Explicit operators and proof-nets

Delia Kesner, Stéphane Lengrand

PPS, Paris VII

18 December 2004

TYPES 2004

- Introduction
- Syntax and Typing in $\lambda\mathsf{lxr}$
- Reduction in $\lambda\mathsf{lxr}$
- Connexion with λ-calculus
- Strong Normalisation of $\lambda\mathsf{lxr}$ & Proof-nets modulo

Explicit substitutions and Strong Normalisation

ES used to compute β-reduction

$(\lambda x.u)v \to u\langle x = v\rangle$ + propagation of $u\langle x = v\rangle$

but final instantiation can be delayed

$\Longrightarrow$ ES used for unification (Dowek, Hardin, Kirchner)

Composition?

$t\langle y = v\rangle\langle x = u\rangle \longrightarrow t\langle x = u\rangle\langle y = v\langle x = u\rangle\rangle$

Not terminating.

even $t\langle y = v\rangle\langle x = u\rangle \longrightarrow t\langle y = v\langle x = u\rangle\rangle$ if $x \notin t$ (Mellies)

β-reduction and cut-elimination in proof-nets

Di Cosmo, Guerrini:

With equivalence classes of proof-nets (associativity of contraction,...), cut-elimination
-is still terminating
-simulates β-reduction

But proof-nets have full composition!!

Full composition=
$t\langle x = v\rangle \longrightarrow^* t\{x = v\}$
even when t contains non-evaluated substitutions

Towards a terminating ES-calc with full composition

λ_{ws}: partial composition + strong normalisation

(Guillaume, David)

Does not simulate β-reduction exactly.

Mellies' counter-example avoided using *labels*

What are labels?

Typed case: label = weakening of linear logic

"resource-aware" logic

Untyped case?... ...Resource control:

weakening $\Longrightarrow$ erasure operator

contraction? $\Longrightarrow$ duplication operator

$\Longrightarrow$ λlxr

Syntax & Typing

Syntax

$(Terms)$	$t, t' ::=$	x	*variable*
	$\mid$	$\lambda x.t$	*abstraction*
	$\mid$	$t\ t'$	*application*
	$\mid$		
	$\mid$		
	$\mid$		

Syntax

$(Terms)$	$t, t' ::=$	x	*variable*
	$\mid$	$\lambda x.t$	*abstraction*
	$\mid$	$t\ t'$	*application*
	$\mid$	$t\langle x = t' \rangle$	*substitution*
	$\mid$		
	$\mid$		

Syntax

$(Terms)$	$t, t' ::=$	x	*variable*
	$\mid$	$\lambda x.t$	*abstraction*
	$\mid$	$t\ t'$	*application*
	$\mid$	$t\langle x = t' \rangle$	*substitution*
	$\mid$		
	$\mid$		

$$(B) \qquad (\lambda x.t)\ u \longrightarrow t\langle x = u \rangle$$

Syntax

$$
\begin{array}{llll}
(Terms) & t, t' ::= & x & \textit{variable} \\
 & \mid & \lambda x.t & \textit{abstraction} \\
 & \mid & t\ t' & \textit{application} \\
 & \mid & t\langle x = t' \rangle & \textit{substitution} \\
 & \mid & W_x(t) & \textit{weakening} \\
 & \mid & C_x^{y,z}(t) & \textit{contraction}
\end{array}
$$

$$(B) \qquad (\lambda x.t)\ u \longrightarrow t\langle x = u \rangle$$

Syntax

$$
\begin{array}{|llll|}
\hline
(Terms) \quad t, t' ::= & x & & variable \\
 & \mid & \lambda x.t & abstraction \\
 & \mid & t\ t' & application \\
 & \mid & t\langle x = t' \rangle & substitution \\
 & \mid & W_x(t) & weakening \\
 & \mid & C_x^{y,z}(t) & contraction \\
\hline
\end{array}
$$

$$(B) \qquad (\lambda x.t)\ u \longrightarrow t\langle x = u \rangle$$

We only consider *well-formed* terms

- Linearity
- Compulsory presence

+ Barendregt's convention

$$\frac{}{x : A \vdash x : A} \quad (Axiom) \qquad \frac{\Gamma, x : B \vdash t : A \qquad \Delta \vdash M : B}{\Gamma, \Delta \vdash t\langle x = M \rangle : A} \quad (Subs)$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Delta \vdash v : A}{\Gamma, \Delta \vdash (t\ v) : B} \quad (App) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B} \quad (Lambda)$$

$$\frac{\Gamma, x : A, y : A \vdash M : B}{\Gamma, z : A \vdash C_z^{x,y}(M) : B} \quad (Cont) \qquad \frac{\Gamma \vdash t : A}{\Gamma, x : B \vdash W_x(t) : A} \quad (Weak)$$

Congruence & reduction

Congruence & propagation

Commutativity of substitutions:

$$t\langle x = u\rangle\langle y = v\rangle \equiv t\langle y = v\rangle\langle x = u\rangle$$

$$\text{if } y \notin FV(u) \;\&\; x \notin FV(v)$$

+ AC of contraction + C of weakening

System x				
(Var)	$x\langle x=u\rangle$	$\longrightarrow$	u	
$(Weak1)$	$W_x(t)\langle x=u\rangle$	$\longrightarrow$	$W_{FV(u)}(t)$	
$(Cont1)$	$C_x^{y,z}(t)\langle x=u\rangle$	$\longrightarrow$	$C_{FV(u)}^{FV(u_1),FV(u_2)}(t\langle y=u_1\rangle\langle z=u_2\rangle)$	
	$C[t]\langle x=u\rangle$	$\longrightarrow$	$C[t\langle x=u\rangle]$	$x\in t$
$(Comp)$	$t\langle y=v\rangle\langle x=u\rangle$	$\longrightarrow$	$t\langle y=v\langle x=u\rangle\rangle$	$x\in FV(v)$

System r				
	$C[W_x(t)]$	$\longrightarrow$	$W_x(C[t])$	x not bound by $C[\ ]$
	$C_x^{y,z}(C[t])$	$\longrightarrow$	$C[C_x^{y,z}(t)]$	$y,z\in t$
$(Merge)$	$C_w^{y,z}(W_y(t))$	$\longrightarrow$	$R_z^w(t)$	
$(Cross)$	$C_w^{y,z}(W_x(t))$	$\longrightarrow$	$W_x(C_w^{y,z}(t))$	$x\neq y,\ \ x\neq z$

Properties

Full composition

Free variables are preserved:

If $s \longrightarrow_{B\mathsf{xr}} s'$, then $FV(s) = FV(s')$. (Interface preserving)

Subject reduction:

If $\Gamma \vdash s : A$ et $s \longrightarrow_{B\mathsf{xr}} s'$, then $\Gamma \vdash s' : A$.

xr is convergent

Bxr is confluent

breaks Melliès' counter-example of non-terminating composition

Confluence on open terms? (terms with meta-variables)

Using composition. still checking all critical pairs

Connexion with λ-calculus

Encodings

$\mathcal{B}()$ hides resource control

$$t \twoheadrightarrow^*_{\mathsf{xr}} W_\Pi(\mathcal{A}(\mathcal{B}(t))) \qquad \lambda\mathsf{lxr} \; \overset{\mathcal{B}()}{\underset{\mathcal{A}()}{\rightleftarrows}} \; \lambda \qquad t = \mathcal{B}(\mathcal{A}(t))$$

$\mathcal{A}()$ introduces resource operators

Simulations (1)

$$\lambda\text{lxr} \qquad\qquad \lambda$$

$$\begin{array}{ccc} t & & \mathcal{B}(t) \\ \downarrow_B & \xrightarrow{\mathcal{B}()} & \downarrow_{\beta*} \\ t' & & \mathcal{B}(t') \end{array}$$

Simulations (2)

$$\lambda\mathsf{lxr} \qquad\qquad \lambda$$

$$\begin{array}{ccc} t & & \mathcal{B}(t) \\ \downarrow_{\mathsf{xr}} & \xrightarrow{\mathcal{B}()} & || \\ t' & & \mathcal{B}(t') \end{array}$$

Simulations (3)

$$\lambda\mathsf{lxr} \qquad\qquad \lambda$$

$$\begin{array}{ccc} \mathcal{A}(t) & & t \\ \downarrow_{B\mathsf{xr}+} & \xleftarrow{\mathcal{A}()} & \downarrow_{\beta} \\ W_{\Pi}(\mathcal{A}(t')) & & t' \end{array}$$

Properties

Preservation of typing:

If $\Gamma, FV(t) \vdash t : A$ then $FV(t) \vdash W_\Gamma(\mathcal{A}(t)) : A$

If $\Gamma \vdash t : A$ then $\Gamma \vdash \mathcal{B}(t) : A$

Preservation of Strong Normalisation (PSN):

If $M \in SN^\beta$ then $\mathcal{A}(M) \in SN^{B\mathsf{xr}}$.

Proof-nets & Strong normalisation

Either directly from PSN, or. . .

Encoding types:

$$
\begin{array}{lcll}
A^* & = & A & \text{if } A \text{ is an atomic type} \\
(A \rightarrow B)^* & = & ?((A^*)^\perp) \parr B^* &
\end{array}
$$

Encoding terms:

. . . Sound and complete encoding $T(-)$ in proof-nets modulo

Simulation

- If $t \equiv t'$ then $T(t) \sim_E T(t')$.
- If $t \longrightarrow_B t'$ then $T(t) {\longrightarrow^+}_{R_E} T(t')$
- If $t \longrightarrow_{\mathsf{xr}} t'$ then $T(t) {\longrightarrow^*}_{R_E} T(t')$

Typed $\lambda\mathsf{lxr}$ is strongly normalising.

Intersection types

Simply-typed +

$$\frac{\Gamma \vdash t : A_1 \quad \Gamma \vdash t : A_2}{\Gamma \vdash t : A_1 \cap A_2} \quad (\cap - I) \qquad \frac{\Gamma \vdash t : A_1 \cap A_2}{\Gamma \vdash t : A_i} \quad (\cap - E_i)$$

to characterise strong normalisation: $t \in SN$ if and only if $\Gamma \vdash t : A$

true in λ-calculus (Dezani, Coppo)

false in λx: $x\langle y = zz\rangle\langle z = \Omega\rangle \longrightarrow x\langle z = \Omega\rangle \longrightarrow x$ is SN

but same typing as $(\lambda y.x)(zz)\langle z = \Omega\rangle \longrightarrow (\lambda y.x)(\Omega\Omega)$ not SN

should be true again in λlxr, because of composition.

Proof: SN from PSN (Proof-nets do not have intersection types)

Conclusion & future work

The λlxr-calculus, in brief:

- An explicit substitution calculus *à la* λx, confluent...

  ...with explicit resource handling.

  ...simulating λ-calculus.

- Convergence of the propagation of resource operators

- Strong version of composition

Typed λlxr-calculus:

Clear interpretation in linear logic - cut-elimination

$\Longrightarrow$ Strong Normalisation

Future works

- Study strategies in λlxr
- Resource control and optimal reductions
- Get rid of α-conversion? (c.f. Van Oostrom)
- Implementation: director strings?

References

[CKP00] Roberto Di Cosmo, Delia Kesner, and Emmanuel Polonovski. Proof nets and explicit substitutions. *Lecture Notes in Computer Science*, 1784:63–??, 2000.

[Gui99] B. Guillaume. *Un calcul de substitution avec Étiquettes*. PhD thesis, 1999.

[HvO03] D. Hendriks and V. van Oostrom. Adbmal. In *Proceedings of the 19th Conference on Automated Deduction (CADE 19), Miami*, pages 136 – 150, July-August 2003. LNAI 2741.

[Mel95] P.-A. Melliès. Typed λ-calculi with explicit substitution may not terminate. In M. Dezani, editor, *TLCA '95—Int. Conf. on Typed Lambda Calculus and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages

328–334, Edinburgh, Scotland, April 1995.
Springer-Verlag.

Congruence

- Associativity and commutativity of contractions:

$$\begin{array}{rcl} C_w^{x,v}(C_x^{y,z}(t)) & \equiv & C_w^{x,y}(C_x^{z,v}(t)) \\ C_x^{y,z}(t) & \equiv & C_x^{z,y}(t) \\ C_{x'}^{y',z'}(C_x^{y,z}(t)) & \equiv & C_x^{y,z}(C_{x'}^{y',z'}(t)) \end{array}$$

- Commutativity of weakenings:

$$W_x(W_y(t)) \equiv W_y(W_x(t))$$

- Commutativity of substitutions:

$$t\langle x = u\rangle\langle y = v\rangle \equiv t\langle y = v\rangle\langle x = u\rangle$$

$$\text{si } y \notin FV(u) \text{ et } x \notin FV(v)$$