Generalized λ-calculi
(Abstract)

Hongwei Xi

Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA
email: hwxi@cs.cmu.edu Fax: +1 412 268 6380

We propose a notion of generalized λ-calculi, which include the usual call-by-name λ-calculus, the usual call-by-value λ-calculus, and many other λ-calculi such as the λ_g-calculus[3], the λ^v_{hd}-calculus[5], etc. We prove the Church-Rosser theorem and the standardization theorem for these generalized λ-calculi. The normalization theorem then follows, which enables us to define evaluation functions for the generalized λ-calculi. Our proof technique mainly establishes on the notion of *separating developments*[4], yielding intuitive and clean inductive proofs.

This work aims at providing a solid foundation for *evaluation under λ-abstraction*, a notion which is pervasive in both partial evaluation and run-time code generation for functional programming languages.

Definition 1. We use the following for λ-terms and contexts:

$$\text{(terms)}\ \ L, M, N \ ::= \ x \mid (\lambda x.M) \mid M(N) \qquad \text{(contexts)}\ \ C \ ::= \ [] \mid (\lambda x.C) \mid M(C) \mid C(M)$$

We use $\mathrm{FV}(M)$ for the set of free variables in M.

Definition 2. (General λ-abstraction) We define function abs on λ-terms as follows:

$$\mathrm{abs}(x) = 0 \quad \mathrm{abs}(\lambda x.M) = \mathrm{abs}(M) + 1 \quad \mathrm{abs}(M(N)) = \mathrm{abs}(M) \dot{-} 1$$

Note $n \dot{-} 1 = n - 1$ if $n > 0$ and $0 \dot{-} 1 = 0$. M is a general λ-abstraction if $\mathrm{abs}(M) > 0$.

We use Λ for the set of λ-terms; **lam** for the set of λ-abstractions; **glam** for the set of general λ-abstractions; **var** for the set of variables.

Definition 3. The body of a general λ-abstraction M is defined as $\mathrm{bd}(M) = \mathrm{gbd}(M, 0)$, where gbd is defined as follows.

$$\mathrm{gbd}(\lambda x.M, 0) = M[x := \bullet] \quad \mathrm{gbd}(\lambda x.M, n+1) = \lambda x.\mathrm{gbd}(M, n) \quad \mathrm{gbd}(M(N), n) = \mathrm{gbd}(M, n+1)(N)$$

A general redex is of form $M(N)$ where M is a general λ-abstraction. The contractum of a general redex $M(N)$ is $\beta(M, N) = \mathrm{bd}(M)[\bullet := N]$.

Definition 4. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be sets of λ-terms; we say $\mathcal{S}_1$ is closed under $\mathcal{S}_2$ if $M[x := N] \in \mathcal{S}_1$ for all $M \in S_1$ and $x \in \mathrm{FV}(M)$ and $N \in \mathcal{S}_2$. $\mathcal{R} = \langle \mathcal{F}, \mathcal{V} \rangle$ is a closed redex set(c.r.s.) if $\mathcal{F}$ contains only general λ-abstractions and both $\mathcal{F}$ and $\mathcal{V}$ are closed under $\mathcal{V}$.

Definition 5. Given a closed redex set $\mathcal{R} = \langle \mathcal{F}, \mathcal{V} \rangle$; $M(N)$ is a $\beta_\mathcal{R}$-redex if $M \in \mathcal{F}$ and $N \in \mathcal{V}$; $M_1 \xrightarrow{\beta}_\mathcal{R} M_2$ if $M_1 = C[M(N)]$ for some $\beta_\mathcal{R}$-redex $M(N)$ and $M_2 = C[\beta(M, N)]$; $\xrightarrow{\beta}\!\!\!\!\rightarrow_\mathcal{R}$ is the reflexive and transitive closure of $\xrightarrow{\beta}_\mathcal{R}$; we use σ for a (finite) $\beta_\mathcal{R}$-reduction sequence, and $\sigma(M)$ for the λ-term to which σ reduces M.

Given a c.r.s. $\mathcal{R}$; the general λ-calculus $\lambda_\mathcal{R}$ studies the reduction $\xrightarrow{\beta}_\mathcal{R}$. We write $\lambda_\mathcal{R} \vdash M \equiv_\mathcal{R} N$ if there exist $M = M_0, M_1, \ldots, M_{2n-2}, M_{2n} = N$ such that $M_{2i+1} \xrightarrow{\beta}\!\!\!\!\rightarrow_\mathcal{R} M_{2i}$ and $M_{2i+1} \xrightarrow{\beta}\!\!\!\!\rightarrow_\mathcal{R} M_{2i+2}$ for $0 \leq i < n$.

Remark. The (usual call-by-name) λ-calculus is $\lambda_\mathcal{R}$ for $\mathcal{R} = \langle \mathbf{lam}, \Lambda \rangle$; the (usual) call-by-value λ-calculus is $\lambda_\mathcal{R}$ for $\mathcal{R} = \langle \mathbf{lam}, \mathbf{lam} \cup \mathbf{var} \rangle$; the λ_g in [3] is $\lambda_\mathcal{R}$ for $\mathcal{R} = \langle \mathbf{glam}, \Lambda \rangle$; the λ^v_{hd} in [5] is $\lambda_\mathcal{R}$ for $\mathcal{R} = \langle \mathbf{ghnf}, \mathbf{ghnf} \rangle$, where **ghnf** is the set of λ-terms in generalized head normal form[5]; the call-by-need λ-calculus[1] closely relates to $\lambda_\mathcal{R}$ for $\mathcal{R} = \langle \mathbf{ghnf}, \mathbf{lam} \cup \mathbf{var} \rangle$. It can be readily verified that every $\mathcal{R}$ mentioned above is a c.r.s.

The notion of *residuals* of a $\beta_\mathcal{R}$-redex under $\beta_\mathcal{R}$-reductions can be defined as usual[2]. Note that the conditions imposed on the definition of closed redex set are crucial for making the definition go through.

Definition 6. (Involvedness) Given a $\beta_{\mathcal{R}}$-reduction sequence σ form M; a $\beta_{\mathcal{R}}$-redex in M is involved in σ if the $\beta_{\mathcal{R}}$-redex or one of its residuals is contracted in σ.

Definition 7. ($\beta_{\mathcal{R}}$-development) Given a λ-term M and a set $\mathcal{S}$ of $\beta_{\mathcal{R}}$-redex in M; $\sigma : M \overset{\beta}{\twoheadrightarrow}_{\mathcal{R}} N$ is a $\beta_{\mathcal{R}}$-development (of $\mathcal{S}$) if it contracts only $\beta_{\mathcal{R}}$-redexes in $\mathcal{S}$ and their residuals.

Lemma 8. *(Separation) Let $M = M_1(M_2)$ be a $\beta_{\mathcal{R}}$-redex and σ be a $\beta_{\mathcal{R}}$-development σ from M in which M is involved; $\sigma(M)$ is of form*

$$\sigma_1(\mathrm{bd}(M_1))[\sigma_{21}(M_2), \ldots, \sigma_{2n}(M_2)],$$

where σ_1 is a $\beta_{\mathcal{R}}$-development from $\mathrm{bd}(M_1)$ and σ_{2i} are $\beta_{\mathcal{R}}$-developments from M_2 for $i = 1, \ldots, n$.

Lemma 8 plays a major rôle in the proofs of the following theorems. Please see [4] for details.

Theorem 9. *(Church-Rosser) For any given c.r.s. $\mathcal{R}$, if $\lambda_{\mathcal{R}} \vdash M_1 \equiv_{\mathcal{R}} M_2$, then there exists N such that $M_i \overset{\beta}{\twoheadrightarrow}_{\mathcal{R}} N$ for $i = 1, 2$.*

Definition 10. Let $\mathcal{R} = \langle \mathcal{F}, \mathcal{V} \rangle$ be a c.r.s. and $\beta_{\mathcal{R}}(M)$ be the set of all $\beta_{\mathcal{R}}$-redexes in M for every λ-term M; a relation on $\beta_{\mathcal{R}}(M)$ is given as follows.

$$\begin{array}{rll}
\lhd_{\mathcal{R}}(M) = & \emptyset & \text{if } M \text{ is a variable;} \\
\lhd_{\mathcal{R}}(\lambda x.M) = & \lhd_{\mathcal{R}}(M) & ; \\
\lhd_{\mathcal{R}}(M(N)) = & \lhd_{\mathcal{R}}(M) \cup \lhd_{\mathcal{R}}(N) \cup (\beta_{\mathcal{R}}(N) \times \beta_{\mathcal{R}}(M)) \cup \{\langle M(N), L\rangle : L \in \beta_{\mathcal{R}}(M) \cup \beta_{\mathcal{R}}(N)\} & \\
 & & \text{if } M(N) \text{ is a } \beta_{\mathcal{R}}\text{-redex;} \\
\lhd_{\mathcal{R}}(M(N)) = & \lhd_{\mathcal{R}}(M) \cup \lhd_{\mathcal{R}}(N) \cup (\beta_{\mathcal{R}}(N) \times \beta_{\mathcal{R}}(M)) & \text{if } M \in \mathcal{F} \text{ and } N \notin \mathcal{V}; \\
\lhd_{\mathcal{R}}(M(N)) = & \lhd_{\mathcal{R}}(M) \cup \lhd_{\mathcal{R}}(N) \cup (\beta_{\mathcal{R}}(M) \times \beta_{\mathcal{R}}(N)) & \text{if } M \notin \mathcal{F}.
\end{array}$$

Note that $\lhd_{\mathcal{R}}(M)$ is a linear order for every M; the standard $\beta_{\mathcal{R}}$ reduction sequences can then be defined accordingly, which leads to the following theorem.

Theorem 11. *(Standardization) Given any $\beta_{\mathcal{R}}$-reduction sequence $\sigma : M \overset{\beta}{\twoheadrightarrow}_{\mathcal{R}} N$; then there exists a standard $\beta_{\mathcal{R}}$-reduction sequence $\mathbf{std}_{\mathcal{R}}(\sigma) : M \overset{\beta}{\twoheadrightarrow}_{\mathcal{R}} N$.*

Let the *first* $\beta_{\mathcal{R}}$-redex in M be the first one according to order $\lhd_{\mathcal{R}}(M)$, then the normalizing strategy is the one which always reduces the first $\beta_{\mathcal{R}}$-redex in a term.

Corollary 12. *(Normalization) If $\lambda_{\mathcal{R}} \vdash M \equiv_{\mathcal{R}} N$ for some N in $\beta_{\mathcal{R}}$-normal form, then the normalizing strategy reduces M to N.*

We can then define a evaluation function for $\lambda_{\mathcal{R}}$ according to the normalizing strategy, establishing a functional programming language upon $\lambda_{\mathcal{R}}$.

In conclusion, we have shown that the generalized λ-calculi can unify many existing λ-calculi. We are currently studying λ_{hd}^{v}, investigating its application to partial-evaluation and run-time code generation.

References

1. Z.M. Ariola, M. Felleisen, J. Maraist, M. Odersky and P. Wadler (1996), A Call-by-Need Lambda Calculus, *Proceedings of the 22nd ACM Symposium on Principles of Programming Languages*, San Francisco, pp. 233-246
2. H.P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, Amsterdam, 1984.
3. F. Kamareddine and R. Nederpelt (1996), A useful lambda-notation, *Theoretical Computer Science*, vol. 155(1), pp 85-109.
4. Hongwei Xi (1996), Separating Developments, *Manuscripts.* Avaiable through: `http://www.cs.cmu.edu/~hwxi/papers/SepDev.ps`
5. Hongwei Xi (1997), Evaluation under lambda-abstraction. In Proceedings of Ninth International Symposium on Programming Languages, Implementations, Logics, and Programs(to appear)
6. N. Yoshida (1993), Optimal reduction in weak λ-calculus with shared environments. In *Proc. ACM conference on Functional Programming Languages and Computer Architecture*, Copenhagen.