Generalized $\lambda$-calculi
(abstract)

Hongwei Xi
Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: hwx@cs.cmu.edu Fax: +1 412 268 6380

We propose a notion of generalized $\lambda$-calculi, which include the usual call-by-name $\lambda$-calculus, the usual call-by-value $\lambda$-calculus, and many other $\lambda$-calculi such as the $\lambda_2$-calculus[3], the $\lambda^h_{\eta^r}$-calculus[5], etc. We prove the Church-Rosser theorem and the standardization theorem for these generalized $\lambda$-calculi. The normalization theorem then follows, which enables us to define evaluation functions for the generalized $\lambda$-calculus. 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 $\lambda$-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 $\lambda$-terms and contexts:

\[
L; M; N \quad ::= \quad x \mid (\lambda x.M) \mid M(N)
\]

(contexts)  \[
C \quad ::= \quad [] \mid (\lambda x.C) \mid M(C) \mid C(M)
\]

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

**Definition 2.** (General $\lambda$-abstraction) We define function abs on $\lambda$-terms as follows:

\[
\text{abs}(x) = 0 \quad \text{abs}(\lambda x.M) = \text{abs}(M) + 1 \quad \text{abs}(M(N)) = \text{abs}(M) - 1
\]

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

We use $\lambda$ for the set of $\lambda$-terms; $\text{lam}$ for the set of $\lambda$-abstractions; $\text{glam}$ for the set of general $\lambda$-abstractions; $\text{var}$ for the set of variables.

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

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

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

**Definition 4.** Let $S_1$ and $S_2$ be sets of $\lambda$-terms; we say $S_1$ is closed under $S_2$ if $M[x := N] \in S_1$ for all $M \in S_1$ and $x \in \text{FV}(M)$ and $N \in S_2$, $R = \{F, V\}$ is a closed redex set (c.r.s.) if $F$ contains only general $\lambda$-abstractions and both $F$ and $V$ are closed under $\text{FV}$.

**Definition 5.** Given a closed redex set $R = \{F, V\}; M(N)$ is a $\beta_R$-redex if $M \in F$ and $N \in V$; $M_1 \beta_R M_2$ if $M_1 = C[M(N)]$ for some $\beta_R$-redex $M(N)$ and $M_2 = C[\beta(M, N)]; \beta_R$ is the reflexive and transitive closure of $\beta_R$; we use $\sigma$ for a (finite) $\beta_R$-reduction sequence, and $\sigma(M)$ for the $\lambda$-term to which $\sigma$ reduces $M$.

Given a c.r.s. $R$; the general $\lambda$-calculus $\lambda_R$ studies the reduction $\beta_R$. We write $\lambda_R \vdash M \equiv_R N$ if there exist $M = M_0, M_1, \ldots, M_{2n-2}, M_{2n} = N$ such that $M_{2i+1} \beta_R M_{2i} \quad M_{2i+1} \beta_R M_{2i+2}$ for $0 \leq i < n$.

**Remark.** The (usual call-by-name) $\lambda$-calculus is $\lambda_R$ for $R = \{\text{lam}, \lambda\}$; the (usual) call-by-value $\lambda$-calculus is $\lambda_R$ for $R = \{\text{lam}, \lambda \cup \text{var}\}$; the $\lambda_2$ in [3] is $\lambda_R$ for $R = \{\text{lam}, \lambda\}$; the $\lambda^h_{\eta^r}$ in [5] is $\lambda_R$ for $R = \{\text{ghnf, ghnf}\}$, where $\text{ghnf}$ is the set of $\lambda$-terms in generalized head normal form[5]; the call-by-need $\lambda$-calculus[1] closely relates to $\lambda_R$ for $R = \{\text{ghnf, lam} \cup \text{var}\}$. It can be readily verified that every $R$ mentioned above is a c.r.s.

The notion of residuals of a $\beta_R$-redex under $\beta_R$-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 $\beta_R$-reduction sequence $\sigma$ form $M$; a $\beta_R$-redex in $M$ is involved in $\sigma$ if the $\beta_R$-redex or one of its residuals is contracted in $\sigma$.

**Definition 7.** ($\beta_R$-development) Given a $\lambda$-term $M$ and a set $S$ of $\beta_R$-redex in $M$; $\sigma : M \xrightarrow{\beta_R} N$ is a $\beta_R$-development (of $S$) if it contracts only $\beta_R$-redexes in $S$ and their residuals.

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

$$\sigma((bd(M_1))[\sigma_1(M_2), \ldots, \sigma_n(M_2)],$$

where $\sigma_i$ is a $\beta_R$-development from $bd(M_1)$ and $\sigma_n$ are $\beta_R$-developments from $M_2$ for $i = 1, \ldots, n$. 

Lemma 8 plays a major role 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 $\lambda R \vdash M_1 \equiv_R M_2$, then there exists $N$ such that $M_1 \xrightarrow{\beta_R} N$ for $i = 1, 2$.

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

- $\mathfrak{q}_R(M) = \emptyset$ if $M$ is a variable;
- $\mathfrak{q}_R(\lambda x. M) = \mathfrak{q}_R(M)$;
- $\mathfrak{q}_R(M(N)) = \mathfrak{q}_R(M) \cup \mathfrak{q}_R(N) \cup (\beta_R(N) \times \beta_R(M)) \cup \{ (M(N), L) : L \in \beta_R(M) \cup \beta_R(N) \}$ if $M(N)$ is a $\beta_R$-redex;
- $\mathfrak{q}_R(M(N)) = \mathfrak{q}_R(M) \cup \mathfrak{q}_R(N) \cup (\beta_R(N) \times \beta_R(M))$ if $M \in \mathcal{F}$ and $N \notin \mathcal{V}$;
- $\mathfrak{q}_R(M(N)) = \mathfrak{q}_R(M) \cup \mathfrak{q}_R(N) \cup (\beta_R(M) \times \beta_R(N))$ if $M \notin \mathcal{F}$.

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

**Theorem 11.** (Standardization) Given any $\beta_R$-reduction sequence $\sigma : M \xrightarrow{\beta_R} N$; then there exists a standard $\beta_R$-reduction sequence $\text{std}_R(\sigma) : M \xrightarrow{\beta_R} N$.

Let the first $\beta_R$-redex in $M$ be the first one according to order $\mathfrak{q}_R(M)$, then the normalizing strategy is the one which always reduces the first $\beta_R$-redex in a term.

**Corollary 12.** (Normalization) If $\lambda R \vdash M \equiv_R N$ for some $N$ in $\beta_R$-normal form, then the normalizing strategy reduces $M$ to $N$.

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

In conclusion, we have shown that the generalized $\lambda$-calculus can unify many existing $\lambda$-calculi. We are currently studying $\lambda^+_R$, investigating its application to partial-evaluation and run-time code generation.

**References**


This article was processed using the LATEX macro package with LLNCS style.