On Continuous Normalization

Klaus Aehlig Felix Joachimski
Mathematisches Institut
LMU München
{aehlig|joachski}@mathematik.uni-muenchen.de

CSL 2002

http://www.mathematik.uni-muenchen.de/~{aehlig|joachski}

Some history

- Cut-elimination: a cut $\dfrac{\neg A \vee \neg B \quad A \wedge B}{\bot}$ is replaced by cuts on A and B.
- This process is defined by induction on (semi-formal) derivations.
- Want to separate the operational definition from the proof-theoretical analysis (well-foundedness of the derivation involves ordinal notation systems and strong means).
- In order to compute the normalized derivation in a primitive recursive way
- Mints 1978 introduced repetition rule* $\dfrac{\Gamma \vdash A}{\Gamma \vdash A}$ to compute the last rule of the normalized derivation ("Please wait; your proof will soon be computed (hopefully)")
- This procedure even applies to non-wellfounded derivations (it is continuous).

* which has the subformula property!

Goal

- Transfer ideas to λ-calculus (as Schwichtenberg 1998 and others did).

Goal

- Transfer ideas to λ-calculus (as Schwichtenberg 1998 and others did).
- Interesting, because some λ-terms (like Y) have infinite "normal forms", or are even diverging (like Ω).

Goal

- Transfer ideas to λ-calculus (as Schwichtenberg 1998 and others did).
- Interesting, because some λ-terms (like Y) have infinite "normal forms", or are even diverging (like Ω).
- Need a coinductive λ-calculus to write out the normal forms!

Goal

- Transfer ideas to λ-calculus (as Schwichtenberg 1998 and others did).
- Interesting, because some λ-terms (like Y) have infinite "normal forms", or are even diverging (like Ω).
- Need a coinductive λ-calculus to write out the normal forms!

Thus: define and analyze
a primitive recursive normalization function
$$(\ )^{\beta} : \Lambda^{(\mathrm{co})} \longrightarrow \Lambda^{\mathrm{co}}_{\mathcal{R}}$$

Goal

- Transfer ideas to λ-calculus (as Schwichtenberg 1998 and others did).
- Interesting, because some λ-terms (like Y) have infinite "normal forms", or are even diverging (like Ω).
- Need a coinductive λ-calculus to write out the normal forms!

> Thus: define and analyze
> a primitive recursive normalization function
> $(\ )^{\beta} : \Lambda^{(\mathrm{co})} \longrightarrow \Lambda^{\mathrm{co}}_{\mathcal{R}}$

In particular, explain why and how many repetition rules are needed.

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$\Lambda \quad \ni \quad r, s ::= \quad x \mid rs \mid \lambda r$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{llll}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r
\end{array}
$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{lcll}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid rs \mid \lambda r \mid \mathcal{R} r
\end{array}
$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{lclcl}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid rs \mid \lambda r \mid \mathcal{R}r \mid \beta r
\end{array}
$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$\begin{array}{lcll}
\Lambda & \ni & r, s ::= & x \mid r\,s \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid r\,s \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid r\,s \mid \lambda r \mid \mathcal{R}r \mid \beta r \\
\Lambda_{\mathcal{R}}^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid r\,s \mid \lambda r \mid \mathcal{R}r \mid \beta r
\end{array}$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{llllllllll}
\Lambda & \ni & r,s ::= & x & \mid & rs & \mid & \lambda r & & \\
\Lambda^{\text{co}} & \ni & r,s ::=^{\text{co}} & x & \mid & rs & \mid & \lambda r & & \\
\Lambda_{\mathcal{R}} & \ni & r,s ::= & x & \mid & rs & \mid & \lambda r & \mid \mathcal{R}r & \mid \beta r \\
\Lambda^{\text{co}}_{\mathcal{R}} & \ni & r,s ::=^{\text{co}} & x & \mid & rs & \mid & \lambda r & \mid \mathcal{R}r & \mid \beta r
\end{array}
$$

Examples.

$\Theta \quad := \quad tt$ with $t = \lambda\lambda.0.110$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{llll}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid rs \mid \lambda r \mid \mathcal{R} r \mid \beta r \\
\Lambda_{\mathcal{R}}^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \mid \mathcal{R} r \mid \beta r
\end{array}
$$

Examples.

$$\Theta \quad := \quad tt \text{ with } t = \lambda\lambda.0.110 \quad = \text{“}\lambda t \lambda f (f (ttf))\text{”}$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{llll}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid rs \mid \lambda r \mid \mathcal{R}r \mid \beta r \\
\Lambda_{\mathcal{R}}^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \mid \mathcal{R}r \mid \beta r
\end{array}
$$

Examples.

$$
\begin{array}{llll}
\Theta & := & tt \text{ with } t = \lambda\lambda.0.110 & = \text{“}\lambda t \lambda f (f (ttf))\text{”} \\
Y & := & \lambda.(\lambda.1.00)(\lambda.1.00) & = \text{“}\lambda f.(\lambda x(f(xx)))(\lambda x(f(xx)))\text{”}
\end{array}
$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$
\begin{array}{lcll}
\Lambda & \ni & r, s ::= & x \mid rs \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= & x \mid rs \mid \lambda r \mid \mathcal{R}r \mid \beta r \\
\Lambda_{\mathcal{R}}^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} & x \mid rs \mid \lambda r \mid \mathcal{R}r \mid \beta r
\end{array}
$$

Examples.

$$
\begin{array}{lcll}
\Theta & := & tt \text{ with } t = \lambda\lambda.0.110 & = \text{``}\lambda t \lambda f(f(ttf))\text{''} \\
Y & := & \lambda.(\lambda.1.00)(\lambda.1.00) & = \text{``}\lambda f.(\lambda x(f(xx)))(\lambda x(f(xx)))\text{''} \\
Y_r^{\mathrm{co}} & := & rY_r^{\mathrm{co}} & = r(r(r \ldots
\end{array}
$$

Terms

Terms. Inductive and coinductive λ-calculus in de Bruijn-notation ($x \in \mathbb{N}$).

$$\begin{array}{lcl}
\Lambda & \ni & r, s ::= x \mid r s \mid \lambda r \\
\Lambda^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} x \mid r s \mid \lambda r \\
\Lambda_{\mathcal{R}} & \ni & r, s ::= x \mid r s \mid \lambda r \mid \mathcal{R} r \mid \beta r \\
\Lambda_{\mathcal{R}}^{\mathrm{co}} & \ni & r, s ::=^{\mathrm{co}} x \mid r s \mid \lambda r \mid \mathcal{R} r \mid \beta r
\end{array}$$

Examples.

$$\begin{array}{lcll}
\Theta & := & tt \text{ with } t = \lambda\lambda.0.110 & = \text{“}\lambda t \lambda f (f(ttf))\text{”} \\
Y & := & \lambda.(\lambda.1.00)(\lambda.1.00) & = \text{“}\lambda f.(\lambda x(f(xx)))(\lambda x(f(xx)))\text{”} \\
Y_r^{\mathrm{co}} & := & r Y_r^{\mathrm{co}} & = r(r(r\ldots
\end{array}$$

Observational equality. $r \simeq_k s$ iff the outermost k constructors are identical. E.g., $1\lambda\lambda 0 \simeq_2 1\lambda(0\,2)$.

Equality. $r = s$ iff $r \simeq_k s$ for all k.

Continuous normalization

Substitution. $r[s]$ is well-defined

Continuous normalization

Substitution. $r[s]$ is well-defined and continuous with identity as modulus of continuity, i.e.,

$$r \simeq_k r' \wedge s \simeq_k s' \Rightarrow r[s] \simeq_k r'[s'].$$

Continuous normalization

Substitution. $r[s]$ is well-defined and continuous with identity as modulus of continuity, i.e.,

$$r \simeq_k r' \wedge s \simeq_k s' \Rightarrow r[s] \simeq_k r'[s'].$$

Reduction. $\to$ is the compatible closure of $(\lambda r)s \to r[s]$.

Continuous normalization

Substitution. $r[s]$ is well-defined and continuous with identity as modulus of continuity, i.e.,

$$r \simeq_k r' \wedge s \simeq_k s' \Rightarrow r[s] \simeq_k r'[s'].$$

Reduction. $\to$ is the compatible closure of $(\lambda r)s \to r[s]$.

Normal forms. $\mathrm{NF} \ni r, s ::=^{\mathrm{co}} x\vec{r} \mid \lambda r \mid \mathcal{R}r \mid \beta r.$

Continuous normalization

Substitution. $r[s]$ is well-defined and continuous with identity as modulus of continuity, i.e.,

$$r \simeq_k r' \wedge s \simeq_k s' \Rightarrow r[s] \simeq_k r'[s'].$$

Reduction. $\to$ is the compatible closure of $(\lambda r)s \to r[s]$.

Normal forms. $\mathrm{NF} \ni r, s ::=^{\mathrm{co}} x\vec{r} \mid \lambda r \mid \mathcal{R}r \mid \beta r$.

Continuous normalization. $r^\beta := r@\varepsilon$
where for $r, \vec{s} \in \Lambda^{\mathrm{co}}$ we define $r@\vec{s} \in \mathrm{NF}$ by

$$\begin{array}{rcl}
(rs)@\vec{s} & := & \mathcal{R}.r@(s, \vec{s}) \\
x@\vec{s} & := & x\vec{s}^{\,\beta} \\
(\lambda r)@\varepsilon & := & \lambda r^\beta \\
(\lambda r)@(s, \vec{s}) & := & \beta.r[s]@\vec{s}
\end{array}$$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$

Examples

- $(SKK)^{\beta} = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ \ K := \lambda\lambda 1.$
  Note that SKK

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0)$

Examples

- $(SKK)^{\beta} = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \rightarrow^2 \lambda.K0(K0) \rightarrow^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda0$.
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\qquad 0(\qquad\qquad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad 0(\qquad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Note that Θf

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \rightarrow^2 \lambda.K0(K0) \rightarrow^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\quad\quad 0(\quad\quad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Note that $\Theta f = (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ \ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$
\begin{array}{rcl}
\mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\qquad 0(\qquad\qquad 0(\ldots \\
\Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\
Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots
\end{array}
$$

Note that

$$
\begin{array}{rcl}
\Theta f & = & (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f \\
 & \to & (\lambda f.f(\Theta f))f
\end{array}
$$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\;\; K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$
\begin{array}{rcl}
\mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\qquad 0(\qquad\qquad 0(\ldots \\
\Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\
Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots
\end{array}
$$

Note that

$$
\begin{array}{rcl}
\Theta f & = & (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f \\
& \to & (\lambda f.f(\Theta f))f \\
& \to & f(\Theta f)
\end{array}
$$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\quad 0(\qquad\quad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Note that
$$\begin{array}{rcl} \Theta f & = & (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f \\ & \to & (\lambda f.f(\Theta f))f \\ & \to & f(\Theta f) \end{array}$$

while Y_f

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ \ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\quad 0(\qquad\quad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Note that

$$\begin{array}{rcl} \Theta f & = & (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f \\ & \to & (\lambda f.f(\Theta f))f \\ & \to & f(\Theta f) \end{array}$$

while

$$Y_f := (\lambda x.f(xx))(\lambda x.f(xx))$$

Examples

- $(SKK)^\beta = \mathcal{R}\mathcal{R}\beta\beta\lambda\mathcal{R}\mathcal{R}\beta\beta 0$ with $S := \lambda\lambda\lambda.2\,0\,.1\,0,\ \ K := \lambda\lambda 1.$
  Note that $SKK \to^2 \lambda.K0(K0) \to^2 \lambda 0.$
- For $\Theta = tt$ with $t = \lambda\lambda.0.110$ and $Y = \lambda.(\lambda.1.00)(\lambda.1.00)$ we get

$$\begin{array}{rcl} \mathrm{nf}\,\Theta = \mathrm{nf}\,Y & = & \lambda.\ 0(\qquad\qquad 0(\qquad\qquad 0(\ldots \\ \Theta^\beta & = & \mathcal{R}\beta\lambda.\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\mathcal{R}\mathcal{R}\beta\beta\mathcal{R}(0(\ldots \\ Y^\beta & = & \lambda.\mathcal{R}\beta\mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ \ \mathcal{R}\beta\ \ \mathcal{R}(0(\ldots \end{array}$$

Note that

$$\begin{array}{rcl} \Theta f & = & (\lambda x, f.f(xxf))(\lambda x, f.f(xxf))f \\ & \to & (\lambda f.f(\Theta f))f \\ & \to & f(\Theta f) \end{array}$$

while

$$Y_f := (\lambda x.f(xx))(\lambda x.f(xx)) \to fY_f.$$

Properties

- Continuity: $r \simeq_k r' \ \wedge \ \vec{s} \simeq_k \vec{s'} \implies r@\vec{s} \simeq_k r'@\vec{s'}$.
  In particular $r^\beta \simeq_k r'^\beta$ for $r \simeq_k r'$.

Properties

- Continuity: $r \simeq_k r' \;\wedge\; \vec{s} \simeq_k \vec{s'} \implies r@\vec{s} \simeq_k r'@\vec{s'}$.
  In particular $r^\beta \simeq_k r'^\beta$ for $r \simeq_k r'$.
- Soundness: $r \in \Lambda \;\wedge\; r^\beta \in \Lambda_\mathcal{R} \implies r \to^* r^{\beta *}$
  where r^* is r without β and $\mathcal{R}$ (for $r \in \Lambda_\mathcal{R}$).

Properties

- Continuity: $r \simeq_k r' \ \wedge\ \vec{s} \simeq_k \vec{s'} \implies r@\vec{s} \simeq_k r'@\vec{s'}$.
  In particular $r^\beta \simeq_k r'^\beta$ for $r \simeq_k r'$.
- Soundness: $r \in \Lambda \ \wedge\ r^\beta \in \Lambda_{\mathcal{R}} \implies r \longrightarrow^* r^{\beta *}$
  where r^* is r without β and $\mathcal{R}$ (for $r \in \Lambda_{\mathcal{R}}$).
- Continuous normal forms: $\exists s \in \Lambda^{\mathrm{co}}.s^\beta = r$ iff $\vdash r$ with

$$\frac{s, \vec{s} \vdash r}{\vec{s} \vdash \mathcal{R}r} \qquad \frac{\vdash \vec{r}}{\vec{r} \vdash x\vec{r}} \qquad \frac{\vdash r}{\vdash \lambda r} \qquad \frac{\vec{s} \vdash r}{s, \vec{s} \vdash \beta r}$$

Properties

- Continuity: $r \simeq_k r' \;\wedge\; \vec{s} \simeq_k \vec{s'} \implies r@\vec{s} \simeq_k r'@\vec{s'}$.
  In particular $r^\beta \simeq_k r'^\beta$ for $r \simeq_k r'$.
- Soundness: $r \in \Lambda \;\wedge\; r^\beta \in \Lambda_{\mathcal{R}} \implies r \to^* r^{\beta *}$
  where r^* is r without β and $\mathcal{R}$ (for $r \in \Lambda_{\mathcal{R}}$).
- Continuous normal forms: $\exists s \in \Lambda^{\mathrm{co}}.s^\beta = r$ iff $\vdash r$ with

$$\frac{s, \vec{s} \vdash r}{\vec{s} \vdash \mathcal{R}r} \qquad \frac{\vdash \vec{r}}{\vec{r} \vdash x\vec{r}} \qquad \frac{\vdash r}{\vdash \lambda r} \qquad \frac{\vec{s} \vdash r}{s, \vec{s} \vdash \beta r}$$

- Analysis: A $\mathcal{R}$ is justified by a β or an application:

$$\vdash s \implies \mathcal{R}_\zeta s \geq \beta_\zeta s + A_\zeta s$$

with $=$ for complete paths ζ

Counting reductions

in the leftmost-outermost reduction strategy

Counting reductions

in the leftmost-outermost reduction strategy

Standardization.

Counting reductions

in the leftmost-outermost reduction strategy

Standardization. If $r \longrightarrow^* s \in \mathrm{NF} \cap \Lambda$ then $r \leadsto s$, given by

$$\frac{\vec{r} \leadsto_{\vec{n}} \vec{s}}{x\vec{r} \leadsto_{\Sigma \vec{n}} x\vec{s}} \qquad \frac{r \leadsto_n s}{\lambda r \leadsto_n \lambda s} \qquad \frac{r[s]\vec{s} \leadsto_n t}{(\lambda r)s\vec{s} \leadsto_{n+1} t}$$

Counting reductions

in the leftmost-outermost reduction strategy

Standardization. If $r \longrightarrow^* s \in \mathrm{NF} \cap \Lambda$ then $r \rightsquigarrow s$, given by

$$\frac{\vec{r} \rightsquigarrow_{\vec{n}} \vec{s}}{x\vec{r} \rightsquigarrow_{\Sigma \vec{n}} x\vec{s}} \qquad \frac{r \rightsquigarrow_n s}{\lambda r \rightsquigarrow_n \lambda s} \qquad \frac{r[s]\vec{s} \rightsquigarrow_n t}{(\lambda r)s\vec{s} \rightsquigarrow_{n+1} t}$$

Lemma. $r \rightsquigarrow_n s \implies r^\beta \rhd^n_n s^\beta$

where $\rhd^k_n$ removes n occurrences of β and k occurences of $\mathcal{R}$.

Counting reductions

in the leftmost-outermost reduction strategy

Standardization. If $r \longrightarrow^* s \in \mathrm{NF} \cap \Lambda$ then $r \rightsquigarrow s$, given by

$$\frac{\vec{r} \rightsquigarrow_{\vec{n}} \vec{s}}{x\vec{r} \rightsquigarrow_{\Sigma \vec{n}} x\vec{s}} \qquad \frac{r \rightsquigarrow_n s}{\lambda r \rightsquigarrow_n \lambda s} \qquad \frac{r[s]\vec{s} \rightsquigarrow_n t}{(\lambda r)s\vec{s} \rightsquigarrow_{n+1} t}$$

Lemma. $r \rightsquigarrow_n s \implies r^\beta \rhd_n^n s^\beta$

where $\rhd_n^k$ removes n occurrences of β and k occurences of $\mathcal{R}$.

Theorem. $r \rightsquigarrow_n s \in \mathrm{NF} \cap \Lambda \implies r^\beta \rhd_n^{n+|s|} s$

where $|s|$ is the number of applications in s.

Conclusions and further work

- Defined and analyzed a primitive recursive normalization function in the coinductive λ-calculus with repetition rule $\mathcal{R}$.
- Explained the use of $\mathcal{R}$, using auxiliary constructor β. This gives information on the normalization process.
- Can be extended to calculi with infinitary branching rules, such as (ω).
- Straightforward implementation in `Haskell`:

```
beta :: Term -> Term
beta r = app r []

app :: Term -> [Term] -> Term
app (Lam r) (s:l) = Bet (app (subst r s 0) l)
app (Lam r)  []   = Lam (beta r)
app (Var k)   l   = foldl App (Var k) (map beta l)
app (App r s) l   = Rep (app r (s:l))
```

Examples in Haskell

```
s = Lam (Lam (Lam (Var 2 `App` (Var 0) `App` (Var 1 `App` (Var 0)))))
k = Lam (Lam (Var 1))
y = Lam (Lam (Var 1 `App` (Var 0 `App` (Var 0))) `App`
        (Lam (Var 1 `App` (Var 0 `App` (Var 0)))))
yco r = r `App` (yco r)
theta = Lam (Lam (Var 0 `App` (Var 1 `App` (Var 1) `App` (Var 0)))) `App`
        Lam (Lam (Var 0 `App` (Var 1 `App` (Var 1) `App` (Var 0))))
church n = Lam (Lam (iterate (App (Var 1)) (Var 0) !! n))
```

Test runs:

```
Main> beta (s `App` k `App` k)
  Rep (Rep (Bet (Bet (Lam (Rep (Rep (Bet (Bet (Var 0)))))))))
Main> beta (yco k)
  Rep (Bet (Lam (Rep (Bet (Lam (Rep (Bet (Lam (Rep (Bet (Lam
              (Rep (Bet (Lam (Rep (Bet (Lam (Rep {Interrupted!}
Main> beta (y `App` k)
  Rep (Bet (Rep (Bet (Rep (Bet (Lam (Rep (Bet (Rep (Bet (Lam
          (Rep (Bet (Rep (Bet (Lam (Rep (Bet (Rep {Interrupted!}
Main> beta (theta `App` k)
  Rep (Rep (Bet (Bet (Rep (Bet
      (Lam (Rep (Rep (Bet (Bet (Rep (Bet (Lam
          (Rep (Rep (Bet (Bet (Rep (Bet (Lam {Interrupted!}
```

More test runs

```
Main> beta (church 2 `App` (church 3))
  Rep (Bet (Lam (Rep (Bet (Lam (Rep (Rep (Bet (Bet (Rep (App (Var 1)
      (Rep (App (Var 1) (Rep (App (Var 1) (Rep (Rep (Bet (Bet (Rep
      (App (Var 1) (Rep (App (Var 1) (Rep (App (Var 1) (Rep (Rep
      (Bet (Bet (Rep (App (Var 1) (Rep (App (Var 1) (Rep (App (Var 1)
      (Var 0))) [..])
Main> beta (church 3 `App` (church 2))
  Rep (Bet (Lam (Rep (Bet (Lam (Rep (Rep (Bet (Bet (Rep (Rep (Bet
      (Bet (Rep (App (Var 1) (Rep (App (Var 1) (Rep (Rep (Bet (Bet (Rep
      (App (Var 1) (Rep (App (Var 1) (Rep (Rep (Bet (Bet (Rep (Rep (Bet
      (Bet (Rep (App (Var 1) (Rep (App (Var 1) (Rep (Rep (Bet (Bet (Rep
      (App (Var 1) (Rep (App (Var 1) (Var 0))) [..] )
```