Formalization & Computation: Categorical Normalization by Evaluation

David G. Berry Marcelo P. Fiore

University of Cambridge

2$^{\text{nd}}$ HoTT Conference
Carnegie Mellon University
May 2023

Outline

Categorical Normalization

1. Normalization (by Evaluation)
2. Naïve Categorical Normalization
3. P-Category Theory & P-Categorical Normalization
4. Correctness

Coq Formalization

1. Design Decisions
2. Basic Constructions
3. Cartesian Structures
4. (Co)Ends & Presheaf Exponential

Why Normalization?

Problem

Decide for STLC.

$$\Gamma \vdash t \equiv_{\beta\eta} t' : T$$

Why Normalization?

Problem

Decide for STLC.

$$\Gamma \vdash t \equiv_{\beta\eta} t' : T$$

Standard Solution

$$\Gamma \vdash \mathsf{nf}(t) \equiv_{\alpha} \mathsf{nf}(t') : T$$

Why Normalization?

Problem

Decide for STLC.

$$\Gamma \vdash t \equiv_{\beta\eta} t' : T$$

Standard Solution

$$\Gamma \vdash \mathsf{nf}(t) \equiv_{\alpha} \mathsf{nf}(t') : T$$

- Characterizes βη-equivalence.
- Clearly decidable.

Why Normalization?

Problem

Decide for STLC.

$$\Gamma \vdash t \equiv_{\beta\eta} t' : T$$

Standard Solution

$$\Gamma \vdash \mathsf{nf}(t) \equiv_{\alpha} \mathsf{nf}(t') : T$$

- Characterizes βη-equivalence.
- Clearly decidable.

Algorithmic Problem

How to compute?

$$\Gamma \vdash t : T \rightsquigarrow \Gamma \vdash \mathsf{nf}(t) : T$$

Normalization by Evaluation

The standard approach after Berger and Schwichtenberg '91 proceeds as follows.

Normalization by Evaluation

The standard approach after Berger and Schwichtenberg '91 proceeds as follows.

1. Define neutral, $\mathcal{M}_{T,\Gamma}$, and normal, $\mathcal{N}_{T,\Gamma}$, terms, as subsets of all terms, $\mathcal{L}_{T,\Gamma}$.
2. Define a particular model for types, $[\![T]\!]_\Gamma$.
3. Interpret terms into the model, $\mathcal{L}_{T,\Gamma} \to [\![T]\!]_\Gamma$.
4. Define maps
   - $q : \mathcal{M}_{T,\Gamma} \to [\![T]\!]_\Gamma$
   - $u : [\![T]\!]_\Gamma \to \mathcal{N}_{T,\Gamma}$
5. Define $\mathsf{nf} : \mathcal{L}_{T,\Gamma} \to \mathcal{N}_{T,\Gamma}$

Normalization by Evaluation

The standard approach after Berger and Schwichtenberg ‘91 proceeds as follows.

1. Define neutral, $\mathcal{M}_{T,\Gamma}$, and normal, $\mathcal{N}_{T,\Gamma}$, terms, as subsets of all terms, $\mathcal{L}_{T,\Gamma}$.
2. Define a particular model for types, $[\![T]\!]_\Gamma$.
3. Interpret terms into the model, $\mathcal{L}_{T,\Gamma} \to [\![T]\!]_\Gamma$.
4. Define maps
   - $q : \mathcal{M}_{T,\Gamma} \to [\![T]\!]_\Gamma$
   - $u : [\![T]\!]_\Gamma \to \mathcal{N}_{T,\Gamma}$
5. Define $\mathsf{nf} : \mathcal{L}_{T,\Gamma} \to \mathcal{N}_{T,\Gamma}$

Mathematical/Categorical Justification?

From where does all this come?

A.H.S. ‘95 provides some categorical justification, using an *ad-hoc* gluing-style argument.

Č.D.S. ‘98 uses an alternative categorical foundation.

Fiore ‘02 provides a fully categorical foundation using gluing.

Naïve Categorical Normalization (I)

Generic Interpretation

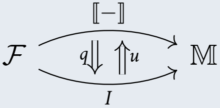
For any Cartesian-closed category, $\mathbb{M}$, there is a universal Cartesian-closed interpretation functor, $[\![-]\!]$, from the free Cartesian-closed category, $\mathcal{F}$, over a basetype:

$$\mathcal{F} \xrightarrow{[\![-]\!]} \mathbb{M}$$

Naïve Categorical Normalization (I)

Generic Interpretation

For any Cartesian-closed category, $\mathbb{M}$, there is a universal Cartesian-closed interpretation functor, $[\![-]\!]$, from the free Cartesian-closed category, $\mathcal{F}$, over a basetype:



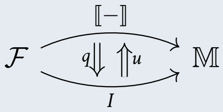
I-Normalization

A normalization function for some model, $\mathbb{M}$, and interpretation functor, $I : \mathcal{F} \to \mathbb{M}$, can be constructed for $\sigma : \mathcal{F}(\Gamma, \Delta)$:

- $u_\Gamma : I(\Gamma) \to [\![\Gamma]\!]$
- $[\![\sigma]\!] : [\![\Gamma]\!] \to [\![\Delta]\!]$
- $q_\Delta : [\![\Delta]\!] \to I(\Delta)$

$\Big\}$ $\mathsf{nf}_I(\sigma) : I(\Gamma) \to I(\Delta)$

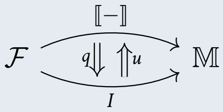
Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/ ✗)

Naïve Categorical Normalization (II)

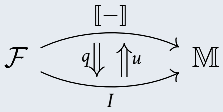
Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/ ✗)
$\mathcal{F}$	Id	✗

Naïve Categorical Normalization (II)

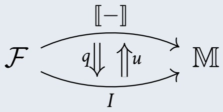
Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/ ✗)
$\mathcal{F}$	Id	✗
$\widehat{\mathcal{F}}$	よ	✗

Naïve Categorical Normalization (II)

Choice of $\mathbb{M}$ and I



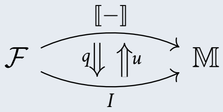
$\mathbb{M}$	I	Utile? (✓/ ✗)
$\mathcal{F}$	Id	✗
$\widehat{\mathcal{F}}$	よ	✗

Problem

In fact, no matter what category we choose for our model all normalization functions will be inutile as they are all extensionally the identity. Following Č.D.S. we switch to a more intensional setting: P-category theory.

Naïve Categorical Normalization (II)

Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/✗)
$\mathcal{F}$	Id	✗
$\widehat{\mathcal{F}}$	よ	✗

Problem

In fact, no matter what category we choose for our model all normalization functions will be inutile as they are all extensionally the identity. Following Č.D.S. we switch to a more intensional setting: P-category theory.

Silver Lining

So far the standard category theory has created a framework for normalization which has avoided defining neutral and normal forms.

Warning!

Nota Bene

In the sequel I use some terminology not in its precise HoTT/UF sense!

P-Category Theory (I) [Č.D.S. '98]

Partial Equivalence Relation (PER)

A relation which is symmetric and transitive.

P-Set

A collection with a given PER.
We denote the underlying collection of a P-set, X, by $|X|$.
We denote the associated PER by $\sim_X$.

P-Category Theory (I) [Č.D.S. '98]

Partial Equivalence Relation (PER)

A relation which is symmetric and transitive.

P-Set

A collection with a given PER.

We denote the underlying collection of a P-set, X, by $|X|$.

We denote the associated PER by $\sim_X$.

Intuition

Think of $(|X|, \sim_X) \simeq \{x : X \mid x \sim x\}/\sim_X$.

This allows simultaneously taking a subset and a quotient.

This provides (co)completeness properties.

P-Category Theory (II)

P-Category

A P-category is given by the following:

- a collection of objects;
- a P-set of arrows between objects;
- a composition operation for arrows; and
- an identity arrow;

} DATA

such that:

- $f \sim f' \wedge g \sim g' \Rightarrow f \circ g \sim f' \circ g'$;
- $f \sim f' \wedge g \sim g' \wedge h \sim h' \Rightarrow (f \circ g) \circ h \sim f' \circ (g' \circ h')$;
- $\mathsf{id}_x \sim \mathsf{id}_x$;
- $f \sim f' \Rightarrow \mathsf{id}_x \circ f \sim f'$; and
- $f \sim f' \Rightarrow f \circ \mathsf{id}_x \sim f'$.

} AXIOMS

P-Category Theory (III)

P-Functor

A P-functor, F, from P-category, $\mathbb{C}$, to P-category, $\mathbb{D}$, is given by the following:

- a map of objects; and
- a P-map of arrows between objects;

} DATA

such that:

- $f \sim f' \Rightarrow F\,f \sim F\,f'$ (this is the P-map condition);
- $f \sim f' \wedge g \sim g' \Rightarrow F\,(f \circ g) \sim F\,f' \circ F\,g'$; and
- $F\,\mathsf{id}_x \sim \mathsf{id}_{F\,x}$.

} AXIOMS

P-Category Theory (IV)

P-Functor Category

The P-functor category, $[\mathbb{C}, \mathbb{D}]$, has:

- P-functors, $\mathbb{C} \to \mathbb{D}$, as objects; and
- *all* transformations as morphisms, where $\alpha \sim \beta$ when:
  - α is P-natural;
  - β is P-natural; and
  - $\alpha_x \sim \beta_x$, for all x.

P-Naturality

α is P-natural when:

$$f \sim f' \Rightarrow (\alpha_y \circ F f) \sim (F f' \circ \alpha_x)$$

P-Category Theory (IV)

P-Functor Category

The P-functor category, $[\mathbb{C}, \mathbb{D}]$, has:

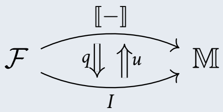
- P-functors, $\mathbb{C} \to \mathbb{D}$, as objects; and
- *all* transformations as morphisms, where $\alpha \sim \beta$ when:
  - α is P-natural;
  - β is P-natural; and
  - $\alpha_x \sim \beta_x$, for all x.

Observation

Note that the PER for the morphisms in functor categories typifies the P-categorical approach of taking subsets, by predicating both α and β.

P-Categorical Normalization

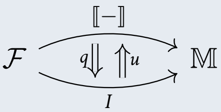
Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/ ✗)
	Id	✗

P-Categorical Normalization

Choice of $\mathbb{M}$ and I



$\mathbb{M}$	I	Utile? (✓/ ✗)
$\mathcal{F}$	Id	✗
$\widehat{\mathcal{F}}$	よ	✓

Success!

By switching into the intensional P-categorical setting we can elucidate the intensional behaviour separately from the extensional properties. We now have a putative computational algorithm: only thing we have to do now is formalize it ...

Correctness

Correctness Properties

Correctness of normalization algorithms arises in these properties:

- $t \equiv_{\beta\eta} t' \Rightarrow \mathsf{nf}(t) \equiv_{\alpha} \mathsf{nf}(t')$;
- $t \equiv_{\beta\eta} \mathsf{nf}(t)$;
- $\mathsf{nf}(t) \in \mathcal{N}$; and
- $t \in \mathcal{N} \Rightarrow t \equiv_{\alpha} \mathsf{nf}(t)$.

Correctness

Correctness Properties

Correctness of normalization algorithms arises in these properties:

- $t \equiv_{\beta\eta} t' \Rightarrow \mathsf{nf}(t) \equiv_{\alpha} \mathsf{nf}(t')$;
- $t \equiv_{\beta\eta} \mathsf{nf}(t)$;
- $\mathsf{nf}(t) \in \mathcal{N}$; and
- $t \in \mathcal{N} \Rightarrow t \equiv_{\alpha} \mathsf{nf}(t)$.

Properties for Free

The P-categorical construction give us the following for free:

- $t \equiv_{\beta\eta} t' \Rightarrow \mathsf{nf}(t) \equiv_{\beta\eta} \mathsf{nf}(t')$; and
- $t \equiv_{\beta\eta} \mathsf{nf}(t)$.

Correctness Proof (Sketch) (I)

Category of Renamings

The category of contexts and context renamings, $\mathcal{R}$, is a subcategory of $\mathcal{F}$ with inclusion functor i.

Presheaves of Neutral and Normal Terms

Neutrality and Normality are preserved under renamings, allowing them to lift to type-indexed families of presheaves.

$$\mathcal{M}, \mathcal{N} : \mathsf{Ty} \to \widehat{\mathcal{R}}$$

Gluing Category

The gluing category, $\mathcal{G} \triangleq \widehat{\mathcal{R}}{\downarrow}i^*$, is Cartesian-closed. Furthermore, the codomain projection functor is Cartesian-closed.

Correctness Proof (Sketch) (II)

Interpretation in $\mathcal{G}$

The interpretation in $\mathcal{G}$ is induced by:

$$\mathcal{G}[\![\iota]\!] \triangleq \begin{array}{c} \mathcal{M}_\iota \\ \downarrow \\ i^*\left(\widehat{\mathcal{F}}[\![\iota]\!]\right) \end{array}$$

We denote the domain presheaf in $\widehat{R}$ by $\mathcal{I}$.

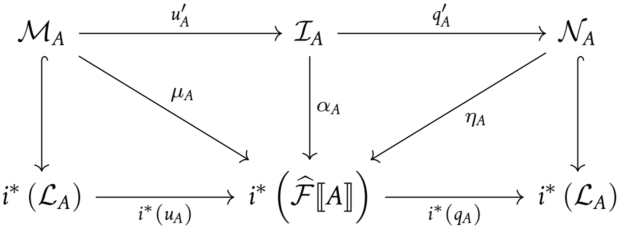
Diagram in $\mathcal{G}$

We define the following type-indexed diagram in $\mathcal{G}$ by induction on A:

$$
\left(\begin{array}{c} \mathcal{M}_A \\ \Big\downarrow \mu_A \\ i^*\left(\widehat{\mathcal{F}}[\![A]\!]\right) \end{array} \right)
\xrightarrow{\mathsf{in}_A}
\left(\begin{array}{c} \mathcal{I}_A \\ \Big\downarrow \alpha_A \\ i^*\left(\widehat{\mathcal{F}}[\![A]\!]\right) \end{array} \right)
\xrightarrow{\mathsf{out}_A}
\left(\begin{array}{c} \mathcal{N}_A \\ \Big\downarrow \eta_A \\ i^*\left(\widehat{\mathcal{F}}[\![A]\!]\right) \end{array} \right)
$$

Correctness Proof (Sketch) (IV)

The following is induced:



Coq Formalization

Design Decisions

- Universe Polymorphism
- Cumulative Records
- Yoneda-Centric Definitions

Coq Formalization: PERs & P-Types

PER

```
Cumulative Record PER@{+i +j} (A : Type@{i}) := Build_PER {
  PER_rel : A -> A -> Type@{j};
  PER_symm : forall {x y}, PER_rel x y -> PER_rel y x;
  PER_trans : forall {x y z}, PER_rel x y -> PER_rel y z -> PER_rel x z;
}.
```

P-Type

```
Cumulative Record PType@{+i +j} : Type := Build_PType {
  PType_type :> Type@{i};
  PType_per :> PER@{i j} PType_type;
}.
```

Coq Formalization: P-Category

P-Category

```
Cumulative Record PCat@{+i +j +k} := Build_PCat {
  PCat_obj :> Type@{i};
  PCat_hom : PCat_obj -> PCat_obj -> PType@{j k};
  PCat_id_mor : forall x, PCat_hom x x;
  PCat_comp : forall {x y z}, PCat_hom y z -> PCat_hom x y -> PCat_hom x z;

  PCat_id_rel : forall x, (PCat_id_mor x) ~ (PCat_id_mor x);
  PCat_comp_rel : forall {x y z f f' g g'},
    f ~ f' -> g ~ g' -> (PCat_comp f g) ~ (PCat_comp f' g');
  ...
}.
```

Coq Formalization: P-Cartesian Structures (I)

P-Terminal Objects

```
Definition IsPTermObj {C : PCat} (term : C) :=
  PNatIso
    (PBiFunPartialRight (@PHomFun C) term)
    (PCompFun (PConstFun (C:=PSet) PUnit) PTermFun).
```

$$\mathsf{IsTerminal}(t) \triangleq \mathsf{Hom}_{\mathbb{C}}(-, t) \cong \Delta_{\{*\}}$$

Coq Formalization: P-Cartesian Structures (II)

P-Cartesian Products

```
Definition IsPCartProd {C : PCat} (prod : C -> C -> C) :=
  forall a b,
    PNatIso
      (PBiFunPartialRight PHomFun (prod a b))
      (PCompFun
        (PBiFunPartialRight (PHomFun (C:=PProdCat C C)) (a, b))
        (POppFun (PPairFun PIdFun PIdFun))
      ).
```

$$\mathsf{IsProduct}(- \times =) \triangleq \prod_{a,b\,:\,\mathbb{C}} \mathsf{Hom}_{\mathbb{C}}(\equiv, a \times b) \cong \mathsf{Hom}_{\mathbb{C}\times\mathbb{C}}((\equiv,\equiv),(a,b))$$

Coq Formalization: P-Cartesian Structures (III)

P-Cartesian Exponentials

```
Definition IsPCartExp {C : PCartCat} (exp : C -> C -> C) :=
  forall a b,
    PNatIso
      (PBiFunPartialRight PHomFun (exp b a))
      (PCompFun
        PHomFun
        (PPairFun
          (POppFun (PBiFunPartialRight PCartProdFun a))
          (PCompFun (PConstFun b) PTermFun))
      ).
```

$$\mathsf{IsExponential}(- \Rightarrow =) \triangleq \prod_{a,b\,:\,\mathbb{C}} \mathsf{Hom}_{\mathbb{C}}(\equiv, a \Rightarrow b) \cong \mathsf{Hom}_{\mathbb{C}\times\mathbb{C}}(\equiv \times a, b)$$

P-Set–Valued P-Ends

P-Ends

For $F : \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathsf{PSet}$ we have:

- $\left| \int_{c:\mathbb{C}} F(c,c) \right| \triangleq \prod_{c:\mathbb{C}} F(c,c)$
- $w \sim w' \triangleq$
    - $\prod_{x,y:\mathbb{C}} \prod_{f,f':x \to y} f \sim f' \Rightarrow F\,(f, \mathsf{id})\,(w\,y) \sim F\,(\mathsf{id}, f')\,(w\,x) \quad \wedge$
    - $\prod_{x,y:\mathbb{C}} \prod_{f,f':x \to y} f \sim f' \Rightarrow F\,(f, \mathsf{id})\,(w'\,y) \sim F\,(\mathsf{id}, f')\,(w'\,x) \quad \wedge$
    - $\prod_{z:\mathbb{C}} w\,z \sim w'\,z$

P-Set–Valued P-Coends

P-Coends

For $F : \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathsf{PSet}$ we have:

- $\left| \int^{c:\mathbb{C}} F(c, c) \right| \triangleq \sum_{c:\mathbb{C}} F(c, c)$
- $w \sim w'$ is inductively generated by the following:
  - $\prod_{z:\mathbb{C}} \prod_{s,s':F\,(z,z)} s \sim s' \Rightarrow (z; s) \sim (z; s')$
  - $\prod_{x,y:\mathbb{C}} \prod_{f,f':y \to x} \prod_{s,s':F\,(x,y)} f \sim f' \Rightarrow s \sim s' \Rightarrow (y; F\,(f, \mathsf{id})\, s) \sim (x; F\,(\mathsf{id}, f')\, s')$
  - $\prod_{x,y:\mathbb{C}} \prod_{f,f':y \to x} \prod_{s,s':F\,(x,y)} f \sim f' \Rightarrow s \sim s' \Rightarrow (x; F\,(\mathsf{id}, f)\, s) \sim (y; F\,(f', \mathsf{id})\, s')$
  - $w_1 \sim w_2 \wedge w_2 \sim w_3 \Rightarrow w_1 \sim w_3$

P-Set–Valued P-(Co)Ends

Properties

- Density Formula for coends.
- Fubini rule for ends.
- Functor Category homs as ends.
- Cocontinuity and Continuity of the Hom-functor.
- Isomorphism under duality of $\mathbb{C}$.

Presheaf Exponential

$$\widehat{\mathbb{C}}\,(K, G^F) \cong \int_c \mathbf{Set}\,(Kc, G^F c) \equiv \int_c \mathbf{Set}\left(Kc, \int_{c'} \mathbb{C}(c', c) \Rightarrow Fc' \Rightarrow Gc'\right)$$
$$\cong \int_c \int_{c'} \mathbf{Set}\Big(Kc, \mathbb{C}(c', c) \Rightarrow Fc' \Rightarrow Gc'\Big)$$
$$\cong \int_{c'} \int_{c} \mathbf{Set}\Big(Kc, \mathbb{C}(c', c) \Rightarrow Fc' \Rightarrow Gc'\Big)$$
$$\cong \int_{c'} \int_{c} \mathbf{Set}\big(Kc \times \mathbb{C}(c', c), Fc' \Rightarrow Gc'\big)$$
$$\cong \int_{c'} \mathbf{Set}\left(\int^{c} Kc \times \mathbb{C}(c', c), Fc' \Rightarrow Gc'\right)$$
$$\cong \int_{c'} \mathbf{Set}(Kc', Fc' \Rightarrow Gc')$$
$$\cong \int_{c'} \mathbf{Set}(Kc' \times Fc', Gc') \cong \widehat{\mathbb{C}}(K \times F, G)$$

Future Work

- Complete formalization of gluing construction
- Move to P-bicategory theory for two-dimensional simple type theory
- Find connections with other categorical/mathematical systems
- Monoidal setting with Day convolution

Any Questions?