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

$$
\begin{array}{lr}
t ::= x \mid \lambda x.t \mid t\, t \mid t[x\backslash t] & \text{terms} \\
\mathtt{C} ::= \Box \mid \lambda x.\mathtt{C} \mid \mathtt{C}\, t \mid t\, \mathtt{C} \mid \mathtt{C}[x\backslash t] \mid t[x\backslash \mathtt{C}] & \text{contexts} \\
\mathtt{L} ::= \Box \mid \mathtt{L}[x\backslash t] & \text{substitution contexts} \\
 & \text{(lists of substitutions)}
\end{array}
$$

Reduction rules

$$
\begin{array}{rclcrr}
(\lambda x.t)\mathtt{L}\, s & \to & t[x\backslash s]\mathtt{L} & & \text{distant beta} & (\mathrm{dB}) \\
\mathtt{C}\langle\!\langle x\rangle\!\rangle[x\backslash t] & \to & \mathtt{C}\langle t\rangle[x\backslash t] & & \text{linear substitution} & (\mathrm{ls}) \\
t[x\backslash s] & \to & t & \text{if } x \notin \mathrm{fv}(t) & \text{garbage collection} & (\mathrm{gc})
\end{array}
$$

Structural equivalence

$$
\begin{array}{rclr}
\lambda x.t[y\backslash s] & \sim & (\lambda x.t)[y\backslash s] & \text{if } x \notin \mathrm{fv}(s) \\
t[x\backslash s]\, u & \sim & (t\, u)[x\backslash s] & \text{if } x \notin \mathrm{fv}(u) \\
t[x\backslash s][y\backslash u] & \sim & t[y\backslash u][x\backslash s] & \text{if } x \notin \mathrm{fv}(u) \text{ and } y \notin \mathrm{fv}(s)
\end{array}
$$

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 coinitial steps $\mathcal{M}$ is a possibly infinite sequence:

$$t_0 \xrightarrow{\mathsf{R}_1} t_1 \xrightarrow{\mathsf{R}_2} t_2 \ldots$$

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

Theorem (Curry)

If $\mathcal{M}$ is a set of coinitial 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:

$$(\lambda x.xy)I \overset{\mathsf{R}}{\to} Iy \overset{\mathsf{S}}{\to} y$$

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 $\rho \mathsf{R}$.
(The last step is singled out).

Definition (Family)

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

$$\begin{tikzcd}
 & & {} \arrow[dl, ->>, "\rho"'] \arrow[dr, ->>, "\sigma"] & & \\
 & {} \arrow[dl, "\mathsf{R}"'] \arrow[rr, dotted, ->>, "\tau"] & & {} \arrow[dr, "\mathsf{S}"] & \\
 {} & & & & {}
\end{tikzcd}$$

Zig-zag $\leftrightsquigarrow$ is the least equivalence relation containing $\leq$.
A **family** is an equivalence class of $\leftrightsquigarrow$.

Finite Family Developments (FFD)

Definition (Family Development)

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

$$\mathsf{R}_1 \mathsf{R}_2 \ldots \mathsf{R}_\mathsf{n} \ldots$$

such that the family of each hstep $\mathsf{R}_1 \ldots \mathsf{R}_\mathsf{n}$ 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 $t \twoheadrightarrow s$ can be factorized as follows:

$$\begin{tikzcd} t \arrow[rr, two heads] \arrow[dr, dotted, two heads, "\neg\mathrm{gc}"'] & & s \\ & {} \arrow[ur, dotted, two heads, "\mathrm{gc}"'] & \end{tikzcd}$$

The LSC with Lévy labels (LLSC)

Syntax

$$
\begin{array}{lll}
\alpha ::= \ \mathbf{a} \ \mid \ \overline{\alpha} \ \mid \ \underline{\alpha} \ \mid \ \alpha\alpha \ \mid \ \mathtt{dB}(\alpha) & & \text{labels} \\
t ::= \ x^{\alpha} \ \mid \ \lambda^{\alpha}x.t \ \mid \ @^{\alpha}(t,t) \ \mid \ t[x \backslash t] & & \text{labeled terms}
\end{array}
$$

Adding a label to a term

$$
\begin{array}{rcl}
\alpha : x^{\beta} & \stackrel{\text{def}}{=} & x^{\alpha\beta} \\
\alpha : \lambda^{\beta}x.t & \stackrel{\text{def}}{=} & \lambda^{\alpha\beta}x.t \\
\alpha : @^{\beta}(t,s) & \stackrel{\text{def}}{=} & @^{\alpha\beta}(t,s) \\
\alpha : t[x \backslash s] & \stackrel{\text{def}}{=} & (\alpha : t)[x \backslash s]
\end{array}
$$

Outermost and innermost sublabel

$$
\begin{array}{rcll}
\uparrow(\alpha_1\alpha_2) & \stackrel{\text{def}}{=} & \uparrow(\alpha_1) & \\
\uparrow(\alpha) & \stackrel{\text{def}}{=} & \alpha & (\alpha \neq \alpha_1\alpha_2) \\
\downarrow(\alpha_1\alpha_2) & \stackrel{\text{def}}{=} & \downarrow(\alpha_1) & \\
\downarrow(\alpha) & \stackrel{\text{def}}{=} & \alpha & (\alpha \neq \alpha_1\alpha_2)
\end{array}
$$

Reduction rules

$$
\begin{array}{rcl|r}
 & & & \text{Name of the step} \\
@^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}, s) & \to & \alpha\overline{\mathtt{dB}(\beta)} : t[x \backslash \underline{\mathtt{dB}(\beta)} : s]\mathtt{L} & \mathtt{dB}(\beta) \\
\mathtt{C}\langle\!\langle x^{\alpha} \rangle\!\rangle[x \backslash t] & \to & \mathtt{C}\langle \alpha\bullet : t \rangle[x \backslash t] & \downarrow(\alpha)\bullet\uparrow(t)
\end{array}
$$

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

$$\begin{array}{ccc}
\underline{x^{\mathbf{a}}}[x\backslash\lambda^{\mathbf{b}}y.t[z\backslash s]] & \xrightarrow{\mathbf{a}\bullet\mathbf{b}} & (\lambda^{\mathbf{a}\bullet\mathbf{b}}y.t[z\backslash s])[x\backslash\lambda^{\mathbf{b}}y.t[z\backslash s]] \\
\sim & & \sim \\
\underline{x^{\mathbf{a}}}[x\backslash(\lambda^{\mathbf{b}}y.t)[z\backslash s]] & \xrightarrow{\mathbf{a}\bullet\mathbf{b}} & (\lambda^{\mathbf{a}\bullet\mathbf{b}}y.t)[z\backslash s][x\backslash(\lambda^{\mathbf{b}}y.t)[z\backslash s]]
\end{array}$$

Key properties of LLSC

Lemma (Copy)

Hsteps in the same family have the same name.

Example

$$
\begin{tikzcd}
x^{\mathbf{a}}[x\backslash y^{\mathbf{b}}][y\backslash z^{\mathbf{c}}] \arrow[r, "\mathbf{a}\bullet\mathbf{b}"] \arrow[dd, "\mathbf{b}\bullet\mathbf{c}"'] & y^{\mathbf{a}\bullet\mathbf{b}}[x\backslash y^{\mathbf{b}}][y\backslash z^{\mathbf{c}}] \arrow[d, "\mathbf{b}\bullet\mathbf{c}"'] \\
 & z^{\mathbf{a}\bullet\mathbf{b}\bullet\mathbf{c}}[x\backslash y^{\mathbf{b}}][y\backslash z^{\mathbf{c}}] \arrow[d, "\mathbf{b}\bullet\mathbf{c}"'] \\
x^{\mathbf{a}}[x\backslash z^{\mathbf{b}\bullet\mathbf{c}}][y\backslash z^{\mathbf{c}}] \arrow[r, "\mathbf{a}\bullet\mathbf{b}"] & z^{\mathbf{a}\bullet\mathbf{b}\bullet\mathbf{c}}[x\backslash z^{\mathbf{b}\bullet\mathbf{c}}][y\backslash z^{\mathbf{c}}]
\end{tikzcd}
$$

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)

$$\mathtt{@}^{\mathbf{a}}(\underline{x^{\mathbf{b}}}, t)[x \backslash \lambda^{\mathbf{c}} y.z^{\mathbf{d}}]$$

$$\xrightarrow{\mathbf{b} \bullet \mathbf{c}} \quad \underline{\mathtt{@}^{\mathbf{a}}((\lambda^{\mathbf{b} \bullet \mathbf{c}} y.z^{\mathbf{d}}), t)}[x \backslash \lambda^{\mathbf{c}} y.z^{\mathbf{d}}]$$

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

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:

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

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 $\rho\mathsf{R}$ 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 $\mathsf{R} : t \to s$ 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 $\mathcal{M}_1 \ldots \mathcal{M}_n$ 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 $\mathcal{M}_1 \ldots \mathcal{M}_n$ reaches a term in X in an optimal number of multisteps.

Corollary (Optimality for LSC)

This holds for LSC taking $X := \{t \mid \mathsf{nf}_{\mathsf{gc}}(t)$ 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:

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

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

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

Theorem (Standardization for LSC without gc)

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

Normalization of linear call-by-need

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

$$\mathtt{N} ::= \Box \mid \mathtt{N}\, t \mid \mathtt{N}[x\backslash t] \mid \mathtt{N}\langle\!\langle x\rangle\!\rangle[x\backslash \mathtt{N}] \qquad \text{evaluation contexts}$$

Reduction rules (closed under evaluation contexts)

$$\begin{array}{rcll} (\lambda x.t)\mathtt{L}\, s & \to_{\mathsf{need}} & t[x\backslash s] & \\ \mathtt{N}\langle\!\langle x\rangle\!\rangle[x\backslash \mathtt{vL}] & \to_{\mathsf{need}} & \mathtt{N}\langle \mathtt{vL}\rangle[x\backslash \mathtt{vL}] & \text{if } \mathtt{v} = \lambda y.t \end{array}$$

Normal forms

$$\begin{array}{rcll} A \in \mathsf{NLNF} & ::= & (\lambda x.t)\mathtt{L} & \text{answers} \\ & \mid & \mathtt{N}\langle\!\langle x\rangle\!\rangle & \text{structures} \end{array}$$

Theorem (Normalization of linear call-by-need)

If $t = s$ for some $s \in \mathsf{NLNF}$ then $\to_{\mathsf{need}}$ 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).