Call-by-value, call-by-name, and strong normalization for the classical sequent calculus
-
Erratum

Stéphane Lengrand

École Normale Supérieure de Lyon
46, Allée d'Italie, 69364 Lyon cedex 07, France
Stephane.Lengrand@ENS-Lyon.fr
and
PPS - Université Denis Diderot
2, Place Jussieu, 75251 Paris cedex 05, France

In [Len03], there was one typo in the CBN-translation of a context in Definition 3.4 and another one in the statement of in Theorem 3.6 (Preservation of types by the CBN-translation of a context). Here is the corrected version.

Those typos were already present in [Len02] (Definition 14, page 21, and Theorem 15, page 22).

3 The $\overline{\lambda}\mu\widetilde{\mu}$-calculus and its semantics

Definition 3.4 We define the semantics of terms as follows:

$$
\begin{array}{rl@{\qquad}rl}
\multicolumn{2}{c}{\textsf{CBV}\text{-translation}} & \multicolumn{2}{c}{\textsf{CBN}\text{-translation}} \\
\overline{x} & := \lambda k.k\ \widetilde{x} & \underline{x} & := \widetilde{x} \\
\overline{\lambda x.v} & := \lambda k.(k\ (\lambda u.\lambda\widetilde{x}.(\overline{v}\ u))) & \underline{\lambda x.v} & := \lambda k.(k\ \lambda\widetilde{x}.\underline{v}) \\
\overline{\mu\alpha.c} & := \lambda\widetilde{\alpha}.\overline{c} & \underline{\mu\alpha.c} & := \lambda\widetilde{\alpha}.\underline{c} \\
\overline{\alpha} & := \widetilde{\alpha} & \underline{\alpha} & := \lambda u.(u\ \widetilde{\alpha}) \\
\overline{v\cdot e} & := \lambda u.(\overline{v}\ (u\ \overline{e})) & \underline{v\cdot e} & := \lambda u.(u\ \lambda h.(\underline{e}\ (h\ \underline{v}))) \\
\overline{\widetilde{\mu}x.c} & := \lambda\widetilde{x}.\overline{c} & \underline{\widetilde{\mu}x.c} & := \lambda\widetilde{x}.\underline{c} \\
\overline{\langle v|\ e\rangle} & := \overline{v}\ \overline{e} & \underline{\langle v|\ e\rangle} & := \underline{e}\ \underline{v}
\end{array}
$$

As we always want to see the connexion with logic, we are concerned with the translation of types. σ denotes a base type (of $\overline{\lambda}\mu\widetilde{\mu}$), R is the type of responses, V_A denotes a type of values, K_A denotes a type of continuations, and C_A denotes a type of computations, all defined as follows:

Definition 3.5

$$
\begin{array}{llll}
\multicolumn{2}{c}{\textsf{CBV}} & \multicolumn{2}{c}{\textsf{CBN}} \\
V_\sigma^{\textsf{CBV}} & := \sigma & V_\sigma^{\textsf{CBN}} & := \sigma \\
V_{A\to B}^{\textsf{CBV}} & := K_B^{\textsf{CBV}} \to K_A^{\textsf{CBV}} & V_{A\to B}^{\textsf{CBN}} & := C_A^{\textsf{CBN}} \to C_B^{\textsf{CBN}} \\
K_A^{\textsf{CBV}} & := V_A^{\textsf{CBV}} \to R & K_A^{\textsf{CBN}} & := V_A^{\textsf{CBN}} \to R \\
C_A^{\textsf{CBV}} & := K_A^{\textsf{CBV}} \to R & C_A^{\textsf{CBN}} & := K_A^{\textsf{CBN}} \to R
\end{array}
$$

Such an interpretation of types enables us to prove the following:

Theorem 3.6 (Preservation of types)

$$
\begin{cases}
\Gamma \vdash v : A|\Delta \Rightarrow & V_\Gamma^{\mathsf{CBV}}, K_\Delta^{\mathsf{CBV}} \vdash \overline{v} : C_A^{\mathsf{CBV}} \\
 & C_\Gamma^{\mathsf{CBN}}, K_\Delta^{\mathsf{CBN}} \vdash \underline{v} : C_A^{\mathsf{CBN}} \\
\Gamma|\ e : B \vdash \Delta \Rightarrow & V_\Gamma^{\mathsf{CBV}}, K_\Delta^{\mathsf{CBV}} \vdash \overline{e} : K_B^{\mathsf{CBV}} \\
 & C_\Gamma^{\mathsf{CBN}}, K_\Delta^{\mathsf{CBN}} \vdash \underline{e} : C_B^{\mathsf{CBN}} \to R \\
c : (\Gamma \vdash \Delta) \Rightarrow & V_\Gamma^{\mathsf{CBV}}, K_\Delta^{\mathsf{CBV}} \vdash \overline{c} : R \\
 & C_\Gamma^{\mathsf{CBN}}, K_\Delta^{\mathsf{CBN}} \vdash \underline{c} : R
\end{cases}
$$

References

[Len02] S. Lengrand. *A computational interpretation of the Cut-rule in classical sequent calculus*. Master thesis, Computing Laboratory, Oxford University, Sept. 2002.

[Len03] S. Lengrand. Call-by-value, call-by-name, and strong normalization for the classical sequent calculus. In B. Gramlich and S. Lucas, editors, *Electronic Notes in Theoretical Computer Science*, volume 86. Elsevier, 2003.