

Optimality and the Linear Substitution Calculus

FSCD 2017

Pablo Barenbaum

Universidad de Buenos Aires
Université Paris 7
CONICET

Eduardo Bonelli

Universidad Nacional de Quilmes
CONICET

Outline

1. **Review: The Linear Substitution Calculus**
2. **Review: Finite Family Developments**
3. **Lévy Labels for the Linear Substitution Calculus**
4. **Applications**
 - **Optimality**
 - **Standardization**
 - **Normalization of a call-by-need strategy**

The Linear Substitution Calculus (LSC)

LSC is a **calculus of explicit substitutions**.

Introduced by Accattoli and Kesner [CSL'10].

Inspired by an earlier calculus of Milner.

Based on **distant interaction** using **contextual rules**.

The Linear Substitution Calculus (LSC)

Syntax

$t ::= x \mid \lambda x.t \mid t t \mid t[x \setminus t]$	terms
$C ::= \square \mid \lambda x.C \mid C t \mid t C \mid C[x \setminus t] \mid t[x \setminus C]$	contexts
$L ::= \square \mid L[x \setminus t]$	substitution contexts (lists of substitutions)

Reduction rules

$(\lambda x.t)L s \rightarrow t[x \setminus s]L$	distant beta	(dB)
$C\langle\langle x \rangle\rangle[x \setminus t] \rightarrow C\langle t \rangle[x \setminus t]$	linear substitution	(ls)
$t[x \setminus s] \rightarrow t$	if $x \notin \text{fv}(t)$	garbage collection (gc)

Structural equivalence

$\lambda x.t[y \setminus s] \sim (\lambda x.t)[y \setminus s]$	if $x \notin \text{fv}(s)$
$t[x \setminus s] u \sim (t u)[x \setminus s]$	if $x \notin \text{fv}(u)$
$t[x \setminus s][y \setminus u] \sim t[y \setminus u][x \setminus s]$	if $x \notin \text{fv}(u)$ and $y \notin \text{fv}(s)$

The Linear Substitution Calculus (LSC)

Known properties of LSC

- ▶ Preservation of strong normalization.
- ▶ Confluence on open terms.
- ▶ Simulation of β -reduction.
- ▶ Full composition.

Various works, culminating in [Accattoli and Kesner, CSL'10].

- ▶ LSC modulo \sim is **isomorphic to its encoding in proof-nets**.

Various works; cf. [Di Cosmo+, FoSSaCS'03] and Accattoli's 2010 PhD thesis.

- ▶ LSC admits a **notion of residuals** and **standardization**.

[Accattoli+, POPL'14].

- ▶ Evaluation strategies correspond to **abstract machines**.

[Accattoli+, ICFP'14].

Finite Developments (FD)

Definition (Development)

A **development** of a set of cointial steps \mathcal{M} is a possibly infinite sequence:

$$t_0 \xrightarrow{R_1} t_1 \xrightarrow{R_2} t_2 \dots$$

such that each R_i is a residual of some step in \mathcal{M} .

Theorem (Curry)

If \mathcal{M} is a set of cointial steps in the λ -calculus:

1. There are no infinite developments of \mathcal{M} .
2. Maximal developments of \mathcal{M} end on the same term.
3. Residuals by maximal developments of \mathcal{M} are the same.

Finite Developments (FD)

Some derivations are not developments:

$$(\lambda x.xy)I \xrightarrow{R} Iy \xrightarrow{S} y$$

The step **S** is **created**, *i.e.* it has no ancestor.

Finite Family Developments (FFD)

FD can be generalized to involve also created steps.

The notion of **residual** is generalized to the notion of **family**.

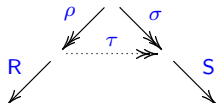
Definition (Step with history)

An **hstep** is a non-empty sequence of steps ρR .

(The last step is singled out).

Definition (Family)

An hstep σS is a **copy** of ρR (written $\rho R \leq \sigma S$) if there exists a derivation τ such that $\rho\tau \equiv \sigma$ and $S \in R/\tau$:



Zig-zag $\Leftarrow\rightsquigarrow$ is the least equivalence relation containing \leq .

A **family** is an equivalence class of $\Leftarrow\rightsquigarrow$.

Finite Family Developments (FFD)

Definition (Family Development)

If \mathcal{F} is a set of coinital families, a **family development** of \mathcal{F} is a possibly infinite sequence:

$$R_1 R_2 \dots R_n \dots$$

such that the family of each hstep $R_1 \dots R_n$ is in \mathcal{F} .

Theorem (Lévy, 1980)

If \mathcal{F} is a finite set of families in the λ -calculus:

1. *There are no infinite family developments of \mathcal{F} .*
2. *Maximal family developments of \mathcal{F} end on the same term.*
3. *Residuals by maximal family developments of \mathcal{F} are the same.*

Redex families for LSC

In this work, we introduce a variant of the LSC with **Lévy labels** to study families and prove FFD.

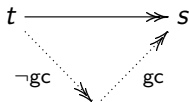
The gc rule poses some problems.

(In this paper) we ignore the gc rule altogether.

For the most part at no loss of generality:

- ▶ gc steps do not interfere with dB or ls steps.
- ▶ gc steps can be postponed.

Every derivation $t \rightarrow s$ can be factorized as follows:



The LSC with Lévy labels (LLSC)

Syntax

$\alpha ::= \mathbf{a} \mid \bar{\alpha} \mid \underline{\alpha} \mid \alpha\alpha \mid \mathbf{dB}(\alpha)$ labels
 $t ::= x^\alpha \mid \lambda^\alpha x.t \mid \mathbb{C}^\alpha(t, t) \mid t[x \setminus t]$ labeled terms

Adding a label to a term

$\alpha : x^\beta \stackrel{\text{def}}{=} x^{\alpha\beta}$
 $\alpha : \lambda^\beta x.t \stackrel{\text{def}}{=} \lambda^{\alpha\beta} x.t$
 $\alpha : \mathbb{C}^\beta(t, s) \stackrel{\text{def}}{=} \mathbb{C}^{\alpha\beta}(t, s)$
 $\alpha : t[x \setminus s] \stackrel{\text{def}}{=} (\alpha : t)[x \setminus s]$

Outermost and innermost sublabel

$\uparrow(\alpha_1\alpha_2) \stackrel{\text{def}}{=} \uparrow(\alpha_1)$
 $\uparrow(\alpha) \stackrel{\text{def}}{=} \alpha \quad (\alpha \neq \alpha_1\alpha_2)$
 $\downarrow(\alpha_1\alpha_2) \stackrel{\text{def}}{=} \downarrow(\alpha_1)$
 $\downarrow(\alpha) \stackrel{\text{def}}{=} \alpha \quad (\alpha \neq \alpha_1\alpha_2)$

Reduction rules

$\mathbb{C}^\alpha((\lambda^\beta x.t)L, s) \rightarrow \alpha \overline{\mathbf{dB}(\beta)} : t[x \setminus \underline{\mathbf{dB}(\beta)} : s]L$

$\mathbf{C}\langle\langle x^\alpha \rangle\rangle[x \setminus t] \rightarrow \mathbf{C}\langle\alpha \bullet : t\rangle[x \setminus t]$

Name of the step

$\mathbf{dB}(\beta)$

$\downarrow(\alpha) \bullet \uparrow(t)$

Key properties of LLSC

Lemma (LLSC is well-defined modulo \sim)

Structural equivalence \sim is a strong bisimulation.

Names of steps are preserved by \sim .

Example

$$\underline{x^a[x \setminus \lambda^b y.t][z \setminus s]} \xrightarrow{a \bullet b} (\lambda^{a \bullet b} y.t[z \setminus s])[x \setminus \lambda^b y.t][z \setminus s]$$

\sim \sim

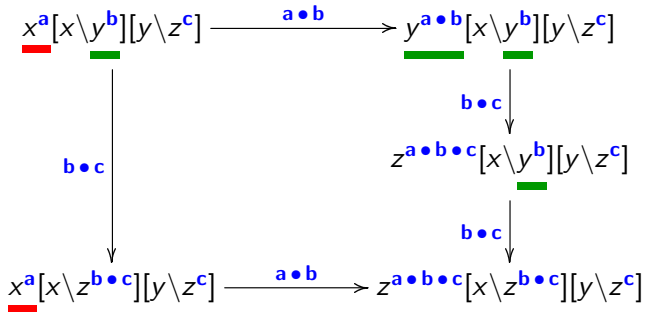
$$\underline{x^a[x \setminus (\lambda^b y.t)][z \setminus s]} \xrightarrow{a \bullet b} (\lambda^{a \bullet b} y.t)[z \setminus s][x \setminus (\lambda^b y.t)][z \setminus s]$$

Key properties of LLSC

Lemma (COPY)

Hsteps in the same family have the same name.

Example



Key properties of LLSC

Lemma (CREATION)

If R creates S , the name of R is a sublabel of the name of S .

Example (ls creates dB)

$$\textcircled{a}(\underline{x^b}, t)[x \setminus \lambda^c y.z^d]$$

$$\xrightarrow{b \bullet c} \textcircled{a}(\underline{(\lambda^{b \bullet c} y.z^d)}, t)[x \setminus \lambda^c y.z^d]$$

$$\xrightarrow{dB(b \bullet c)} z^{\overline{a \, dB(b \bullet c)} \, d} [y \setminus \underline{dB(b \bullet c)} : t][x \setminus \lambda^c y.z^d]$$

Key properties of LLSC

Proposition (Bounded termination)

Reduction in LLSC is SN if the height of the names is bounded.

The proof relies on Klop–Nederpelt's lemma:

$$\text{Inc} \wedge \text{WCR} \wedge \text{WN} \implies \text{SN}$$

Corollary (FFD)

*Finite Family Developments holds for the **unlabeled** LSC.*

Corollary (Confluence)

Labeled reduction is confluent.

A consequence of WCR and SN for bounded names.

Key properties of LLSC

Proposition (CONTRIBUTION)

The following are equivalent:

Syntactic contribution

A name M is a sublabel of a name N .

Semantic contribution

For every hstep ρR such that the name of R is N there is a step S in ρ whose name is M .

Key properties of LLSC

The properties above can be summed up as follows:

Theorem

*LSC forms a **Deterministic Family Structure (DFS)** as defined in [Glauert and Khasidashvili, '96].*

Applications

Optimal reduction

Definition

A step $R : t \rightarrow s$ is **X -needed** if every reduction $t \twoheadrightarrow t' \in X$ contracts a residual of R .

Theorem (Glauert and Khasidashvili '96, generalizing Lévy '80)

If X is a stable set of terms in a DFS, and $\mathcal{M}_1 \dots \mathcal{M}_n$ such that:

- ▶ Each \mathcal{M}_i is a maximal set of steps in the same family.
- ▶ Each \mathcal{M}_i contains at least a X -needed step.
- ▶ The target is a term in X .

Then $\mathcal{M}_1 \dots \mathcal{M}_n$ reaches a term in X in an optimal number of multisteps.

Corollary (Optimality for LSC)

This holds for LSC taking $X := \{t \mid \text{nf}_{\text{gc}}(t) \text{ is in normal form}\}$.

Standardization by selection

Definition (Selection strategy)

For each term t let $<_t$ be any **strict partial order** on the set steps going out from t .

If ρ is a non-empty derivation, $\mathbb{M}(\rho)$ selects a multistep:

$$\mathbb{M}(\rho) \stackrel{\text{def}}{=} \{R \mid R/\rho = \emptyset \text{ and } R \text{ is minimal for } <_{\text{src}(\rho)}\}$$

If ρ is a derivation, $\mathbb{M}^*(\rho)$ builds a sequence of multisteps:

$$\begin{aligned} \mathbb{M}^*(\epsilon) &\stackrel{\text{def}}{=} \epsilon \\ \mathbb{M}^*(\rho) &\stackrel{\text{def}}{=} \mathbb{M}(\rho) \mathbb{M}^*(\rho/\mathbb{M}(\rho)) \quad \text{if } \rho \text{ is non-empty} \end{aligned}$$

Theorem (Standardization for LSC without gc)

1. $\mathbb{M}^*(\rho)$ is well-defined and computable. **(Relies on FFD).**
2. If $\rho \equiv \sigma$ then $\mathbb{M}^*(\rho) = \mathbb{M}^*(\sigma)$.
3. For every ρ there is a unique σ such that $\rho \equiv \sigma$ and $\mathbb{M}^*(\sigma) = \sigma$.

Normalization of linear call-by-need

The **linear call-by-need** strategy is an evaluation strategy for LSC.

$N ::= \square \mid N t \mid N[x \setminus t] \mid N \langle\langle x \rangle\rangle [x \setminus N]$ evaluation contexts

Reduction rules (closed under evaluation contexts)

$$\begin{array}{l} (\lambda x.t)L s \rightarrow_{\text{need}} t[x \setminus s] \\ N \langle\langle x \rangle\rangle [x \setminus vL] \rightarrow_{\text{need}} N \langle vL \rangle [x \setminus vL] \quad \text{if } v = \lambda y.t \end{array}$$

Normal forms

$$A \in \text{NLNF} ::= \begin{array}{l} (\lambda x.t)L \quad \text{answers} \\ \mid N \langle\langle x \rangle\rangle \quad \text{structures} \end{array}$$

Theorem (Normalization of linear call-by-need)

If $t = s$ for some $s \in \text{NLNF}$ then $\rightarrow_{\text{need}}$ terminates and reaches a term in NLNF.

The proof relies on FFD.

Conclusions

LSC (without gc) can be endowed with a notion of Lévy labels.

More abstractly, LSC forms a Deterministic Family Structure.

This can be exploited to prove further results:

- Optimality.
- Standardization.
- Normalization of strategies (in particular: linear call-by-need).