

Factoring Derivation Spaces via Intersection Types

APLAS 2018

Pablo Barenbaum

Gonzalo Ciruelos

Derivation spaces

$$\mathbb{D}[1 + 1] = \left(1 + 1 \longrightarrow 2 \right)$$

$$\mathbb{D}[3 + 4 * 5] = \left(3 + 4 * 5 \longrightarrow 3 + 20 \longrightarrow 23 \right)$$

$$\mathbb{D}[(1 + 1, 3 + 4 * 5)] =$$

$$\begin{array}{ccccc} (1 + 1, 3 + 4 * 5) & \longrightarrow & (1 + 1, 3 + 20) & \longrightarrow & (1 + 1, 23) \\ \downarrow & & \downarrow & & \downarrow \\ (2, 3 + 4 * 5) & \longrightarrow & (2, 3 + 20) & \longrightarrow & (2, 23) \end{array}$$

$$\underbrace{\mathbb{D}[(A, B)] \simeq \mathbb{D}[A] \times \mathbb{D}[B]}_{\text{isomorphism of lattices}}$$

Derivation spaces in the λ -calculus

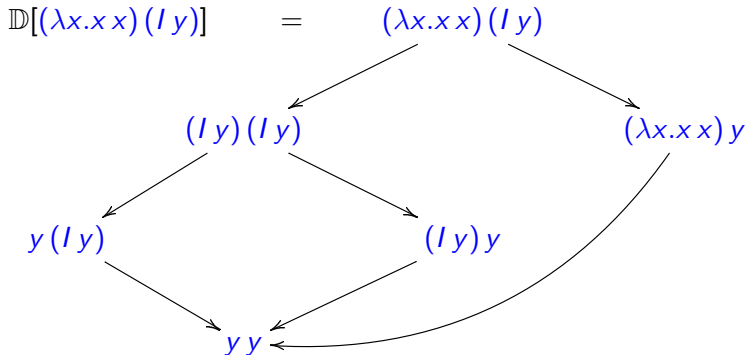
Creation.

$$\begin{aligned} \mathbb{D}[(\lambda x.f(x x)) \lambda x.f(x x)] &= (\lambda x.f(x x)) \lambda x.f(x x) \\ &\quad \downarrow \\ &f((\lambda x.f(x x)) \lambda x.f(x x)) \\ &\quad \downarrow \\ &f(f((\lambda x.f(x x)) \lambda x.f(x x))) \\ &\quad \downarrow \\ &\quad \vdots \end{aligned}$$

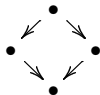
$\mathbb{D}[\lambda x.f(x x)]$ is finite but $\mathbb{D}[(\lambda x.f(x x)) \lambda x.f(x x)]$ is infinite.

Derivation spaces in the λ -calculus

Duplication.



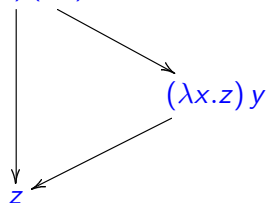
$$\mathbb{D}[(\lambda x.x x) (I y)] \not\cong \underbrace{\mathbb{D}[(\lambda x.x x)\square] \times \mathbb{D}[I y]}$$



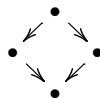
Derivation spaces in the λ -calculus

Erasure.

$$\mathbb{D}[(\lambda x.z)(I y)] = (\lambda x.z)(I y)$$



$$\mathbb{D}[(\lambda x.z)(I y)] \not\approx \underbrace{\mathbb{D}[(\lambda x.z)\square]}_{\text{diamond}} \times \mathbb{D}[I y]$$



Derivation spaces in the λ -calculus

Definition (Derivation space)

If t is a term, $\mathbb{D}[t]$ is the set of **reduction sequences**¹ from t :

$$\{\rho \mid \rho : t \rightarrow^* s \text{ is a sequence of rewrite steps}\} / \equiv$$

Partially ordered by the **prefix order**:

$$[\rho] \sqsubseteq [\sigma] \quad \stackrel{\text{def}}{\iff} \quad \rho/\sigma = \epsilon$$

¹Modulo permutation equivalence.

Derivation spaces in the λ -calculus

Theorem (J.-J. Lévy)

In the λ -calculus, $\mathbb{D}[t]$ forms an **upper semilattice** with:

$$[\rho] \sqcup [\sigma] \stackrel{\text{def}}{=} [\rho(\sigma/\rho)]$$

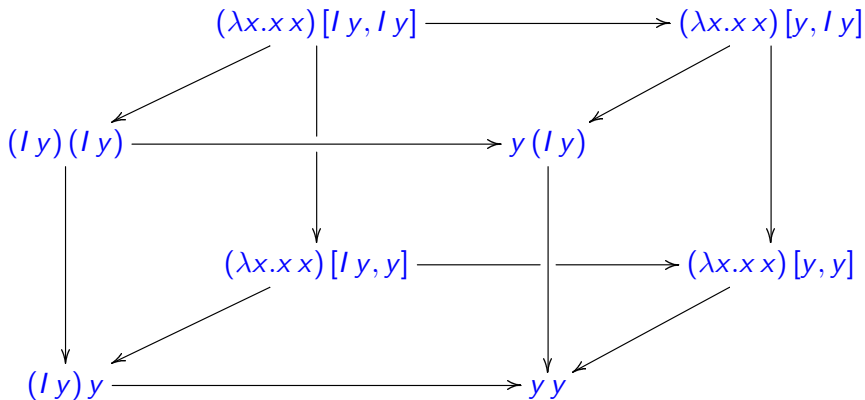
It is not necessarily a lattice.

(Soft) goals and hypotheses

Goal: **understanding derivation spaces.**

Hypothesis: **explicit resource management may be helpful.**

$\mathbb{D}[(\lambda x.x x) [! y, ! y]] =$



The distributive λ -calculus ($\lambda^\#$)

Proof-term notation for the non-idempotent intersection type system \mathcal{W} .

Definition (Proto- $\lambda^\#$)

Syntax.

Terms	$t ::= x^A \mid \lambda x.t \mid t \vec{t}$	Lists of terms	$\vec{t} ::= [t_1, \dots, t_n]$
Types	$A ::= \alpha \mid \mathcal{M} \rightarrow A$	Multisets of types	$\mathcal{M} ::= [A_1, \dots, A_n]$
Contexts	$\Gamma ::= (\cdot) \mid \Gamma, x : \mathcal{M}$		

Typing.

$$\frac{}{x : [A] \vdash x^A : A} \quad \frac{\Gamma, x : \mathcal{M} \vdash t : A}{\Gamma \vdash \lambda x.t : \mathcal{M} \rightarrow A} \quad \frac{\Gamma \vdash t : [A_1, \dots, A_n] \rightarrow B \quad (\Delta_i \vdash s_i : A_i)_{i=1}^n}{\Gamma +_{i=1}^n \Delta_i \vdash t[s_1, \dots, s_n] : B}$$

Reduction.

$$(\lambda x.t)[s_1, \dots, s_n] \longrightarrow_{\#} t\{x := [s_1, \dots, s_n]\}$$

Each free occurrence of x^A consumes exactly one argument s_i of type A .

The distributive λ -calculus ($\lambda^\#$)

Proof-term notation for the non-idempotent intersection type system \mathcal{W} .

Definition (Proto- $\lambda^\#$)

Syntax.

Terms	$t ::= x^A \mid \lambda x.t \mid t \vec{t}$	Lists of terms	$\vec{t} ::= [t_1, \dots, t_n]$
Types	$A ::= \alpha \mid \mathcal{M} \rightarrow A$	Multisets of types	$\mathcal{M} ::= [A_1, \dots, A_n]$
Contexts	$\Gamma ::= (\cdot) \mid \Gamma, x : \mathcal{M}$		

Typing.

$$\frac{}{x : [A] \vdash x^A : A} \quad \frac{\Gamma, x : \mathcal{M} \vdash t : A}{\Gamma \vdash \lambda x.t : \mathcal{M} \rightarrow A} \quad \frac{\Gamma \vdash t : [A_1, \dots, A_n] \rightarrow B \quad (\Delta_i \vdash s_i : A_i)_{i=1}^n}{\Gamma +_{i=1}^n \Delta_i \vdash t[s_1, \dots, s_n] : B}$$

Reduction.

$$(\lambda x.t)[s_1, \dots, s_n] \longrightarrow_{\#} t\{x := [s_1, \dots, s_n]\}$$

Each free occurrence of x^A consumes exactly one argument s_i of type A .

(Non-confluent).

$$(\lambda x.(\lambda y.f y x)x)[a, b]$$

The distributive λ -calculus ($\lambda^\#$)

Definition ($\lambda^\#$)

Syntax.

Terms	$t ::= x^A \mid \lambda^L x.t \mid t \vec{t}$	Lists of terms	$\vec{t} ::= [t_1, \dots, t_n]$
Types	$A ::= \alpha^L \mid \mathcal{M} \xrightarrow{L} A$	Multisets of types	$\mathcal{M} ::= [A_1, \dots, A_n]$
Contexts	$\Gamma ::= (\cdot) \mid \Gamma, x : \mathcal{M}$		

Typing.

$$\frac{}{x : [A] \vdash x^A : A} \quad \frac{\Gamma, x : \mathcal{M} \vdash t : A}{\Gamma \vdash \lambda^L x.t : \mathcal{M} \xrightarrow{L} A} \quad \frac{\Gamma \vdash t : [A_1, \dots, A_n] \xrightarrow{L} B \quad (\Delta_i \vdash s_i : A_i)_{i=1}^n}{\Gamma +_{i=1}^n \Delta_i \vdash t[s_1, \dots, s_n] : B}$$

Reduction.

$$(\lambda^L x.t)[s_1, \dots, s_n] \xrightarrow{L}_\# t\{x := [s_1, \dots, s_n]\}$$

Each free occurrence of x^A consumes exactly one argument s_i of type A .

The distributive λ -calculus ($\lambda^\#$)

Remark (Unique typing)

If $\Gamma \vdash t : A$ is derivable, there is a unique typing derivation for t .

Definition (Correct terms)

A typable term t is **correct** if:

- ▶ Different lambdas are decorated with different labels.
- ▶ Given a multiset of types $[A_1, \dots, A_n]$ occurring as a subformula anywhere in the typing derivation of t , if $i \neq j$ then A_i and A_j are decorated with different labels at the root.

Lemma (Subject reduction)

If $\Gamma \vdash t : A$, the term t is correct and $t \rightarrow_\# s$
then $\Gamma \vdash s : A$ and s is correct.

The distributive λ -calculus ($\lambda^\#$)

Proposition (Confluence)

The $\lambda^\#$ -calculus has the Church–Rosser property.

$$\begin{array}{ccc} (\lambda^1 x. (\lambda^2 y. f^3 y^4 x^5) x^4) [a^4, b^5] & \xrightarrow{1} & (\lambda^2 y. f^3 y^4 b^5) a^4 \\ \downarrow 2 & & \downarrow 2 \\ (\lambda^1 x. f^3 x^4 x^5) [a^4, b^5] & \xrightarrow{1} & f^3 a^4 b^5 \end{array}$$

The distributive λ -calculus ($\lambda^\#$)

Proposition (Strong normalization)

There is no infinite reduction sequence $t_1 \rightarrow_\# t_2 \rightarrow_\# \dots$

[cf. System \mathcal{W}]

Residuals can be defined in $\lambda^\#$ using the labels over the lambdas.

Lemma

There is no duplication nor erasure in $\lambda^\#$.

Proposition

In the $\lambda^\#$ -calculus, $\mathbb{D}[t]$ is a distributive lattice.

There are joins (\sqcup) and meets (\sqcap) that distribute over each other.

Simulation

Definition (Refinement)

Refinement (\bowtie) relates correct $\lambda^\#$ -terms and λ -terms:

$$\frac{}{x^A \bowtie x} \quad \frac{t' \bowtie t}{\lambda^L x. t' \bowtie \lambda x. t} \quad \frac{t' \bowtie t \quad (s'_i \bowtie s)_{i=1}^n}{t' [s'_1, \dots, s'_n] \bowtie t s}$$

A λ -term may have many refinements:

$$\begin{aligned} & (\lambda^1 x. y^2 [] []) [] \quad \bowtie \quad (\lambda x. y x x) z \\ & (\lambda^1 x. y^2 [] [x^3]) [z^3] \quad \bowtie \quad (\lambda x. y x x) z \\ & (\lambda^1 x. y^2 [x^3] [x^4, x^5]) [z^5, z^3, z^4] \quad \bowtie \quad (\lambda x. y x x) z \\ & \dots \end{aligned}$$

Simulation

Proposition (Simulation)

Forward. If $t' \times t \rightarrow_{\beta} s$ there is a term s' such that:

$$\begin{array}{ccc} t & \xrightarrow{\beta} & s \\ \times & & \times \\ t' & \xrightarrow{\#} & s' \end{array}$$

Reverse. If $t \times t' \rightarrow_{\#} s'$ there are terms s, s'' such that:

$$\begin{array}{ccc} t & \xrightarrow{\beta} & s \\ \times & & \times \\ t' & \xrightarrow{\#} s' & \xrightarrow{\#} s'' \end{array}$$

Simulation

Proposition (Refinement characterizes head normalization)

The following are equivalent:

1. The term t has a refinement $t' \times t$.
2. The term t has a head normal form.

[cf. System \mathcal{W}]

Simulation

Proposition (Algebraic simulation)

For each refinement $t' \times t$ the construction given by the Simulation result is a morphism of upper semilattices:

$$\begin{aligned} \mathbb{D}[t] &\rightarrow \mathbb{D}[t'] \\ \rho &\mapsto \rho/t' \end{aligned}$$

Its definition and properties resemble residual theory.

E.g. there is a “cube lemma”:

$$(\rho/t')/(\sigma/t') \equiv (\rho/\sigma)/(t'/\sigma)$$

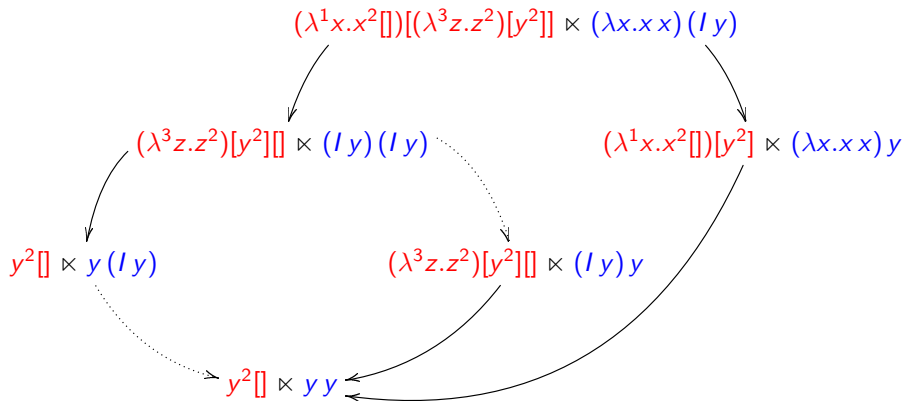
Garbage and factorization

Definition (Garbage)

Let $t' \times t$. A derivation $\rho : t \rightarrow_{\beta}^* s$ is t' -**garbage** if $\rho/t' = \epsilon$.

The notion of garbage depends on the choice of t' .

The dotted steps are garbage:



Garbage and factorization

Theorem (Factorization)

If $t' \times t$ there is an isomorphism of upper semilattices:

$$\mathbb{D}[t] \simeq \int_{\mathcal{F}} \mathcal{G}$$

where:

- ▶ $\int_{\mathcal{F}} \mathcal{G}$ is the Grothendieck construction.
- ▶ \mathcal{F} is the lattice of **garbage-free** derivations.
- ▶ $\mathcal{G} : \mathcal{F} \rightarrow \text{Semilattice}$ is a functor.
For each $\rho : t \rightarrow_{\beta}^* s$ in \mathcal{F} , we write $\mathcal{G}(\rho)$ for the semilattice of **garbage** derivations starting at s .

In particular, for any derivation $\rho : t \rightarrow_{\beta}^* s$ there is a **unique** way to factor $\rho \equiv \rho_1 \rho_2$ such that ρ_1 is garbage-free and ρ_2 is garbage.

Future work

- ▶ Generalize to other rewriting systems.
- ▶ Show that the notion of garbage is not *ad hoc*.
- ▶ Relate with Mellès external–internal factorization.
- ▶ Use other resource calculi instead of $\lambda^\#$.