The $\lambda\sigma$-Calculus and Strong Normalization

Anders Schack-Nielsen

IT University Technical Report Series **TR-2010-132**

ISSN 1600–6100 **October 2010**

ISSN 1600–6100

ISBN 9788779492257

Copies may be obtained by contacting:

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S
Denmark

Telephone:	**+45 72 18 50 00**
Telefax:	**+45 72 18 50 01**
Web	**`www.itu.dk`**

The $\lambda\sigma$-Calculus and Strong Normalization

Anders Schack-Nielsen

October, 2010

Abstract

The general $\lambda\sigma$-calculus is not strongly normalizing [Mel95], but we show how a small restriction in the reduction rules allows us to prove strong normalization.

The proof has been formalized in the proof assistant Abella.

1 Introduction

The $\lambda\sigma$-calculus has a lot of nice properties. It allows us to represent the intermediate stages of β-reduction in the λ-calculus, thereby allowing an easy implementation of lazy β-reduction with the added benefit that we can compose substitutions and reduce the number of passes required to evaluate several β-reduction steps. Additionally, an explicit substitution calculus such as $\lambda\sigma$ is useful for reasoning about and implementing higher-order unification.

However, the calculus also has some theoretical short-comings. The system does not preserve strong normalization, i.e. a strongly normalizing λ-term might not be strongly normalizing as a $\lambda\sigma$-term [Mel95].

In this report we show how a small restriction in the reduction rules allows us to prove strong normalization.

The entire development is formalized in the proof assistant Abella, and each theorem is marked with the corresponding Abella file and theorem name. The Abella source files can be found at `http://www.itu.dk/people/anderssn/exsub-sn.tgz`.

2 The $\lambda\sigma$-calculus

The syntax of the $\lambda\sigma$-calculus consists of terms and substitutions written in de Bruijn notation:

Terms:	$M, N ::= 1 \mid M[s] \mid \lambda M \mid M\ N$
Substitutions:	$s, t ::= \mathsf{id} \mid \uparrow \mid M \,.\, s \mid s \circ t$

The variable 1 refers to the innermost λ-binder. Other variables are represented with a closure and a sequence of shifts, e.g. $3 = 1[\uparrow \circ \uparrow]$.

The intuition behind each of the substitution constructs is the following: A term under an identity is supposed to reduce to itself. A shift applied to a term increments all freely occurring variables by one. An extension $M \,.\, s$ will substitute M for the variable 1, decrement all other freely occurring variables, and then apply s to them. Finally a composition of two substitutions $s \circ t$ represents the substitution that first applies s and then t, i.e. $M[s \circ t]$ is supposed to reduce to the same term as reducing each closure individually in $M[s][t]$.

We will use $\uparrow^n$ where $n \geq 0$ as a short-hand for n compositions of shift, i.e. $\uparrow \circ (\uparrow \circ (\ldots \circ (\uparrow \circ \uparrow) \ldots))$, where $\uparrow^0$ means id. Additionally, de Bruijn indices n with $n > 1$ are short-hand for $1[\uparrow^{n-1}]$.

The reduction rules are shown in Figure 1. In addition to these rules we are also going to include every possible congruence rule, i.e. we allow rewrites to occur anywhere inside a term or substitution.

The rule **beta** corresponds to the ordinary β-step in λ-calculus and introduces an explicit substitution inside a closure. The rest of the rules are called σ-rules and details how to evaluate closures and substitution compositions. The fragment of the rules that excludes the **beta** rule is called the σ-fragment and the corresponding relation is written $\rightarrow_\sigma$.

beta	$(\lambda M)\ N$	$\to$	$M[N\ .\ \mathsf{id}]$
clos-var	$1[M\ .\ s]$	$\to$	M
clos-clos	$M[s][t]$	$\to$	$M[s \circ t]$
clos-lam	$(\lambda M)[s]$	$\to$	$\lambda(M[1\ .\ (s \circ \uparrow)])$
clos-app	$(M\ N)[s]$	$\to$	$M[s]\ N[s]$
clos-var-id	$1[\mathsf{id}]$	$\to$	1
comp-id-L	$\mathsf{id} \circ s$	$\to$	s
comp-shift-id	$\uparrow \circ\ \mathsf{id}$	$\to$	$\uparrow$
comp-shift	$\uparrow \circ\ (M\ .\ s)$	$\to$	s
comp-cons	$(M\ .\ s) \circ t$	$\to$	$M[t]\ .\ (s \circ t)$
comp-comp	$(s_1 \circ s_2) \circ s_3$	$\to$	$s_1 \circ (s_2 \circ s_3)$

Figure 1: Basic reduction rules

2.1 Variations of $\lambda\sigma$

Several variations of $\lambda\sigma$ have been treated in the literature. The most important distinction is whether or not we include term and substitution meta-variables. In the presence of meta-variables the system presented above is not even locally confluent, since the critical pair $((\lambda M)N)[s]$ can reduce to both $M[N[s]\ .\ s]$ and $M[N[s]\ .\ (s \circ \mathsf{id})]$. Closing the critical pairs leads to the following four additional rules:

comp-id-R	$s \circ \mathsf{id}$	$\to$	s
clos-id	$M[\mathsf{id}]$	$\to$	M
var-shift	$1\ .\uparrow$	$\to$	id
s-cons	$1[s]\ .\ (\uparrow \circ\ s)$	$\to$	s

The first two rules are the general right-identity rules and thus replace **clos-var-id** and **comp-shift-id**. The **var-shift** and **s-cons** rules can be seen as η-contraction rules on substitutions. The resulting system is easily seen to be locally confluent by checking all the critical pairs.

Constants can be added to the calculus with just a single additional rule:

clos-const $\quad c[s] \quad \to \quad c$

We can also represent de Bruijn indices and sequences of shifts directly in the syntax with the following five rules:

clos-var-dot2	$(n+1)[M\ .\ s]$	$\to$	$n[s]$
clos-var-shift	$n[\uparrow^m]$	$\to$	$n+m$
comp-shift-dot	$\uparrow^{n+1} \circ\ (M\ .\ s)$	$\to$	$\uparrow^n \circ\ s$
comp-shift-shift	$\uparrow^n \circ \uparrow^m$	$\to$	$\uparrow^{n+m}$
eta-subst	$(n+1)\ .\uparrow^{n+1}$	$\to$	$\uparrow^n$

These rules are essentially shortcuts in the sense that the first four can be obtained by a sequence of the reduction steps shown in Figure 1 when n and $\uparrow^n$ are syntactic short-hands. The last rule generalizes **var-shift** and corresponds to **s-cons** in the case when $s = \uparrow^n$.

We will not discuss meta-variables in this report. For the remainder of this report we will therefore define $\lambda\sigma$ to be:

Terms:	$M, N ::= n \mid c \mid M[s] \mid \lambda M \mid M\ N$
Substitutions:	$s, t ::= \uparrow^n \mid M\ .\ s \mid s \circ t$

where de Bruijn indices n have $n \geq 1$ and shifts $\uparrow^n$ have $n \geq 0$. The reduction rules are given in Figure 2. We have excluded the two general right-identity rules, as we can easily prove them admissible for the transitive closure:

Theorem 2.1 (`beta.thm:clos_id_ext`). *For all M and s we have $M[\uparrow^0] \to^* M$ and $s \circ \uparrow^0 \to^* s$.*

Proof. The proof is an easy mutual induction over M and s. □

We are not going to say much about the congruence rules, so for now we will just assume that we can perform any reduction step anywhere within a term or substitution. We will return to this matter below in section 4.

beta	$(\lambda M)\ N$	$\rightarrow$	$M[N\ .\ \mathsf{id}]$
clos-const	$c[s]$	$\rightarrow$	c
clos-var-dot1	$1[M\ .\ s]$	$\rightarrow$	M
clos-var-dot2	$(n+1)[M\ .\ s]$	$\rightarrow$	$n[s]$
clos-var-shift	$n[\uparrow^m]$	$\rightarrow$	$n+m$
clos-clos	$M[s][t]$	$\rightarrow$	$M[s \circ t]$
clos-lam	$(\lambda M)[s]$	$\rightarrow$	$\lambda(M[1\ .\ (s \circ \uparrow^1)])$
clos-app	$(M\ N)[s]$	$\rightarrow$	$M[s]\ N[s]$
comp-id-L	$\uparrow^0 \circ s$	$\rightarrow$	s
comp-cons	$(M\ .\ s) \circ t$	$\rightarrow$	$M[t]\ .\ (s \circ t)$
comp-shift-dot	$\uparrow^{n+1} \circ (M\ .\ s)$	$\rightarrow$	$\uparrow^n \circ s$
comp-shift-shift	$\uparrow^n \circ \uparrow^m$	$\rightarrow$	$\uparrow^{n+m}$
comp-comp	$(s_1 \circ s_2) \circ s_3$	$\rightarrow$	$s_1 \circ (s_2 \circ s_3)$
eta-subst	$(n+1)\ .\ \uparrow^{n+1}$	$\rightarrow$	$\uparrow^n$

Figure 2: Reduction rules

3 A non-terminating example

Mellies showed in [Mel95] that the simply typed term

$$\lambda v.(\lambda x.(\lambda y.y)((\lambda z.z)x))((\lambda w.w)v)$$

has an infinite reduction sequence in $\lambda\sigma$. This is very counter-intuitive since the ordinary simply typed λ-calculus is strongly normalizing and the σ-fragment of $\lambda\sigma$ is also strongly normalizing [CHR92] (even on untyped terms).

We will sketch the idea of the counter-example here. Consider a β-redex under a closure:

$$((\lambda M)\ N)[s]$$

If we evaluate the redex first and then the substitution composition we arrive at $M[N[s].s]$. If we instead begin by pushing the substitution through the application we can get the following reduction sequence:

$$\begin{aligned} ((\lambda M)\ N)[s] &\rightarrow (\lambda M)[s]\ N[s] \\ &\rightarrow (\lambda M[1\ .\ s \circ \uparrow])\ N[s] \\ &\rightarrow M[1\ .\ s \circ \uparrow^1][N[s]\ .\ \uparrow^0] \\ &\rightarrow^* M[N[s]\ .\ s \circ (\uparrow^1 \circ (N[s]\ .\ \uparrow^0))] \end{aligned}$$

Here we see a substitution $s' = \uparrow^1 \circ (N[s]\ .\ \uparrow^0)$, which contains s, being applied to s itself. Of course s' can reduce in one step to $\uparrow^0$, but if we carefully avoid that specific reduction step and if s contains a β-redex then we can push s' into s and through the redex in s and thus replicate the situation above with s' instead of s. Now since s' contains s and therefore also a β-redex we can keep on doing this (see [Mel95] for all the details).

The term that we end up creating in this way consists of a sequence of closures nested arbitrarily deep:

$$M[\ldots M[\ldots M[\ldots M[\ldots]\ldots]\ldots]\ldots]$$

And consequently the reduction steps that we perform are similarly nested deeper and deeper through an arbitrary number of closures and substitutions.

This also highlights the brittle nature of the counter-example. If we at any time were to reduce any of the arising s's to $\uparrow^0$ the entire thing would normalize. Thus we have an indication that a suitable minor tweak to the reduction strategy will give us strong normalization (as opposed to the current non-deterministic, everything-is-allowed reduction strategy).

4 Revisiting the congruence rules

Let us present the congruence rules that up until now have remained implicit:

$$\frac{M \to M'}{M\,.\,s \to M'\,.\,s} \qquad \frac{s \to s'}{M\,.\,s \to M\,.\,s'} \qquad \frac{s \to s'}{s \circ t \to s' \circ t} \qquad \frac{t \to t'}{s \circ t \to s \circ t'}$$

$$\frac{M \to M'}{\lambda M \to \lambda M'} \qquad \frac{M \to M'}{M\ N \to M'\ N} \qquad \frac{N \to N'}{M\ N \to M\ N'}$$

$$\frac{M \to M'}{M[s] \to M'[s]} \qquad \frac{s \to s'}{M[s] \to M[s']}\,(*)$$

If we get rid of the second congruence rule for closures $(*)$ then certainly we will have a strongly normalizing calculus, but this is overly restrictive. If we instead replace it by a version that only allows σ-steps then since the σ-fragment by itself is strongly normalizing, we can hope for strong normalization. We are therefore going to use the following congruence rule in place of $(*)$:

$$\frac{s \to_\sigma s'}{M[s] \to_\sigma M[s']}$$

The resulting rewrite system is indeed strongly normalizing for simply typed terms as we will see below.

5 Strong normalization of $\lambda\sigma$

The proof of strong normalization can be summarized to the following: Take any reduction sequence and divide it into groups of σ-steps and applications of **beta**. Since the σ-fragment is strongly normalizing we can focus on the **beta** steps. If we relate each term to its σ-normal form then we can show that every **beta** step corresponds to a regular β-reduction step in the λ-calculus, and we therefore get preservation of strong normalization.

The complete proof is formalized in the proof assistant Abella. Below are the central theorems along with references to the Abella source files.

5.1 Strong normalization of the σ-fragment

In [CHR92] the σ-fragment of $\lambda\sigma$ was proved strongly normalizing. The central part of the proof consists of showing that if e is strongly normalizing then $e[\uparrow^1]$ is also strongly normalizing, where e is an expression (defined below). The proof we give in this section share the same overall structure, but our proof of the strong normalization of $e[\uparrow^1]$ given strong normalization of e is greatly simplified (Lemma 5.5 and Lemma 5.6 below).

In order to give a simpler proof of strong normalization of the σ-fragment, we collapse the syntactic classes of terms and substitutions into a single class called expressions:

$$\textbf{Expressions:} \qquad e ::= n \mid c \mid \uparrow^n \mid e_1[e_2] \mid \lambda e \mid e_1\,.\,e_2$$

The set of reduction rules for expressions is given in Figure 3. We use the following translation from terms and substitutions into expressions:

$$\mathcal{E}(n) = n \qquad \mathcal{E}(c) = c \qquad \mathcal{E}(M[s]) = \mathcal{E}(M)[\mathcal{E}(s)]$$

$$\mathcal{E}(\lambda M) = \lambda\mathcal{E}(M) \qquad \mathcal{E}(M\ N) = \mathcal{E}(M)\,.\,\mathcal{E}(N) \qquad \mathcal{E}(\uparrow^n) = \uparrow^n$$

$$\mathcal{E}(M\,.\,s) = \mathcal{E}(M)\,.\,\mathcal{E}(s) \qquad \mathcal{E}(s \circ t) = \mathcal{E}(s)[\mathcal{E}(t)]$$

It is easy to see that σ-reductions are preserved by the translation, and therefore strong normalization of expressions implies strong normalization of the σ-fragment.

We shall write $\mathcal{SN}$ for the set of strongly normalizing expressions, and then aim to prove $e \in \mathcal{SN}$ for all expressions e.

Strong normalization of the cases λe and $e_1\,.\,e_2$ are fairly easy.

$$\begin{array}{llllll}
c[e] & \to & c & 1[e_1 \,.\, e_2] & \to & e_1 \\
(n+1)[e_1 \,.\, e_2] & \to & n[e_2] & n[\uparrow^m] & \to & n+m \\
e_1[e_2][e_3] & \to & e_1[e_2[e_3]] & (\lambda e_1)[e_2] & \to & \lambda e_1[1 \,.\, e_2[\uparrow^1]] \\
(e_1 \,.\, e_2)[e_3] & \to & e_1[e_3] \,.\, e_2[e_3] & \uparrow^0[e] & \to & e \\
\uparrow^{n+1}[e_1 \,.\, e_2] & \to & \uparrow^n[e_2] & \uparrow^n[\uparrow^m] & \to & \uparrow^{n+m} \\
(n+1) \,.\, \uparrow^{n+1} & \to & \uparrow^n & & &
\end{array}$$

$$\frac{e_1 \to e_1'}{e_1[e_2] \to e_1'[e_2]} \qquad \frac{e_2 \to e_2'}{e_1[e_2] \to e_1[e_2']} \qquad \frac{e \to e'}{\lambda e \to \lambda e'}$$

$$\frac{e_1 \to e_1'}{e_1 \,.\, e_2 \to e_1' \,.\, e_2} \qquad \frac{e_2 \to e_2'}{e_1 \,.\, e_2 \to e_1 \,.\, e_2'}$$

Figure 3: Expression reduction rules

Lemma 5.1 (`sigma-strong.thm:esn_lam`). *If $e \in \mathcal{SN}$ then $\lambda e \in \mathcal{SN}$.*

Lemma 5.2 (`sigma-strong.thm:esn_dot,esn_dot_inv`). *$e_1 \,.\, e_2 \in \mathcal{SN}$ if and only if $e_1 \in \mathcal{SN}$ and $e_2 \in \mathcal{SN}$.*

Closures are a lot more difficult due to the rewrite $(\lambda e_1)[e_2] \to \lambda e_1[1 \,.\, e_2[\uparrow^1]]$. We will therefore split the proof depending on whether the given expressions contain λs. We write the exclusion of λs from an expression e as $\lambda \notin e$.[1]

Lemma 5.3 (`sigma-strong.thm:esn_clos_nolam`). *If $e_1 \in \mathcal{SN}$, $e_2 \in \mathcal{SN}$, and $\lambda \notin e_1$ then $e_1[e_2] \in \mathcal{SN}$.*

These lemmas can be put together to prove strong normalization in the absence of λs:

Theorem 5.4 (`sigma-strong.thm:nolam_esn`). *If $\lambda \notin e$ then $e \in \mathcal{SN}$.*

In order to prove the general statement we are going to need a version of Lemma 5.3 without the restriction on e_1. The tricky part is proving that $e \in \mathcal{SN}$ implies $e[\uparrow^1] \in \mathcal{SN}$.

If we do an intuitive comparison between reduction sequences for e and $e[\uparrow^1]$ it seems that any reduction in $e[\uparrow^1]$ either can be mimicked by a reduction in e or consists of pushing the $\uparrow^1$ through e. We will formalize this intuitive idea by building a relation $e \overset{\lhd}{\sim} e'$, such that $e \overset{\lhd}{\sim} e[\uparrow^1]$ for any e, and whenever $e_1' \to e_2'$ with $e_1 \overset{\lhd}{\sim} e_1'$ then either $e_1 \overset{\lhd}{\sim} e_2'$ or $e_1 \to e_2$ with $e_2 \overset{\lhd}{\sim} e_2'$. In the former case the reduction $e_1' \to e_2'$ is intended to be one of the steps associated with pushing the $\uparrow^1$ through the structure of the expression and thus the number of such steps should be bounded by the structure of the expression. Given such a relation we would get the desired lemma as a corollary to the fact that $e \in \mathcal{SN}$ and $e \overset{\lhd}{\sim} e'$ implies $e' \in \mathcal{SN}$.

In order to build the relation such that it is closed under the conditions stated above, we need it to contain $e \overset{\lhd}{\sim} e[e']$ where e' can be $\uparrow^1$ and is closed under at least $1 \,.\, \cdot \circ \uparrow^1$ and reduction. Instead of characterizing this class of expressions we can simply take expressions without λ as we already know this class to be in $\mathcal{SN}$.

The relation is defined as follows:

$$\frac{}{c \overset{\lhd}{\sim} c} \qquad \frac{\lambda \notin e \quad \lambda \notin e'}{e \overset{\lhd}{\sim} e'} \qquad \frac{e_2 \overset{\lhd}{\sim} e_2'}{e_1[e_2] \overset{\lhd}{\sim} e_1[e_2']} \qquad \frac{e_1 \overset{\lhd}{\sim} e_1' \quad e_2 \overset{\lhd}{\sim} e_2'}{e_1 \,.\, e_2 \overset{\lhd}{\sim} e_1' \,.\, e_2'}$$

$$\frac{e \overset{\lhd}{\sim} e'}{\lambda e \overset{\lhd}{\sim} \lambda e'} \qquad \frac{\lambda \notin e'}{e \overset{\lhd}{\sim} e[e']} \qquad \frac{e_1 \overset{\lhd}{\sim} e_1' \quad \lambda \notin e_2}{e_1[e_2] \overset{\lhd}{\sim} e_1'[e_2]}$$

The following theorem states the desired simulation properties.

[1]The formalized proof represents the exclusion of λ by a predicate `nolam`. This predicate also excludes constants to reduce the number of cases needed in the proofs, but it could just as well have included them, and the general theorem is proved later anyway. Thus, whenever we write $\lambda \notin e$ we will also exclude occurrences of constants in e.

Lemma 5.5 (`sigma-strong.thm:is_esn_rel_under_shift`). *If $e_0 \overset{\lhd}{\sim} e_0'$ then there exists a k such that for all reductions $e_0' \to e_1' \to \cdots \to e_n'$ in n steps there exists a sequence $e_0, e_1, \ldots, e_n$ such that*

1. *either $e_i \to e_{i+1}$ or $e_i = e_{i+1}$ for $0 \leq i < n$,*
2. *$e_i \overset{\lhd}{\sim} e_i'$ for $0 \leq i \leq n$, and*
3. *$n \geq k$ implies $e_i \to e_{i+1}$ for some i.*

It is noteworthy to consider how Lemma 5.5 is encoded in Abella. The statement of the lemma is kind of like strong normalization in the sense that some limit k exists after which any reduction sequence will bottom out in some way; in this case, have a corresponding reduction in the $\overset{\lhd}{\sim}$-related expression. The Abella-encoding of $\mathcal{SN}$ is the following inductive definition:

$$e \in \mathcal{SN} \quad := \quad \forall e'.\, e \to e' \supset e' \in \mathcal{SN}$$

We can then give a similar, but slightly more complicated, inductive definition of a relation $\mathcal{SN}[e]$ (denoted `esn_rel_under_shift` in the formalization).

$$\begin{aligned} e_1' \in \mathcal{SN}[e_1] \quad := \quad & e_1 \overset{\lhd}{\sim} e_1' \land \forall e_2'.\, e_1' \to e_2' \supset \\ & (\exists e_2.\, e_1 \to e_2 \land e_2 \overset{\lhd}{\sim} e_2') \lor e_2' \in \mathcal{SN}[e_1] \end{aligned}$$

Lemma 5.5 can now be represented as the statement $e \overset{\lhd}{\sim} e' \supset e' \in \mathcal{SN}[e]$.

By a nested induction on the strong normalization of e and the k from Lemma 5.5 we get the following lemma:

Lemma 5.6 (`sigma-strong.thm:exp_under_shift_esn`). *If $e \in \mathcal{SN}$ and $e \overset{\lhd}{\sim} e'$ then $e' \in \mathcal{SN}$.*

As an immidiate corollary we get that $e \in \mathcal{SN}$ implies $e[\uparrow^1] \in \mathcal{SN}$. This gives us the desired version of Lemma 5.3 without the restriction on e_1.

Lemma 5.7 (`sigma-strong.thm:esn_clos`). *If $e_1 \in \mathcal{SN}$ and $e_2 \in \mathcal{SN}$ then $e_1[e_2] \in \mathcal{SN}$.*

Together Lemma 5.1, Lemma 5.2, and Lemma 5.7 give the strong normalization theorem:

Theorem 5.8 (`sigma-strong.thm:exp_esn`). *Every expression is strongly normalizing: $e \in \mathcal{SN}$ for all e.*

And thus, we have strong normalization of the σ-fragment of $\lambda\sigma$.

Theorem 5.9 (`lambda-sigma.thm:tm_sn_su, sub_sns_su`). *The reduction relation $\to_\sigma$ is strongly normalizing for terms and substitutions.*

5.2 Confluence of the σ-fragment

With strong normalization we can easily prove confluence of the σ-fragment by first proving local confluence.

Theorem 5.10 (`conf.thm:local_conf`). *The σ-fragment is locally confluent.*

1. *If $M_1 \;{}_\sigma\!\leftarrow M \to_\sigma M_2$ then there exists an M' such that $M_1 \to_\sigma^* M' \;{}_\sigma^*\!\leftarrow M_2$.*
2. *If $s_1 \;{}_\sigma\!\leftarrow s \to_\sigma s_2$ then there exists an s' such that $s_1 \to_\sigma^* s' \;{}_\sigma^*\!\leftarrow s_2$.*

Theorem 5.11 (`conf.thm:conf_tm, conf_sub`). *The σ-fragment is confluent.*

1. *If $M_1 \;{}_\sigma^*\!\leftarrow M \to_\sigma^* M_2$ then there exists an M' such that $M_1 \to_\sigma^* M' \;{}_\sigma^*\!\leftarrow M_2$.*
2. *If $s_1 \;{}_\sigma^*\!\leftarrow s \to_\sigma^* s_2$ then there exists an s' such that $s_1 \to_\sigma^* s' \;{}_\sigma^*\!\leftarrow s_2$.*

Together confluence and strong normalization give us the existence of unique σ-normal forms. We shall denote the σ-normal form of a term M or substitution s as $\sigma(M)$ and $\sigma(s)$, respectively.

$$\frac{}{\Gamma, A \vdash 1 : A} \qquad \frac{\Gamma \vdash n : A}{\Gamma, B \vdash n+1 : A} \qquad \frac{\Gamma \vdash s : \Gamma' \quad \Gamma' \vdash M : A}{\Gamma \vdash M[s] : A}$$

$$\frac{\Sigma(c) = A}{\Gamma \vdash c : A} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B} \qquad \frac{\Gamma, A \vdash M : B}{\Gamma \vdash \lambda M : A \to B}$$

$$\frac{}{\Gamma \vdash \uparrow^0 : \Gamma} \qquad \frac{\Gamma \vdash \uparrow^n : \Gamma'}{\Gamma, A \vdash \uparrow^{n+1} : \Gamma'}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash s : \Gamma'}{\Gamma \vdash M \,.\, s : \Gamma', A} \qquad \frac{\Gamma \vdash t : \Gamma'' \quad \Gamma'' \vdash s : \Gamma'}{\Gamma \vdash s \circ t : \Gamma'}$$

Figure 4: Types for $\lambda\sigma$

5.3 Preservation of strong normalization

With the σ-fragment covered, we turn our attention to the **beta** steps. Let $\to_\beta$ denote a **beta** step. Then the reduction relation $\to$ is the disjoint union of $\to_\sigma$ and $\to_\beta$. The σ-normal forms correspond to the ordinary lambda calculus, so for such terms we can define ordinary β-reduction in the following way:

$$M \Rightarrow_\beta M' \quad := \quad M \to_\beta M'' \wedge M' = \sigma(M'')$$

Preservation of strong normalization (PSN) is the property that strong normalization of $\sigma(M)$ with respect to $\Rightarrow_\beta$ implies strong normalization of M with respect to $\to$.

Our restriction of the congruence rules allows us to prove that **beta** steps correspond to β-steps in the ordinary lambda calculus:

Theorem 5.12 (`beta.thm:project_beta`). *If $M \to_\beta M'$ then $\sigma(M) \Rightarrow_\beta \sigma(M')$.*

Together with confluence and strong normalization of the σ-fragment, we immidiately get PSN:

Theorem 5.13 (`strong-norm.thm:psn`). *If $\sigma(M)$ is strongly normalizing with respect to $\Rightarrow_\beta$ then M is strongly normalizing with respect to $\to$.*

5.4 Strong normalization of simply typed $\lambda\sigma$

So far, we have only considered untyped terms and substitutions. So before we can talk about strong normalization of well-typed terms and substitutions, we need to introduce the type system. The typing rules are standard for $\lambda\sigma$ and given in Figure 4.

Theorem 5.14 (`typing.thm:of_step_ext`). *Subject reduction.*

1. *If $\Gamma \vdash M : A$ and $M \to M'$ then $\Gamma \vdash M' : A$.*
2. *If $\Gamma \vdash s : \Gamma'$ and $s \to s'$ then $\Gamma \vdash s' : \Gamma'$.*

Since well-typed σ-normal forms are exactly the simply typed lambda calculus, we have the usual proof of strong normalization using a logical relation. The Abella proof is adapted from Girard's proof of strong normalization in the example suite of Abella [Abe].

Theorem 5.15 (`beta-sn.thm:strong_beta`). *Let M be a σ-normal form with $\Gamma \vdash M : A$. Then M is strongly normalizing with respect to $\Rightarrow_\beta$.*

Now PSN (Theorem 5.13) gives us strong normalization of simply typed terms:

Theorem 5.16 (`strong-norm.thm:strong_tm`). *If $\Gamma \vdash M : A$ then M is strongly normalizing with respect to $\to$.*

We can adapt the proof of Lemma 5.3 to prove strong normalization of compositions. We will not have to consider λs in this case, since for terms we can simply appeal to Theorem 5.16.

Lemma 5.17 (`strong-norm.thm:sns_clos`). *If $\Gamma \vdash t : \Gamma''$, $\Gamma'' \vdash s : \Gamma'$, and s and t are strongly normalizing then $s \circ t$ is strongly normalizing.*

As an immidiate consequence we get strong normalization of substitutions:

Theorem 5.18 (`strong-norm.thm:strong_sub`). *If $\Gamma \vdash s : \Gamma'$ then s is strongly normalizing.*

6 Extending the strong normalization proof

So far we have achieved strong normalization with the restricted congruence rule:

$$\frac{s \to_\sigma s'}{M[s] \to_\sigma M[s']}$$

With Theorem 5.18 we showed that substitutions are strongly normalizing, and it is therefore plausible that we could allow a single application of the unrestricted congruence rule in each step without breaking strong normalization. The intuition is that whenever we take a step inside a substitution we maintain the overall structure of the term disregarding the contents of substitutions. Thus, a nested induction on the strong normalization of the term and the strong normalization of all substitutions occurring within the term could presumably extend the strong normalization to this slightly more general reduction relation.

Going further, we can consider an extension of the reduction relation that allows the unrestricted congruence rule at most k times in each step for some fixed number k. As we saw in the counter-example, the non-terminating reduction sequence involved deeper and deeper nesting of **beta** steps using the unrestricted closure congruence rule. This means that having such a fixed limit k is still going to rule out this particular counter-example.

In this section we will formalize these ideas and prove the extended reduction relation strongly normalizing on well-typed terms and substitutions, thereby pushing the boundary of strong normalization right up to the limit set by the counter-example.

We will in the following assume that every term and substitution is well-typed.

6.1 The extended reduction relation

We will define a reduction relation $\xrightarrow{k}$ for each natural number $k \geq 0$ that allows k applications of the unrestricted congruence rule.

The reduction relation $\xrightarrow{k}$ is shown in its entirety in Figure 5. Notice that, for $k = 0$ the relation coincides with our regular reduction relation $\xrightarrow{0} = \to$, and of course a larger value of k allows more reductions $\xrightarrow{k} \subset \xrightarrow{k+1}$.

We will denote the subrelation that uses one of the rules **clos1** or **clos2** at least once as $\xrightarrow{k}_{[]}$. The subrelation that uses neither **clos1** nor **clos2** is denoted $\xrightarrow{k}_{\not[]}$, such that $\xrightarrow{k} = \xrightarrow{k}_{[]} \cup \xrightarrow{k}_{\not[]}$. Since $\xrightarrow{k}_{\not[]}$ is independent of the value of k we will also write this relation as $\to_{\not[]}$.

6.2 Strong normalization of the extended reduction relation

We will prove $\xrightarrow{k}$ strongly normalizing for terms and substitutions by induction on k. For $k = 0$ the relation is equal to $\to$ and therefore strongly normalizing by Theorem 5.16 and Theorem 5.18.

In the following we will prove the induction step.

If we first consider $\xrightarrow{k+1}_{[]}$ then every step uses **clos1**. And this relation is therefore strongly normalizing by the simultaneous induction on the strong normalization of every substitution occurring inside the term, since these substitutions are in turn strongly normalizing by the induction on k.

Theorem 6.1 (`strong-ext.thm:is_sn1_clo,is_sn1s_clo`). *Assuming that $\xrightarrow{k}$ is strongly normalizing on substitutions, the relation $\xrightarrow{k+1}_{[]}$ is strongly normalizing on terms and substitutions.*

Assume we have $M_1 \xrightarrow{k+1}{}^{*}_{[]} M_2$ and we wish to prove M_2 strongly normalizing with respect to $\xrightarrow{k+1}$. Any $\xrightarrow{k+1}$ step taken by M_2 is either a $\xrightarrow{k+1}_{[]}$ step or a $\xrightarrow{k+1}_{\not[]}$ step. In the first case we again have $M_1 \xrightarrow{k+1}{}^{*}_{[]} M_2'$ with the same M_1, and this case cannot happen indefinitely since $\xrightarrow{k+1}_{[]}$ is strongly normalizing. In the second case we have $M_1 \xrightarrow{k+1}{}^{*}_{[]} M_2 \to_{\not[]} M_2'$. Since the steps from M_1 to M_2 only occur inside substitutions and the step from M_2 to M_2' only occurs outside substitutions, the idea is to prove that these two parts commute in some sense. That

beta	$(\lambda M)\ N$	$\xrightarrow{k}$	$M[N\,.\,\mathsf{id}]$
clos-const	$c[s]$	$\xrightarrow{k}$	c
clos-var-dot1	$1[M\,.\,s]$	$\xrightarrow{k}$	M
clos-var-dot2	$(n+1)[M\,.\,s]$	$\xrightarrow{k}$	$n[s]$
clos-var-shift	$n[\uparrow^m]$	$\xrightarrow{k}$	$n+m$
clos-clos	$M[s][t]$	$\xrightarrow{k}$	$M[s \circ t]$
clos-lam	$(\lambda M)[s]$	$\xrightarrow{k}$	$\lambda(M[1\,.\,(s \circ \uparrow^1)])$
clos-app	$(M\ N)[s]$	$\xrightarrow{k}$	$M[s]\ N[s]$
comp-id-L	$\uparrow^0 \circ s$	$\xrightarrow{k}$	s
comp-cons	$(M\,.\,s) \circ t$	$\xrightarrow{k}$	$M[t]\,.\,(s \circ t)$
comp-shift-dot	$\uparrow^{n+1} \circ (M\,.\,s)$	$\xrightarrow{k}$	$\uparrow^n \circ s$
comp-shift-shift	$\uparrow^n \circ \uparrow^m$	$\xrightarrow{k}$	$\uparrow^{n+m}$
comp-comp	$(s_1 \circ s_2) \circ s_3$	$\xrightarrow{k}$	$s_1 \circ (s_2 \circ s_3)$
eta-subst	$(n+1)\,.\,\uparrow^{n+1}$	$\xrightarrow{k}$	$\uparrow^n$

$$\frac{M \xrightarrow{k} M'}{M\,.\,s \xrightarrow{k} M'\,.\,s} \qquad \frac{s \xrightarrow{k} s'}{M\,.\,s \xrightarrow{k} M\,.\,s'} \qquad \frac{s \xrightarrow{k} s'}{s \circ t \xrightarrow{k} s' \circ t} \qquad \frac{t \xrightarrow{k} t'}{s \circ t \xrightarrow{k} s \circ t'}$$

$$\frac{M \xrightarrow{k} M'}{\lambda M \xrightarrow{k} \lambda M'} \qquad \frac{M \xrightarrow{k} M'}{M\ N \xrightarrow{k} M'\ N} \qquad \frac{N \xrightarrow{k} N'}{M\ N \xrightarrow{k} M\ N'}$$

$$\frac{M \xrightarrow{k} M'}{M[s] \xrightarrow{k} M'[s]} \qquad \frac{s \xrightarrow{k} s'}{M[s] \xrightarrow{k+1} M[s']}\ \textbf{clos1} \qquad \frac{s \to_\sigma s'}{M[s] \xrightarrow{0} M[s']}\ \textbf{clos2}$$

Figure 5: Reduction rules

is, if we could prove $M_1 \to M_1' \xrightarrow{k+1}{}^*_{[]} M_2'$ then we could appeal to induction on the strong normalization of M_1 with respect to $\to$. Unfortunately this is not always the case, but this idea will lead us to something similar, which in the end will get us there.

One of the hard cases turns out to be $1[s_1] \xrightarrow{k+1}{}^*_{[]} 1[M_2'\,.\,s_2] \to_{\not{[]}} M_2'$ since this only gives us $s_1 \xrightarrow{k}{}^* M_2'\,.\,s_2$ to work with. To deal with this case (and a few similar cases) we will introduce a weak head normalization function for substitution compositions.

The functions $\mathsf{wcomp}(s)$ and $\mathsf{wcomp}(n;s)$ compute a weak head normal form of s and $\uparrow^n \circ s$, respectively. They are defined as follows and easily seen to be total.

$$\begin{array}{lcl}
\mathsf{wcomp}(s) & = & \mathsf{wcomp}(0;s) \\
\mathsf{wcomp}(n_1;\uparrow^{n_2}) & = & \uparrow^{n_1+n_2} \\
\mathsf{wcomp}(0;M\,.\,s) & = & M\,.\,s \\
\mathsf{wcomp}(n+1;M\,.\,s) & = & \mathsf{wcomp}(n;s) \\
\mathsf{wcomp}(n;s_1 \circ s_2) & = & \textbf{case } \mathsf{wcomp}(n;s_1) \textbf{ of} \\
 & & \quad \uparrow^{n'} \Rightarrow \mathsf{wcomp}(n';s_2) \\
 & & \quad M\,.\,s \Rightarrow M[s_2]\,.\,(s \circ s_2)
\end{array}$$

Since wcomp plays a bit with the associativity of composition, the following theorems are not entirely trivial, but nevertheless true.

Theorem 6.2 (`strong-ext.thm:wcomp_to_msteps_su0`)**.** *If* $\mathsf{wcomp}(n;s) = s'$ *then* $\uparrow^n \circ s \to_\sigma^* s'$.

Theorem 6.3 (`strong-ext.thm:wcomp0_to_msteps_su0`)**.** *If* $\mathsf{wcomp}(s) = s'$ *then* $s \to_\sigma^* s'$.

The definition of wcomp is designed to do as little as possible. In particular, for $s_1 \circ s_2$ it avoids any reduction in s_2 whenever possible. This means that wcomp computes the least possibly reduced weak head normal form in a sense made precise by the following theorem. In particular, any reduction to a substitution s_2 with $\mathsf{wcomp}(s_2) = s_2$ factors through wcomp.

Theorem 6.4 (`strong-ext.thm:commute_wcomp_mstep1`). *If $s_1 \xrightarrow{m}{}^{*} s_2$ then $\mathsf{wcomp}(n; s_1) \xrightarrow{m+1}{}^{*} \mathsf{wcomp}(n; s_2)$.*

Returning our attention to the case $1[s_1] \xrightarrow{m}{}^{*}_{[]} 1[M \,.\, s_2] \to_{\not{[]}} M$, we can apply Theorem 6.4 to $s_1 \xrightarrow{m}{}^{*} M \,.\, s_2$ and thereby get a $\to$ reduction step of $1[s_1]$ to some M' with $M' \xrightarrow{m+1}{}^{*} M$. This deals with most of the otherwise problematic cases and allows us to prove the following theorem.

Theorem 6.5 (`strong-ext.thm:commute_clo_noc`). *If $M_1 \xrightarrow{m}{}^{*}_{[]} M_2 \to_{\not{[]}} M_3$ then there exists an M such that $M_1 \to^{+} M \xrightarrow{m+1}{}^{*} M_3$.*

Theorem 6.5 presents the following diagram:

$$\begin{tikzcd}
M_1 \arrow[r, "m"] \arrow[d, "+"] & {}^{*}_{[]}\, M_2 \arrow[d, "\not{[]}"] \\
M \arrow[r, "m+1"] & {}^{*}\, M_3
\end{tikzcd}$$

Unfortunately, the bottom arrow is $M \xrightarrow{m+1}{}^{*} M_3$ and not $M \xrightarrow{m+1}{}^{*}_{[]} M_3$. The reduction sequence from M to M_3 can be divided into $\xrightarrow{m+1}_{[]}$ steps and $\to_{\not{[]}}$ steps, so if it is not entirely consisting of $\xrightarrow{m+1}_{[]}$ steps, we can split it as $M \xrightarrow{m+1}{}^{*}_{[]} M_4 \to_{\not{[]}} M_5 \xrightarrow{m+1}{}^{*} M_3$ and apply Theorem 6.5 to the left half:

$$\begin{tikzcd}
M \arrow[r, "m+1"] \arrow[d, "+"] & {}^{*}_{[]}\, M_4 \arrow[d, "\not{[]}"] & \\
M' \arrow[r, "m+2"] & {}^{*}\, M_5 \arrow[r, "m+1"] & {}^{*}\, M_3
\end{tikzcd}$$

We can repeat this construction on $M' \xrightarrow{m+2}{}^{*} M_3$ and since M is strongly normalizing with respect to $\to$ we will eventually reach $M \to^{+} M'' \xrightarrow{m'}{}^{*}_{[]} M_3$ for some m'. Thus, we have strengthened Theorem 6.5 into:

Theorem 6.6 (`strong-ext.thm:commute_clo_noc2`). *If $M_1 \xrightarrow{m}{}^{*}_{[]} M_2 \to_{\not{[]}} M_3$ then there exists m' and M such that $M_1 \to^{+} M \xrightarrow{m'}{}^{*}_{[]} M_3$.*

Now we can prove $\xrightarrow{k+1}$ strongly normalizing for some given term M_2 by a nested induction on the strong normalization of M_1 with respect to $\to$ and the strong normalization of M_2 with respect to $\xrightarrow{k+1}_{[]}$ and the invariant $M_1 \xrightarrow{m}{}^{*}_{[]} M_2$ for some m.

Theorem 6.7 (`strong-ext.thm:is_sn1`). *If $\xrightarrow{k+1}_{[]}$ is strongly normalizing then $\xrightarrow{k+1}$ is strongly normalizing for terms.*

Is it easy to refit the proofs of Lemma 5.17 and Theorem 5.18 to yield strong normalization for substitutions with respect to $\xrightarrow{k+1}$ given strong normalization for terms.

With Theorem 6.1 and Theorem 6.7 we now have all the pieces to finish the induction on k and prove $\xrightarrow{k}$ strongly normalizing for all k.

Theorem 6.8 (`strong-ext.thm:strong_N`). *The reduction relation $\xrightarrow{k}$ is strongly normalizing for well-typed terms and substitutions for all $k \geq 0$.*

7 Conclusion

We have shown how a small restriction in the congruence rules of $\lambda\sigma$ gives a strongly normalizing calculus. In addition to general insight into the normalization properties of explicit substitution calculi, this result also provides a very flexible foundation for the design of normalization procedures in any λ-calculus-based implementation, e.g. logical frameworks and proof assistants.

References

[Abe] Girard's proof of strong normalization of the simply-typed lambda-calculus. `http://abella.cs.umn.edu/examples/`. Abella examples.

[CHR92] P.-L. Curien, T. Hardin, and A. Ríos. Strong Normalization of Substitutions. *Mathematical Foundations of Computer Science 1992*, pages 209–217, 1992.

[Mel95] Paul-André Mellies. Typed λ-calculi with explicit substitutions may not terminate. *Typed Lambda Calculi and Applications*, pages 328–334, 1995.