.

[9] H. Zantema. Termination of term rewriting: interpretation and type elimination. *J. Symbolic Computation*, 17(1):23–50, 1994.

[10] H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995.