Expansion in a Calculus with Explicit Substitutions

Ana Jorge Almeida, Sandra Alves, and Mário Florido

LIACC, Departamento de Ciência de Computadores
Faculdade de Ciências, Universidade do Porto

Term expansion was first defined in [8] to relate terms typed in an intersection type system with linear terms. Recently, new applications of *term expansion* include the relation with other substructural type systems (relevant and ordered type systems) [2] and quantitative types [3]. Here we define *term expansion* for a calculus with explicit substitutions and apply it to relate a λ-calculus with explicit substitutions [9] with a resource calculus [5], where the argument of a function is a bag of resources, that is, a multiset of terms.

1 Explicit substitutions

Although the λ-calculus [4, 6] is a convenient model for computational functions, it lacks the means for observing operational properties of the execution of such algorithms, mainly due to its implicit β-contraction, which is a meta-operation. There was a necessity to explicitly deal with substitutions, in order to bridge the gap between theory and implementation, allow efficient reduction in implementations, and avoid variable capture and scope issues [1].

Here we will be using a modification of the explicit substitution calculus presented in [9], the λxgc-calculus, which is an adaptation of $\lambda\sigma$ [1], that retains variable names instead of using indices *à la Bruijn* [7], and preserves strong-normalisation.

Definition 1 (λx-preterms)**.** The λx-preterms are the extension of the λ-preterms defined inductively by

$$M ::= x \mid \lambda x.M \mid MN \mid M < x := N >$$

In this calculus, explicit substitution is given highest precedence. It also has explicit garbage collection [10], which is useful and easy to specify using names.

Definition 2 (Evaluation in the λxgc-calculus)**.** The reduction is defined as $\underset{\text{bxgc}}{\longrightarrow} = \underset{\text{b}}{\rightarrow} \cup \underset{\text{x}}{\rightarrow} \cup \underset{\text{gc}}{\longrightarrow}$.

$$(\lambda x.M)N \underset{\text{b}}{\rightarrow} M < x := N > \qquad\qquad x < y := N > \underset{\text{xvgc}}{\longrightarrow} x \text{ if } x \not\equiv y$$

$$x < x := N > \underset{\text{xv}}{\longrightarrow} N \qquad\qquad M < x := N > \underset{\text{gc}}{\longrightarrow} M \text{ if } x \notin fv(M)$$

$$(M_1 M_2) < x := N > \underset{\text{xap}}{\longrightarrow} M_1 < x := N > M_2 < x := N >$$

$$\frac{M \underset{\text{bxgc}}{\longrightarrow} M'}{MN \underset{\text{bxgc}}{\longrightarrow} M'N} \qquad\qquad \frac{(\lambda x.M)N_2 \underset{\text{bxgc}}{\longrightarrow} M' \quad \text{if } y \notin fv(N_2)}{((\lambda x.M) < y := N_1 >)N_2 \underset{\text{bxgc}}{\longrightarrow} M' < y := N_1 >}$$

Example 1. Consider the λx-term $(\lambda x.xx)I$, where $I \equiv \lambda z.z$.

$$\begin{aligned}(\lambda x.xx)I \underset{\text{bxgc}}{\longrightarrow} (xx) < x := I > \underset{\text{bxgc}}{\longrightarrow} x < x := I > x < x := I > \underset{\text{bxgc}}{\longrightarrow} II \quad &\underset{\text{bxgc}}{\longrightarrow} \quad z < z := I > \\ &\underset{\text{bxgc}}{\longrightarrow} \quad I\end{aligned}$$

2 Term Expansion

We now relate expansion with a resource calculus [5], where the standard λ-calculus application MN, is denoted by MN^{∞}, to indicate that the argument N is always available for function M.

Our main focus now is to develop a relation between terms typable by idempotent intersection types and a subset of Boudol's language, which we will refer to as Λ^{∞}, allowing us to express only multiplicities ∞, hence each bag of resources is used without restriction.

Definition 3 (Boudol's terms)**.** The following terms are the syntax of Boudol's λ-calculus with multiplicities.

$$
\begin{array}{rcll}
M & ::= & x \mid \lambda x.M \mid (MP) \mid (M < P/x >) & \text{terms} \\
P & ::= & 1 \mid M \mid (P \mid P) \mid M^{\infty} & \text{bags of terms} \\
V & ::= & \lambda x.M \mid V < P/x > & \text{values}
\end{array}
$$

Definition 4 (Expansion)**.** Given a pair $M : \sigma$, where M is a λ-term and σ an intersection type, and a term N, we define a relation $\mathcal{E}(M : \sigma) \lhd N$, which we call *expansion*:

$$
\begin{array}{rcll}
\mathcal{E}(x : \tau) & \lhd & x & \\
\mathcal{E}(\lambda x.M : \tau_1 \cap \cdots \cap \tau_n \to \sigma) & \lhd & \lambda x.M^* & \text{if } x \in fv(M) \text{ and } \mathcal{E}(M : \sigma) \lhd M^* \\
\mathcal{E}(\lambda x.M : \tau \to \sigma) & \lhd & \lambda x.M^* & \text{if } x \notin fv(M) \text{ and } \mathcal{E}(M : \sigma) \lhd M^* \\
\mathcal{E}(MN : \sigma) & \lhd & (M^*(P_1^{m_1} \mid \cdots \mid P_k^{m_k})) & \text{if for some } k > 0 \text{ and } \tau_1 \ldots \tau_k \text{ such} \\
 & & & \text{that } \mathcal{E}(M : \tau_1 \cap \cdots \cap \tau_k \to \sigma) \lhd M^* \\
 & & & \text{and } \mathcal{E}(N : \tau_i) \lhd P_i^{m_i} \text{ for } 1 \leq i \leq k \\
\mathcal{E}(M < x := N >: \sigma) & \lhd & (M^* < (P_1^{m_1} \mid \cdots \mid P_k^{m_k})/x >) & \text{if for some } k > 0 \text{ and } \tau_1 \ldots \tau_k \text{ such} \\
 & & & \text{that } \mathcal{E}(M : \tau_1 \cap \cdots \cap \tau_k \to \sigma) \lhd M^* \\
 & & & \text{and } \mathcal{E}(N : \tau_i) \lhd P_i^{m_i} \text{ for } 1 \leq i \leq k
\end{array}
$$

Since we are working with Λ^{∞}, we have that for some $k > 0$ and $1 \leq i \leq k$, $m_i = \infty$.

Theorem 1 (Expansion and Multiplicities)**.** Given a λx-term M and a type σ, such that $\mathcal{E}(M : \sigma) \lhd M^*$, if $M \underset{\text{bxgc}}{\twoheadrightarrow} V_1$ then $M^* \twoheadrightarrow_{\text{B}} V_2$ and $\mathcal{E}(V_1 : \sigma) \lhd V_2$.

This theorem is proved by induction on the definition of expansion, and it shows that, if we evaluate a λx-term until it reaches a value, then we are able to expand that initial term, evaluate it in Boudol's system and its result is an expansion of the value obtained in the λx evaluation.

Example 2. Using the term in Example 1, we have

$$\mathcal{E}((\lambda x.xx)I : \sigma) \lhd ((\lambda x.(xx^{\infty}))I^{\infty})$$

because $\mathcal{E}(\lambda x.xx : ((\sigma \to \sigma) \cap \sigma) \to \sigma) \lhd \lambda x.(xx^{\infty})$ and $\mathcal{E}(I : (\sigma \to \sigma) \cap \sigma) \lhd I$.

$$
\begin{array}{rcl}
((\lambda x.(xx^{\infty}))I^{\infty}) \to (xx^{\infty}) < I^{\infty}/x > \to (Ix^{\infty}) < I^{\infty}/x > & \to & z < x^{\infty}/z >< I^{\infty}/x > \\
 & \to & x < x^{\infty}/z >< I^{\infty}/x > \\
 & \to & I < x^{\infty}/z >< I^{\infty}/x > \equiv I
\end{array}
$$

and $\mathcal{E}(I : \sigma) \lhd I$.

3 Conclusion and Future Work

Here we have proved that there exists a relation between ACI-intersection types and a resource calculus that deals with multiplicities $m = \infty$ (infinitely available resources).

This serves as preliminary work towards proving that there exists a relation between AC-intersection types and a resource calculus of finite multiplicities. We also wish to look into an extension of this calculus that deals with α-conversion.

Acknowledgements This work was financially supported by: UID/00027 of the LIACC - Artificial Intelligence and Computer Science Laboratory - funded by Fundação para a Ciência e a Tecnologia, I.P./ MCTES through the national funds.

References

[1] Martin Abadi, Luca Cardelli, P-L Curien, and J-J Lévy. Explicit substitutions. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 31–46, 1989.

[2] Sandra Alves and Mário Florido. Structural rules and algebraic properties of intersection types. In *International Colloquium on Theoretical Aspects of Computing*, pages 60–77. Springer, 2022.

[3] Sandra Alves and Daniel Ventura. Quantitative weak linearisation. In *International Colloquium on Theoretical Aspects of Computing*, pages 78–95. Springer, 2022.

[4] Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985.

[5] Gérard Boudol. The lambda-calculus with multiplicities. In *CONCUR'93: 4th Intrenational Conference on Concurrency Theory Hildesheim, Germany, August 23–26, 1993 Proceedings 4*, pages 1–6. Springer, 1993.

[6] Alonzo Church. A set of postulates for the foundation of logic. *Annals of mathematics*, 33(2):346–366, 1932.

[7] Nicolaas Govert De Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes mathematicae (proceedings)*, volume 75, pages 381–392. Elsevier, 1972.

[8] Mario Florido and Luis Damas. Linearization of the lambda-calculus and its relation with intersection type systems. *Journal of Functional Programming*, 14(5):519–546, 2004.

[9] Kristoffer H Rose. *Explicit substitution: tutorial & survey.* Computer Science Department, 1996.

[10] Kristoffer Høgsbro Rose. Explicit cyclic substitutions. In *International Workshop on Conditional Term Rewriting Systems*, pages 36–50. Springer, 1992.