Russian Constructivism in a Prefascist Theory

Pierre-Marie Pédrot

Gallinette, INRIA

LICS'20

It's Time to CIC Ass

CIC, the Calculus of Inductive Constructions.

It's Time to CIC Ass

CIC, the Calculus of Inductive Constructions.

CIC, a very fancy **intuitionistic** **logical system**.

- Not just higher-order logic, not just first-order logic
- First class notion of computation and crazy inductive types

It's Time to CIC Ass

CIC, the Calculus of Inductive Constructions.

CIC, a very fancy **intuitionistic** **logical system**.

- Not just higher-order logic, not just first-order logic
- First class notion of computation and crazy inductive types

CIC, a very powerful **functional** **programming language**.

- Finest types to describe your programs
- No clear phase separation between runtime and compile time

It's Time to CIC Ass

CIC, the Calculus of Inductive Constructions.

CIC, a very fancy **intuitionistic** **logical system**.

- Not just higher-order logic, not just first-order logic
- First class notion of computation and crazy inductive types

CIC, a very powerful **functional** **programming language**.

- Finest types to describe your programs
- No clear phase separation between runtime and compile time

The Pinnacle of the Curry-Howard correspondence

It's Time to CIC Ass

CIC, the Calculus of Inductive Constructions.

CIC, a very fancy **intuitionistic** **logical system**.

- Not just higher-order logic, not just first-order logic
- First class notion of computation and crazy inductive types

CIC, a very powerful **functional** **programming language**.

- Finest types to describe your programs
- No clear phase separation between runtime and compile time

The Pinnacle of the Curry-Howard correspondence

Extending Coq

Our mission: to boldly extend CIC with new principles

Extending Coq

Our mission: to boldly extend CIC with new principles

$\rightsquigarrow$ we need to design models for that.

$\rightsquigarrow$ and ensure they satisfy *the good properties*.

- Consistency
- Canonicity
- Decidable type-checking
- Strong normalization

Extending Coq

Our mission: to boldly extend CIC with new principles

$\rightsquigarrow$ we need to design models for that.

$\rightsquigarrow$ and ensure they satisfy *the good properties*.

- Consistency
- Canonicity
- Decidable type-checking
- Strong normalization

Today we will focus on a specific family of models...

PRESHEAVES!

- Bread and Butter of Model Construction
- Proof-relevant Kripke semantics a.k.a. Intuitionistic Forcing

All Your Base Category Are Belong to Us

Definition

Let $\mathbb{P}$ be a category. A presheaf over $\mathbb{P}$ is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$.

(In what follows we will fix the base category $\mathbb{P}$ once and for all.)

All Your Base Category Are Belong to Us

Definition

Let $\mathbb{P}$ be a category. A presheaf over $\mathbb{P}$ is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$.

(In what follows we will fix the base category $\mathbb{P}$ once and for all.)

Presheaves with nat. transformations as morphisms form a category $\mathsf{Psh}(\mathbb{P})$.

Objects: A presheaf $(\mathbf{A}, \theta_{\mathbf{A}})$ is given by

- A family of $\mathbb{P}$-indexed sets $\mathbf{A}_p : \mathbf{Set}$
- Restriction morphisms $\theta_{\mathbf{A}} : \prod_{p,q} (\alpha \in \mathbb{P}(q,p)).\ \mathbf{A}_p \to \mathbf{A}_q$ (+ functoriality)

Morphisms: A morphism from $(\mathbf{A}, \theta_{\mathbf{A}})$ to $(\mathbf{B}, \theta_{\mathbf{B}})$ is given by

- A family of $\mathbb{P}$-indexed functions $f_p : \mathbf{A}_p \to \mathbf{B}_p$ which is natural in p

All Your Base Category Are Belong to Us

Definition

Let $\mathbb{P}$ be a category. A presheaf over $\mathbb{P}$ is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$.

(In what follows we will fix the base category $\mathbb{P}$ once and for all.)

Presheaves with nat. transformations as morphisms form a category $\mathsf{Psh}(\mathbb{P})$.

Objects: A presheaf $(\mathbf{A}, \theta_{\mathbf{A}})$ is given by

- A family of $\mathbb{P}$-indexed sets $\mathbf{A}_p : \mathbf{Set}$
- Restriction morphisms $\theta_{\mathbf{A}} : \prod_{p,q} (\alpha \in \mathbb{P}(q,p)).\ \mathbf{A}_p \to \mathbf{A}_q$ (+ functoriality)

Morphisms: A morphism from $(\mathbf{A}, \theta_{\mathbf{A}})$ to $(\mathbf{B}, \theta_{\mathbf{B}})$ is given by

- A family of $\mathbb{P}$-indexed functions $f_p : \mathbf{A}_p \to \mathbf{B}_p$ which is natural in p

Theorem

$\mathsf{Psh}(\mathbb{P})$ is a model of CIC.

Cantor's Hell

Let's have a look at the good properties we long for.

Let's have a look at the good properties we long for.

Consistency There is no proof of `False`. ☺

Cantor's Hell

Let's have a look at the good properties we long for.

Consistency There is no proof of `False`. ☺

Canonicity Closed integers are integers... are they?

$$\vdash M : \mathbb{N} \quad \text{“(C)ZF-implies”} \quad M \equiv \mathtt{S} \ldots \mathtt{S}\,\mathtt{0}$$ 😕

Cantor's Hell

Let's have a look at the good properties we long for.

Consistency There is no proof of `False`. ☺

Canonicity Closed integers are integers... are they?

$$\vdash M : \mathbb{N} \quad \text{“(C)ZF-implies”} \quad M \equiv \mathtt{S} \ldots \mathtt{S}\,\mathtt{0}$$

Implementability Type-checking is **not** decidable. ☹

Cantor’s Hell

Let’s have a look at the good properties we long for.

Consistency There is no proof of `False`.

Canonicity Closed integers are integers... are they?

$$\vdash M : \mathbb{N} \quad \text{“(C)ZF-implies”} \quad M \equiv \texttt{S} \ldots \texttt{S}\,\texttt{0}$$

Implementability Type-checking is **not** decidable.

Reduction Never heard of that. What’s syntax already?

Cantor's Hell

Let's have a look at the good properties we long for.

Consistency There is no proof of `False`.

Canonicity Closed integers are integers... are they?

$$\vdash M : \mathbb{N} \quad \text{"(C)ZF-implies"} \quad M \equiv \mathtt{S} \ldots \mathtt{S}\,\mathtt{0}$$

Implementability Type-checking is **not** decidable.

Reduction Never heard of that. What's syntax already?

Phenomenological Law

Set-theoretical models suck.

Down With Semantics

Instead

Syntactic Models

compilation

$\vdash_{\mathcal{S}} M : A$ implies $\vdash_{\mathcal{T}} [M] : [\![A]\!]$

Down With Semantics

Instead

Syntactic Models

compilation

$\vdash_{\mathcal{S}} M : A$ implies $\vdash_{\mathcal{T}} [M] : [\![A]\!]$

- Does not require non-type-theoretical foundations (*monism*)
- Can be implemented in Coq (*software monism*)
- Automatically inherit the good properties from CIC

Persevere Diabolicum

Is it possible to see presheaves as a syntactic model?

Is it possible to see presheaves as a syntactic model?

2012	2016	2020

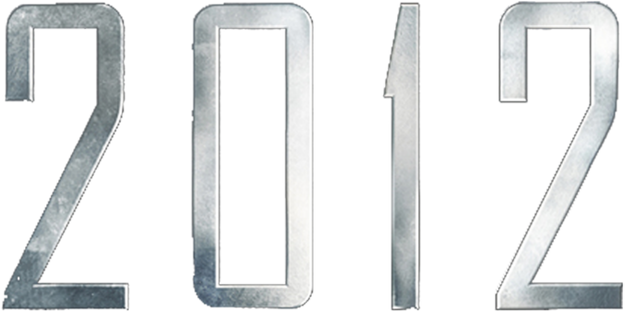
Extending Type Theory with Forcing (LICS, Jaber, Tabareau, Sozeau)	The Definitional Side of the Forcing (LICS, Jaber, Lewertowski, Pédrot, Tabareau, Sozeau)	Russian Constructivism in a Prefascist Theory (LICS, Pédrot)
FAIL	**FAIL**	**YAY?**

Persevere Diabolicum

Is it possible to see presheaves as a syntactic model?

2012	2016	2020
Extending Type Theory with Forcing (LICS, Jaber, Tabareau, Sozeau)	The Definitional Side of the Forcing (LICS, Jaber, Lewertowski, Pédrot, Tabareau, Sozeau)	Russian Constructivism in a Prefascist Theory (LICS, Pédrot)
FAIL	**FAIL**	**YAY?**

It is the journey, not the destination



(We were warned.)

Syntactic Presheaves, 2012 Edition

"A presheaf is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$."

Easy peasy: just replace $\mathbf{Set}$ everywhere with CIC.

Syntactic Presheaves, 2012 Edition

"A presheaf is just a functor $\mathbb{P}^{\mathsf{op}} \rightarrow \mathbf{Set}$."

Easy peasy: just replace $\mathbf{Set}$ everywhere with CIC.

$$\texttt{Cat} : \square \quad := \quad \left\{ \begin{array}{l} \mathbb{P} : \square \\ \leq : \mathbb{P} \rightarrow \mathbb{P} \rightarrow \square \\ \texttt{id} : \Pi p.\, p \leq p \\ \circ : \Pi p\, q\, r.\, p \leq q \rightarrow q \leq r \rightarrow p \leq r \\ \texttt{eqn} : \ldots ; \end{array} \right\}$$

$$\texttt{Psh} : \square \quad := \quad \left\{ \begin{array}{l} \mathbf{A} : \mathbb{P} \rightarrow \square \\ \theta_{\mathbf{A}} : \Pi (p\, q : \mathbb{P})\, (\alpha : q \leq p).\, \mathbf{A}_p \rightarrow \mathbf{A}_q \\ \texttt{eqn} : \ldots ; \end{array} \right\}$$

Syntactic Presheaves, 2012 Edition

"A presheaf is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$."

Easy peasy: just replace $\mathbf{Set}$ everywhere with CIC.

$$\mathtt{Cat} : \square \quad := \quad \left\{ \begin{array}{l} \mathbb{P} : \square \\ \leq : \mathbb{P} \to \mathbb{P} \to \square \\ \mathtt{id} : \Pi p.\, p \leq p \\ \circ : \Pi p\, q\, r.\, p \leq q \to q \leq r \to p \leq r \\ \mathtt{eqn} : \ldots ; \end{array} \right\}$$

$$\mathtt{Psh} : \square \quad := \quad \left\{ \begin{array}{l} \mathbf{A} : \mathbb{P} \to \square \\ \theta_{\mathbf{A}} : \Pi (p\, q : \mathbb{P})\, (\alpha : q \leq p).\, \mathbf{A}_p \to \mathbf{A}_q \\ \mathtt{eqn} : \ldots ; \end{array} \right\}$$

This almost works...

Syntactic Presheaves, 2012 Edition

"A presheaf is just a functor $\mathbb{P}^{\mathsf{op}} \to \mathbf{Set}$."

Easy peasy: just replace **Set** everywhere with CIC.

$$\mathtt{Cat} : \square \quad := \quad \left\{ \begin{array}{l} \mathbb{P} : \square \\ \leq : \mathbb{P} \to \mathbb{P} \to \square \\ \mathtt{id} : \Pi p.\, p \leq p \\ \circ : \Pi p\, q\, r.\, p \leq q \to q \leq r \to p \leq r \\ \mathtt{eqn} : \ldots ; \end{array} \right\}$$

$$\mathtt{Psh} : \square \quad := \quad \left\{ \begin{array}{l} \mathbf{A} : \mathbb{P} \to \square \\ \theta_{\mathbf{A}} : \Pi (p\, q : \mathbb{P})\, (\alpha : q \leq p).\, \mathbf{A}_p \to \mathbf{A}_q \\ \mathtt{eqn} : \ldots ; \end{array} \right\}$$

This almost works... except that equations are propositional !!!

$$\vdash_{\mathsf{CIC}} M \equiv N \quad \not\longrightarrow \quad \vdash [M] \equiv [N]$$
$$\vdash_{\mathsf{CIC}} M \equiv N \quad \longrightarrow \quad \vdash e : [M] = [N]$$

You need to introduce rewriting everywhere

Equality is Too Serious a Matter

"The Coherence Hell": the target theory must be **EXTENSIONAL**

$$\frac{\Gamma \vdash e : M = N}{\Gamma \vdash M \equiv N}$$

Equality is Too Serious a Matter

"The Coherence Hell": the target theory must be **EXTENSIONAL**

$$\frac{\Gamma \vdash e : M = N}{\Gamma \vdash M \equiv N}$$

- Arguably better than ZFC ("constructive")
- ... but undecidable type-checking
- ... computation destroyed, e.g. β-reduction is undecidable
- See Théo Winterhalter's soon to be defended PhD for more horrors

Equality is Too Serious a Matter

"The Coherence Hell": the target theory must be **EXTENSIONAL**

$$\frac{\Gamma \vdash e : M = N}{\Gamma \vdash M \equiv N}$$

- Arguably better than ZFC ("constructive")
- … but undecidable type-checking
- … computation destroyed, e.g. β-reduction is undecidable
- See Théo Winterhalter's soon to be defended PhD for more horrors

Bold Claim

ETT is not really a type theory, so we don't have a syntactic model.

2016

(Make conversion great again, and break everything else.)

Squaring the Circle

Key Observation 1

Presheaves factorize in CBPV through a *call-by-value* decomposition

They only satisfy definitionally the CBV equational theory generated by

$$(\lambda x.\, t)\; V \equiv_{\beta v} t\{x := V\}$$

Squaring the Circle

Key Observation 1

Presheaves factorize in CBPV through a *call-by-value* decomposition

They only satisfy definitionally the CBV equational theory generated by

$$(\lambda x.\, t)\; V \equiv_{\beta v} t\{x := V\}$$

Key Observation 2

Type theory is *call-by-name*!

$$\frac{\Gamma \vdash M : B \qquad \Gamma \vdash A \equiv_{\beta} B}{\Gamma \vdash M : A}\;(\text{Conv})$$

Squaring the Circle

Key Observation 1

Presheaves factorize in CBPV through a *call-by-value* decomposition

They only satisfy definitionally the CBV equational theory generated by

$$(\lambda x.\, t)\ V \equiv_{\beta v} t\{x := V\}$$

Key Observation 2

Type theory is *call-by-name*!

$$\frac{\Gamma \vdash M : B \qquad \Gamma \vdash A \equiv_{\beta} B}{\Gamma \vdash M : A}\ \text{(Conv)}$$

Someone Had To Say It

CBV and CBN are not the same

If There is No Solution, There is No Problem

Easy solution! Pick the **call-by-name** decomposition instead.

$$\text{CBV} \quad [\![A \to B]\!]_p \quad := \quad \Pi(q \le p).\, ([\![A]\!]_q \ \to \ [\![B]\!]_q)$$

$$\text{CBN} \quad [\![A \to B]\!]_p \quad := \quad (\Pi(q \le p).\, [\![A]\!]_q) \ \to \ [\![B]\!]_p$$

If There is No Solution, There is No Problem

Easy solution! Pick the **call-by-name** decomposition instead.

$$\text{CBV} \quad [\![A \to B]\!]_p \quad := \quad \Pi(q \leq p).\,([\![A]\!]_q \;\to\; [\![B]\!]_q)$$

$$\text{CBN} \quad [\![A \to B]\!]_p \quad := \quad (\Pi(q \leq p).\,[\![A]\!]_q) \;\to\; [\![B]\!]_p$$

- In CBN, types are not interpreted as functors in general
- Functoriality given freely by **thunking** over all lower conditions
- This adapts straightforwardly to the dependently-typed setting.

If There is No Solution, There is No Problem

Easy solution! Pick the **call-by-name** decomposition instead.

$$\text{CBV} \quad [\![A \to B]\!]_p \quad := \quad \Pi(q \leq p).\, ([\![A]\!]_q \;\to\; [\![B]\!]_q)$$

$$\text{CBN} \quad [\![A \to B]\!]_p \quad := \quad (\Pi(q \leq p).\, [\![A]\!]_q) \;\to\; [\![B]\!]_p$$

- In CBN, types are not interpreted as functors in general
- Functoriality given freely by **thunking** over all lower conditions
- This adapts straightforwardly to the dependently-typed setting.

Theorem (Jaber & al. 2016)

There is a syntactic CBN presheaf model of CC^ω *into* CIC.

where CC^ω is CIC without inductive types.

If There is No Solution, There is No Problem

Easy solution! Pick the **call-by-name** decomposition instead.

$$\text{CBV} \quad [\![A \to B]\!]_p \quad := \quad \Pi(q \leq p).\, ([\![A]\!]_q \;\to\; [\![B]\!]_q)$$

$$\text{CBN} \quad [\![A \to B]\!]_p \quad := \quad (\Pi(q \leq p).\, [\![A]\!]_q) \;\to\; [\![B]\!]_p$$

- In CBN, types are not interpreted as functors in general
- Functoriality given freely by **thunking** over all lower conditions
- This adapts straightforwardly to the dependently-typed setting.

Theorem (Jaber & al. 2016)

There is a syntactic CBN presheaf model of CC^ω *into* CIC.

where CC^ω is CIC without inductive types.

... but the model disproves dependent elimination!

We still don't have a syntactic presheaf model.

Interlude

Interlude

Puzzle

Why does $\mathsf{Psh}(\mathbb{P})$ interpret full β-conversion (although **only extensionally**)?

Interlude

Puzzle

Why does $\mathsf{Psh}(\mathbb{P})$ interpret full β-conversion (although **only extensionally**)?

Answer

This is because of the **naturality** requirement on functions.

Interlude

Puzzle

Why does $\text{Psh}(\mathbb{P})$ interpret full β-conversion (although **only extensionally**)?

Answer

This is because of the **naturality** requirement on functions.

Theorem (Pédrot-Tabareau '20)

Naturality in CBV presheaves corresponds to Führmann's ***thunkability****.*

- This is a well-known **systematic** construction from realizability
- $\text{Psh}(\mathbb{P})$ is the **pure fragment** of an effectful CBV language
- In CBV, effects break functions, in CBN they break inductive types
- We were missing the equivalent in the CBN presheaves!

Interlude

Puzzle

Why does $\mathrm{Psh}(\mathbb{P})$ interpret full β-conversion (although **only extensionally**)?

Answer

This is because of the **naturality** requirement on functions.

Theorem (Pédrot-Tabareau '20)

*Naturality in CBV presheaves corresponds to Führmann's **thunkability**.*

- This is a well-known **systematic** construction from realizability
- $\mathrm{Psh}(\mathbb{P})$ is the **pure fragment** of an effectful CBV language
- In CBV, effects break functions, in CBN they break inductive types
- We were missing the equivalent in the CBN presheaves!

Theorem (Bernardy-Lasson '11)

*The CBN equivalent is **parametricity**. It is a syntactic model!*

On Parametric Presheaves

What does parametricity look like on the CBN presheaf model?

$$x : \mathbb{B} \quad \longrightarrow \quad \left\{ \begin{array}{l} x : \Pi(q \leq p).\, \mathbb{B} \\ x_\varepsilon : \mathbb{B}_\varepsilon\ p\ x \end{array} \right.$$

On Parametric Presheaves

What does parametricity look like on the CBN presheaf model?

$$x : \mathbb{B} \quad \longrightarrow \quad \begin{cases} x : \Pi(q \leq p).\, \mathbb{B} \\ x_\varepsilon : \mathbb{B}_\varepsilon \; p \; x \end{cases}$$

We have a bit of constraints. To get dependent elimination we need:

1. $\mathbb{B}_\varepsilon \; p \; x$ iff $(x = \lambda q\, \alpha.\, \mathtt{tt})$ or $(x = \lambda q\, \alpha.\, \mathtt{ff})$
2. in a **unique** way, i.e. $b_1, b_2 : \mathbb{B}_\varepsilon \; p \; x \vdash b_1 = b_2$ (i.e. a HoTT proposition)

On Parametric Presheaves

What does parametricity look like on the CBN presheaf model?

$$x : \mathbb{B} \quad \longrightarrow \quad \begin{cases} x : \Pi(q \leq p).\, \mathbb{B} \\ x_\varepsilon : \mathbb{B}_\varepsilon\ p\ x \end{cases}$$

We have a bit of constraints. To get dependent elimination we need:

1. $\mathbb{B}_\varepsilon\ p\ x$ iff $(x = \lambda q\, \alpha.\, \mathtt{tt})$ or $(x = \lambda q\, \alpha.\, \mathtt{ff})$
2. in a **unique** way, i.e. $b_1, b_2 : \mathbb{B}_\varepsilon\ p\ x \vdash b_1 = b_2$ (i.e. a HoTT proposition)

But we also critically need to be compatible with the presheaf structure!

3. That is, $\theta_{\mathbb{B}_\varepsilon}\ (\alpha : q \leq p) : \mathbb{B}_\varepsilon\ p\ x \to \mathbb{B}_\varepsilon\ q\ (\alpha \cdot x)$
4. with further **definitional** functoriality to avoid coherence issues

On Parametric Presheaves

What does parametricity look like on the CBN presheaf model?

$$x : \mathbb{B} \quad \longrightarrow \quad \begin{cases} x : \Pi(q \leq p).\, \mathbb{B} \\ x_\varepsilon : \mathbb{B}_\varepsilon\ p\ x \end{cases}$$

We have a bit of constraints. To get dependent elimination we need:

1. $\mathbb{B}_\varepsilon\ p\ x$ iff $(x = \lambda q\, \alpha.\, \mathtt{tt})$ or $(x = \lambda q\, \alpha.\, \mathtt{ff})$
2. in a **unique** way, i.e. $b_1, b_2 : \mathbb{B}_\varepsilon\ p\ x \vdash b_1 = b_2$ (i.e. a HoTT proposition)

But we also critically need to be compatible with the presheaf structure!

3. That is, $\theta_{\mathbb{B}_\varepsilon}\ (\alpha : q \leq p) : \mathbb{B}_\varepsilon\ p\ x \rightarrow \mathbb{B}_\varepsilon\ q\ (\alpha \cdot x)$
4. with further **definitional** functoriality to avoid coherence issues

You cannot have both at the same time in CIC

😱 This is exactly the CBV vs. CBN conundrum **one level higher** 😱

(On the virtues of Authoritarianism.)

It is a Revolution

Essentially, we were blocked on this issue since then. When suddenly...

It is a Revolution

Essentially, we were blocked on this issue since then. When suddenly...

Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau.
Definitional proof-irrelevance without K.
Proc. ACM Program. Lang., 3(POPL):3:1–3:28, 2019.

It is a Revolution

Essentially, we were blocked on this issue since then. When suddenly...

Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau.
Definitional proof-irrelevance without K.
Proc. ACM Program. Lang., 3(POPL):3:1–3:28, 2019.

They introduce a new sort SProp of **strict propositions**.

$$M, N : A : \texttt{SProp} \quad \longrightarrow \quad \vdash M \equiv N$$

- A well-behaved subset of Prop compatible with HoTT
- It enjoys all good syntactic properties

It is a Revolution

Essentially, we were blocked on this issue since then. When suddenly...

Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau.
Definitional proof-irrelevance without K.
Proc. ACM Program. Lang., 3(POPL):3:1–3:28, 2019.

They introduce a new sort `SProp` of **strict propositions**.

$$M, N : A : \texttt{SProp} \quad \longrightarrow \quad \vdash M \equiv N$$

- A well-behaved subset of `Prop` compatible with HoTT
- It enjoys all good syntactic properties

$\rightsquigarrow$ `SProp` is closed under products.

$$\vdash A : \square, \qquad x : A \vdash B : \texttt{SProp} \qquad \longrightarrow \qquad \vdash \Pi(x : A).\, B : \texttt{SProp}$$

$\rightsquigarrow$ Only `False` is eliminable from `SProp` into `Type`.

A Strict Doctrine

Possible Extension

$\mathfrak{s}$CIC additionally allows the elimination of `eq` from `SProp` to `Type`

This gives rise to a **strict equality**, i.e. $\mathfrak{s}$CIC has definitional UIP.

A Strict Doctrine

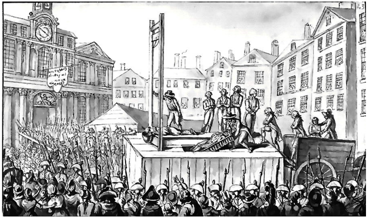
Possible Extension

$\mathfrak{s}$CIC additionally allows the elimination of `eq` from `SProp` to `Type`

This gives rise to a **strict equality**, i.e. $\mathfrak{s}$CIC has definitional UIP.

When the libertarian HoTT freely adds infinite towers of equalities...

... the authoritarian $\mathfrak{s}$CIC will instead **guillotine** all higher equalities.



Art. 1. *All humans are born uniquely equal in rights.*

Strict Parametricity

In the parametric presheaf translation

Strict equality is the authoritarian way to solve the coherence hell.

- make the parametricity predicate **free** $\rightsquigarrow$ **definitional functoriality**
- require it to be a **strict** proposition $\rightsquigarrow$ **proof uniqueness**

$$x : A \quad \longrightarrow \quad \begin{cases} x : \Pi(q \leq p).\, [\![A]\!]_q \\ x_\varepsilon : \Pi(q \leq p).\, [\![A]\!]_\varepsilon\ q\ (\alpha \cdot x) \end{cases}$$

where critically $[\![A]\!]_\varepsilon\ p\ x : \texttt{SProp}$.

Strict Parametricity

In the parametric presheaf translation

Strict equality is the authoritarian way to solve the coherence hell.

- make the parametricity predicate **free** $\leadsto$ **definitional functoriality**
- require it to be a **strict** proposition $\leadsto$ **proof uniqueness**

$$x : A \quad \longrightarrow \quad \begin{cases} x : \Pi(q \leq p).\, [\![A]\!]_q \\ x_\varepsilon : \Pi(q \leq p).\, [\![A]\!]_\varepsilon\ q\ (\alpha \cdot x) \end{cases}$$

where critically $[\![A]\!]_\varepsilon\ p\ x : \texttt{SProp}$.

We call the result the **prefascist translation**. (lat. *fascis* : sheaf)

Strict Parametricity

In the parametric presheaf translation

Strict equality is the authoritarian way to solve the coherence hell.

- make the parametricity predicate **free** $\leadsto$ **definitional functoriality**
- require it to be a **strict** proposition $\leadsto$ **proof uniqueness**

$$x : A \quad \longrightarrow \quad \left\{ \begin{array}{l} x : \Pi(q \leq p).\, [\![A]\!]_q \\ x_\varepsilon : \Pi(q \leq p).\, [\![A]\!]_\varepsilon\ q\ (\alpha \cdot x) \end{array} \right.$$

where critically $[\![A]\!]_\varepsilon\ p\ x : \texttt{SProp}$.

We call the result the **prefascist translation**. (lat. *fascis* : sheaf)

Theorem

The prefascist translation is a syntactic model of CIC *into* $\mathfrak{s}$CIC.

- Full conversion, full dependent elimination.
- The actual construction is a tad involved, but boils down to the above.
- Unsurprinsingly, UIP is required to interpret universes (tricky!).

No Pain, No Gain

$\mathfrak{s}$CIC is way weaker than ETT

$\mathfrak{s}$CIC is **conjectured** to enjoy the usual good syntactic properties.

- Canonicity seems relatively easy to show
- UIP makes reduction depend on conversion though
- SN is problematic, e.g. $\mathfrak{s}$CIC + an impredicative universe is **not** SN
- Hoping that SN holds in the predicative case, decidability follows

No Pain, No Gain

$\mathfrak{s}$CIC is way weaker than ETT

$\mathfrak{s}$CIC is **conjectured** to enjoy the usual good syntactic properties.

- Canonicity seems relatively easy to show
- UIP makes reduction depend on conversion though
- SN is problematic, e.g. $\mathfrak{s}$CIC + an impredicative universe is **not** SN
- Hoping that SN holds in the predicative case, decidability follows

We don't rely on impredicativity in the prefascist model

We would inherit the purported good properties $\mathfrak{s}$CIC for free.

Back to Set

$\mathbf{Set}$ is a model of $\mathfrak{s}$CIC

Thus, the prefascist model can also be described set-theoretically.

Back to Set

$\mathbf{Set}$ is a model of $\mathfrak{s}$CIC

Thus, the prefascist model can also be described set-theoretically.

Theorem

*Prefascist sets over $\mathbb{P}$ form a category $\mathbf{Pfs}(\mathbb{P})$ with **definitional** laws.*

$\rightsquigarrow$ they have a distinct realizability flavour compared to presheaves

Back to Set

$\mathbf{Set}$ is a model of $\mathfrak{s}$CIC

Thus, the prefascist model can also be described set-theoretically.

Theorem

*Prefascist sets over $\mathbb{P}$ form a category $\mathbf{Pfs}(\mathbb{P})$ with **definitional** laws.*

$\rightsquigarrow$ they have a distinct realizability flavour compared to presheaves

Theorem

As categories, $\mathbf{Psh}(\mathbb{P})$ and $\mathbf{Pfs}(\mathbb{P})$ are equivalent.

Back to Set

$\mathbf{Set}$ is a model of $\mathfrak{s}$CIC

Thus, the prefascist model can also be described set-theoretically.

Theorem

Prefascist sets over $\mathbb{P}$ form a category $\mathbf{Pfs}(\mathbb{P})$ with ***definitional*** *laws.*

$\rightsquigarrow$ they have a distinct realizability flavour compared to presheaves

Theorem

As categories, $\mathbf{Psh}(\mathbb{P})$ and $\mathbf{Pfs}(\mathbb{P})$ are equivalent.

- Proving this requires extensionality principles
- Yet, $\mathbf{Pfs}(\mathbb{P})$ is better behaved in an intensional setting
- This could come in handy for higher category theory...

Back to Set

$\mathbf{Set}$ is a model of $\mathfrak{s}$CIC

Thus, the prefascist model can also be described set-theoretically.

Theorem

*Prefascist sets over $\mathbb{P}$ form a category $\mathbf{Pfs}(\mathbb{P})$ with **definitional** laws.*

$\leadsto$ they have a distinct realizability flavour compared to presheaves

Theorem

As categories, $\mathbf{Psh}(\mathbb{P})$ and $\mathbf{Pfs}(\mathbb{P})$ are equivalent.

- Proving this requires extensionality principles
- Yet, $\mathbf{Pfs}(\mathbb{P})$ is better behaved in an intensional setting
- This could come in handy for higher category theory...

Takeaway: prefascist sets are a better presentation of presheaves

APPLICATION

RUSSIAN CONSTRUCTIVISM

Russian Constructivist School

A splinter group of constructivists, whose core tenet can be summarized as:

Proofs are Kleene realizers

Russian Constructivist School

A splinter group of constructivists, whose core tenet can be summarized as:

Proofs are Kleene realizers

Thus, the principle that puts it apart both from Brouwer **and** Bishop:

Markov's Principle (MP)

$$\forall (f : \mathbb{N} \to \mathbb{B}).\, \neg\neg(\exists n : \mathbb{N}.\, f\ n = \texttt{tt}) \to \exists n : \mathbb{N}.\, f\ n = \texttt{tt}$$

- Semi-classical: $\mathbf{HA}^\omega \subsetneq \mathbf{HA}^\omega + \mathsf{MP} \subsetneq \mathbf{PA}^\omega$
- Known to preserve existence property (i.e. canonicity)
- Often required to prove various completeness results

Russian Constructivist School

A splinter group of constructivists, whose core tenet can be summarized as:

Proofs are Kleene realizers

Thus, the principle that puts it apart both from Brouwer **and** Bishop:

Markov's Principle (MP)

$$\forall (f : \mathbb{N} \rightarrow \mathbb{B}).\, \neg\neg(\exists n : \mathbb{N}.\, f\; n = \mathtt{tt}) \rightarrow \exists n : \mathbb{N}.\, f\; n = \mathtt{tt}$$

- Semi-classical: $\mathbf{HA}^\omega \subsetneq \mathbf{HA}^\omega + \mathsf{MP} \subsetneq \mathbf{PA}^\omega$
- Known to preserve existence property (i.e. canonicity)
- Often required to prove various completeness results

What if we tried to extend CIC with MP through a syntactic model?

MP in Kleene Realizability

Let's look at the realizer

$$\forall (f : \mathbb{N} \to \mathbb{B}).\, \neg\neg(\exists n : \mathbb{N}.\, f\; n = \texttt{tt}) \to \exists n : \mathbb{N}.\, f\; n = \texttt{tt}$$

```
let mp f _ :=
  let n := ref 0 in
  while true do
    if f !n then return n else n := n + 1
  done
```

MP in Kleene Realizability

Let's look at the realizer

$$\forall (f : \mathbb{N} \to \mathbb{B}). \neg\neg(\exists n : \mathbb{N}. f\, n = \texttt{tt}) \to \exists n : \mathbb{N}. f\, n = \texttt{tt}$$

```
let mp f _ :=
  let n := ref 0 in
  while true do
    if f !n then return n else n := n + 1
  done
```

Proving $\texttt{mp} \Vdash$ MP needs MP in the meta-theory!

- As such, this is **cheating**
- The realizer doesn't use the doubly-negated proof
- Relies on unbounded loops in realizers
- We have little hope to implement this in CIC with a syntactic model

Let's look at the realizer

$$\forall (f : \mathbb{N} \to \mathbb{B}).\, \neg\neg(\exists n : \mathbb{N}.\, f\, n = \mathtt{tt}) \to \exists n : \mathbb{N}.\, f\, n = \mathtt{tt}$$

```
let mp f _ :=
  let n := ref 0 in
  while true do
    if f !n then return n else n := n + 1
  done
```

Proving mp $\Vdash$ MP needs MP in the meta-theory!

- As such, this is **cheating**
- The realizer doesn't use the doubly-negated proof
- Relies on unbounded loops in realizers
- We have little hope to implement this in CIC with a syntactic model

We need something else...

What Else?

Not one, but at least **two** alternatives!

What Else?

Not one, but at least **two** alternatives!

- Coquand-Hofmann's syntactic model for $\mathbf{HA}^{\omega} + \mathrm{MP}$
- Herbelin's direct style proof using static exceptions

What Else?

Not one, but at least **two** alternatives!

- Coquand-Hofmann's syntactic model for $\mathbf{HA}^\omega + \mathrm{MP}$
- Herbelin's direct style proof using static exceptions

CH's model is a mix of Kripke semantics and Friedman's A-translation

- Kripke semantics $\leadsto$ global cell $p : \mathbb{N} \to \mathbb{B}$ where

$$q \leq p \quad := \quad \forall n : \mathbb{N}.\, p\ n = \mathtt{tt} \to q\ n = \mathtt{tt} \qquad (q \text{ truer than } p)$$

- A-translation $\leadsto$ exceptions of type $A_p := \exists n : \mathbb{N}.\, p\ n = \mathtt{tt}$

What Else?

Not one, but at least **two** alternatives!

- Coquand-Hofmann's syntactic model for $\mathbf{HA}^\omega + \mathrm{MP}$
- Herbelin's direct style proof using static exceptions

CH's model is a mix of Kripke semantics and Friedman's A-translation

- Kripke semantics $\leadsto$ global cell $p : \mathbb{N} \to \mathbb{B}$ where

$$q \leq p \quad := \quad \forall n : \mathbb{N}.\, p\ n = \mathtt{tt} \to q\ n = \mathtt{tt} \qquad (q \text{ truer than } p)$$

- A-translation $\leadsto$ exceptions of type $A_p := \exists n : \mathbb{N}.\, p\ n = \mathtt{tt}$

The secret sauce is that the exception type depends on the current p

Pipelining

Coquand-Hofmann's model is a bit ad-hoc

Pipelining

Coquand-Hofmann's model is a bit ad-hoc

Instead, we define the *Calculus of Constructions with Completeness Principles* as

$$\mathsf{CCCP} \quad (\supseteq \mathsf{CIC}) \quad \xrightarrow{\mathbf{Exn}} \quad \mathsf{CIC} + \mathcal{E} \quad \xrightarrow{\mathbf{Pfs}} \quad \mathfrak{s}\mathsf{CIC}$$

- $\mathbf{Pfs}$ is the prefascist model described before
- $\mathbf{Exn}$ is the exceptional model, a CIC-worthy A-translation

Theorem

If $\mathfrak{s}$CIC *enjoys the good properties then so does* CCCP.

Pipelining

Coquand-Hofmann's model is a bit ad-hoc

Instead, we define the *Calculus of Constructions with Completeness Principles* as

$$\mathsf{CCCP} \quad (\supseteq \mathsf{CIC}) \quad \xrightarrow{\mathbf{Exn}} \quad \mathsf{CIC} + \mathcal{E} \quad \xrightarrow{\mathbf{Pfs}} \quad \mathfrak{s}\mathsf{CIC}$$

- $\mathbf{Pfs}$ is the prefascist model described before
- $\mathbf{Exn}$ is the exceptional model, a CIC-worthy A-translation

Theorem

If $\mathfrak{s}$CIC *enjoys the good properties then so does* CCCP*.*

$\mathbf{Exn}$ is a very simple syntactic model of CIC

Pick a fixed type $\mathcal{E}$ of **exceptions** in the target theory.

$$\vdash_{\mathcal{S}} A : \square \qquad \longrightarrow \qquad \vdash_{\mathcal{T}} [\![A]\!]_{\mathcal{E}} : \square \quad + \quad \vdash_{\mathcal{T}} [A]_{\mathcal{E}}^{\varnothing} : \mathcal{E} \to [\![A]\!]_{\mathcal{E}}$$

In particular $\qquad [\![\neg A]\!]_{\mathcal{E}} \quad \cong \quad [\![A]\!]_{\mathcal{E}} \to \mathcal{E}$

Somebody Set Up Us The Bomb

We perform the exceptional translation over an **exotic** type of exceptions

$$\mathsf{CCCP} \quad \xrightarrow{\textbf{Exn}} \quad \mathsf{CIC} + \mathcal{E} \quad \xrightarrow{\textbf{Pfs}} \quad \mathfrak{s}\mathsf{CIC}$$

In the the prefascist model over $\mathbb{N} \rightarrow \mathbb{B}$, $\qquad \mathcal{E}_p := \Sigma n : \mathbb{N}.\, p\ n = \mathtt{tt}$

Somebody Set Up Us The Bomb

We perform the exceptional translation over an exotic type of exceptions

$$\text{CCCP} \xrightarrow{\textbf{Exn}} \text{CIC} + \mathcal{E} \xrightarrow{\textbf{Pfs}} \mathfrak{s}\text{CIC}$$

In the the prefascist model over $\mathbb{N} \to \mathbb{B}$, $\quad \mathcal{E}_p := \Sigma n : \mathbb{N}.\, p\ n = \mathtt{tt}$

We also have a modality in CIC $+ \mathcal{E}$

$$\begin{array}{lcl} \mathtt{local} & : & (\mathbb{N} \to \mathbb{B}) \to \square \to \square \\ [\mathtt{local}\ \varphi\ A]_p & :\stackrel{\sim}{=} & [A]_{p \wedge \varphi} \end{array}$$

- $\mathtt{return} : A \to \mathtt{local}\ \varphi\ A$
- `local` commutes to arrows and positive types
- $\mathtt{local}\ \varphi\ \mathcal{E} \quad \cong \quad \mathcal{E} + (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{tt})$

Somebody Set Up Us The Bomb

We perform the exceptional translation over an **exotic** type of exceptions

$$\text{CCCP} \xrightarrow{\textbf{Exn}} \text{CIC} + \mathcal{E} \xrightarrow{\textbf{Pfs}} \mathfrak{s}\text{CIC}$$

In the the prefascist model over $\mathbb{N} \to \mathbb{B}$, $\mathcal{E}_p := \Sigma n : \mathbb{N}.\, p\ n = \mathtt{tt}$

We also have a modality in CIC $+ \mathcal{E}$

$$\begin{array}{lcl} \mathtt{local} & : & (\mathbb{N} \to \mathbb{B}) \to \square \to \square \\ [\mathtt{local}\ \varphi\ A]_p & :\stackrel{\sim}{=} & [A]_{p \wedge \varphi} \end{array}$$

- $\mathtt{return} : A \to \mathtt{local}\ \varphi\ A$
- local commutes to arrows and positive types
- $\mathtt{local}\ \varphi\ \mathcal{E} \quad \cong \quad \mathcal{E} + (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{tt})$

Theorem

CCCP *validates* MP.

Proof by symbol pushing in CIC $+ \mathcal{E}$ by the above and $[\![\neg A]\!]_{\mathcal{E}} \cong [\![A]\!]_{\mathcal{E}} \to \mathcal{E}$.

A Computational Analysis of MP

Every time we go under `local` we get new exceptions!

$$\texttt{local}\ \varphi\ \mathcal{E} \quad \cong \quad \mathcal{E} + (\Sigma n : \mathbb{N}.\, \varphi\ n = \texttt{tt})$$

`return` is a delimited continuation prompt / static exception binder.

A Computational Analysis of MP

Every time we go under `local` we get new exceptions!

$$\texttt{local}\ \varphi\ \mathcal{E} \quad \cong \quad \mathcal{E} + (\Sigma n : \mathbb{N}.\, \varphi\ n = \texttt{tt})$$

`return` is a delimited continuation prompt / static exception binder.

The structure of the realizer thus follows closely Herbelin's proof.

$$\begin{array}{l} \texttt{mp}\ (p : \neg\neg(\exists n.\, f\ n = \texttt{tt})) := \\ \qquad \texttt{try}_\alpha\ \bot_e\ (p\ (\lambda k.\, k\ (\lambda n.\, \texttt{raise}_\alpha\ n)))\ \texttt{with}\ \alpha\ n \mapsto n \end{array}$$

In particular p can raise exceptions from outside, which is reflected here.

A Computational Analysis of MP

Every time we go under `local` we get new exceptions!

$$\texttt{local}\ \varphi\ \mathcal{E} \quad \cong \quad \mathcal{E} + (\Sigma n : \mathbb{N}.\ \varphi\ n = \texttt{tt})$$

`return` is a delimited continuation prompt / static exception binder.

The structure of the realizer thus follows closely Herbelin's proof.

$$\begin{array}{l} \texttt{mp}\ (p : \neg\neg(\exists n.\, f\ n = \texttt{tt})) := \\ \qquad \texttt{try}_\alpha\ \bot_e\ (p\ (\lambda k.\, k\ (\lambda n.\, \texttt{raise}_\alpha\ n)))\ \texttt{with}\ \alpha\ n \mapsto n \end{array}$$

In particular p can raise exceptions from outside, which is reflected here.

Thus, Herbelin's proof is the direct style variant of Coquand-Hofmann

Conclusion

On presheaves:

- Presheaves are a purified sublanguage of a monotonic reader effect
- We have given a better-behaved presentation of presheaves
- It is a syntactic model that relies on strict equality in the target
- Provides for free extensions of CIC with SN, canonicity and the like
- ... assuming $\mathfrak{s}$CIC enjoys this ($\dagger$)

Conclusion

On presheaves:

- Presheaves are a purified sublanguage of a monotonic reader effect
- We have given a better-behaved presentation of presheaves
- It is a syntactic model that relies on strict equality in the target
- Provides for free extensions of CIC with SN, canonicity and the like
- ... assuming $\mathfrak{s}$CIC enjoys this $(\dagger)$

On MP:

- Composition of the prefascist model with another model of ours
- This provides a computational extension of CIC that validates MP
- Once again, good properties for free

Conclusion

On presheaves:

- Presheaves are a purified sublanguage of a monotonic reader effect
- We have given a better-behaved presentation of presheaves
- It is a syntactic model that relies on strict equality in the target
- Provides for free extensions of CIC with SN, canonicity and the like
- ... assuming $\mathfrak{s}$CIC enjoys this $(\dagger)$

On MP:

- Composition of the prefascist model with another model of ours
- This provides a computational extension of CIC that validates MP
- Once again, good properties for free

TODO:

- Implement cubical type theory in this model

Thanks for your attention.