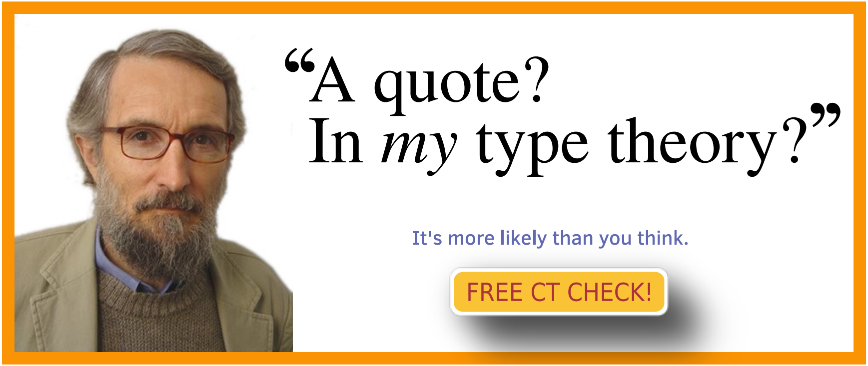

Pierre-Marie Pédrot

INRIA

2023/12/01

Today's Focus

Church's thesis!

All reasonable computational models are equivalent.

Today's Focus

All reasonable computational models are equivalent.

Today's Focus

~~[illegible]~~

All reasonable computational models are equivalent.

The internal Church thesis in a theory $\mathcal{T}$!

From within $\mathcal{T}$, "all functions $\mathbb{N} \to \mathbb{N}$ are computable".

Horrible Encodings Ahead

Let's fix a simple type theory $\mathcal{T}$ containing arithmetic.

$\leadsto$ One can define the (decidable) Turing predicate:

$$\frac{p : \mathbb{N} \qquad n : \mathbb{N} \qquad k : \mathbb{N}}{\mathsf{T}(p, n, k) : \mathsf{Prop}}$$

"$\mathsf{T}(p, n, k)$ holds iff the Turing machine p returns n in $\leq k$ steps."

Horrible Encodings Ahead

Let's fix a simple type theory $\mathcal{T}$ containing arithmetic.

$\rightsquigarrow$ One can define the (decidable) Turing predicate:

$$\frac{p : \mathbb{N} \qquad n : \mathbb{N} \qquad k : \mathbb{N}}{\mathsf{T}(p, n, k) : \mathsf{Prop}}$$

"$\mathsf{T}(p, n, k)$ holds iff the Turing machine p returns n in $\leq k$ steps."

[NB: for readability, I'll henceforth write $\mathbb{P} := \mathbb{N}$ to indicate numbers coding programs]

Horrible Encodings Ahead

Let's fix a simple type theory $\mathcal{T}$ containing arithmetic.

$\rightsquigarrow$ One can define the (decidable) Turing predicate:

$$\frac{p : \mathbb{N} \qquad n : \mathbb{N} \qquad k : \mathbb{N}}{\mathsf{T}(p, n, k) : \mathsf{Prop}}$$

"$\mathsf{T}(p, n, k)$ holds iff the Turing machine p returns n in $\leq k$ steps."

[NB: for readability, I'll henceforth write $\mathbb{P} := \mathbb{N}$ to indicate numbers coding programs]

$\rightsquigarrow$ We say that $f : \mathbb{N} \to \mathbb{N}$ is computed by $p : \mathbb{P}$, written $\mathbf{calc}\ f\ p$ when

$$\vdash_{\mathcal{T}} \forall n : \mathbb{N}.\, \exists k : \mathbb{N}.\, \mathsf{T}(p \bullet n, f\ n, k)$$

Horrible Encodings Ahead

Let's fix a simple type theory $\mathcal{T}$ containing arithmetic.

$\rightsquigarrow$ One can define the (decidable) Turing predicate:

$$\frac{p : \mathbb{N} \qquad n : \mathbb{N} \qquad k : \mathbb{N}}{\mathsf{T}(p, n, k) : \mathsf{Prop}}$$

"$\mathsf{T}(p, n, k)$ holds iff the Turing machine p returns n in $\leq k$ steps."

[NB: for readability, I'll henceforth write $\mathbb{P} := \mathbb{N}$ to indicate numbers coding programs]

$\rightsquigarrow$ We say that $f : \mathbb{N} \to \mathbb{N}$ is computed by $p : \mathbb{P}$, written $\mathbf{calc}\ f\ p$ when

$$\vdash_{\mathcal{T}} \forall n : \mathbb{N}.\, \exists k : \mathbb{N}.\, \mathsf{T}(p \bullet n, f\ n, k)$$

Internal CT

$\mathcal{T}$ validates CT if $\vdash_{\mathcal{T}} \forall f : \mathbb{N} \to \mathbb{N}.\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p$

Alonzo in Maŝinmondo

CT is a weird principle!

- Implies a mechanical world
- A staple of Russian constructivism
- In presence of choice, incompatible with funext
- In presence of choice, incompatible with classical logic

CT is a weird principle!

- Implies a mechanical world
- A staple of Russian constructivism
- In presence of choice, incompatible with funext
- In presence of choice, incompatible with classical logic

A fleeting panic

Is it actually consistent?

Alonzo in Maŝinmondo

CT is a weird principle!

- Implies a mechanical world
- A staple of Russian constructivism
- In presence of choice, incompatible with funext
- In presence of choice, incompatible with classical logic

A fleeting panic

Is it actually consistent?

Ja.

(Mumble something about The Effective Topos™ being a model of HOL + CT.)

MARTIN-LÖF'S TYPE THEORY

MARTIN-LÖF'S TYPE THEORY

Is this a logical foundation?

MARTIN-LÖF'S TYPE THEORY

Is this a logical foundation?

Is this a programming language?

MARTIN-LÖF'S TYPE THEORY

Is this a logical foundation?

Is this a programming language?

Is this crypto-realizability?

MARTIN-LÖF'S TYPE THEORY

Is this a logical foundation?

Is this a programming language?

Is this crypto-realizability?

Is this Coq?

Blanket Propaganda

MARTIN-LÖF'S TYPE THEORY

Is this a logical foundation?

Is this a programming language?

Is this crypto-realizability?

Is this Coq?

All of this and much more!

We Need to Go Deeper

A Legitimate Question

"Can we extend Martin-Löf's Type Theory with CT?"

We Need to Go Deeper

A Legitimate Question

"Can we extend Martin-Löf's Type Theory with CT?"

An Even More Legitimate Question

"Why would I do that?"

We Need to Go Deeper

A Legitimate Question

"Can we extend Martin-Löf's Type Theory with CT?"

An Even More Legitimate Question

"Why would I do that?"

- This watch does not smell of mustard.
- Simple type theory is cool, but a bit old-fashioned and limited
- In MLTT, functions are *already* programs
- MLTT + CT is the foundation for **synthetic computability**

Synthetic Computability

Never suffer with Turing machines again!

Synthetic Computability

Never suffer with Turing machines again!

- Another instance of the synthetic trend
- Prove computability results (almost) pain-free in Coq!
- Synthetic homotopy: MLTT terms are paths
- Synthetic computability: MLTT terms are programs

Synthetic Computability

Never suffer with Turing machines again!

- Another instance of the synthetic trend
- Prove computability results (almost) pain-free in Coq!
- Synthetic homotopy: MLTT terms are paths
- Synthetic computability: MLTT terms are programs

The one missing primitive: inspecting the code of a program.

Synthetic Computability

Never suffer with Turing machines again!

- Another instance of the synthetic trend
- Prove computability results (almost) pain-free in Coq!
- Synthetic homotopy: MLTT terms are paths
- Synthetic computability: MLTT terms are programs

The one missing primitive: inspecting the code of a program.

That's exactly what CT gives you.

$$\vdash \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma(p : \mathbb{P}).\, \mathbf{calc}\ f\ p$$

Never suffer with Turing machines again!

- Another instance of the synthetic trend
- Prove computability results (almost) pain-free in Coq!
- Synthetic homotopy: MLTT terms are paths
- Synthetic computability: MLTT terms are programs

The one missing primitive: inspecting the code of a program.

That's exactly what CT gives you.

$$\vdash \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma(p : \mathbb{P}).\, \mathbf{calc}\ f\ p$$

Several people doing SCT in Coq
$\rightsquigarrow$ including this guy next door to me

Today's Definitely Legitimate Question

"Can we extend Martin-Löf's Type Theory with CT?"

I think, Therefore I merely am

In dependent type theories, existing is a complex matter

$\Sigma x : A.\, B$	v.s.	$\exists x : A.\, B$
actual existence		mere existence
proof relevant		proof-irrelevant
choice built-in		no choice *a priori*
in `Type`		in `Prop`

I think, Therefore I merely am

In dependent type theories, existing is a complex matter

$\Sigma x : A.\, B$	v.s.	$\exists x : A.\, B$
actual existence		mere existence
proof relevant		proof-irrelevant
choice built-in		no choice *a priori*
in `Type`		in `Prop`

We have not one, but *two* theses.

$$\begin{array}{lcl} \mathsf{CT}_\exists & := & \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p \\ \mathsf{CT}_\Sigma & := & \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma p : \mathbb{P}.\, \mathbf{calc}\ f\ p \end{array}$$

I think, Therefore I merely am

In dependent type theories, existing is a complex matter

$\Sigma x : A.\, B$	v.s.	$\exists x : A.\, B$
actual existence		mere existence
proof relevant		proof-irrelevant
choice built-in		no choice *a priori*
in `Type`		in `Prop`

We have not one, but *two* theses.

$$\begin{array}{lcl} \mathsf{CT}_\exists & := & \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p \\ \mathsf{CT}_\Sigma & := & \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma p : \mathbb{P}.\, \mathbf{calc}\ f\ p \end{array}$$

Which do we want?

I choose you, Sigma-chu

$$\mathsf{CT}_\exists \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- $\Pi(x : A).\, \exists(y : B).\, P$ does not magically turn into a function
- i.e. choice does not hold over $\exists$ / $\exists$ is non-computational
- does **not** endanger function extensionality
- MLTT $+$ $\mathsf{CT}_\exists$ is known to be consistent

I choose you, Sigma-chu

$$\mathsf{CT}_\exists \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- $\Pi(x : A).\, \exists(y : B).\, P$ does not magically turn into a function
- i.e. choice does not hold over $\exists$ / $\exists$ is non-computational
- does **not** endanger function extensionality
- MLTT $+$ $\mathsf{CT}_\exists$ is known to be consistent (The Effective Topos™)

I choose you, Sigma-chu

$$\mathsf{CT}_\exists \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- $\Pi(x : A).\, \exists(y : B).\, P$ does not magically turn into a function
- i.e. choice does not hold over $\exists$ / $\exists$ is non-computational
- does **not** endanger function extensionality
- MLTT + $\mathsf{CT}_\exists$ is known to be consistent (The Effective Topos™)

$$\mathsf{CT}_\Sigma \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- Dually, CT_Σ is the hallmark of weird crap going on
- Intuitionistic non-choice gives a quote function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$
- Consistency of MLTT + CT_Σ is not established

I choose you, Sigma-chu

$$\mathsf{CT}_{\exists} \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- $\Pi(x : A).\, \exists(y : B).\, P$ does not magically turn into a function
- i.e. choice does not hold over $\exists$ / $\exists$ is non-computational
- does **not** endanger function extensionality
- MLTT $+$ $\mathsf{CT}_{\exists}$ is known to be consistent (The Effective Topos™)

$$\mathsf{CT}_{\Sigma} \quad := \quad \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma p : \mathbb{P}.\, \mathbf{calc}\ f\ p$$

- Dually, CT_{Σ} is the hallmark of weird crap going on
- Intuitionistic non-choice gives a quote function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$
- Consistency of MLTT $+$ CT_{Σ} is not established

This is the one we really want to have in MLTT!

Second-hand "Quotes" from Anonymous Experts**

M.E. (Birmingham)

"MLTT is obviously inconsistent with $\mathbf{CT}_\Sigma$"

"I believe that MLTT cannot validate $\mathbf{CT}_\Sigma$"

T.S. (Darmstadt)

** All these quotes are a pure work of fiction. Serving suggestion. May contain phthalates.

Somebody is Wrong on the Internet

Are you seriously kidding me?

Somebody is Wrong on the Internet

Are you seriously kidding me?

- In MLTT, functions are **already** frigging programs!
- CT_Σ holds externally, it's called extraction (duh)

$$\text{for all} \quad \vdash f : \mathbb{N} \rightarrow \mathbb{N} \quad \text{there is} \quad \vdash p : \mathbb{P} \quad \text{s.t.} \quad \vdash \mathbf{calc}\ f\ p$$

- It is hence **obvious** that CT_Σ is compatible with MLTT
- We just have to handle those pesky **open** terms!

Somebody is Wrong on the Internet

Are you seriously kidding me?

- In MLTT, functions are **already** frigging programs!
- CT_Σ holds externally, it's called extraction (duh)

$$\text{for all} \quad \vdash f : \mathbb{N} \to \mathbb{N} \quad \text{there is} \quad \vdash p : \mathbb{P} \quad \text{s.t.} \quad \vdash \mathbf{calc}\ f\ p$$

- It is hence **obvious** that CT_Σ is compatible with MLTT
- We just have to handle those pesky **open** terms!

A tiny detail...?

No: many properties are true externally but negated internally.

The Plot Thickens

The literature is not very engaging either!

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of mTT + CT.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of mTT + CT.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of $\mathrm{mTT} + \mathrm{CT}$.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

Another line of work claims to prove the consistency of $\mathrm{MLTT} + \mathrm{CT}$.

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of $\mathsf{mTT} + \mathsf{CT}$.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

Another line of work claims to prove the consistency of $\mathsf{MLTT} + \mathsf{CT}$.

- Not even clear what theory they implement 😕

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of $\mathrm{mTT} + \mathrm{CT}$.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

Another line of work claims to prove the consistency of $\mathrm{MLTT} + \mathrm{CT}$.

- Not even clear what theory they implement 😕
- Proved in three papers, not peer-reviewed, totalling > 120 pages! 😱

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of $\mathrm{mTT} + \mathrm{CT}$.

$\rightsquigarrow$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

Another line of work claims to prove the consistency of $\mathrm{MLTT} + \mathrm{CT}$.

- Not even clear what theory they implement
- Proved in three papers, not peer-reviewed, totalling > 120 pages!
- Using game semantics!

The Plot Thickens

The literature is not very engaging either!

One line of work (implying T.S.) proves the consistency of $\mathrm{mTT} + \mathrm{CT}$.

$\leadsto$ a strict subset of MLTT, without the ξ rule.

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

... this is throwing the baby with the bathwater

Another line of work claims to prove the consistency of $\mathrm{MLTT} + \mathrm{CT}$.

- Not even clear what theory they implement
- Proved in three papers, not peer-reviewed, totalling > 120 pages!
- Using game semantics!

ABANDON THREAD

The Yes Needs The No To Win Against The No

But consistency of $\mathrm{MLTT} + \mathrm{CT}_\Sigma$ is *obviously* trivial...

The Yes Needs The No To Win Against The No

But consistency of $\mathrm{MLTT} + \mathrm{CT}_\Sigma$ is *obviously* trivial...

Only one way out: prove that I am right!

- Define an extension of MLTT proving CT_Σ
- Prove it's consistent / canonical / strongly normalizing / ...
- Formalize this in Coq otherwise nobody believes you

The Yes Needs The No To Win Against The No

But consistency of $\mathrm{MLTT} + \mathrm{CT}_\Sigma$ is *obviously* trivial...

Only one way out: prove that I am right!

- Define an extension of MLTT proving CT_Σ
- Prove it's consistent / canonical / strongly normalizing / ...
- Formalize this in Coq otherwise nobody believes you

Spoiler alert: we will sketch that in the rest of the talk.

“MLTT”

We define “MLTT” as the extension of MLTT with two new primitives.

$$M, N := \ldots \mid \text{⚲}\ M \mid \text{♀}\ M\ N$$

“MLTT”

We define “MLTT” as the extension of MLTT with two new primitives.

$$M, N := \ldots \mid \text{⚲}\ M \mid \text{♀}\ M\ N$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{⚲}\ M : \mathbb{P}}$$

“MLTT”

We define “MLTT” as the extension of MLTT with two new primitives.

$$M, N := \ldots \mid \text{⚲}\ M \mid \text{♀}\ M\ N$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{⚲}\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N : \mathtt{eval}\ (\text{⚲}\ M)\ N\ (M\ N)}$$

where

$$\begin{array}{lcl} \mathtt{eval} & : & \mathbb{P} \to \mathbb{N} \to \mathbb{N} \to \square \\ \mathtt{eval}\ P\ N\ V & \sim & \text{program } P \text{ applied to } N \text{ normalizes to } V \end{array}$$

“MLTT”

We define “MLTT” as the extension of MLTT with two new primitives.

$$M, N := \ldots \mid \text{♀}\ M \mid \text{♀}\ M\ N$$

$$\frac{\Gamma \vdash M : \mathbb{N} \rightarrow \mathbb{N}}{\Gamma \vdash \text{♀}\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \rightarrow \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N : \texttt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

where

$$\begin{array}{lcl} \texttt{eval} & : & \mathbb{P} \rightarrow \mathbb{N} \rightarrow \mathbb{N} \rightarrow \square \\ \texttt{eval}\ P\ N\ V & \sim & \text{program } P \text{ applied to } N \text{ normalizes to } V \end{array}$$

The system is parameterized by a *computation model*, given by:

- A meta-function $\lceil \cdot \rceil : \texttt{term} \Rightarrow \mathbf{N}$ (your favourite Gödel numbering)

“MLTT”

We define “MLTT” as the extension of MLTT with two new primitives.

$$M, N := \ldots \mid \text{?}\ M \mid \text{?}\ M\ N$$

$$\frac{\Gamma \vdash M : \mathbb{N} \rightarrow \mathbb{N}}{\Gamma \vdash \text{?}\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \rightarrow \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{?}\ M\ N : \texttt{eval}\ (\text{?}\ M)\ N\ (M\ N)}$$

where

$$\begin{array}{lcl} \texttt{eval} & : & \mathbb{P} \rightarrow \mathbb{N} \rightarrow \mathbb{N} \rightarrow \square \\ \texttt{eval}\ P\ N\ V & \sim & \text{program } P \text{ applied to } N \text{ normalizes to } V \end{array}$$

The system is parameterized by a *computation model*, given by:

- A meta-function $\ulcorner \cdot \urcorner : \texttt{term} \Rightarrow \mathbf{N}$ (your favourite Gödel numbering)
- An MLTT function $\vdash \texttt{run} : \mathbb{P} \rightarrow \mathbb{N} \rightarrow \mathfrak{P}(\mathbb{N})$

where $\mathfrak{P}(A) := \mathbb{N} \rightarrow \texttt{option}\ A$ is the partiality monad

and `eval` is derived from `run` through standard combinators

The Hard Part

What is the hard part?

The Hard Part

What is the hard part?

Conversion!

The Hard Part

What is the hard part?

Conversion!

$$\frac{\Gamma \vdash M : B \qquad \Gamma \vdash A \equiv B}{\Gamma \vdash M : A}$$

In MLTT the type system embeds the runtime.

What is the hard part?

Conversion!

$$\frac{\Gamma \vdash M : B \qquad \Gamma \vdash A \equiv B}{\Gamma \vdash M : A}$$

In MLTT the type system embeds the runtime.

We need to ensure that convertible terms are quoted to the same number.

Remember that CT_Σ is inconsistent with funext.

Thankfully conversion is intensional in MLTT...

Naive Solution

We need to ensure that convertible terms are quoted to the same number.

Naive Solution

We need to ensure that convertible terms are quoted to the same number.

Assume we can magically choose one representative per convertibility class.

$$\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N} \quad \text{iff} \quad \lceil \varepsilon(M) \rceil = \lceil \varepsilon(N) \rceil$$

Unfortunately, this is not going to be stable by substitution.

$$\varepsilon(M\{x := N\}) \neq \varepsilon(M)\{x := \varepsilon(N)\}$$

Immediate breakage of conversion!

The Sigh Rule

This is the problem solved by mTT.

Remember the ξ rule:

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

The Sigh Rule

This is the problem solved by mTT.

Remember the ξ rule:

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

No ξ rule $\sim$ conversion on functions implies syntactic equality

The Sigh Rule

This is the problem solved by mTT.

Remember the ξ rule:

$$\frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash \lambda x : A.\, M \equiv \lambda x : A.\, N : \Pi x : A.\, B}$$

No ξ rule $\sim$ conversion on functions implies syntactic equality

A violent way to resolve the problem!

We need something else.

The major insight for "MLTT"

Open terms are a lie! It's a conspiracy from Big Variable!

The major insight for "MLTT"

Open terms are a lie! It's a conspiracy from Big Variable!

(Source: $\mathbb{X}$.)

The major insight for “MLTT”

Open terms are a lie! It's a conspiracy from Big Variable!

(Source: $\mathbb{X}$.)

ϙ and Ϙ will only compute on (deep normal) **closed** terms

The major insight for "MLTT"

OPEN TERMS ARE A LIE! IT'S A CONSPIRACY FROM BIG VARIABLE!

(Source: $\mathbb{X}$.)

♀ and ☿ will only compute on (deep normal) **closed** terms

$$\frac{\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{♀}\ M \equiv \text{♀}\ N : \mathbb{P}} \qquad \frac{\Gamma \vdash M \equiv M' : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N \equiv N' : \mathbb{N}}{\Gamma \vdash \text{☿}\ M\ N \equiv \text{☿}\ M'\ N' : \texttt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

The major insight for “MLTT”

OPEN TERMS ARE A LIE! IT'S A CONSPIRACY FROM BIG VARIABLE!

(Source: $\mathbb{X}$.)

⚲ and ♀ will only compute on (deep normal) closed terms

$$\frac{\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{⚲}\ M \equiv \text{⚲}\ N : \mathbb{P}} \qquad \frac{\Gamma \vdash M \equiv M' : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N \equiv N' : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N \equiv \text{♀}\ M'\ N' : \texttt{eval}\ (\text{⚲}\ M)\ N\ (M\ N)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{⚲}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

The major insight for "MLTT"

OPEN TERMS ARE A LIE! IT'S A CONSPIRACY FROM BIG VARIABLE!

(Source: $\mathbb{X}$.)

♀ and ⚲ will only compute on (deep normal) closed terms

$$\frac{\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{♀}\ M \equiv \text{♀}\ N : \mathbb{P}} \qquad \frac{\Gamma \vdash M \equiv M' : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N \equiv N' : \mathbb{N}}{\Gamma \vdash \text{⚲}\ M\ N \equiv \text{⚲}\ M'\ N' : \texttt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{♀}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M, P \text{ closed dnf} \qquad n \in \mathbf{N} \qquad \Gamma \vdash P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}{\Gamma \vdash \text{⚲}\ M\ \overline{n} \equiv P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}$$

The major insight for "MLTT"

OPEN TERMS ARE A LIE! IT'S A CONSPIRACY FROM BIG VARIABLE!

(Source: $\mathbb{X}$.)

⚲ and ♀ will only compute on (deep normal) closed terms

$$\frac{\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{⚲}\ M \equiv \text{⚲}\ N : \mathbb{P}} \qquad \frac{\Gamma \vdash M \equiv M' : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N \equiv N' : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N \equiv \text{♀}\ M'\ N' : \texttt{eval}\ (\text{⚲}\ M)\ N\ (M\ N)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{⚲}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M, P \text{ closed dnf} \qquad n \in \mathbf{N} \qquad \Gamma \vdash P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}{\Gamma \vdash \text{♀}\ M\ \overline{n} \equiv P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}$$

This One Weird Trick

Closed terms are stable by substitution!

The major insight for “MLTT”

OPEN TERMS ARE A LIE! IT'S A CONSPIRACY FROM BIG VARIABLE!

(Source: $\mathbb{X}$.)

♀ and ♀ will only compute on (deep normal) **closed** terms

$$\frac{\Gamma \vdash M \equiv N : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{♀}\ M \equiv \text{♀}\ N : \mathbb{P}} \qquad \frac{\Gamma \vdash M \equiv M' : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N \equiv N' : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N \equiv \text{♀}\ M'\ N' : \texttt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{♀}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M, P \text{ closed dnf} \qquad n \in \mathbf{N} \qquad \Gamma \vdash P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}{\Gamma \vdash \text{♀}\ M\ \overline{n} \equiv P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}$$

This One Weird Trick

Closed terms are stable by substitution!

(Some additional technicalities to validate η-laws.)

The Secret Sauce

These functions return hereditarily positive types (aka Σ^0_1 formulae)

The Secret Sauce

These functions return hereditarily positive types (aka Σ^0_1 formulae)

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{♀}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

The Secret Sauce

These functions return hereditarily positive types (aka Σ^0_1 formulae)

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{♀}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M, P \text{ closed dnf} \qquad n \in \mathbf{N} \qquad \Gamma \vdash P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}{\Gamma \vdash \text{♀}\ M\ \overline{n} \equiv P : \texttt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n})}$$

$$\begin{array}{lcl} \texttt{eval} & : & \mathbb{P} \to \mathbb{N} \to \mathbb{N} \to \square \\ \texttt{eval}\ f\ n\ v & := & \Sigma k : \mathbb{N}.\, \texttt{step}\ k\ (\texttt{run}\ f\ n)\ v \end{array}$$

$$\begin{array}{lcl} \texttt{step} & : & \mathbb{N} \to (\mathbb{N} \to \texttt{option}\ \mathbb{N}) \to \mathbb{N} \to \square \\ \texttt{step}\ \mathsf{O}\ p\ v & := & p\ \mathsf{O} = \texttt{Some}\ v \\ \texttt{step}\ (\mathsf{S}\ k)\ p\ v & := & (p\ k = \texttt{None}) \times (\texttt{step}\ k\ (p \circ \mathsf{S})\ v) \end{array}$$

The Secret Sauce

These functions return hereditarily positive types (aka Σ^0_1 formulae)

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ closed dnf}}{\Gamma \vdash \text{♀}\ M \equiv \ulcorner M \urcorner : \mathbb{P}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M, P \text{ closed dnf} \qquad n \in \mathbf{N} \qquad \Gamma \vdash P : \texttt{eval}\ \ulcorner M \urcorner\ \overline{n}\ (M\ \overline{n})}{\Gamma \vdash \text{♀}\ M\ \overline{n} \equiv P : \texttt{eval}\ \ulcorner M \urcorner\ \overline{n}\ (M\ \overline{n})}$$

$$\begin{array}{lcl} \texttt{eval} & : & \mathbb{P} \to \mathbb{N} \to \mathbb{N} \to \square \\ \texttt{eval}\ f\ n\ v & := & \Sigma k : \mathbb{N}.\, \texttt{step}\ k\ (\texttt{run}\ f\ n)\ v \end{array}$$

$$\begin{array}{lcl} \texttt{step} & : & \mathbb{N} \to (\mathbb{N} \to \texttt{option}\ \mathbb{N}) \to \mathbb{N} \to \square \\ \texttt{step}\ \mathsf{O}\ p\ v & := & p\ \mathsf{O} = \texttt{Some}\ v \\ \texttt{step}\ (\mathsf{S}\ k)\ p\ v & := & (p\ k = \texttt{None}) \times (\texttt{step}\ k\ (p \circ \mathsf{S})\ v) \end{array}$$

These types have canonical “absolute” values!

The Basic Model

A variant of Abel's style NbE model in (small) IR

The Basic Model

A variant of Abel's style NbE model in (small) IR

- Type formation is defined inductively: $\Gamma \Vdash A$

$$\frac{A \Rightarrow^{*}_{\mathsf{wh}} \mathbb{N}}{\mathfrak{r}_{\mathbb{N}} : \Gamma \Vdash A} \qquad \frac{A \Rightarrow^{*}_{\mathsf{wh}} \Pi(x : X).\, Y \qquad p : \Gamma \Vdash X \qquad q : \Gamma, x : X \Vdash Y}{\mathfrak{r}_{\Pi}\ p\ q : \Gamma \Vdash A} \qquad \ldots$$

The Basic Model

A variant of Abel's style NbE model in (small) IR

- Type formation is defined inductively: $\Gamma \Vdash A$

$$\frac{A \Rightarrow^*_{\mathsf{wh}} \mathbb{N}}{\mathfrak{r}_{\mathbb{N}} : \Gamma \Vdash A} \qquad \frac{A \Rightarrow^*_{\mathsf{wh}} \Pi(x : X).\, Y \qquad p : \Gamma \Vdash X \qquad q : \Gamma, x : X \Vdash Y}{\mathfrak{r}_{\Pi}\ p\ q : \Gamma \Vdash A} \qquad \cdots$$

- Term typedness $\Gamma \Vdash M : A \mid p$ is defined by recursion on $p : \Gamma \Vdash A$

$$\begin{array}{lcl} \Gamma \Vdash M : \mathbb{N} \mid \mathfrak{r}_{\mathbb{N}} & := & \Gamma \Vdash M \in \mathbb{N} \\ \Gamma \Vdash M : \Pi(x : A).\, B \mid \mathfrak{r}_{\Pi}\ p\ q & := & \\ \quad \Pi(\rho : \Delta \leq \Gamma).\, (\Delta \Vdash a : A\langle\rho\rangle \mid p) \rightarrow \Delta \Vdash M\langle\rho\rangle\ a : B\{\rho, a\} \mid q & & \end{array}$$

$$\frac{M \Rightarrow^*_{\mathsf{wh}} \mathsf{O}}{\Gamma \Vdash M \in \mathbb{N}} \qquad \frac{M \Rightarrow^*_{\mathsf{wh}} \mathsf{S}\ N \qquad \Gamma \Vdash N \in \mathbb{N}}{\Gamma \Vdash M \in \mathbb{N}} \qquad \frac{\Gamma \vdash n : \mathbb{N} \qquad \mathsf{wne}\,(n)}{\Gamma \Vdash n \in \mathbb{N}}$$

A variant of Abel's style NbE model in (small) IR

- Type formation is defined inductively: $\Gamma \Vdash A$

$$\frac{A \Rightarrow^{*}_{\mathsf{wh}} \mathbb{N}}{\mathfrak{r}_{\mathbb{N}} : \Gamma \Vdash A} \qquad \frac{A \Rightarrow^{*}_{\mathsf{wh}} \Pi(x : X).\, Y \qquad p : \Gamma \Vdash X \qquad q : \Gamma, x : X \Vdash Y}{\mathfrak{r}_{\Pi}\ p\ q : \Gamma \Vdash A} \qquad \ldots$$

- Term typedness $\Gamma \Vdash M : A \mid p$ is defined by recursion on $p : \Gamma \Vdash A$

$$\begin{array}{lcl} \Gamma \Vdash M : \mathbb{N} \mid \mathfrak{r}_{\mathbb{N}} & := & \Gamma \Vdash M \in \mathbb{N} \\ \Gamma \Vdash M : \Pi(x : A).\, B \mid \mathfrak{r}_{\Pi}\ p\ q & := & \\ \quad \Pi(\rho : \Delta \leq \Gamma).\, (\Delta \Vdash a : A\langle\rho\rangle \mid p) \to \Delta \Vdash M\langle\rho\rangle\ a : B\{\rho, a\} \mid q & & \end{array}$$

$$\frac{M \Rightarrow^{*}_{\mathsf{wh}} \mathsf{O}}{\Gamma \Vdash M \in \mathbb{N}} \qquad \frac{M \Rightarrow^{*}_{\mathsf{wh}} \mathsf{S}\ N \qquad \Gamma \Vdash N \in \mathbb{N}}{\Gamma \Vdash M \in \mathbb{N}} \qquad \frac{\Gamma \vdash n : \mathbb{N} \qquad \mathsf{wne}\,(n)}{\Gamma \Vdash n \in \mathbb{N}}$$

(ditto for conversion) (+ second layer of *validity* aka closure by substitution)

The Basic Model for Dummies

This is just realizability with (a lot of) bells and whistles

$$\Gamma \Vdash M : A \mid p_A \quad \sim \quad M \text{ wh-normalizes to a value at type } A$$

The Basic Model for Dummies

This is just realizability with (a lot of) bells and whistles

$$\Gamma \Vdash M : A \mid p_A \quad \sim \quad M \text{ wh-normalizes to a value at type } A$$

Being a value at type A is defined by case-analysis on A.

The Basic Model for Dummies

This is just realizability with (a lot of) bells and whistles

$$\Gamma \Vdash M : A \mid p_A \quad \sim \quad M \text{ wh-normalizes to a value at type } A$$

Being a value at type A is defined by case-analysis on A.

Importantly, all well-typed neutrals are values of the corresponding type.

$$\frac{}{\mathsf{wne}\,(x)} \qquad \frac{\mathsf{wne}\,(n)}{\mathsf{wne}\,(n\ M)} \qquad \frac{\mathsf{wne}\,(n)}{\mathsf{wne}\,(\mathbb{N}_{\mathbf{rec}}\ P\ T_{\mathsf{O}}\ T_{\mathsf{S}}\ n)} \qquad \ldots$$

This is the standard and correct way to handle open terms.

The Basic Model for Dummies

This is just realizability with (a lot of) bells and whistles

$$\Gamma \Vdash M : A \mid p_A \quad \sim \quad M \text{ wh-normalizes to a value at type } A$$

Being a value at type A is defined by case-analysis on A.

Importantly, all well-typed neutrals are values of the corresponding type.

$$\frac{}{\mathsf{wne}\,(x)} \qquad \frac{\mathsf{wne}\,(n)}{\mathsf{wne}\,(n\ M)} \qquad \frac{\mathsf{wne}\,(n)}{\mathsf{wne}\,(\mathbb{N}_{\mathbf{rec}}\ P\ T_{\mathsf{O}}\ T_{\mathsf{S}}\ n)} \qquad \ldots$$

This is the standard and correct way to handle open terms.

(They don't exist anyways, remember?)

Comparison with standard NbE

Type interpretation unchanged

- No funny business with effects or whatnot
- In particular we have the same canonicity properties

Comparison with standard NbE

Type interpretation unchanged

- No funny business with effects or whatnot
- In particular we have the same canonicity properties

Differences with Abel's model

$\rightsquigarrow$ annotate reducibility proofs with deep normalization

$$\Gamma \Vdash M : A \mid p_A \qquad \text{implies} \qquad M \Downarrow_{\mathsf{deep}} M_0 \quad \text{with} \quad \Gamma \vdash M \equiv M_0 : A$$

$\rightsquigarrow$ normal / neutral terms generalized into deep and weak-head variants

$\rightsquigarrow$ extend neutrals to contain quotes blocked on open terms

$$\frac{\mathsf{dnf}\,(M) \qquad M \text{ not closed}}{\mathsf{wne}\,(\text{♀}\ M)} \qquad \frac{\mathsf{dnf}\,(M) \qquad \mathsf{dnf}\,(N) \qquad M \text{ or } N \text{ not closed}}{\mathsf{wne}\,(\text{♀}\ M\ N)}$$

Comparison with standard NbE

Type interpretation unchanged

- No funny business with effects or whatnot
- In particular we have the same canonicity properties

Differences with Abel's model

$\leadsto$ annotate reducibility proofs with deep normalization

$$\Gamma \Vdash M : A \mid p_A \qquad \text{implies} \qquad M \Downarrow_{\mathsf{deep}} M_0 \quad \text{with} \quad \Gamma \vdash M \equiv M_0 : A$$

$\leadsto$ normal / neutral terms generalized into deep and weak-head variants

$\leadsto$ extend neutrals to contain quotes blocked on open terms

$$\frac{\mathsf{dnf}\,(M) \qquad M \text{ not closed}}{\mathsf{wne}\,(\text{♀}\ M)} \qquad \frac{\mathsf{dnf}\,(M) \qquad \mathsf{dnf}\,(N) \qquad M \text{ or } N \text{ not closed}}{\mathsf{wne}\,(\text{♀}\ M\ N)}$$

... and that's about it.

Some Dust under the Rug

"MLTT" is reduction-free. I didn't define properly reduction!

- For the MLTT fragment, weak-head reduction is standard.
- Deep reduction is just iterated weak-head reduction.

$$\frac{M\ N \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} R} \qquad \frac{\mathsf{wne}\,(M) \qquad M \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} R\ N} \qquad \frac{\mathsf{dne}\,(M) \qquad N \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} M\ R}$$

- In particular, it is deterministic (critical!)

Some Dust under the Rug

“MLTT” is reduction-free. I didn't define properly reduction!

- For the MLTT fragment, weak-head reduction is standard.
- Deep reduction is just iterated weak-head reduction.

$$\frac{M\ N \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} R} \qquad \frac{\mathsf{wne}\,(M) \qquad M \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} R\ N} \qquad \frac{\mathsf{dne}\,(M) \qquad N \Rightarrow_{\mathsf{wh}} R}{M\ N \Rightarrow_{\mathsf{deep}} M\ R}$$

- In particular, it is deterministic (critical!)

Reduction for ⚲ is straightforward

$$\frac{M \Rightarrow_{\mathsf{deep}} R}{\text{⚲}\ M \Rightarrow_{\mathsf{wh}} \text{⚲}\ R} \qquad \frac{M\ \text{closed dnf}}{\text{⚲}\ M \Rightarrow_{\mathsf{wh}} \ulcorner M \urcorner}$$

Don't Quote Me on That

The only tricky case is the rule for ♀

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N : \mathtt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

Don't Quote Me on That

The only tricky case is the rule for Ϙ

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{Ϙ}\ M\ N : \texttt{eval}\ (\text{Ϙ}\ M)\ N\ (M\ N)}$$

$\leadsto$ congruence as expected

$$\frac{M \Rightarrow_{\mathsf{deep}} R}{\text{Ϙ}\ M\ N \Rightarrow_{\mathsf{wh}} \text{Ϙ}\ R\ N} \qquad \frac{M\ \mathsf{dnf} \qquad N \Rightarrow_{\mathsf{deep}} R}{\text{Ϙ}\ M\ N \Rightarrow_{\mathsf{wh}} \text{Ϙ}\ M\ R}$$

Don't Quote Me on That

The only tricky case is the rule for ♀

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{♀}\ M\ N : \mathtt{eval}\ (\text{♀}\ M)\ N\ (M\ N)}$$

$\rightsquigarrow$ congruence as expected

$$\frac{M \Rightarrow_{\mathsf{deep}} R}{\text{♀}\ M\ N \Rightarrow_{\mathsf{wh}} \text{♀}\ R\ N} \qquad \frac{M\ \mathsf{dnf} \qquad N \Rightarrow_{\mathsf{deep}} R}{\text{♀}\ M\ N \Rightarrow_{\mathsf{wh}} \text{♀}\ M\ R}$$

$\rightsquigarrow$ for the actual reduction, basically compute the unique fuel

$$\frac{M\ \text{closed, dnf} \qquad k\ \text{smallest integer s.t.} \qquad M\ \overline{n}\ \Downarrow^{k}\ \overline{v}}{\text{♀}\ M\ \overline{n} \Rightarrow_{\mathsf{wh}} (\overline{k}, \mathtt{refl}, \ldots, \mathtt{refl})}$$

Fact 1: $\text{♀}\ M\ \overline{n} : \mathtt{eval}\ \lceil M \rceil\ \overline{n}\ (M\ \overline{n}) \equiv \Sigma k : \mathbb{N}.\ \mathtt{step}\ k\ (\mathtt{run}\ \lceil M \rceil\ \overline{n})\ (M\ \overline{n})$

Fact 2: $\mathtt{step}\ \overline{k}\ p\ v := (p\ \mathsf{O} = \mathsf{None}) \times \ldots \times (p\ \overline{k-1} = \mathsf{None}) \times (p\ \overline{k} = \mathsf{Some}\ \overline{v})$

L'η de droit

Some technical details to handle η-laws

$$\frac{\Gamma \vdash M : \Pi(x : A).\, B \qquad x \notin M}{\Gamma \vdash \lambda x : A.\, M\; x \equiv M : \Pi(x : A).\, B} \qquad \frac{\Gamma \vdash M : \Sigma(x : A).\, B}{\Gamma \vdash \langle M.\mathtt{fst}, M.\mathtt{snd} \rangle_{A,B} \equiv M : \Sigma(x : A).\, B}$$

L'η de droit

Some technical details to handle η-laws

$$\frac{\Gamma \vdash M : \Pi(x : A).\, B \qquad x \notin M}{\Gamma \vdash \lambda x : A.\, M\, x \equiv M : \Pi(x : A).\, B} \qquad \frac{\Gamma \vdash M : \Sigma(x : A).\, B}{\Gamma \vdash \langle M.\mathtt{fst}, M.\mathtt{snd} \rangle_{A,B} \equiv M : \Sigma(x : A).\, B}$$

Quoting has to be stable by these η laws!

- We have to pick a canonical representative for dnf up to η
- We cannot infer the type annotations
- Thankfully they are not computationally relevant

L'η de droit

Some technical details to handle η-laws

$$\frac{\Gamma \vdash M : \Pi(x : A).\, B \qquad x \notin M}{\Gamma \vdash \lambda x : A.\, M\ x \equiv M : \Pi(x : A).\, B} \qquad \frac{\Gamma \vdash M : \Sigma(x : A).\, B}{\Gamma \vdash \langle M.\mathtt{fst}, M.\mathtt{snd}\rangle_{A,B} \equiv M : \Sigma(x : A).\, B}$$

Quoting has to be stable by these η laws!

- We have to pick a canonical representative for dnf up to η
- We cannot infer the type annotations
- Thankfully they are not computationally relevant

Simple* solution

Quoting performs η-reduction and erasure of type annotations.

The Theorems

We say that the computation model $(\lceil\cdot\rceil, \mathtt{run})$ is adequate when:
for all $M \in \mathtt{term}$ and $n, r, k \in \mathbf{N}$, $M\,\overline{n} \Downarrow^k\ \overline{r}$ implies

- $\mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k} \Downarrow \mathsf{Some}\ \overline{r}$
- $\mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k'} \Downarrow \mathsf{None}$ for all $k' < k$

The Theorems

We say that the computation model $(\lceil\cdot\rceil, \mathtt{run})$ is adequate when:
for all $M \in \mathtt{term}$ and $n, r, k \in \mathbf{N}$, $M\,\overline{n} \Downarrow^{k}\ \overline{r}$ implies

- $\mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k} \Downarrow \text{Some}\ \overline{r}$
- $\mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k'} \Downarrow \text{None}$ for all $k' < k$

Theorem

If the model is adequate, the logical relation is sound and complete.

A Taste of Not Nice and Dire

A sketch of why ⫯ and ⟟ validate their type

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_Σ.

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_{Σ}.

Reminder

Soundness means that a typable term is in the logical relation.

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_Σ*.*

Reminder

Soundness means that a typable term is in the logical relation.

Theorem (Consistency)

There is no closed term of type $\bot$ *in "MLTT".*

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_Σ*.*

Reminder

Soundness means that a typable term is in the logical relation.

Theorem (Consistency)

There is no closed term of type $\perp$ *in "MLTT".*

Theorem (Canonicity)

All closed terms of type $\mathbb{N}$ *in "MLTT" reduce to an integer.*

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_Σ.

Reminder

Soundness means that a typable term is in the logical relation.

Theorem (Consistency)

There is no closed term of type $\bot$ *in "MLTT".*

Theorem (Canonicity)

All closed terms of type $\mathbb{N}$ *in "MLTT" reduce to an integer.*

Theorem (Normalization)

All typed terms of "MLTT" weak-head normalize.

Formalization

Based on Adjedj et al. CPP'24 "Martin-Löf à la Coq" (using small IR)

Formalization

Based on Adjedj et al. CPP'24 "Martin-Löf à la Coq" (using small IR)

The base theory contains one universe, Π / Σ types with η-laws, $\bot$, $\mathbb{N}$, $\mathbf{Id}$

MLTT + ϙ fully formalized in Coq

As of 2023/12/01, ϙ basically done, only annoying stuff remains

Formalization

Based on Adjedj et al. CPP'24 "Martin-Löf à la Coq" (using small IR)

The base theory contains one universe, Π / Σ types with η-laws, $\bot$, $\mathbb{N}$, $\mathbf{Id}$

MLTT + ♀ fully formalized in Coq

As of 2023/12/01, ♀ basically done, only annoying stuff remains

No expected surprise, just tedious proofs on untyped reduction.

- A weak form of confluence
- Proving that erasure is computationally harmless

Formalization

Based on Adjedj et al. CPP'24 "Martin-Löf à la Coq" (using small IR)

The base theory contains one universe, Π / Σ types with η-laws, $\bot$, $\mathbb{N}$, $\mathbf{Id}$

MLTT + ♀ fully formalized in Coq

As of 2023/12/01, ♀ basically done, only annoying stuff remains

No expected surprise, just tedious proofs on untyped reduction.

- A weak form of confluence
- Proving that erasure is computationally harmless

Nightmare stuff I'm not gonna prove: the existence of adequate models

Typical instance of "conceptually trivial but practically impossible".

Conclusion

- $\mathsf{MLTT} + \mathsf{CT}_\Sigma$ is obviously consistent, obviously
- The model is a trivial adaptation of standard NbE models
- Open terms do not exist. I have met them.
- A sizable chunk proved in Coq
- I must be missing something from our anonymous experts

Thanks for your attention.