Notes on $\lambda\mathtt{x}$: a calculus with explicit substitutions

Delia Kesner
kesner@pps.jussieu.fr
http://www.pps.jussieu.fr/~kesner

Syntax

$$M ::= x \mid (MM) \mid \lambda x.M \mid M[x/M]$$

A term is called *pure* if it has no explicit substitution operator [ ]. Indeed, the term $x[x/M]$ is not pure while $\lambda x.(xy)$ is pure. The pure terms of $\lambda\mathtt{x}$ are the terms of the λ-calculus.

The symbols λ and [/] are bind constructors. We work modulo α-conversion so that renaming of *bound* variables is systematically used when necessary. Thus for example, $x[x/y] =_\alpha z[z/y]$, $w[x/y] =_\alpha w[z/y]$, $\lambda x.(xz) =_\alpha \lambda y.(yz)$, $(\lambda x.x)x =_\alpha (\lambda y.y)x$ and $x[x/x] =_\alpha y[y/x]$.

Typing Rules

$$\frac{}{\Gamma, x : A \vdash_{\lambda\mathtt{x}} x : A}\ (Ax) \qquad \frac{\Gamma, x : B \vdash_{\lambda\mathtt{x}} M : A \quad \Gamma \vdash_{\lambda\mathtt{x}} N : B}{\Gamma \vdash_{\lambda\mathtt{x}} M[x/N] : A}\ (Cut)$$

$$\frac{\Gamma \vdash_{\lambda\mathtt{x}} M : B \to A \quad \Gamma \vdash_{\lambda\mathtt{x}} N : B}{\Gamma \vdash_{\lambda\mathtt{x}} (MN) : A}\ (\to e) \qquad \frac{\Gamma, x : A \vdash_{\lambda\mathtt{x}} M : B}{\Gamma \vdash_{\lambda\mathtt{x}} \lambda x : A.M : B \to A}\ (\to i)$$

We use $\Gamma \vdash_{\lambda\mathtt{x}} M : A$ to denote typability in $\lambda\mathtt{x}$ and $\Gamma \vdash M : A$ to denote standard typability in the simply typed λ-calculus.

Reduction Rules

$$
\begin{array}{llll}
(\lambda x.M)N & \longrightarrow_B & M[x/N] & \\
(MN)[x/L] & \longrightarrow_{App} & (M[x/L]N[x/L]) & \\
(\lambda y.M)[x/L] & \longrightarrow_{Lamb} & \lambda y.M[x/L] & \text{if } x \neq y \text{ and } y \notin FV(L) \\
y[x/M] & \longrightarrow_{Var1} & y & \text{if } x \neq y \\
x[x/M] & \longrightarrow_{Var2} & M &
\end{array}
$$

The system $\lambda\mathtt{x}$ contains the rules $\{B, App, Lamb, Var1, Var2\}$ and the sub-system $\mathtt{x}$ only contains $\lambda\mathtt{x} \setminus \{B\}$. An example of $\lambda\mathtt{x}$-rewriting is

$$(\lambda x.xz)y \longrightarrow_B (xz)[x/y] \longrightarrow_{App} (x[x/y]z[x/y]) \longrightarrow_{Var2} (yz[x/y]) \longrightarrow_{Var1} (yz)$$

Rewriting is considered modulo α-conversion, another example of $\lambda\mathtt{x}$-rewriting is

$$
\begin{array}{l}
(\lambda w.(\lambda x.xz))(xy) \longrightarrow_B (\lambda x.xz)[w/xy] =_\alpha (\lambda m.mz)[w/xy] \longrightarrow_{Lamb} \\
\lambda m.(mz)[w/xy] \longrightarrow_{App} \lambda m.(m[w/xy]z[w/xy]) \longrightarrow^*{}_{Var1,Var2} \lambda m.(mz)
\end{array}
$$

Subject Reduction

Proposition 0.1 If $\Gamma \vdash_{\lambda\mathtt{x}} M : A$ is derivable and $M \longrightarrow_{\lambda\mathtt{x}} M'$, then $\Gamma \vdash_{\lambda\mathtt{x}} M' : A$ is derivable.

Proof. By induction on the structure of M. The base case is $M = x$ and the property vacuously holds for this case. The inductive cases (to be done) are:

- $M = (LR)$. We must consider $M' = (L'R)$, where $L \longrightarrow_{\lambda\mathtt{x}} L'$, $M' = (LR')$, where $R \longrightarrow_{\lambda\mathtt{x}} R'$, and $M' = N[x/R]$ where $L = \lambda x.N$.
- $M = \lambda x.R$. We must consider $M' = \lambda x.R'$, where $R \longrightarrow_{\lambda\mathtt{x}} R'$.
- $M = L[x/R]$. We must consider $M' = L'[x/R]$, where $L \longrightarrow_{\lambda\mathtt{x}} L'$, $M' = L[x/R']$, where $R \longrightarrow_{\lambda\mathtt{x}} R'$, and the remaining cases of reduction at the root position of M:
  - $M = x[x/R] \longrightarrow_{Var1} R = M'$.
  - $M = y[x/R] \longrightarrow_{Var1} y = M'$.
  - $M = (M_1M_2)[x/R] \longrightarrow_{Var1} (M_1[x/R]M_2[x/R]) = M'$.
  - $M = (\lambda y.M_1)[x/R] \longrightarrow_{Var1} \lambda y.M_1[x/R] = M'$.

∎

Confluence

Lemma 0.2 The system $\mathtt{x}$ is terminating and locally confluent.

Proof. Termination can be proved for example with the RPO technique. Local confluence can be proved by analyzing all the cases. ▪

By Newman's Lemma we obtain:

Corollary 0.3 The system $\mathtt{x}$ is confluent and thus it defines a function.

Proposition 0.4 If M is a pure term, then $\mathtt{x}(M[y/N]) = \mathtt{x}(M)\{y \leftarrow \mathtt{x}(N)\}$.

Proof. By induction on pure terms. ▪

Proposition 0.5 The $\mathtt{x}$-normal forms of $\lambda\mathtt{x}$-terms are pure terms.

Proof. By induction on terms. For the case $M = N[y/L]$ we have : $\mathtt{x}(M[y/N]) = \mathtt{x}(\mathtt{x}(M)[y/N])$. By Proposition 0.5 $\mathtt{x}(M)$ is a pure term and thus by Proposition 0.4 $\mathtt{x}(\mathtt{x}(M)[y/N]) = \mathtt{x}(\mathtt{x}(M))\{y \leftarrow \mathtt{x}(N)\} = \mathtt{x}(M)\{y \leftarrow \mathtt{x}(N)\}$ which is a pure term since $\mathtt{x}(N)$ is also a pure term by Proposition 0.5.

▪

Lemma 0.6 $\mathtt{x}(M[y/N]) = \mathtt{x}(M)\{y \leftarrow \mathtt{x}(N)\}$.

Proof. We have $\mathtt{x}(M[y/N]) = \mathtt{x}(\mathtt{x}(M)[y/N])$ and we know that $\mathtt{x}(M)$ is a pure term (Proposition 0.5). We then apply Proposition 0.4 and obtain $\mathtt{x}(\mathtt{x}(M)[y/N]) = \mathtt{x}(\mathtt{x}(M))\{y \leftarrow \mathtt{x}(N)\} = \mathtt{x}(M)\{y \leftarrow \mathtt{x}(N)\}$.

▪

Thus for example:

$$\begin{array}{lcl} \mathtt{x}((yn)[y/\lambda z.z[z/m]]) & = & ((\lambda z.m)n) \\ \mathtt{x}(yn)\{y \leftarrow \mathtt{x}(\lambda z.z[z/m]])\} & = & (yn)\{y \leftarrow \lambda z.m\} = ((\lambda z.m)n) \end{array}$$

Lemma 0.7 If $M \longrightarrow_{\lambda\mathtt{x}} N$, then $\mathtt{x}(M) {\longrightarrow^*}_\beta\, \mathtt{x}(N)$.

Proof. By induction on the structure of M. The case $M = (\lambda x.L)R \longrightarrow_B L[y/R] = N$ uses Lemma 0.6. ▪

Lemma 0.8 If M is a pure term and $M \longrightarrow_\beta N$, then $M {\longrightarrow^*}_{\lambda\mathtt{x}} N$.

Proof. By induction on the structure of M. The case $M = (\lambda x.L)R \longrightarrow_\beta L\{x \leftarrow R\} = N$ uses Lemma 0.6. ∎

Lemma 0.9 The calculus $\lambda\mathtt{x}$ is confluent.

Proof. Let M be a $\lambda\mathtt{x}$-term such that $M \longrightarrow^*_{\lambda\mathtt{x}} N$ and $M \longrightarrow^*_{\lambda\mathtt{x}} R$. We want to show that we can close this diagram, that is, that there exists a $\lambda\mathtt{x}$-term L such that $N \longrightarrow^*_{\lambda\mathtt{x}} L$ and $R \longrightarrow^*_{\lambda\mathtt{x}} L$.

We use the interpretation method. Let us take the images of M, N and R by the function $\mathtt{x}$, denoted by $\mathtt{x}(M), \mathtt{x}(N)$ and $\mathtt{x}(R)$ respectively. By Lemma 0.7 we have $\mathtt{x}(M) \longrightarrow^*_\beta \mathtt{x}(N)$ and $\mathtt{x}(M) \longrightarrow^*_\beta \mathtt{x}(R)$. Since the λ-calculus is confluent, we know that there exists a pure term L such that $\mathtt{x}(N) \longrightarrow^*_\beta L$ and $\mathtt{x}(R) \longrightarrow^*_\beta L$, and since $\mathtt{x}(M)$ and $\mathtt{x}(N)$ are pure terms then we know by Lemma 0.8 that $\mathtt{x}(N) \longrightarrow^*_{\lambda\mathtt{x}} L$ and $\mathtt{x}(R) \longrightarrow^*_{\lambda\mathtt{x}} L$. The diagram can then be closed by the $\lambda\mathtt{x}$-reduction sequences $N \longrightarrow^*_{\mathtt{x}} \mathtt{x}(N) \longrightarrow^*_{\lambda\mathtt{x}} L$ and $R \longrightarrow^*_{\mathtt{x}} \mathtt{x}(R) \longrightarrow^*_{\lambda\mathtt{x}} L$. ∎

Preservation of β-Strong Normalization (PSN)

We will prove in this section the PSN property stated as follows: If M is a pure term in SN_β, then M is in $SN_{\lambda\mathtt{x}}$. We use the semantic RPO technique.

0.0.1 RPO

We consider a partial order $\rhd$ (called *precedence*) on a (possibly infinite) set of function symbols $f, g, h, \ldots$, each one equipped with a status in $\{lex, mul\}$. If two symbols are equivalent w.r.t the order $\rhd$ then they have the same status.

Definition 0.1 (RPO) Given a term $s = f(s_1, \ldots, s_m)$ and another term t, we say that $s >_{rpo} t$ iff

- $\exists i$ such that $s_i \geq_{rpo} t$, or
- $t = g(t_1, \ldots, t_n)$ and
  - $f \rhd g$ and $\forall j \;\; s >_{rpo} t_j$
  - $f = g$ and $\forall j \;\; s >_{rpo} t_j$ and $\langle s_1, \ldots, s_m \rangle >^{status(f)}_{rpo} \langle t_1, \ldots, t_n \rangle$.

Example 0.10 Let us consider the function symbols f, g, a, having *lex* status and equipped with the following precedence: $f \triangleright g \triangleright a$. Let us suppose that f and g are binary symbols and a is a constant. We then have $f(a, g(x, y)) >_{rpo} x$, $f(a, x) >_{rpo} g(a, a)$ and $f(g(a, a), x) >_{rpo} f(a, x)$.

Remark 0.11 RPO has the subterm property: $C[t] >_{rpo} t$, for every term t and every non empty context C.

Theorem 0.12 If $\triangleright$ is well-founded, then RPO is well-founded, that is, there exist no infinite chain $t_1 >_{rpo} t_2 >_{rpo} t_3 >_{rpo} \ldots$.

0.0.2 The proof of PSN

We denote by SN_z the set of strongly normalizing terms with respect to the reduction relation z.

Let us define the following set:

$$\mathcal{F} = \{M \text{ is a } \lambda\mathtt{x}\text{-term} \mid \text{for each subterm } N \subseteq M \text{ we have } \mathtt{x}(N) \in SN_\beta\}$$

Thus for example, the term $(xx)[x/(\lambda y.yy)]$ is not in $\mathcal{F}$ and $x[x/y] \in \mathcal{F}$.

Lemma 0.13 If $M \in \mathcal{F}$ and $M \longrightarrow_{\lambda\mathtt{x}} N$, then $N \in \mathcal{F}$.

Proof. By induction on the structure of M. ∎

Definition 0.2 (The term-algebra $\mathcal{T}_\Sigma$) Let us consider the infinite signature

$$\Sigma = \{\star, lamb\} \cup \{app^n, subs^n \mid n \in \mathbb{N}\}$$

where $\star$ is a constant, *lamb* is a unary symbol and all the app^n and $subs^n$ symbols are binary. We suppose that they all have multiset status and that they are equipped with the following precedence on Σ:

$$app^{n+1} \triangleright subs^n \triangleright app^n \triangleright lamb, \star, \text{ for every } n \in \mathbb{N}$$

We remark that this definition of precedence implies that $subs^m \triangleright subs^n$ whenever $m > n$.

The term-algebra $\mathcal{T}_\Sigma$ is the one generated by the signature Σ. Thus for example, $app^n(subs^m(\star, \star), lamb(\star))$ is in $\mathcal{T}_\Sigma$.

Definition 0.3 (The translation $\mathcal{T}$) The translation $\mathcal{T}$ from terms in $\mathcal{F}$ to terms in $\mathcal{T}_\Sigma$ is defined by induction as follows:

$$\begin{array}{lcl}
\mathcal{T}(x) & = & \star \\
\mathcal{T}(MN) & = & app^n(\mathcal{T}(M), \mathcal{T}(N)), \text{ where } n = \nu(\mathtt{x}(MN)) \\
\mathcal{T}(\lambda x.M) & = & lamb(\mathcal{T}(M)) \\
\mathcal{T}(M[x/N]) & = & subs^n(\mathcal{T}(M), \mathcal{T}(N)), \text{ where } n = \nu(\mathtt{x}(M[x/N]))
\end{array}$$

where $\nu(M)$ is the maximal length of a β-reduction sequence starting at M.

As an example, $\mathcal{T}(x[x/(yz)]) = subs^0(\star, app^0(\star, \star))$ and $\mathcal{T}((\lambda x.x)(y[y/z])) = app^1(lamb(\star), subs^0(\star, \star))$.

Proposition 0.14 If $M \in \mathcal{F}$ and $M \longrightarrow_{\lambda\mathtt{x}} N$, then $\mathcal{T}(M) >_{rpo} \mathcal{T}(N)$.

Proof. First of all we remark that $N \in \mathcal{F}$ by Lemma 0.13 so that $\mathcal{T}(M)$ and $\mathcal{T}(N)$ are well-defined. In order to show the property we proceed by induction on the structure of M. We only show here the cases of reduction at the root position of M:

- $M = x[x/L] \longrightarrow_{\lambda\mathtt{x}} L = N$. Then

$$\mathcal{T}(M) = subs^n(\star, \mathcal{T}(L)) >_{rpo} \mathcal{T}(L) = \mathcal{T}(N)$$

- $M = y[x/L] \longrightarrow_{\lambda\mathtt{x}} y = N$. Then

$$\mathcal{T}(M) = subs^n(\star, \mathcal{T}(L)) >_{rpo} \star = \mathcal{T}(N)$$

- $M = (M_1M_2)[x/L] \longrightarrow_{\lambda\mathtt{x}} (M_1[x/L]M_2[x/L]) = N$. Then

$$\begin{array}{l}
\mathcal{T}(M) = subs^n(app^k(\mathcal{T}(M_1), \mathcal{T}(M_2)), \mathcal{T}(L)) \\
>_{rpo} \\
app^n(subs^{n_1}(\mathcal{T}(M_1), \mathcal{T}(L)), subs^{n_2}(\mathcal{T}(M_2), \mathcal{T}(L))) = \mathcal{T}(N)
\end{array}$$

since $n_1, n_2 \leq n$ and $k \leq n$.

- $M = (\lambda y.M_1)[x/L] \longrightarrow_{\lambda\mathtt{x}} (\lambda y.M_1[x/L]) = N$. Then

$$\mathcal{T}(M) = subs^n(lamb(\mathcal{T}(M_1)), \mathcal{T}(L)) >_{rpo} lamb(subs^n(\mathcal{T}(M_1), \mathcal{T}(L))) = \mathcal{T}(N)$$

- $M = (\lambda y.L)R \longrightarrow_{\lambda\mathtt{x}} L[x/R] = N$. Then

  $$\mathcal{T}(M) = app^m(lamb(\mathcal{T}(L)), \mathcal{T}(R)) >_{rpo} subs^n(\mathcal{T}(L), \mathcal{T}(R)) = \mathcal{T}(N)$$

  since $m > n$.

∎

Proposition 0.15 If $M \in \mathcal{F}$, then $M \in SN_{\lambda\mathtt{x}}$.

Proof. Suppose $M \in \mathcal{F}$, and $M \notin SN_{\lambda\mathtt{x}}$. Then there is an infinite $\lambda\mathtt{x}$-reduction sequence of the form

$$M \longrightarrow_{\lambda\mathtt{x}} M_1 \longrightarrow_{\lambda\mathtt{x}} M_2 \longrightarrow_{\lambda\mathtt{x}} \ldots$$

Since $M \in \mathcal{F}$, then by Lemma 0.13 all the terms M_i are in $\mathcal{F}$ and so $\mathcal{T}$ is defined for all of them. We can then construct, by Proposition 0.14, an infinite RPO-chain

$$\mathcal{T}(M) >_{rpo} \mathcal{T}(M_1) >_{rpo} \mathcal{T}(M_2) >_{rpo} \ldots$$

which leads to a contradiction with Theorem 0.12. ∎

Theorem 0.16 (PSN) If M is a pure term in SN_β, then M is in $SN_{\lambda\mathtt{x}}$.

Proof. If M is a pure term in SN_β, then all its subterms are in SN_β, and then M is in $\mathcal{F}$. By application of Proposition 0.15 we can conclude that $M \in SN_{\lambda\mathtt{x}}$. ∎

How PSN can be lost

Let us consider the following rewriting rule:

$$(C)\;\; M[x/N][y/L] \longrightarrow^* M[x/N[y/L]],\;\text{ if } y \notin FV(M)$$

We write $\lambda\mathtt{xc}$ for the system $\lambda\mathtt{x} \cup \{C\}$.

Notation 0.17 We define $\alpha_0 =_{def} [x/(\lambda y.u)u]$ and $\alpha_{m+1} =_{def} [x/u\alpha_m]$.

Remark 0.18

1. $u\alpha_0\alpha_{m+1} \longrightarrow^*_{\lambda\mathtt{xc}} \ldots u\alpha_{m+1}\alpha_{m+2} \ldots$.

Proof.

$$
\begin{array}{ll}
u\alpha_0\alpha_{m+1} & = \\
u[x/(\lambda y.u)u][x'/u\alpha_m] & \longrightarrow_C \\
u[x/((\lambda y.u)u)[x'/u\alpha_m]] & \longrightarrow_{App} \\
u[x/(\lambda y.u)[x'/u\alpha_m]u[x'/u\alpha_m]] & \longrightarrow_{Lamb} \\
u[x/(\lambda y.u[x'/u\alpha_m])u[x'/u\alpha_m]] & \longrightarrow_B \\
u[x/u[x'/u\alpha_m][y/u[x'/u\alpha_m]]] & = \\
u[x/u\alpha_{m+1}\alpha_{m+2}] &
\end{array}
$$

∎

2. $u\alpha_{m+1}\alpha_{n+1} \longrightarrow^*_{\lambda\mathtt{xc}} \ldots u\alpha_m\alpha_{n+1} \ldots$.

Proof.

$$
\begin{array}{ll}
u\alpha_{m+1}\alpha_{n+1} & = \\
u[x/u\alpha_m][x'/u\alpha_n] & \longrightarrow_C \\
u[x/u\alpha_m[x'/u\alpha_n]] & = \\
u[x/u\alpha_m\alpha_{n+1}] &
\end{array}
$$

∎

Lemma 0.19 The calculus $\lambda\mathtt{xc}$ does not preserve β-strong normalization.

Proof. Let $M = \lambda u.(\lambda x.(\lambda y.u)u)((\lambda x.u)u)$. This term can be well-typed in the simply typed λ-calculus using for example the type $A \rightarrow A$. The term M is β-strongly normalizing. Let us show that M is not $\lambda\mathtt{xc}$-strongly normalizing. We have the following infinite $\lambda\mathtt{xc}$-reduction sequence:

$$
\begin{array}{ll}
\lambda u.(\lambda x.(\lambda y.u)u)((\lambda x.u)u) & \longrightarrow_B \\
\lambda u.((\lambda y.u)u)[x/(\lambda x.u)u] & \longrightarrow^*{}_{\mathtt{x}} \\
\lambda u.((\lambda y.u[x/(\lambda x.u)u])u[x/(\lambda x.u)u]) & = \\
\lambda u.((\lambda y.u\alpha_0)u\alpha_0) & \longrightarrow_B \\
\lambda u.(u\alpha_0)[y/u\alpha_0] & = \\
\lambda u.(u\alpha_0\alpha_1) & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_1\alpha_2 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_0\alpha_2 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_2\alpha_3 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_1\alpha_3 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_0\alpha_3 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_3\alpha_4 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_2\alpha_4 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_1\alpha_4 \ldots & \longrightarrow^*{}_{\lambda\mathtt{xc}} \\
\ldots u\alpha_0\alpha_4 \ldots & \ldots
\end{array}
$$

■

λx-Strong Normalization

The $\lambda\mathtt{x}$-calculus is not strongly normalizing as the following example shows:

$$
\begin{array}{ll}
(\lambda y.yy)(\lambda y.yy) & \longrightarrow_B \\
yy[y/(\lambda y.yy)] & \longrightarrow_{App} \\
y[y/(\lambda y.yy)]y[y/(\lambda y.yy)] & \longrightarrow_{Var1} \\
(\lambda y.yy)y[y/(\lambda y.yy)] & \longrightarrow_{Var1} \\
(\lambda y.yy)(\lambda y.yy) &
\end{array}
$$

However, we can expect $\lambda\mathtt{x}$ to be terminating on *well-typed* terms. We will prove in this section that every typable term is strongly normalizing.

Remark 0.20 If $\Gamma \vdash M : A$, then $\Gamma \vdash_{\lambda\mathtt{x}} M : A$.

Lemma 0.21 If $\Gamma \vdash_{\lambda\mathtt{x}} M : A$, then $\Gamma \vdash \mathtt{x}(M) : A$.

Proof. By induction on the structure of M. ■

First Proof

Theorem 0.22 (Strong Normalization) Every typable $\lambda\mathtt{x}$-term M is in $SN_{\lambda\mathtt{x}}$.

Proof.[By Herbelin] Let us define the following translation $\mathcal{B}()$ from $\lambda\mathtt{x}$-terms to λ-terms:

$$\begin{array}{lcl} \mathcal{B}(x) & = & x \\ \mathcal{B}(MN) & = & \mathcal{B}(M)\mathcal{B}(N) \\ \mathcal{B}(\lambda x.M) & = & \lambda x.\mathcal{B}(M) \\ \mathcal{B}(M[x/N]) & = & (\lambda x.\mathcal{B}(M))\mathcal{B}(N) \end{array}$$

Thus for example, $\mathcal{B}((x[x/y]z)[w/(w_1w_2)]) = (\lambda w.((\lambda x.x)y)z)(w_1w_2)$.

We remark that for every $\lambda\mathtt{x}$-term one has $\mathcal{B}(M) \longrightarrow^*_{\lambda\mathtt{x}} M$. We also remark that when M is typable in $\lambda\mathtt{x}$, then also $\mathcal{B}(M)$ is typable in $\lambda\mathtt{x}$. By Lemma 0.21 the term $\mathtt{x}(\mathcal{B}(M)) = \mathcal{B}(M)$ is also typable in simply typed λ-calculus and thus it is in SN_β. As a consequence we have that $\mathcal{B}(M)$ is in $SN_{\lambda\mathtt{x}}$ by Theorem 0.16 and thus M is necessarily in $SN_{\lambda\mathtt{x}}$. ▪

Remark: This proof technique, which is very simple in this case, needs some additional work to be applied to other calculi.

Second Proof

Remark 0.23 If M is a typable term in $\lambda\mathtt{x}$, then every subterm of M is also typable in $\lambda\mathtt{x}$.

Theorem 0.24 (Strong Normalization) Every typable $\lambda\mathtt{x}$-term M is in $SN_{\lambda\mathtt{x}}$.

Proof. [By Bloo] Let us consider a typable $\lambda\mathtt{x}$-term M, that is, we have that $\Gamma \vdash_{\lambda\mathtt{x}} M : A$ is derivable. We want to show that $M \in \mathcal{F}$. Take any subterm N of M. We know that N is also typable in $\lambda\mathtt{x}$, that is, we have that $\Delta \vdash_{\lambda\mathtt{x}} N : B$ is derivable. By Lemma 0.21 we know that $\Delta \vdash \mathtt{x}(N) : B$ is derivable so that $\mathtt{x}(N) \in SN_\beta$ since typable terms are strongly normalizing in the simply typed λ-calculus. By definition the $\lambda\mathtt{x}$-term M is in $\mathcal{F}$ and by Proposition 0.15 we have $M \in SN_{\lambda\mathtt{x}}$ which concludes the proof of this Theorem. ▪

Remark: This proof technique applies more directly to other calculi.