Géocalisation 2015, Chambéry

Reduction strategies for the differential lambda-calculus

Marie Kerjean

PPS Laboratory
Paris 7 University

June 12, 2015

Section 1

The differential λ-calculus

Differentiation in the syntax

Differentiation : linearisation. What is linear in the λ-calculus ?

Fundamental intuition

In $x, f \mapsto f(x)$, f is linear, not x.

$$(f + g)(x) = f(x) + g(x)$$
$$f(x + y) \neq f(x) + f(y)$$

Differential λ-calculus is a resource-sensitive calculus.

The syntax

$$\Lambda^d : S, T, U, V ::= 0 \mid s \mid s + T$$

$$\Lambda^s : s, t, u, v ::= x \mid \lambda x.s \mid sT \mid \mathrm{D}s \cdot t$$

Λ^d is the set of all differential λ-terms, et Λ^s is the set of simple terms.

There two distinct application, and two corresponding reduction rule :

- The usual non-linear application $(\lambda x.s)T$.

$$(\lambda x.s)T \to_\beta s[T/x]$$

- The linear application $\mathrm{D}(\lambda x.s) \cdot t$.

$$\mathrm{D}(\lambda x.s) \cdot t \to_{\beta_D} \lambda x.\frac{\partial s}{\partial x} \cdot t$$

The linear substitution ...

... which is not exactly a substitution

$$\frac{\partial y}{\partial x} \cdot T = \begin{cases} T \textit{ if } x = y \\ 0 \textit{ otherwise} \end{cases} \qquad \frac{\partial}{\partial x}(sU) \cdot T = (\frac{\partial s}{\partial x} \cdot T)U + (\mathrm{D}s \cdot (\frac{\partial U}{\partial x} \cdot T))U$$

$$\frac{\partial}{\partial x}(\lambda y.s) \cdot T = \lambda y.\frac{\partial s}{\partial x} \cdot T \qquad \frac{\partial}{\partial x}(\mathrm{D}s \cdot u) \cdot T = \mathrm{D}(\frac{\partial s}{\partial x} \cdot T) \cdot u + \mathrm{D}s \cdot (\frac{\partial u}{\partial x} \cdot T)$$

$$\frac{\partial 0}{\partial x} \cdot T = 0 \qquad \frac{\partial}{\partial x}(s + U) \cdot T = \frac{\partial s}{\partial x} \cdot T + \frac{\partial U}{\partial x} \cdot T$$

$\frac{\partial s}{\partial x} \cdot t$ represents s where x is linearly (i.e. one time) substituted by t.

Example

We want a differentiation operator in the syntax :

$$
\begin{aligned}
\frac{\partial (x)(x)y}{\partial x} \cdot u =& (\frac{\partial x}{\partial x} \cdot u)(x)y + (\mathrm{D}x \cdot (\frac{\partial (x)y}{\partial x} \cdot u))(x)y \\
=& (u)(x)y + (\mathrm{D}x \cdot [(\frac{\partial x}{\partial x} \cdot u)y) + (\mathrm{D}x \cdot (\frac{\partial y}{\partial x} u))y](x)y \\
=& (u)(x)y + (\mathrm{D}x \cdot (u)y)(x)y + ((\mathrm{D}x \cdot 0)y)(x)y
\end{aligned}
$$

Ehrhard, Regnier, The differential λ-calculus, 2004

Section 2

A call by-name differential λ-calculus

Motivations

We want to break the non-determinism induced by the sum.

We want to implement a reduction strategy for the differential lambda-calculus.

We do so by redefining the linear application for a call-by-name or call-by-value lambda-calculus.

We choose to substitute the first occurrence of the variable encountered in the reduction process : in call-by-name $\frac{\partial(x)x}{\partial x} u \to (u)x$.

Choose the occurrence of the substituted variable

We choose to substitute the first variable encountered in the reduction process, that is the first one to be isolated from the context.

In call-by-name :

$(\lambda z.xz)x \to_n (x)x = E_n[x]$ with $E_n = [\ ]x$.

$\frac{\partial(\lambda z.xz)x}{\partial x} u \to_n \frac{\partial(x)x}{\partial x} u \to_n (u)x$

In call-by-value :

$(\lambda z.xz)x = E_v[x]$ with $E_v = (\lambda z.xz)[\ ]$

$\frac{\partial(\lambda z.xz)x}{\partial x} u \to_n (D(\lambda z.xz) \cdot u)x \to (\lambda z.\frac{\partial xz}{\partial z} u)x \to (\lambda z.xu)x$

Deterministic differential λ-calculus.

We change slightly the syntax $:= x|\lambda x.t|tu|\frac{\partial t}{\partial x}$.

We want to know when we will be able to substitute a variable. It must be free, and not under an abstraction.

Linear free variables

$SL(x) = \{x\}$, $SL(\lambda x.a) = \emptyset$ $SL(ab) = SL(a) \bigcup SL(b)$,
$SL(\frac{\partial a}{\partial x}) = SL(a) \setminus \{x\}$.

Free variables

$Fv(x) = \{x\}$, $Fv(\lambda x.a) = Fv(a) \setminus \{x\}$ $Fv(ab) = Fv(a) \bigcup SL(b)$,
$Fv(\frac{\partial a}{\partial x}) = Fv(a) \setminus \{x\}$.

A cbn strategy for differential λ-calculus

Simple rules :

$(\lambda x.a)b \rightarrow_n a[b/x].$

$$\frac{a \rightarrow_v a'}{ab \rightarrow_v a'b}$$

Differential rules :

$\frac{\partial x}{\partial x} u \rightarrow_n u$

$\frac{\partial (\lambda y.a)b}{\partial x} u \rightarrow_n \frac{\partial a[b/y]}{\partial x} t$

$$\frac{\frac{\partial a}{\partial x} t \rightarrow_n a'}{\frac{\partial ab}{\partial x} t \rightarrow_n a'b}$$

$$\frac{x \notin SL(a) \qquad a \rightarrow_n a'}{\frac{\partial ab}{\partial x} t \rightarrow_n \frac{\partial a'b}{\partial x} t}$$

Theorem

It is a deterministic procedure

Examples

$$\frac{\partial(\lambda z.xz)x}{\partial x}u \to_n \frac{\partial(x)x}{\partial x}u \to_n (u)x.$$

$$\frac{\partial(x)(x)y}{\partial x}u \to_n (u)(x)y.$$

$$\frac{\partial(\lambda x.(x)x)y}{\partial y}u \to_n \frac{\partial(y)y}{\partial y}u \to_n (u)y.$$

Boudol, Curien, Lavatelli. A semantics for λ-calculus with ressources.

The KAM

Our definition is justified by the easiness of its adaptation to the KAM.

	Machine state before			Machine state after		
	Term	Env	Stack	Code	Env	Stack
Eval	x	$e_{\ x \mapsto (t, e')}$	π	t	e'	π
Pop	$\lambda x.t$	e	$c :: \pi$	t	$e + (x = c)$	π
Push	$(a)b$	e	π	a	e	$(b, e) :: \pi$

It is enough to add a Temporary and Prioritary stack.

The KAM for differential λ-calculus

	Machine state before				Machine state after			
	Term	Env	Stack	Temp	Code	Env	Stack	Temp
Eval	x	e $x \mapsto (t, e')$	π	T $(x, \ ) \notin T$	t	e'	π	T
Pop	$\lambda x.t$	e	$c :: \pi$	T	t	$e + (x = c)$	π	T
Push	$(a)b$	e	π	T	a	e	$(b, e) :: \pi$	T
Push	$\frac{\partial a}{\partial x} b$	e	π	T	$\frac{\partial a}{\partial x}$	e	$(b, e) :: \pi$	T
Dpop	$\frac{\partial a}{\partial x}$	e	$c :: \pi$	T	a	e	π	$(x, c) :: T$
Deval	x	e	π	T $x \mapsto (t, e')$	t	e'	π	$T - (x, c)$

Theorem

If a is a differential λ-term, if the *KAM* goes to the state $a, \emptyset, \emptyset, \emptyset$ to the state $a', e, \pi, \emptyset$, then a reduces to $a'\{e\}\tilde{\pi}$.

A translation in π-calculus.

Remember the translation of the call-by-name λ-calculus into π-calculus by Milner:

$$[|x|]_v = \bar{x}(v)$$
$$[|\lambda x.a|]_u = u(x,v)[|a|]_v$$
$$[|(a)b|]_u = \nu v([|a|]_v|\bar{v}(x,u).!(x(w).[|b|]_w))$$

$$\begin{aligned}[|(\lambda x.a)b|]_u &= \nu v(v(x,z)[|a|]_z|\bar{v}(x,u).!(x(w).[|b|]_w)) \\ &\to [|a|]_u|!(x(w).[|b|]_w)\end{aligned}$$

Let us define : $\left[\left|\frac{\partial a}{\partial x}t\right|\right]_u = [|a|]_u|(x(w).[|b|]_w)$.

Theorem

If x is free in a, and if $\frac{\partial a}{\partial x}t \to_n^* a'$n then there is c, $\bar{b}$, $\bar{y}$, y' such that $a' \equiv \lambda y'.c[\bar{b}/\bar{y}]$ and $[|\frac{\partial a}{\partial x}t|]_u \to \nu\bar{y}(([|\lambda y'.C|]_u|[|\bar{y} := \bar{b}|]_u)$.

From deterministic differential λ-calculus to the non-deterministic one.

Proposition

If a is a differential λ-term, if $a \to_n a'$, then $a \to a' + \sum_i a'_i$ in the differential λ-calculus.

How do we choose ?

$\frac{\partial ab}{\partial x} u = \frac{\partial a}{\partial x} ub + Da. \left(\frac{\partial b}{\partial x} u\right) b$

The substitution $\frac{\partial a}{\partial x} t$ in the differential λ-calculus corresponds to the cbn traduction $!A \to B$ of the λ-calculus into the differential nets (T. Ehrhard). It acts differently as it allows the strong reduction.

A cbv strategy for differential λ-calculus

Now we want to follow the call-by-value strategy. We want to substitute linearly the first occurrence to be isolated from the context.

$$\frac{\partial(\lambda y.a)x}{\partial x}u \to (\lambda y.a)u \to a[u/y]$$

A cbv strategy for differential λ-calculus

Now we want to follow the call-by-value strategy. We want to substitute linearly the first occurrence to be isolated from the context.

$$\frac{\partial(\lambda y.a)x}{\partial x}u \to (\lambda y.a)u \to a[u/y]$$

$$\frac{\partial(\lambda y.a)x}{\partial x}u \to_n \left(\lambda y.\frac{\partial a}{\partial y}u\right)x$$

A cbv strategy for differential λ-calculus

$$\text{Values}: \; v = x | \lambda x.a$$

$$\frac{\partial x}{\partial x} u \to_v u \qquad \frac{\partial (\lambda z.c) v}{\partial x} t \to_v \frac{\partial c[v/z]}{\partial x} t \qquad (\lambda x.a)v \to_v a[v/x].$$

$$\frac{a \to_v a'}{ab \to_v a'b} \qquad \frac{b \to_v b'}{(\lambda x.a)b \to_v (\lambda x.a)b'}$$

$$\frac{\frac{\partial a}{\partial x} t \to_v a'}{\frac{\partial ab}{\partial x} t \to_v a'b} \qquad \frac{\frac{\partial b}{\partial x} t \to_v b' \qquad \frac{\partial a}{\partial z} b' \to_v a'}{\frac{\partial (\lambda z.a) b}{\partial x} t \to_v (\lambda z.a')b}$$

$$\frac{x \notin SL(a) \qquad a \to_v a'}{\frac{\partial ab}{\partial x} t \to_v \frac{\partial a'b}{\partial x} t} \qquad \frac{x \notin SL(b) \qquad b \to_v b'}{\frac{\partial (\lambda x.a) b}{\partial x} t \to_v \frac{\partial (\lambda x.a) b'}{\partial x} t}$$

An example

$$\frac{\partial(\lambda z.xz)x}{\partial x}u \to_v (\lambda z.\frac{\partial xz}{\partial z}u)x.$$

$$\frac{\partial(\lambda x.(x)x)y}{\partial y}u \to_v (\lambda x.\frac{\partial(x)x}{\partial x}u)y.$$

This calculus is not as clean as the cbn one.

Differentiating $\tilde{\mu}$

We want to reunite these two approaches into the $\bar{\lambda}\mu\tilde{\mu}$-calculus.

$$\frac{\partial <x|e>}{\partial x} u = < u|\partial x. < x|e >> \to < u|e >$$

$$\begin{aligned} \frac{\partial < \lambda z.v_1|v_2.e >}{\partial x} u &= < u|\partial x. < \lambda z.v_1|v_2.e >> \\ &\to < u|\partial x. < v_2|\tilde{\mu} z. < v_1|e >> \\ &\to_n < u|\partial x. < v_1|e > [v_2/z] > \end{aligned}$$

$$\frac{\partial <\lambda z.v_1|v_2.e>}{\partial x} u \to_v$$
$$< v_2|\tilde{\mu} z. <\mu\alpha. <u|\partial x. < v_2|\alpha > >|\partial z. < v_1|e > > >$$

Perspectives

- Expressing differentiation into $\bar{\lambda}\mu\tilde{\mu}$.
- A differential semantics in more traditional models ?
- Proofs-nets and cbn/cbv of LJ into LL

Merci