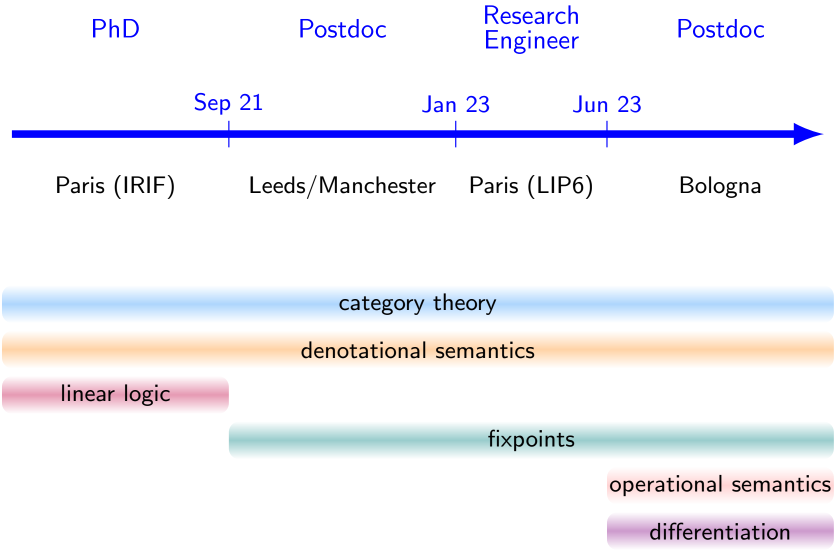
CNRS Audition, Zeinab Galal



General setting

Syntax Curry-Howard-Lambek **Semantics**

program interpretation

$A \vdash P : B$ $\llbracket A \rrbracket \xrightarrow{\llbracket P \rrbracket} \llbracket B \rrbracket$

mathematical object
with some structure

General setting

$$\begin{tikzcd}[column sep=huge] \textbf{Syntax} \arrow[r, bend left] \arrow[r, phantom, "\text{Curry-Howard-Lambek}"] & \textbf{Semantics} \arrow[l, bend left] \end{tikzcd}$$

program

$A \vdash P : B$

interpretation

$$[\![A]\!] \xrightarrow{[\![P]\!]} [\![B]\!]$$

map preserving
the structure

General setting

$$\begin{tikzcd} \textbf{Syntax} \arrow[rr, bend left] \arrow[rr, phantom, "\text{Curry-Howard-Lambek}"] & & \textbf{Semantics} \arrow[ll, bend left] \end{tikzcd}$$

general dynamic

more constructs on programs $\leftrightsquigarrow$ **richer models**

- finer invariants

cost, resource consumption
probability of termination,...

- quantitative semantics

graded, differential
probabilistic models, ...

- reason directly on execution steps

$$P \xrightarrow{\text{execution}} Q$$

- 2-dimensional semantics

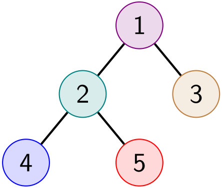
$$\begin{tikzcd} \llbracket A \rrbracket \arrow[rr, bend left, "\llbracket P \rrbracket"] \arrow[rr, bend right, "\llbracket Q \rrbracket"'] \arrow[rr, phantom, "\Downarrow \llbracket \text{execution} \rrbracket"] & & \llbracket B \rrbracket \end{tikzcd}$$

An important model from combinatorics

- **Joyal:** species of structures

set of structures

$$\{1, 2, 3, 4, 5\}$$



count the number of structures

generating series $f : x \mapsto 1 + x + 2x^2 + 5x^3 + 14x^4 + \dots$

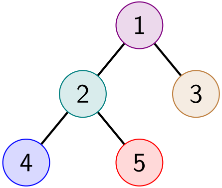
$$f + g = g + f \qquad (f \circ g) \circ h = f \circ (g \circ h)$$

An important model from combinatorics

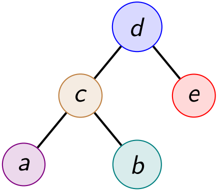
- **Joyal:** species of structures

set of structures $+$ relabelling action

$$\{1,2,3,4,5\} \xrightarrow[\sigma]{\sim} \{a,b,c,d,e\}$$



$$F(\sigma)\ \xrightarrow{\sim}$$



count the number of structures $+$ relabelling action

generating functor $X \mapsto 1 + X + 2X^2 + 5X^3 + 14X^4 + \dots$

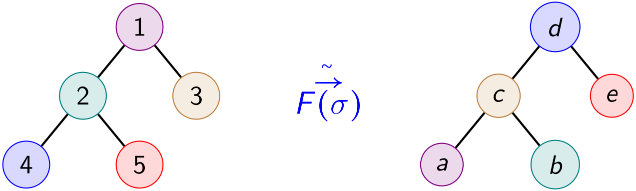
$$F + G \xRightarrow[\text{isomorphism}]{\text{rewriting}} G + F \qquad (F \circ G) \circ H \xRightarrow[\text{isomorphism}]{\text{rewriting}} F \circ (G \circ H)$$

An important model from combinatorics

- **Joyal:** species of structures

set of structures $\quad + \quad$ relabelling action

$$\{1,2,3,4,5\} \quad \overset{\sim}{\underset{\sigma}{\longrightarrow}} \quad \{a,b,c,d,e\}$$



- **Fiore et al:** 2-dimensional model of generalized species

$$P \xrightarrow{\beta} Q \quad \rightsquigarrow \quad [\![P]\!] \xRightarrow{[\![\beta]\!]} [\![Q]\!]$$

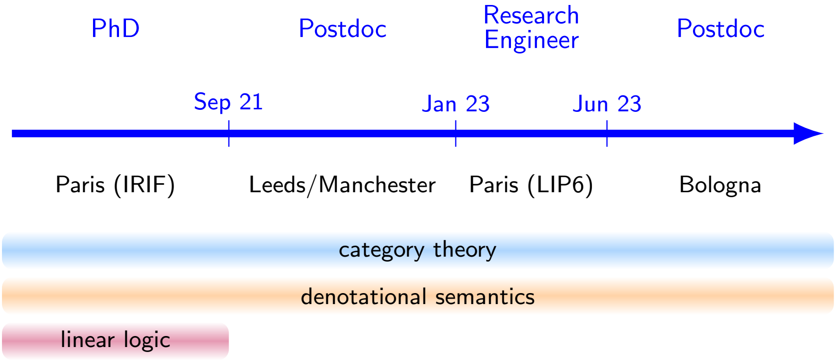
execution step in λ-calculus

higher order species

rewriting 2-map

PhD thesis

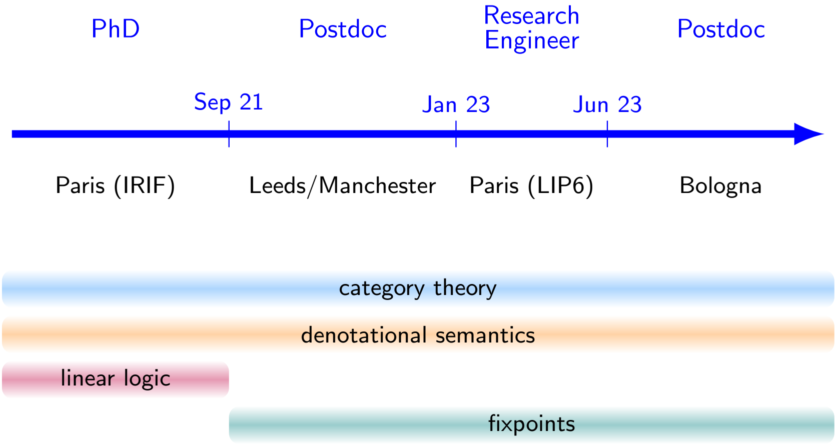
My thesis: generalize linear logic duality methods to dimension 2 for the species model where we need to take into account the reductions.



- FSCD 20, G.
- FSCD 21, G.
- FSCD 22, Fiore, G., Paquet
- LMCS 24, Fiore, G., Paquet

1st visit to M. Fiore in Cambridge

First postdoc



- MFPS 23, Fiore, G., Jafarrahmani 2nd visit to M. Fiore in Cambridge
- LICS 23, G.

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

fixpoint unfolding

$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

fixpoint unfolding

$$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$$

variable elimination

$$\mathbf{fix}_x\, \mathbf{fix}_y\, f(x, y) \longrightarrow \mathbf{fix}_x\, f(x, x)$$

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

system of equations

$$\mathbf{fix}_{x,y}(f(x,y), g(x,y)) \longrightarrow (\mathbf{fix}_x\, f(x, \mathbf{fix}_y\, .g(x,y)), \mathbf{fix}_y\, .g(\mathbf{fix}_x\, .f(x,y), y))$$

fixpoint unfolding

$$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$$

variable elimination

$$\mathbf{fix}_x\, \mathbf{fix}_y\, f(x,y) \longrightarrow \mathbf{fix}_x\, f(x,x)$$

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

system of equations

$$\begin{array}{r}\mathbf{fix}_{x,y}(f(x,y), \\ g(x,y))\end{array} \longrightarrow \begin{array}{l}(\mathbf{fix}_x\, f(x, \mathbf{fix}_y .g(x,y)), \\ \mathbf{fix}_y .g(\mathbf{fix}_x .f(x,y), y))\end{array}$$

fixpoint unfolding

$$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$$

variable elimination

$$\mathbf{fix}_x\, \mathbf{fix}_y\, f(x,y) \longrightarrow \mathbf{fix}_x\, f(x,x)$$

Fixpoint operator in **dimension 1**:

$$\begin{aligned}\mathbf{fix}(f) &= f(\mathbf{fix}\, f) \\ \mathbf{fix}\,\mathbf{fix}(f) &= \mathbf{fix}(f \circ \Delta)\end{aligned}$$

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

system of equations

$$\mathbf{fix}_{x,y}(f(x,y), g(x,y)) \longrightarrow (\mathbf{fix}_x\, f(x, \mathbf{fix}_y\, .g(x,y)), \mathbf{fix}_y\, .g(\mathbf{fix}_x\, .f(x,y), y))$$

fixpoint unfolding

$$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$$

variable elimination

$$\mathbf{fix}_x\, \mathbf{fix}_y\, f(x,y) \longrightarrow \mathbf{fix}_x\, f(x,x)$$

Fixpoint operator in **dimension 2**:

$$\mathbf{fix}\, f \xRightarrow{\text{unfold}} f(\mathbf{fix}\, f)$$

$$\mathbf{fix}\,\mathbf{fix}(f) \xRightarrow{\text{diag}} \mathbf{fix}(f \circ \Delta)$$

First posdoc: Fixpoint reductions

Main question: what is the equational theory of term fixpoint reductions?

fixpoint unfolding

$$\mathbf{fix}_x\, f(x) \longrightarrow f(\mathbf{fix}_x\, f(x))$$

system of equations

$$\mathbf{fix}_{x,y}(f(x,y), g(x,y)) \longrightarrow (\mathbf{fix}_x\, f(x, \mathbf{fix}_y\, .g(x,y)), \mathbf{fix}_y\, .g(\mathbf{fix}_x\, .f(x,y), y))$$

variable elimination

$$\mathbf{fix}_x\, \mathbf{fix}_y\, f(x,y) \longrightarrow \mathbf{fix}_x\, f(x,x)$$

Fixpoint operator in **dimension 2**:

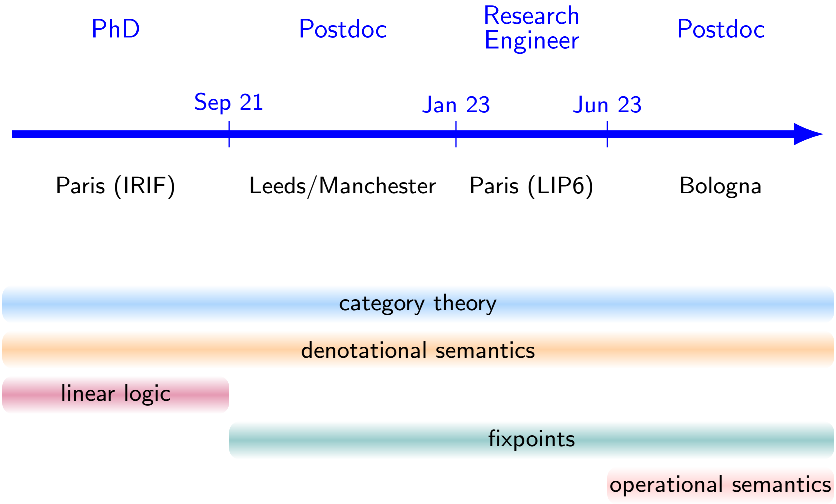
$$\mathbf{fix}\, f \xRightarrow{\text{unfold}} f(\mathbf{fix}\, f)$$

$$\mathbf{fix}\,\mathbf{fix}(f) \xRightarrow{\text{diag}} \mathbf{fix}(f \circ \Delta)$$

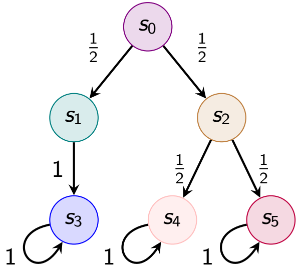
Main difficulty: how to come up with the right equations and ensure that we have them all?

LICS 2023, G. *Fixpoint operators for 2-categorical structures.*

Current postdoc



Current postdoc: coinduction and behavioral metrics

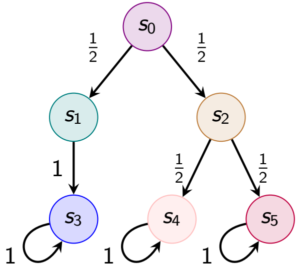


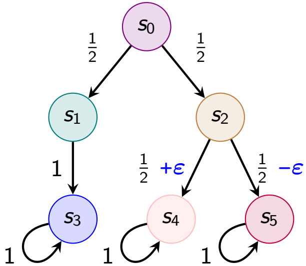
behavioral equivalence

$$s_1 \sim s_2$$

$$\sim \subseteq \mathrm{States} \times \mathrm{States}$$

Current postdoc: coinduction and behavioral metrics





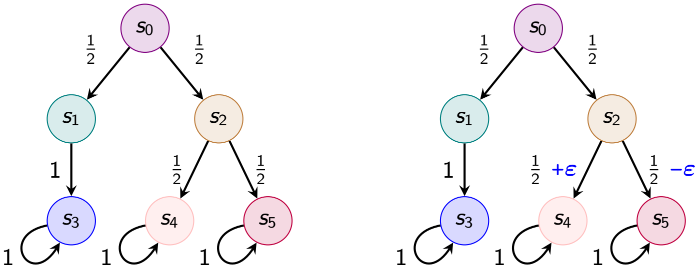
behavioral equivalence $\rightsquigarrow$ **behavioral distance**

$s_1 \sim s_2$ $\qquad$ $d(s_1, s_2) = \varepsilon$

$\sim \subseteq \mathrm{States} \times \mathrm{States}$ $\qquad$ $d : \mathrm{States} \times \mathrm{States} \to [0, 1]$

probabilistic systems, cost analysis, differential privacy, sensitivity, ...

Current postdoc: coinduction and behavioral metrics

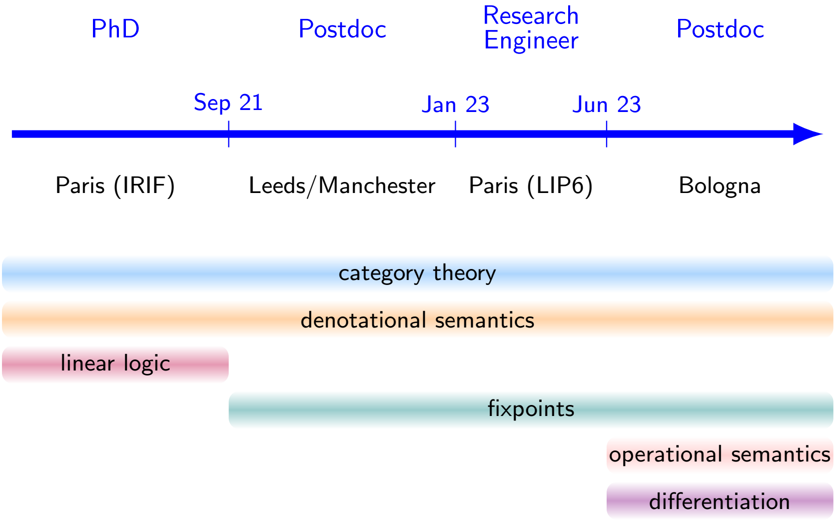


behavioral equivalence $\rightsquigarrow$ **behavioral distance**

Current postdoc: joint work with Ugo dal Lago

- Bisimulation metric for computational indistinguishability
  $\rightsquigarrow$ family of distances dependent on the security parameter
- Higher order behavioral metrics
  main difficulty higher order programs can arbitrarily amplify distances

Current postdoc



- Submitted, G., Lemay

Detailed result 1: fixpoints and derivatives

with Jean-Simon Pacaud Lemay. "Combining fixpoint and differentiation theory." *Submitted*.

Syntax: differential λ-calculus

$$\frac{\partial P}{\partial x} \cdot Q$$

linear substitution

Taylor expansion

Main question: what is the meaning of $\mathbf{fix}\,\dfrac{\partial M}{\partial x}$ or $\dfrac{\partial\,\mathbf{fix}\,M}{\partial x}$?

Detailed result 1: fixpoints and derivatives

with Jean-Simon Pacaud Lemay. "Combining fixpoint and differentiation theory." *Submitted*.

Syntax: differential λ-calculus

$$\frac{\partial P}{\partial x} \cdot Q$$

linear substitution

Taylor expansion

Main question: what is the meaning of $\mathbf{fix}\,\dfrac{\partial M}{\partial x}$ or $\dfrac{\partial\,\mathbf{fix}\,M}{\partial x}$?

Detailed result 1: fixpoints and derivatives

with Jean-Simon Pacaud Lemay. "Combining fixpoint and differentiation theory." *Submitted*.

Syntax: differential λ-calculus

$$\frac{\partial P}{\partial x} \cdot Q$$

linear substitution

Taylor expansion

Main question: what is the meaning of $\mathbf{fix}\,\dfrac{\partial M}{\partial x}$ or $\dfrac{\partial\,\mathbf{fix}\,M}{\partial x}$?

Semantics: Cartesian (closed) differential categories

differentiation operator $\mathbf{D}$

smooth functions

power series

Main question: what is the meaning of $\mathbf{fix}\,\mathbf{D}\,f$ or $\mathbf{D}\,\mathbf{fix}\,f$?

Detailed result 2: fixpoints and derivatives

$$\frac{f : X \to Y}{\mathbf{D}(f) : X \times X \to Y}$$

Directional derivative

$$(x, a) \longmapsto f'(x) \cdot a$$

Detailed result 2: fixpoints and derivatives

$$\frac{f : X \to Y}{\mathbf{D}(f) : X \times X \to Y} \quad \rightsquigarrow \quad \frac{f : X \to Y}{\mathbf{T}(f) : X \times X \to Y \times Y}$$

Directional derivative

$$(x, a) \longmapsto f'(x) \cdot a$$

Tangent bundle

$$(x, a) \longmapsto (f(x), f'(x) \cdot a)$$

Detailed result 2: fixpoints and derivatives

$$\frac{f : X \to Y}{\mathbf{D}(f) : X \times X \to Y} \rightsquigarrow \frac{f : X \to Y}{\mathbf{T}(f) : X \times X \to Y \times Y}$$

Directional derivative

Tangent bundle

$$\frac{f : \Gamma \times X \to X}{\mathbf{fix}(f) : \Gamma \to X}$$

fixpoint

Detailed result 2: fixpoints and derivatives

$$\frac{f : X \to Y}{\mathbf{D}(f) : X \times X \to Y} \quad \rightsquigarrow \quad \frac{f : X \to Y}{\mathbf{T}(f) : X \times X \to Y \times Y}$$

Directional derivative — Tangent bundle

$$\frac{f : \Gamma \times X \to X}{\mathbf{fix}(f) : \Gamma \to X}$$

fixpoint

Computing the **derivative of the fixpoint** is equivalent to computing the **fixpoint of the tangent**.

$$\mathbf{D}\,\mathbf{fix}(f) = \mathbf{fix}(\mathbf{T}(f)c)$$

Detailed result 2: fixpoints and derivatives

$$\frac{f : X \to Y}{\mathbf{D}(f) : X \times X \to Y} \quad \rightsquigarrow \quad \frac{f : X \to Y}{\mathbf{T}(f) : X \times X \to Y \times Y}$$

Directional derivative

Tangent bundle

$$\frac{f : \Gamma \times X \to X}{\mathbf{fix}(f) : \Gamma \to X}$$

fixpoint

Computing the **derivative of the fixpoint** is equivalent to computing the **fixpoint of the tangent**.

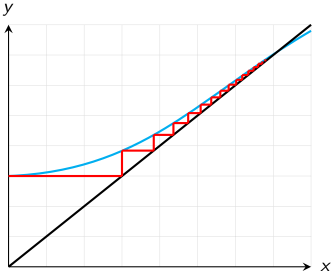
$$\mathbf{D}\,\mathbf{fix}(f) = \mathbf{fix}(\mathbf{T}(f)c)$$

Main result:

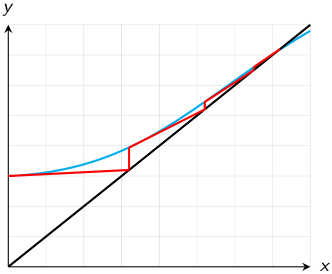
- Axiomatize Cartesian (closed) differential fixpoint categories
- Canonical models of Cartesian (closed) differential categories where we can compute fixpoints are examples

Detailed result 3: Newton-Raphson iteration

Fast computation of fixpoints using derivatives



Kleene iteration



Newton iteration

$$K_0 = 0$$
$$K_{i+1} = f(K_i)$$

$$N_0 = 0$$
$$N_{i+1} = \frac{1}{1 - f'(N_i)}(f(N_i) - f'(N_i) \cdot N_i)$$

Detailed result 3: Newton-Raphson iteration

- **Combinatorial species** Labelle, Lacoste, Pivoteau, Salvy, Soria, ...

Detailed result 3: Newton-Raphson iteration

- **Combinatorial species** $\mathbf{BinTree}(A) = \mathbf{fix}_X(1 + AX^2)$

species defined implicitly $\xrightarrow[\text{using}]{\text{fast computation}}$ Strahler trees + species derivative

Detailed result 3: Newton-Raphson iteration

- **Combinatorial species** $\mathbf{BinTree}(A) = \mathbf{fix}_X(1 + AX^2)$

species defined implicitly $\xrightarrow[\text{using}]{\text{fast computation}}$ Strahler trees + species derivative

- **Power series over a continuous semi-ring** Etessami, Yannakakis, Esparza, Kiefer, Luttenberger, Schlund, ...

Detailed result 3: Newton-Raphson iteration

- **Combinatorial species** $\mathbf{BinTree}(A) = \mathbf{fix}_X(1 + AX^2)$

species defined implicitly	$\xrightarrow[\text{using}]{\text{fast computation}}$	Strahler trees	+	species derivative

- **Power series over a continuous semi-ring** $\mathbf{G} : X \to \epsilon \mid AXX$

language generated by a context-free grammar	$\xrightarrow[\text{using}]{\text{fast computation}}$	regular grammars	+	Brzozowski derivative

Detailed result 3: Newton-Raphson iteration

- **Combinatorial species** $\mathbf{BinTree}(A) = \mathbf{fix}_X(1 + AX^2)$

species defined implicitly $\xrightarrow[\text{using}]{\text{fast computation}}$ Strahler trees + species derivative

- **Power series over a continuous semi-ring** $\mathbf{G} : X \to \epsilon \mid AXX$

language generated by a context-free grammar $\xrightarrow[\text{using}]{\text{fast computation}}$ regular grammars + Brzozowski derivative

Main questions:

- Can we do the same in an arbitrary Cartesian differential fixpoint category?
- How do we measure convergence rates at this level of generality?

Detailed result 4: Newton-Raphson iteration

$$\begin{array}{c}\text{non-linear}\\ \text{fixpoint operator}\end{array} \xrightarrow[\text{using}]{\text{fast computation}} \begin{array}{c}\text{linear}\\ \text{fixpoint operator}\end{array} + \begin{array}{c}\text{differentiation}\\ \text{operator}\end{array}$$

Detailed result 4: Newton-Raphson iteration

$$\begin{array}{c}\text{non-linear}\\ \text{fixpoint operator}\end{array} \xrightarrow[\text{using}]{\text{fast computation}} \begin{array}{c}\text{linear}\\ \text{fixpoint operator}\end{array} + \begin{array}{c}\text{differentiation}\\ \text{operator}\end{array}$$

We assume that:

- the **fixpoint-differentiation axiom** holds
- it is a **Taylor category**

Detailed result 4: Newton-Raphson iteration

$$\begin{array}{c}\text{non-linear}\\ \text{fixpoint operator}\end{array} \xrightarrow[\text{using}]{\text{fast computation}} \begin{array}{c}\text{linear}\\ \text{fixpoint operator}\end{array} + \begin{array}{c}\text{differentiation}\\ \text{operator}\end{array}$$

We assume that:

- the **fixpoint-differentiation axiom** holds
- it is a **Taylor category**
  - $\rightsquigarrow$ Taylor expansion: for $f : X \to Y$, $f(a+b) = f(a) + \frac{\partial f}{\partial x}(a) \cdot b + \dots$
  - $\rightsquigarrow$ canonical metric for $f, g : X \to Y$,

$$d(f,g) = \frac{1}{2^k} \qquad \Leftrightarrow \qquad \text{“} fx - gx = o(x^k) \text{”}$$

Detailed result 4: Newton-Raphson iteration

$$\begin{array}{c}\text{non-linear}\\ \text{fixpoint operator}\end{array} \xrightarrow[\text{using}]{\text{fast computation}} \begin{array}{c}\text{linear}\\ \text{fixpoint operator}\end{array} + \begin{array}{c}\text{differentiation}\\ \text{operator}\end{array}$$

We assume that:

- the **fixpoint-differentiation axiom** holds
- it is a **Taylor category**

Kleene-Scott fixpoint

$$\mathbf{K}_0 = 0$$

$$\mathbf{K}_{i+1} = f(\mathbf{K}_i)$$

$$d(\mathbf{fix}\, f, \mathbf{K}_{i+1}) = \frac{1}{2} d(\mathbf{fix}\, f, \mathbf{K}_i)$$

Newton-Raphson fixpoint

$$\mathbf{N}_0 = 0$$

$$\mathbf{N}_{i+1} = \mathbf{fixlin}(\tfrac{\partial f}{\partial x}(\mathbf{N}_i)) \cdot (f(\mathbf{N}_i) \ominus \tfrac{\partial f}{\partial x}(\mathbf{N}_i) \cdot \mathbf{N}_i)$$

$$d(\mathbf{fix}\, f, \mathbf{N}_{i+1}) = \frac{1}{2} (d(\mathbf{fix}\, f, \mathbf{N}_i))^2$$

Project 1: differential equations of programs

What does it mean for a program to be a solution of a differential equation?

Project 1: differential equations of programs

What does it mean for a program to be a solution of a differential equation?

Long term: use semantics as a guideline to design languages with differentiation and recursion

- More general iterative schemes: solve or approximate solutions of differential equations
- Generalize existing methods from combinatorics
- Computational meaning in terms of resource usage
- Feedback into the syntax

Project 1: differential equations of programs

What does it mean for a program to be a solution of a differential equation?

Long term: use semantics as a guideline to design languages with differentiation and recursion

- More general iterative schemes: solve or approximate solutions of differential equations
- Generalize existing methods from combinatorics
- Computational meaning in terms of resource usage
- Feedback into the syntax

LIP (Plume)	LIPN (LoVe)	LIS (LSC)
Doumane, Kuperberg, Pous	Kerjean, Mazza	Santocanale
Pagani, Pistone	Breuvart	ANR LamdaComb
ARIC team: Salvy	CALIN team: Banderier, Bodini	I2M team

Project 1: differential equations of programs

What does it mean for a program to be a solution of a differential equation?

Long term: use semantics as a guideline to design languages with differentiation and recursion

- More general iterative schemes: solve or approximate solutions of differential equations
- Generalize existing methods from combinatorics
- Computational meaning in terms of resource usage
- Feedback into the syntax

Short term:

- develop a notion of local fixpoints to combine it with reverse differentiation
- potential application: AD for functions defined implicitly

Project 2: fixpoints in dimension 2

Solving/approximating systems of least and greatest fixpoint equations is important for concurrency, static analysis, program analysis, games, etc.

- existing methods: mainly for lattices, systems weighted over reals

Project 2: fixpoints in dimension 2

Solving/approximating systems of least and greatest fixpoint equations is important for concurrency, static analysis, program analysis, games, etc.

- existing methods: mainly for lattices, systems weighted over reals

General goal: fixpoint approximation and optimization in dimension 2

- finer notions of approximation: more complex quantitative systems

Project 2: fixpoints in dimension 2

Solving/approximating systems of least and greatest fixpoint equations is important for concurrency, static analysis, program analysis, games, etc.

- existing methods: mainly for lattices, systems weighted over reals

General goal: fixpoint approximation and optimization in dimension 2

- finer notions of approximation: more complex quantitative systems
- generalize and transport proof methods by duality

$$\begin{tikzcd} \textbf{Induction} \arrow[r, bend left] \arrow[r, phantom, "\text{duality}"] & \textbf{Coinduction} \arrow[l, bend left] \end{tikzcd}$$

Induction	Coinduction
sound approximations of least fixpoints from above	sound approximations of greatest fixpoints from below
e.g. abstract interpretation	e.g. up-to methods for bisimulation

Project 2: fixpoints in dimension 2

Solving/approximating systems of least and greatest fixpoint equations is important for concurrency, static analysis, program analysis, games, etc.

- existing methods: mainly for lattices, systems weighted over reals

General goal: fixpoint approximation and optimization in dimension 2

- finer notions of approximation: more complex quantitative systems
- generalize and transport proof methods by duality

LIP (Plume)	LIPN (LoVe)	LIS (LSC)
Pous, Vignudelli	Mazza	Santocanale
Mio, Pistone, Vignudelli	Kerjean, Mazza	Clairambault
Laurent, Riba	Seiller, Breuvart	Crubillé

Project 2: fixpoints in dimension 2

Solving/approximating systems of least and greatest fixpoint equations is important for concurrency, static analysis, program analysis, games, etc.

- existing methods: mainly for lattices, systems weighted over reals

General goal: fixpoint approximation and optimization in dimension 2

- finer notions of approximation: more complex quantitative systems
- generalize and transport proof methods by duality

Ongoing work:

- guarded fixpoints (ensure progress, safety)
- systems of fixpoint equations combining least and greatest fixpoint reductions in dimension 2

Project 3: generalized metrics and convergence

Program analysis
Combinatorics
Numerical methods

Categorical semantics

wide range of optimization
approximation methods

$\xrightarrow{\text{internalize}}$

reasoning principles
e.g. universal properties

higher order viewpoint
e.g. Newton approximant is
a term $\mathbf{N} : A \Rightarrow A \to A \Rightarrow A$

advanced complexity and
asymptotic analysis

$\xrightarrow[\text{content}]{\text{computational}}$

richer notions of metrics
and convergence

LIP (Plume)

Pagani

Mio, Pistone, Vignudelli

ARIC: Salvy

LIPN (LoVe)

Kerjean, Mazza

Breuvart

CALIN: Banderier, Bodini

LIS (LSC)

Clairambault

Crubillé

Project 3: generalized metrics and convergence

Program analysis

Combinatorics

Numerical methods

Categorical semantics

wide range of optimization approximation methods $\overset{\text{proof methods}}{\leftsquigarrow}$ reasoning principles e.g. universal properties

$\overset{\text{connect and unify}}{\underset{\text{different areas}}{\leftsquigarrow}}$ higher order viewpoint e.g. Newton approximant is a term $\mathbf{N} : A \Rightarrow A \to A \Rightarrow A$

advanced complexity and asymptotic analysis $\overset{\text{feedback into design}}{\underset{\text{finer invariants}}{\leftsquigarrow}}$ richer notions of metrics and convergence

LIP (Plume)	LIPN (LoVe)	LIS (LSC)
Pagani	Kerjean, Mazza	Clairambault
Mio, Pistone, Vignudelli	Breuvart	Crubillé
ARIC: Salvy	CALIN: Banderier, Bodini	

Thank you