A Differential Model Theory for Resource Lambda Calculi - Part I

Giulio Manzonetto

`g.manzonetto@cs.ru.nl`

Intelligent Systems
Radboud University – Nijmegen

FMCS 2011 - Calgary - 11th June 2011

The resource calculus Λ^r

Non-lazy version of Boudol's resource calculus

- Λ^r extends the notion of the λ-calculus application along two directions:

$$MN$$

  1. a term is applied to a multiset of resources, called bag of resources
  2. the resources can be reusable (available at will) or linear (to be used once)
- Ancestors:
  - the λ-calculus with multiplicities by Gérard Boudol (1993) introduced to study the observational semantics induced on the lazy λ-calculus by Milner's translation into the π-calculus;
  - the differential λ-calculus by Thomas Ehrhard and Laurent Regnier (2003) designed starting from a denotational model of linear logic.
- Syntax formalized by Paolo Tranquilli (2008)
  - it corresponds to the minimal fragment of the differential extension of linear logic by Thomas Ehrhard and Laurent Regnier (2006)

The resource calculus Λ^r

Non-lazy version of Boudol's resource calculus

- Λ^r extends the notion of the λ-calculus application along two directions:

$$M[N_1, \ldots, N_n]$$

  1. a term is applied to a multiset of resources, called bag of resources
  2. the resources can be reusable (available at will) or linear (to be used once)
- Ancestors:
  - the λ-calculus with multiplicities by Gérard Boudol (1993) introduced to study the observational semantics induced on the lazy λ-calculus by Milner's translation into the π-calculus;
  - the differential λ-calculus by Thomas Ehrhard and Laurent Regnier (2003) designed starting from a denotational model of linear logic.
- Syntax formalized by Paolo Tranquilli (2008)
  - it corresponds to the minimal fragment of the differential extension of linear logic by Thomas Ehrhard and Laurent Regnier (2006)

The resource calculus Λ^r

Non-lazy version of Boudol's resource calculus

- Λ^r extends the notion of the λ-calculus application along two directions:

$$M[L_1, \ldots, L_\ell, N_1^!, \ldots, N_n^!]$$

  1. a term is applied to a multiset of resources, called bag of resources
  2. the resources can be reusable (available at will) or linear (to be used once)
- Ancestors:
  - the λ-calculus with multiplicities by Gérard Boudol (1993) introduced to study the observational semantics induced on the lazy λ-calculus by Milner's translation into the π-calculus;
  - the differential λ-calculus by Thomas Ehrhard and Laurent Regnier (2003) designed starting from a denotational model of linear logic.

- Syntax formalized by Paolo Tranquilli (2008)
  - it corresponds to the minimal fragment of the differential extension of linear logic by Thomas Ehrhard and Laurent Regnier (2006)

The resource calculus Λ^r

Non-lazy version of Boudol's resource calculus

- Λ^r extends the notion of the λ-calculus application along two directions:

$$M[L_1, \dots, L_\ell, N_1^!, \dots, N_n^!]$$

1. a term is applied to a multiset of resources, called bag of resources
2. the resources can be reusable (available at will) or linear (to be used once)

- Ancestors:
  - the λ-calculus with multiplicities by Gérard Boudol (1993) introduced to study the observational semantics induced on the lazy λ-calculus by Milner's translation into the π-calculus;
  - the differential λ-calculus by Thomas Ehrhard and Laurent Regnier (2003) designed starting from a denotational model of linear logic.

- Syntax formalized by Paolo Tranquilli (2008)
  - it corresponds to the minimal fragment of the differential extension of linear logic by Thomas Ehrhard and Laurent Regnier (2006)

The resource calculus Λ^r

Non-lazy version of Boudol's resource calculus

- Λ^r extends the notion of the λ-calculus application along two directions:

$$M[L_1, \ldots, L_\ell, N_1^!, \ldots, N_n^!]$$

  1. a term is applied to a multiset of resources, called bag of resources
  2. the resources can be reusable (available at will) or linear (to be used once)
- Ancestors:
  - the λ-calculus with multiplicities by Gérard Boudol (1993) introduced to study the observational semantics induced on the lazy λ-calculus by Milner's translation into the π-calculus;
  - the differential λ-calculus by Thomas Ehrhard and Laurent Regnier (2003) designed starting from a denotational model of linear logic.
- Syntax formalized by Paolo Tranquilli (2008)
  - it corresponds to the minimal fragment of the differential extension of linear logic by Thomas Ehrhard and Laurent Regnier (2006)

Just another syntax for Differential Lambda Calculus

Precise link between:

Resource Calculus	↔	Differential λ-Calculus
linear resources	↔	syntactic differentiation
$M[L_1,\dots,L_\ell,N_1^!,\dots,N_n^!]$	↔	$\mathrm{D}^\ell(M)\cdot(L_1,\dots,L_\ell)(\sum_i N_i)$
$M[L_1,\dots,L_\ell]$	↔	$\mathrm{D}^\ell(M)\cdot(L_1,\dots,L_\ell)(0)$

Taylor Expansion Formula

$$(MN)^\circ = \sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{D}^n(M)\cdot(\underbrace{N,\dots,N}_{n\ \text{times}})(0)$$

Just another syntax for Differential Lambda Calculus

Precise link between:

Resource Calculus	$\leftrightarrow$	Differential λ-Calculus
linear resources	$\leftrightarrow$	syntactic differentiation
$M[L_1, \dots, L_\ell, N_1^!, \dots, N_n^!]$	$\leftrightarrow$	$\mathrm{D}^\ell(M) \cdot (L_1, \dots, L_\ell)(\sum_i N_i)$
$M[L_1, \dots, L_\ell]$	$\leftrightarrow$	$\mathrm{D}^\ell(M) \cdot (L_1, \dots, L_\ell)(0)$

Taylor Expansion Formula

$$(MN)^\circ = \sum_{n=0}^{\infty} \frac{1}{n!} \mathrm{D}^n(M) \cdot (\underbrace{N, \dots, N}_{n \text{ times}})(0)$$

The syntax of Λ^r

There are *three* syntactic categories:

- terms are in functional positions,
- bags of resources are in argument position and represent multisets of linear and reusable terms,
- sums of terms are results of the computation.

The syntax of Λ^r

There are *three* syntactic categories:

- terms are in functional positions,
- bags of resources are in argument position and represent multisets of linear and reusable terms,
- sums of terms are results of the computation.

Formally:

$$\begin{array}{lll} M, N, L & := x \mid \lambda x.M \mid MP & \text{terms} \\ P, Q, R & := [] \mid [M] \mid [M^!] \mid P \uplus Q & \text{bags} \\ \mathbb{M}, \mathbb{N} & := 0 \mid M \mid \mathbb{M} + \mathbb{N} & \text{sums} \end{array}$$

The syntax of Λ^r

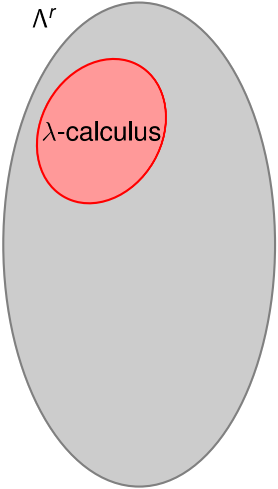
It contains λ-calculus

There are *three* syntactic categories:

- terms are in functional positions,
- bags of resources are in argument position and represent multisets of linear and reusable terms,
- sums of terms are results of the computation.

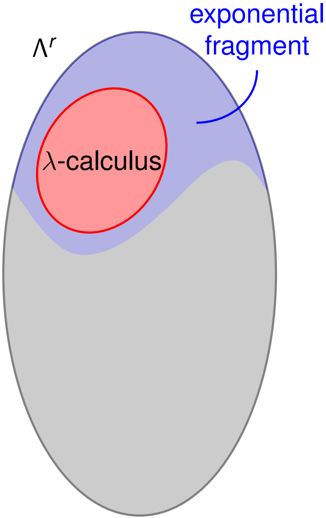
Formally:

$$\begin{array}{lll} M, N, L & := x \mid \lambda x.M \mid MP & \text{terms} \\ P, Q, R & := [] \mid [M] \mid [M^!] \mid P \uplus Q & \text{bags} \\ \mathbb{M}, \mathbb{N} & := 0 \mid M \mid \mathbb{M} + \mathbb{N} & \text{sums} \end{array}$$



The syntax of Λ^r

It contains λ-calculus and a nondeterministic extension of λ-calculus



There are *three* syntactic categories:

- terms are in functional positions,
- bags of resources are in argument position and represent multisets of linear and reusable terms,
- sums of terms are results of the computation.

Formally:

$$\begin{array}{lll} M, N, L & ::= x \mid \lambda x.M \mid MP & \text{terms} \\ P, Q, R & ::= [] \mid [M] \mid [M^!] \mid P \uplus Q & \text{bags} \\ \mathbb{M}, \mathbb{N} & ::= 0 \mid M \mid \mathbb{M} + \mathbb{N} & \text{sums} \end{array}$$

The syntax of Λ^r

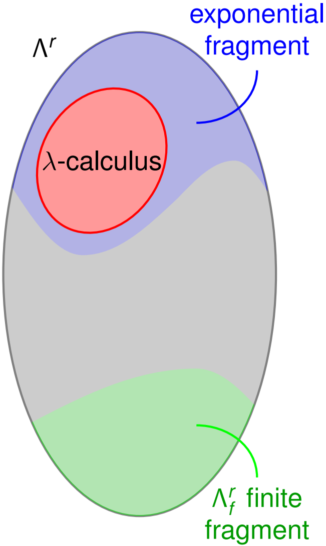
It contains λ-calculus and a nondeterministic extension of λ-calculus and a finite resource calculus

There are *three* syntactic categories:

- terms are in functional positions,
- bags of resources are in argument position and represent multisets of linear and reusable terms,
- sums of terms are results of the computation.

Formally:

$$\begin{array}{lll} M, N, L & := x \mid \lambda x.M \mid MP & \text{terms} \\ P, Q, R & := [] \mid [M] \mid [M^!] \mid P \uplus Q & \text{bags} \\ \mathbb{M}, \mathbb{N} & := 0 \mid M \mid \mathbb{M} + \mathbb{N} & \text{sums} \end{array}$$



Will we have sums everywhere?

Nope! All operators are linear…

$$\begin{array}{rcl} \lambda x.(\sum_i M_i) & := & \sum_i \lambda x.M_i \\ (\sum_i M_i)P & := & \sum_i M_i P \\ M(\sum_i P_i) & := & \sum_i MP_i \\ M([\sum_i N_i] \uplus P) & := & \sum_i M([N_i] \uplus P) \end{array}$$

except the $(\cdot)^!$:

$$M([(\sum_i N_i)^!] \uplus P) := M([N_1^!, \ldots, N_n^!] \uplus P)$$

0 annihilates everything (except under $(\cdot)^!$)

$$\lambda x.0 = 0 \qquad M([0] \uplus P) = 0 \qquad 0P = 0$$

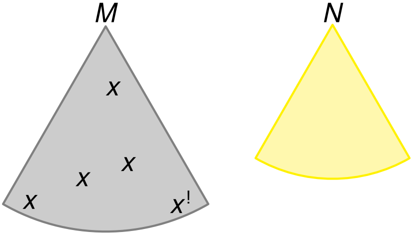
$$M([0^!] \uplus P) = MP$$

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



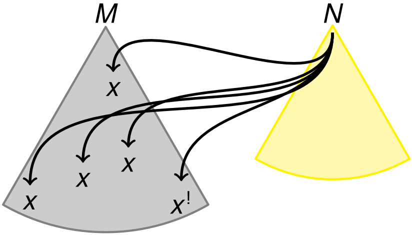
Usual Substitution
substitute all occurrences!

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



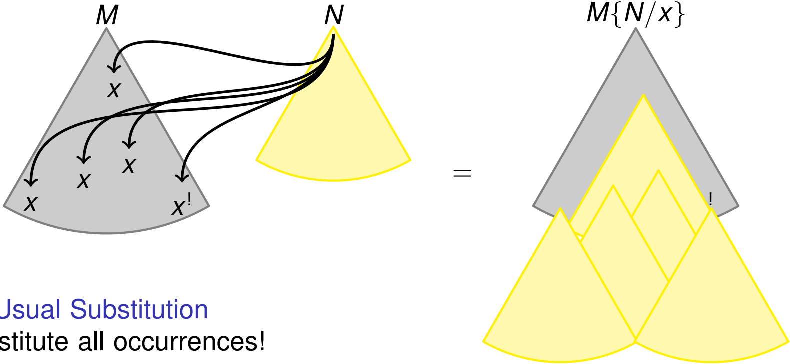
Usual Substitution
substitute all occurrences!

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



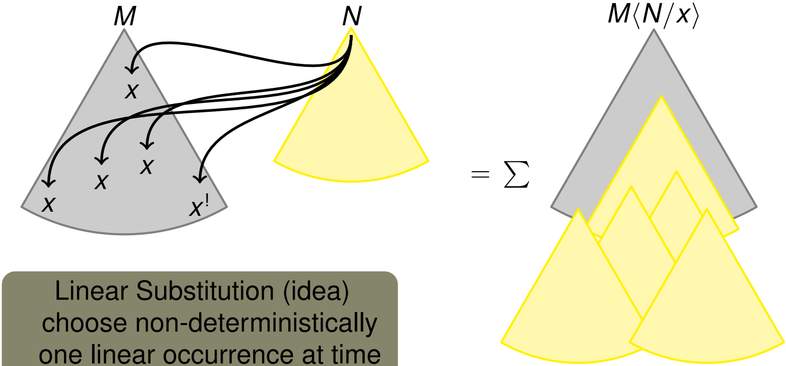
Usual Substitution
substitute all occurrences!

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



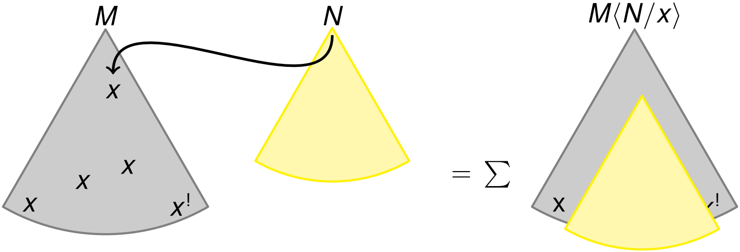
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



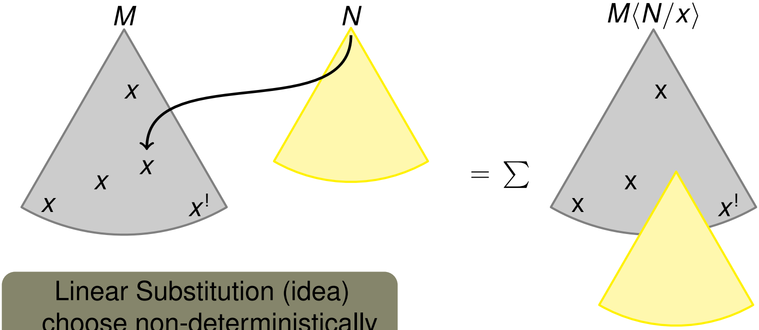
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



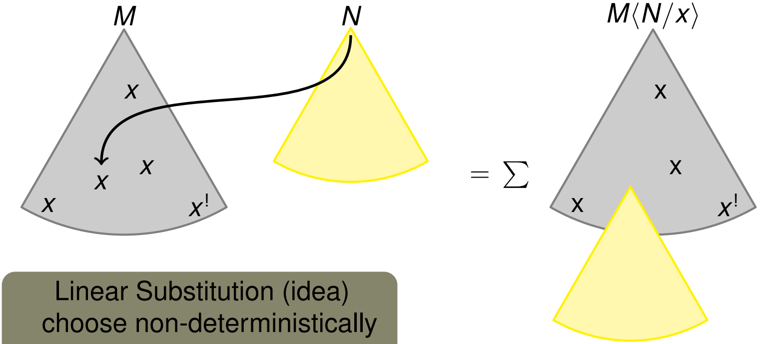
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



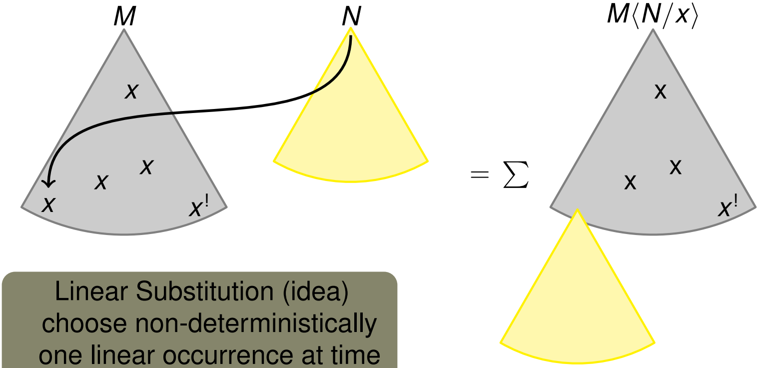
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



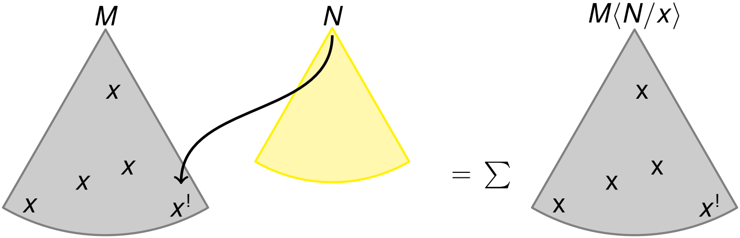
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



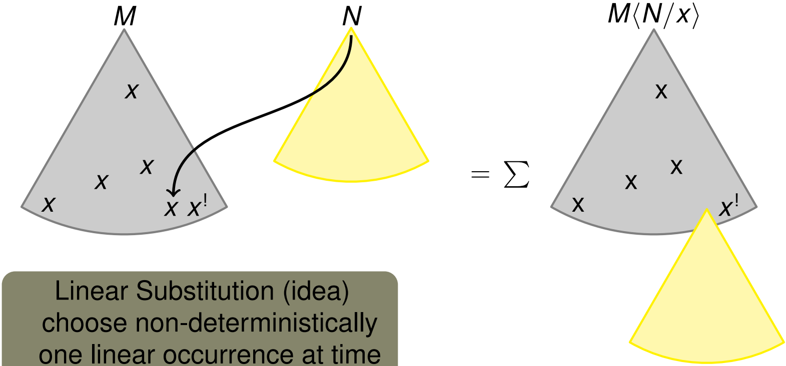
Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Two kind of substitutions

Usual and Linear Substitution

Two kinds of resources ⇒ two kinds of substitution:

- $M\{N/x\}$: usual capture free substitution,
- $M\langle N/x\rangle$: *linear* substitution, N is substituted for **exactly one linear** occurrence of x in M.



Linear Substitution (idea)
choose non-deterministically
one linear occurrence at time
(≅ differentiation)

Linear Substitution (formally)

$M\langle N/x\rangle$: *linear* substitution

On terms:

$$y\langle N/x\rangle = \begin{cases} N & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

$$(\lambda y.M)\langle N/x\rangle = \lambda y.M\langle N/x\rangle$$

$$(MP)\langle N/x\rangle = M\langle N/x\rangle P \;+\; M(P\langle N/x\rangle)$$

On Bags:

$$[]\langle N/x\rangle = 0$$

$$[M]\langle N/x\rangle = [M\langle N/x\rangle]$$

$$[M^!]\langle N/x\rangle = [M\langle N/x\rangle, M^!]$$

$$(P \uplus R)\langle N/x\rangle = P\langle N/x\rangle \uplus R \;+\; P \uplus (R\langle N/x\rangle)$$

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1, \ldots, L_\ell, N_1^!, \ldots, N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle \cdots \langle L_\ell/x\rangle \{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1,\ldots,L_\ell,N_1^!,\ldots,N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle\cdots\langle L_\ell/x\rangle\,\{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1,\dots,L_\ell,N_1^!,\dots,N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle\cdots\langle L_\ell/x\rangle\,\{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

Example

Let $\mathbf{I} := \lambda x.x$:

$(\lambda x.x)[\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\,\{0/x\} \equiv \mathbf{I}$ nice term

$(\lambda x.x)[] \to x\,\{0/x\} \equiv 0$ starvation

$(\lambda x.x)[\mathbf{I},\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv 0$ surfeit

$(\lambda x.y[x][x][x])[\mathbf{I},z^!] \xrightarrow{\beta *} y[\mathbf{I}][z][z] + y[z][\mathbf{I}][z] + y[z][z][\mathbf{I}]$ nondeterminism

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1,\ldots,L_\ell,N_1^!,\ldots,N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle\cdots\langle L_\ell/x\rangle\,\{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

Example

Let $\mathbf{I} := \lambda x.x$:

$(\lambda x.x)[\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\,\{0/x\} \equiv \mathbf{I}$ nice term

$(\lambda x.x)[] \to x\,\{0/x\} \equiv 0$ starvation

$(\lambda x.x)[\mathbf{I},\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv 0$ surfeit

$(\lambda x.y[x][x][x])[\mathbf{I},z^!] \xrightarrow{\beta *} y[\mathbf{I}][z][z] + y[z][\mathbf{I}][z] + y[z][z][\mathbf{I}]$ nondeterminism

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1, \dots, L_\ell, N_1^!, \dots, N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle \cdots \langle L_\ell/x\rangle \{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

Example

Let $\mathbf{I} := \lambda x.x$:

$(\lambda x.x)[\mathbf{I}] \to x\langle \mathbf{I}/x\rangle \{0/x\} \equiv \mathbf{I}\{0/x\} \equiv \mathbf{I}$ nice term

$(\lambda x.x)[] \to x\{0/x\} \equiv 0$ starvation

$(\lambda x.x)[\mathbf{I}, \mathbf{I}] \to x\langle \mathbf{I}/x\rangle\langle \mathbf{I}/x\rangle \{0/x\} \equiv \mathbf{I}\langle \mathbf{I}/x\rangle \{0/x\} \equiv 0$ surfeit

$(\lambda x.y[x][x][x])[\mathbf{I}, z^!] \xrightarrow{\beta *} y[\mathbf{I}][z][z] + y[z][\mathbf{I}][z] + y[z][z][\mathbf{I}]$ nondeterminism

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1,\dots,L_\ell,N_1^!,\dots,N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle\cdots\langle L_\ell/x\rangle\,\{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x\notin \mathrm{FV}(M)$$

Example

Let $\mathbf{I} := \lambda x.x$:

$(\lambda x.x)[\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\,\{0/x\} \equiv \mathbf{I}$ nice term

$(\lambda x.x)[] \to x\,\{0/x\} \equiv 0$ starvation

$(\lambda x.x)[\mathbf{I},\mathbf{I}] \to x\langle \mathbf{I}/x\rangle\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv \mathbf{I}\langle \mathbf{I}/x\rangle\,\{0/x\} \equiv 0$ surfeit

$(\lambda x.y[x][x][x])[\mathbf{I},z^!] \xrightarrow{\beta *} y[\mathbf{I}][z][z] + y[z][\mathbf{I}][z] + y[z][z][\mathbf{I}]$ nondeterminism

The operational semantics of Λ^r

β- and η- reductions

The β-reduction:

$$(\lambda x.M)[L_1, \ldots, L_\ell, N_1^!, \ldots, N_n^!] \xrightarrow{\beta} M\langle L_1/x\rangle \cdots \langle L_\ell/x\rangle \{\Sigma_i N_i/x\}$$

The η-reduction:

$$\lambda x.M[x^!] \xrightarrow{\eta} M, \text{ where } x \notin \mathrm{FV}(M)$$

Theorem [Pagani-Tranquilli APLAS'09]

- $\to_\beta$ is confluent.
- $\to_\beta$ enjoys a standardization property.

Simple Type System

$$\text{Types: } \sigma, \tau ::= \alpha \mid \sigma \to \tau$$

$$(\mathrm{Rx})\,\frac{\Gamma(x) = \sigma}{\Gamma \vdash_R x : \sigma} \qquad (\mathrm{R}\lambda)\,\frac{\Gamma, x : \sigma \vdash_R M : \tau}{\Gamma \vdash_R \lambda x.M : \sigma \to \tau}$$

$$(\mathrm{R@})\,\frac{\Gamma \vdash_R M : \sigma \to \tau \quad \Gamma \vdash_R P : \sigma}{\Gamma \vdash_R MP : \tau}$$

$$(\mathrm{Rb})\,\frac{\Gamma \vdash_R N : \sigma \quad \Gamma \vdash_R P : \sigma}{\Gamma \vdash_R [N^{(!)}] \uplus P : \sigma} \qquad (\mathrm{R}[])\,\frac{}{\Gamma \vdash_R [] : \sigma}$$

$$(\mathrm{R}+)\,\frac{\Gamma \vdash_R A_i : \sigma \qquad \text{for all } i}{\Gamma \vdash_R \sum_i A_i : \sigma}$$

Remark

Sums and bags are typed uniformly…

The differential λ-calculus: Syntax

Differential Lambda Terms:

$$s, t, u ::= x \mid \lambda x.s \mid sT \mid D(s) \cdot t$$

$$S, T, U ::= s \mid s + T \mid 0$$

Reduction Rules ($\to_D = \to_\beta \cup \to_{\beta_D}$):

$$\begin{array}{ll} (\beta) & (\lambda x.s)t \to_\beta s\{t/x\} \\ (\beta_D) & D(\lambda x.s) \cdot t \to_{\beta_D} \lambda x. \frac{\partial s}{\partial x} \cdot t \end{array}$$

Ideas:

- st = usual application of λ-calculus $(\cong s[t^!])$
- $D(\cdots(D(s) \cdot t_1) \cdot \cdots) \cdot t_k$ = linear application $(\cong s[t_1, \ldots, t_k])$
- $\frac{\partial s}{\partial x} \cdot t$ = differential substitution $(\cong s\langle t/x \rangle)$
  - $\frac{\partial (sU)}{\partial x} \cdot t = (\frac{\partial s}{\partial x} \cdot t)U + (D(s) \cdot (\frac{\partial U}{\partial x} \cdot t))U$
    $(\cong \; (s[U^!])\langle t/x \rangle = s\langle t/x \rangle[U^!] + s[U\langle t/x \rangle, U^!])$

The differential λ-calculus: Syntax

Differential Lambda Terms:

$$s, t, u ::= x \mid \lambda x.s \mid sT \mid D(s) \cdot t$$

$$S, T, U ::= s \mid s + T \mid 0$$

Reduction Rules ($\to_D = \to_\beta \cup \to_{\beta_D}$):

$$\begin{array}{ll} (\beta) & (\lambda x.s)t \to_\beta s\{t/x\} \\ (\beta_D) & D(\lambda x.s) \cdot t \to_{\beta_D} \lambda x.\frac{\partial s}{\partial x} \cdot t \end{array}$$

Ideas:

- st = usual application of λ-calculus ($\cong s[t^!]$)
- $D(\cdots(D(s) \cdot t_1) \cdot \cdots) \cdot t_k$ = linear application ($\cong s[t_1, \ldots, t_k]$)
- $\frac{\partial s}{\partial x} \cdot t$ = differential substitution ($\cong s\langle t/x\rangle$)
  - $\frac{\partial(sU)}{\partial x} \cdot t = (\frac{\partial s}{\partial x} \cdot t)U + (D(s) \cdot (\frac{\partial U}{\partial x} \cdot t))U$
    ($\cong$ $(s[U^!])\langle t/x\rangle = s\langle t/x\rangle[U^!] + s[U\langle t/x\rangle, U^!]$)

The differential λ-calculus: Syntax

Differential Lambda Terms:

$$s, t, u ::= x \mid \lambda x.s \mid sT \mid D(s) \cdot t$$

$$S, T, U ::= s \mid s + T \mid 0$$

Reduction Rules ($\to_D = \to_\beta \cup \to_{\beta_D}$):

$$\begin{array}{ll} (\beta) & (\lambda x.s)t \to_\beta s\{t/x\} \\ (\beta_D) & D(\lambda x.s) \cdot t \to_{\beta_D} \lambda x.\frac{\partial s}{\partial x} \cdot t \end{array}$$

Ideas:

- st = usual application of λ-calculus ($\cong s[t^!]$)
- $(D(\cdots(D(s) \cdot t_1) \cdot \cdots) \cdot t_k)0$ = linear application ($\cong s[t_1, \ldots, t_k]$)
- $\frac{\partial s}{\partial x} \cdot t$ = differential substitution ($\cong s\langle t/x\rangle$)
  - $\frac{\partial (sU)}{\partial x} \cdot t = (\frac{\partial s}{\partial x} \cdot t)U + (D(s) \cdot (\frac{\partial U}{\partial x} \cdot t))U$
    ($\cong$ $(s[U^!])\langle t/x\rangle = s\langle t/x\rangle[U^!] + s[U\langle t/x\rangle, U^!]$)

The differential λ-calculus: Syntax

Differential Lambda Terms:

$$s, t, u ::= x \mid \lambda x.s \mid sT \mid D(s) \cdot t$$

$$S, T, U ::= s \mid s + T \mid 0$$

Reduction Rules ($\to_D = \to_\beta \cup \to_{\beta_D}$):

$$\begin{array}{ll} (\beta) & (\lambda x.s)t \to_\beta s\{t/x\} \\ (\beta_D) & D(\lambda x.s) \cdot t \to_{\beta_D} \lambda x. \frac{\partial s}{\partial x} \cdot t \end{array}$$

Ideas:

- $st =$ usual application of λ-calculus ($\cong s[t^!]$)
- $D(\cdots(D(s) \cdot t_1) \cdot \cdots) \cdot t_k =$ linear application ($\cong s[t_1, \ldots, t_k]$)
- $\frac{\partial s}{\partial x} \cdot t =$ differential substitution ($\cong s\langle t/x \rangle$)
  - $\frac{\partial (sU)}{\partial x} \cdot t = (\frac{\partial s}{\partial x} \cdot t)U + (D(s) \cdot (\frac{\partial U}{\partial x} \cdot t))U$
    ($\cong$ $(s[U^!])\langle t/x \rangle = s\langle t/x \rangle [U^!] + s[U\langle t/x \rangle, U^!]$)

Translation between the two calculi

We can define a translation map

$$(\cdot)^\circ : \text{Resource calculus} \to \text{Differential } \lambda\text{-calculus}$$

- $x^\circ = x$,
- $(\lambda x.M)^\circ = \lambda x.M^\circ$,
- $(M[\vec{L}, \vec{N}^!])^\circ = (D^k(M^\circ) \cdot L_1^\circ \cdots L_k^\circ)(\sum_i N_i^\circ)$,
- $0^\circ = 0$,
- $(\sum_i M_i)^\circ = \sum_i M_i^\circ$.

The translation is ‘faithful’

For M, N resource terms: $M \to_\beta N$ implies $M^\circ \to_D^\star N^\circ$

Simple Types in Differential Calculus

$$x\,\frac{\Gamma(x)=\sigma}{\Gamma\vdash_D x:\sigma}\qquad \lambda\,\frac{\Gamma;x:\sigma\vdash_D s:\tau}{\Gamma\vdash_D \lambda x.s:\sigma\to\tau}$$

$$@\,\frac{\Gamma\vdash_D s:\sigma\to\tau\quad \Gamma\vdash_D t:\sigma}{\Gamma\vdash_D st:\tau}\qquad D\,\frac{\Gamma\vdash_D s:\sigma\to\tau\quad \Gamma\vdash_D t:\sigma}{\Gamma\vdash_D D(s)\cdot t:\sigma\to\tau}$$

$$0\,\frac{}{\Gamma\vdash_D 0:\sigma}\qquad \text{sum}\,\frac{\Gamma\vdash_D s_i:\sigma\ \text{ for all } i}{\Gamma\vdash_D \sum_i s_i:\sigma}$$

Remark: Linear application does not decrease types.

The translation remains ‘faithful’

Let M be a resource term. If $\Gamma\vdash_R M:\sigma$ then $\Gamma\vdash_D M^\circ:\sigma$

Corollary

Every model of the (typed/untyped) differential λ-calculus will also be a model of the resource calculus.

Simple Types in Differential Calculus

$$x\,\frac{\Gamma(x) = \sigma}{\Gamma \vdash_D x : \sigma} \qquad \lambda\,\frac{\Gamma ; x : \sigma \vdash_D s : \tau}{\Gamma \vdash_D \lambda x.s : \sigma \to \tau}$$

$$@\,\frac{\Gamma \vdash_D s : \sigma \to \tau \quad \Gamma \vdash_D t : \sigma}{\Gamma \vdash_D st : \tau} \qquad D\,\frac{\Gamma \vdash_D s : \sigma \to \tau \quad \Gamma \vdash_D t : \sigma}{\Gamma \vdash_D D(s) \cdot t : \sigma \to \tau}$$

$$0\,\frac{}{\Gamma \vdash_D 0 : \sigma} \qquad \mathrm{sum}\,\frac{\Gamma \vdash_D s_i : \sigma \;\; \text{for all } i}{\Gamma \vdash_D \sum_i s_i : \sigma}$$

Remark: Linear application does not decrease types.

The translation remains 'faithful'

Let M be a resource term. If $\Gamma \vdash_R M : \sigma$ then $\Gamma \vdash_D M^\circ : \sigma$

Corollary

Every model of the (typed/untyped) differential λ-calculus will also be a model of the resource calculus.

Taylor Expansion: intuition

Lambda Calculus: Taylor Expansion Formula

For λ-terms M, N we have

$$(MN)^\circ = \sum_{n=0}^{\infty} \frac{1}{n!}(\mathrm{D}^{\mathrm{n}}(M) \cdot (\underbrace{N, \dots, N}_{n \text{ times}}))(0)$$

Taylor Expansion: intuition

Lambda Calculus: Taylor Expansion Formula

For λ-terms M, N we have

$$(MN)^\circ = \sum_{n=0}^{\infty} \frac{1}{n!} M[\underbrace{N, \ldots, N}_{n \text{ times}}]$$

Extension: From λ-calculus to resource calculus…

Taylor Expansion: intuition

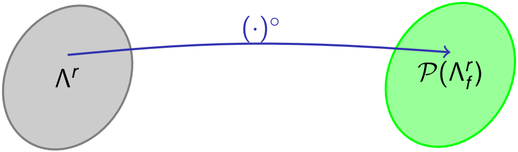
Lambda Calculus: Taylor Expansion Formula

For λ-terms M, N we have

$$(MN)^\circ = \bigcup_{n=0}^{\infty} \{M[\underbrace{N, \dots, N}_{n \text{ times}}]\}$$

Extension: From λ-calculus to resource calculus...
For the sake of simplicity we consider an idempotent sum

Resource Calculus: Full Taylor Expansion



The (support of the full) Taylor Expansion M° of a term M:

- $x^\circ = \{x\}$,
- $(\lambda x.M)^\circ = \lambda x.M^\circ \qquad (= \{\lambda x.M' : M' \in M^\circ\})$,
- $(M[L_1,\dots,L_\ell,N_1^!,\dots,N_n^!])^\circ = \cup_{P\in\mathcal{M}_f(\cup_i N_i{}^\circ)} M^\circ([L_1{}^\circ,\dots,L_\ell{}^\circ]\cdot P)$,

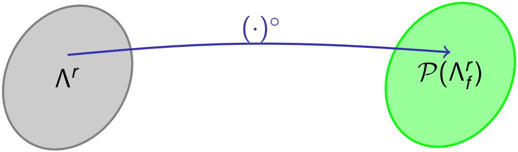
It is extended to sums $\mathbb{M}$ by setting:

- $(\sum_i M_i)^\circ = \cup_i M_i{}^\circ$.

Example

- $\mathbb{M}^\circ := (\lambda z.x[z^!])^\circ = \{\lambda z.x[z^n] : n \in \mathrm{Nat}\}$,
- $\mathbb{N}^\circ := (\lambda z.x[] + \lambda z.x[z,z^!])^\circ = \{\lambda z.x[]\} \cup \{\lambda z.x[z^{n+1}] : n \in \mathrm{Nat}\}$.

Resource Calculus: Full Taylor Expansion



The (support of the full) Taylor Expansion M° of a term M:

- $x^\circ = \{x\}$,
- $(\lambda x.M)^\circ = \lambda x.M^\circ \qquad (= \{\lambda x.M' : M' \in M^\circ\})$,
- $(M[L_1, \dots, L_\ell, N_1^!, \dots, N_n^!])^\circ = \cup_{P \in \mathcal{M}_f(\cup_i N_i{}^\circ)} M^\circ([L_1{}^\circ, \dots, L_\ell{}^\circ] \cdot P)$,

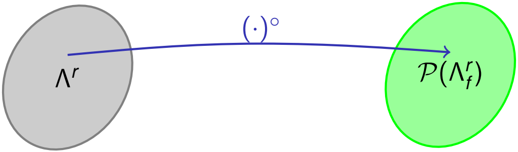
It is extended to sums $\mathbb{M}$ by setting:

- $(\sum_i M_i)^\circ = \cup_i M_i{}^\circ$.

Example

- $\mathbb{M}^\circ := (\lambda z.x[z^!])^\circ = \{\lambda z.x[z^n] : n \in \mathrm{Nat}\}$,
- $\mathbb{N}^\circ := (\lambda z.x[] + \lambda z.x[z, z^!])^\circ = \{\lambda z.x[]\} \cup \{\lambda z.x[z^{n+1}] : n \in \mathrm{Nat}\}$.

Resource Calculus: Full Taylor Expansion



The (support of the full) Taylor Expansion M° of a term M:

- $x^\circ = \{x\}$,
- $(\lambda x.M)^\circ = \lambda x.M^\circ \qquad (= \{\lambda x.M' : M' \in M^\circ\})$,
- $(M[L_1,\dots,L_\ell,N_1^!,\dots,N_n^!])^\circ = \cup_{P\in\mathcal{M}_f(\cup_i {N_i}^\circ)} M^\circ([{L_1}^\circ,\dots,{L_\ell}^\circ]\cdot P)$,

It is extended to sums $\mathbb{M}$ by setting:

- $(\sum_i M_i)^\circ = \cup_i {M_i}^\circ$.

Example

- $\mathbb{M}^\circ := (\lambda z.x[z^!])^\circ = \{\lambda z.x[z^n] : n \in \mathrm{Nat}\}$,
- $\mathbb{N}^\circ := (\lambda z.x[] + \lambda z.x[z,z^!])^\circ = \{\lambda z.x[]\} \cup \{\lambda z.x[z^{n+1}] : n \in \mathrm{Nat}\}$.

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL. . . however their semantical investigations are only at the beginning!

Categorical description:

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL… however their semantical investigations are only at the beginning!

Categorical description:

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL. . . however their semantical investigations are only at the beginning!

Categorical description:

$$\lambda\text{-calculus} \Longleftrightarrow \text{CCC}$$

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL. . . however their semantical investigations are only at the beginning!

Categorical description:

$$\begin{tikzcd}
\lambda\text{-calculus} \arrow[rr, Leftrightarrow] & & \text{CCC} \\
 & & \\
 & & \text{SMCC} \arrow[uu, dashed, two heads, "\text{coKleisli}"]
\end{tikzcd}$$

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL. . . however their semantical investigations are only at the beginning!

Categorical description:

$$\begin{tikzcd}
\text{differential } \lambda\text{-calculus} \arrow[r, Leftrightarrow] & \text{CCC + differential structure} \\
& \text{SMCC + differential structure} \arrow[u, dashed, two heads, "\text{coKleisli}"]
\end{tikzcd}$$

What about semantics?

Semantics

The differential/resource calculi are born from the analysis of the semantics of LL. . . however their semantical investigations are only at the beginning!

Categorical description:

$$\begin{tikzcd}
\text{differential } \lambda\text{-calculus} \arrow[r, Leftrightarrow] & \text{CCC + differential structure} \\
& \text{SMCC + differential structure} \arrow[u, dashed, two heads, "\text{coKleisli}"]
\end{tikzcd}$$

Differential Categories (Blute, Cockett & Seely ’06)

Axiomatic characterization of a **derivative operator** in (possibly non-closed) Symmetric Monoidal Categories + the “!” is not necessarily monoidal.

- SMCCs + monoidal “!” constitute interesting instances.

(Monoidal) Differential Categories

Additive Symmetric Monoidal Categories

Sum on terms ↦ sum on morphisms

A symmetric monoidal category is **additive** if all homsets are enriched with commutative monoids:

$$(f+g);h = f;h + g;h \qquad h;(f+g) = h;f + h;g \qquad 0;f = 0 = f;0$$

The tensor product preserves the commutative monoid structure:

$$(f+g)\otimes h = f\otimes h + g\otimes h \qquad 0\otimes f = 0$$

Remark

“Additive” becomes *left* additive in the coKleisli.

(Monoidal) Differential Categories

Coalgebra Modalities

A *comonad* $(!, \delta, \varepsilon)$ is a **coalgebra modality** if each $!A$ comes equipped with a natural coalgebra structure

$$\Delta : !A \to !A \otimes !A, \qquad e : !A \to \top, \text{ such that:}$$

1. $(!, \Delta, e)$ is a *comonoid*

$$\begin{tikzcd}
& !A \arrow[dl, "\mathrm{Id}"'] \arrow[d, "\Delta"] \arrow[dr, "\mathrm{Id}"] & \\
!A & !A \otimes !A \arrow[l, "e\otimes\mathrm{Id}"'] \arrow[r, "\mathrm{Id}\otimes e"] & !A
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & !A \otimes !A \arrow[d, "\Delta\otimes\mathrm{Id}"] \\
!A \otimes !A \arrow[r, "\mathrm{Id}\otimes\Delta"'] & !A \otimes !A \otimes !A
\end{tikzcd}$$

2. δ is a morphism of comonoids:

$$\begin{tikzcd}
!A \arrow[rr, "\delta"] \arrow[dr, "e"'] & & !!A \arrow[dl, "e"] \\
& \top &
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\delta"] \arrow[d, "\Delta"'] & !!A \arrow[d, "\Delta"] \\
!A \otimes !A \arrow[r, "\delta\otimes\delta"'] & !!A \otimes !!A
\end{tikzcd}$$

(Monoidal) Differential Categories

Coalgebra Modalities

A *comonad* $(!, \delta, \varepsilon)$ is a **coalgebra modality** if each $!A$ comes equipped with a natural coalgebra structure

$$\Delta : !A \to !A \otimes !A, \qquad e : !A \to \top, \text{ such that:}$$

1. $(!, \Delta, e)$ is a *comonoid*

$$\begin{tikzcd}
 & !A \arrow[dl, "\mathrm{Id}"'] \arrow[d, "\Delta"] \arrow[dr, "\mathrm{Id}"] & \\
!A & !A \otimes !A \arrow[l, "e\otimes\mathrm{Id}"'] \arrow[r, "\mathrm{Id}\otimes e"] & !A
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & !A \otimes !A \arrow[d, "\Delta\otimes\mathrm{Id}"] \\
!A \otimes !A \arrow[r, "\mathrm{Id}\otimes\Delta"'] & !A \otimes !A \otimes !A
\end{tikzcd}$$

2. δ is a morphism of comonoids:

$$\begin{tikzcd}
!A \arrow[rr, "\delta"] \arrow[dr, "e"'] & & !!A \arrow[dl, "e"] \\
 & \top &
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\delta"] \arrow[d, "\Delta"'] & !!A \arrow[d, "\Delta"] \\
!A \otimes !A \arrow[r, "\delta\otimes\delta"'] & !!A \otimes !!A
\end{tikzcd}$$

(Monoidal) Differential Categories

Coalgebra Modalities

A *comonad* $(!, \delta, \varepsilon)$ is a **coalgebra modality** if each $!A$ comes equipped with a natural coalgebra structure

$$\Delta : !A \to !A \otimes !A, \qquad e : !A \to \top, \text{ such that:}$$

1. $(!, \Delta, e)$ is a *comonoid*

$$\begin{tikzcd}
& !A \arrow[dl, "\mathrm{Id}"'] \arrow[d, "\Delta"] \arrow[dr, "\mathrm{Id}"] & \\
!A & !A \otimes !A \arrow[l, "e\otimes\mathrm{Id}"'] \arrow[r, "\mathrm{Id}\otimes e"] & !A
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & !A \otimes !A \arrow[d, "\Delta\otimes\mathrm{Id}"] \\
!A \otimes !A \arrow[r, "\mathrm{Id}\otimes\Delta"'] & !A \otimes !A \otimes !A
\end{tikzcd}$$

2. δ is a morphism of comonoids:

$$\begin{tikzcd}
!A \arrow[rr, "\delta"] \arrow[dr, "e"'] & & !!A \arrow[dl, "e"] \\
& \top &
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\delta"] \arrow[d, "\Delta"'] & !!A \arrow[d, "\Delta"] \\
!A \otimes !A \arrow[r, "\delta\otimes\delta"'] & !!A \otimes !!A
\end{tikzcd}$$

(Monoidal) Differential Categories

Coalgebra Modalities

The “!” is not (necessarily) monoidal!

A *con* with a natura

$$\Delta : !A \to !A \otimes !A, \qquad e : !A \to \top, \text{ such that:}$$

1. $(!, \Delta, e)$ is a *comonoid*

$$\begin{tikzcd}
 & !A \arrow[dl, "\mathrm{Id}"'] \arrow[d, "\Delta"] \arrow[dr, "\mathrm{Id}"] & \\
!A & !A \otimes !A \arrow[l, "e\otimes\mathrm{Id}"'] \arrow[r, "\mathrm{Id}\otimes e"] & !A
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & !A \otimes !A \arrow[d, "\Delta\otimes\mathrm{Id}"] \\
!A \otimes !A \arrow[r, "\mathrm{Id}\otimes\Delta"'] & !A \otimes !A \otimes !A
\end{tikzcd}$$

2. δ is a morphism of comonoids:

$$\begin{tikzcd}
!A \arrow[rr, "\delta"] \arrow[dr, "e"'] & & !!A \arrow[dl, "e"] \\
 & \top &
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\delta"] \arrow[d, "\Delta"'] & !!A \arrow[d, "\Delta"] \\
!A \otimes !A \arrow[r, "\delta\otimes\delta"'] & !!A \otimes !!A
\end{tikzcd}$$

(Monoidal) Differential Categories

Differential Combinator

Intuitions:

- A map $f : A \to B \cong$ linear map,
- A coKleisli map $f : !A \to B \cong$ abstract differentiable map from A to B,
- coKleisli $\mathbf{C}_! \cong$ category of abstract differentiable maps.

Differential Combinator

$D^{\otimes} : \mathbf{C}(A, B) \to \mathbf{C}(A \otimes !A, B)$

$$\frac{f : !A \to B}{D^{\otimes}(f) : A \otimes !A \to B}\ D^{\otimes}$$

that must satisfy suitable equations. . . (see later)

(Monoidal) Differential Categories

Differential Combinator - Additivity & functoriality

Additivity:

$$D^{\otimes}(0) = 0, \qquad D^{\otimes}(f+g) = D^{\otimes}(f) + D^{\otimes}(g).$$

Functoriality:

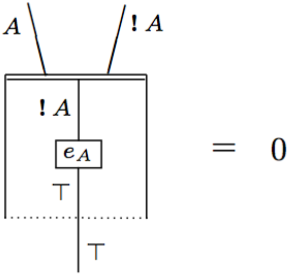
$$\begin{tikzcd}
!A \arrow[r, "f"] \arrow[d, "!u"'] & B \arrow[d, "v"] & & A \otimes !A \arrow[r, "D^{\otimes}(f)"] \arrow[d, "u\otimes !u"'] & B \arrow[d, "v"] \\
!C \arrow[r, "g"] & D & \mapsto & C \otimes !C \arrow[r, "D^{\otimes}(g)"] & D
\end{tikzcd}$$

(Monoidal) Differential Categories

Differential Combinator - Axiom D1

[D1] Constant maps:

$$D^{\otimes}(e_A) = 0$$



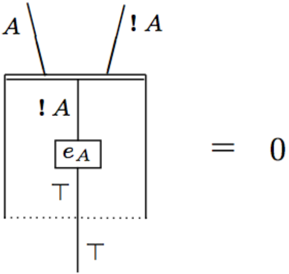
(Monoidal) Differential Categories

Differential Combinator - Axiom D1

[D1] Constant maps:

$$D^{\otimes}(e_A) = 0$$

Constant functions have derivative 0.

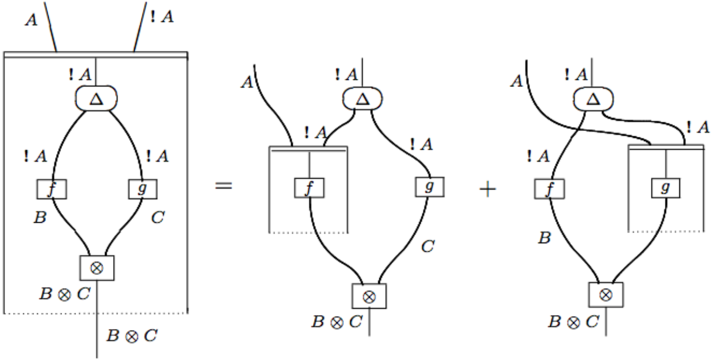


(Monoidal) Differential Categories

Differential Combinator - Axiom D2

[D2] Product rule:

$$D^{\otimes}(\Delta;(f \otimes g)) = (\mathrm{Id} \otimes \Delta);(D^{\otimes}(f) \otimes g) + (\mathrm{Id} \otimes \Delta);(f \otimes D^{\otimes}(g))$$



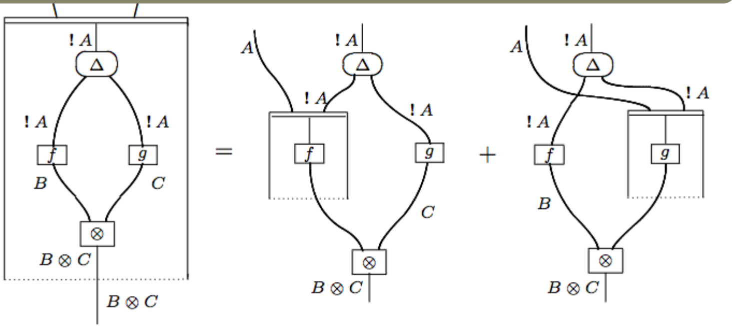
(Monoidal) Differential Categories

Differential Combinator - Axiom D2

[D2] Product rule:

$$D^{\otimes}(\Delta; (f \otimes g)) = (\mathrm{Id} \otimes \Delta); (D^{\otimes}(f) \otimes g) + (\mathrm{Id} \otimes \Delta); (f \otimes D^{\otimes}(g))$$

The tensor of two functions on the same arguments is morally the product of two functions.

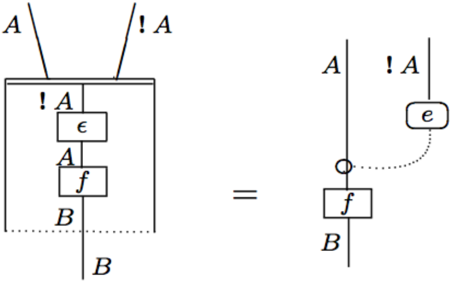


(Monoidal) Differential Categories

Differential Combinator - Axiom D3

[D3] Linear maps:

$$D^{\otimes}(\varepsilon_A; f) = (\mathrm{Id} \otimes e_A); f$$



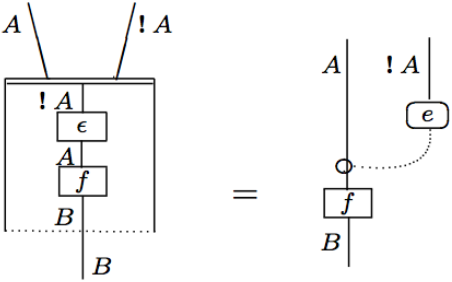
(Monoidal) Differential Categories

Differential Combinator - Axiom D3

[D3] Linear maps:

$$D^{\otimes}(\varepsilon_A; f) = (\mathrm{Id} \otimes e_A); f$$

The derivative of a map which is linear is constant.

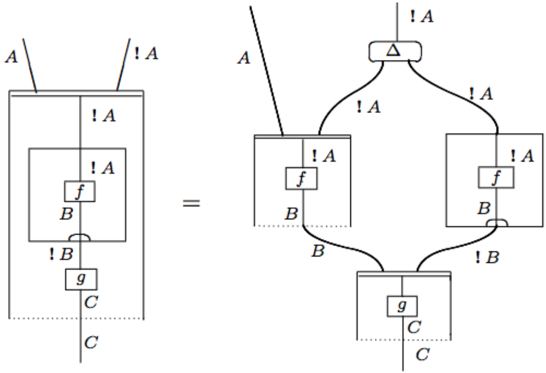


(Monoidal) Differential Categories

Differential Combinator - Axiom D4

[D4] The chain rule:

$$D^{\otimes}(\delta; !f; g) = (\mathrm{Id} \otimes \Delta); (D^{\otimes}(f) \otimes \delta; !f); D^{\otimes}(g)$$



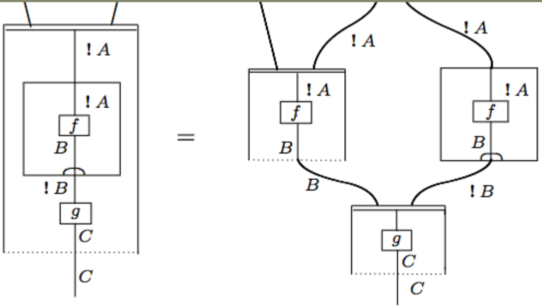
(Monoidal) Differential Categories

Differential Combinator - Axiom D4

[D4] The chain rule:

$$D^{\otimes}(\delta; !f; g) = (\mathrm{Id} \otimes \Delta); (D^{\otimes}(f) \otimes \delta; !f); D^{\otimes}(g)$$

The derivative of the composite of f and g is the derivative of f composed with the derivative of g



Differential Combinator - a simpler characterization

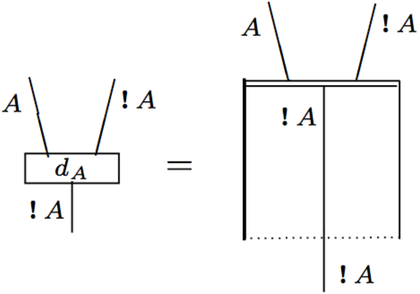
The whole differential structure is generated by the derivative of the identity!

$$\begin{tikzcd}
!A \arrow[r, "\mathrm{Id}_{!A}"] \arrow[d, "!\mathrm{Id}_A"'] & !A \arrow[d, "f"] & & A \otimes !A \arrow[r, "D^{\otimes}(\mathrm{Id}_{!A})"] \arrow[d, "\mathrm{Id}_A \otimes !\mathrm{Id}_A"'] & !A \arrow[d, "f"] \\
!A \arrow[r, "f"] & !A & \mapsto & A \otimes !A \arrow[r, "D^{\otimes}(f)"] & !A
\end{tikzcd}$$

$d_A = D^{\otimes}(\mathrm{Id}_{!A})$ is a **deriving transformation**, namely a transformation (natural in A) satisfying [D1-D4] rephrased.

Differential Categories - a simpler characterization

This is the reason why the differential box is “bottomless”:



Differential Category - Def. 2

A **differential category** is an additive symmetric monoidal category with a deriving transformation $d_A : A \otimes !A \to !A$.

Examples of Differential Categories

- **Finiteness Spaces**,
- **Sets and relations** + the "bag functor" $M_f(\cdot)$ (finite multisets):

$$d_A : A \otimes !A \to !A : a_0, [a_1, \dots, a_n] \mapsto [a_0, a_1, \dots, a_n]$$

$\bigcap$

- **Sup-lattices** + dual of the free $\oplus$-algebra (see later)
- **Vector spaces**$^{op}_K$ + opposite of free commutative algebra monad
- **Convenient differential category**
- **Harmer-McCusker's category of games**
- **Other categories of games. . .**

Differential Storage Categories

Storage modality

The comonad $(!, \delta, e)$ is a **storage modality** if:

- it is symmetric monoidal,
- $(!A, \Delta, e)$ commutative monoids,
- the comonoid is a morphism of the coalgebras for the comonad.

Differential storage category

A **storage differential category** is a differential category if:

- it has products,
- it has a storage modality.

Theorem [Blute, Cockett, Seely'09]

The coKleisly $\mathbf{C}_!$ of a (monoidal closed) differential storage category $\mathbf{C}$ is a Cartesian (closed) differential category. What's that?

We will see after the break...

Differential Storage Categories

Storage modality

The comonad $(!, \delta, e)$ is a **storage modality** if:

- it is symmetric monoidal,
- $(!A, \Delta, e)$ commutative monoids,
- the comonoid is a morphism of the coalgebras for the comonad.

Differential storage category

A **storage differential category** is a differential category if:

- it has products,
- it has a storage modality.

Theorem [Blute, Cockett, Seely'09]

The coKleisly $\mathbf{C}_!$ of a (monoidal closed) differential storage category $\mathbf{C}$ is a Cartesian (closed) differential category. What's that?

We will see after the break. . .

Conclusions

We have:

- Defined the resource calculus Λ^r,
- Shown the relationship with the differential λ-calculus,
- Explained the notion of a (monoidal) differential category (not enough to model differential λ-calculus!)

After the break we will:

- Introduced the Cartesian closed differential categories,
- Show that they model the differential λ-calculus $\Rightarrow$ the resource calculus,
- Give types/untyped models modeling the Taylor expansion,
- Give a canonical construction SMC $\mapsto$ Differential CCC,
- Apply the construction to categories of games
  - full abstraction of **MRel** for Resource PCF.

Thanks for your attention!

Questions?