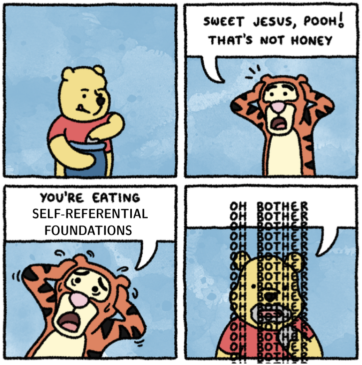
“Proofs *are* programs” in MLTT



Pierre-Marie Pédrot

INRIA

TYPES'24

Proofs *are* programs in MLTT!

Proofs *are* programs in MLTT!

Beyond obvious.

- Proofs are first-class syntactic objects
- Extension of the λ-calculus
- The equational theory can be derived from β-reduction
- Strong normalization and canonicity
- No need for *post-hoc* realizability

Proofs *are* programs in MLTT!

Beyond obvious.

- Proofs are first-class syntactic objects
- Extension of the λ-calculus
- The equational theory can be derived from β-reduction
- Strong normalization and canonicity
- No need for *post-hoc* realizability

MLTT: The Ultimate Synthetic Monistic Curry-Howard System

Proofs *are* programs in MLTT?

Proofs *are* programs in MLTT?

On second thought, it is not so clear.

- MLTT has uncomputational models
- For instance, in $\mathbf{Set}$ functions are ZFC functional graphs
- Is our Curry-Howard faith grounded in reality?

A Hint of Heresy

Our credo is just an **external** statement!

Proofs *are* programs in MLTT

The Problem

Our credo is just an **external** statement!

~~**Proofs *are* programs in** MLTT~~

The Problem

Our credo is just an **external** statement!

~~**Proofs *are* programs in** MLTT~~

What we really want is an **internal** statement.

"Proofs *are* programs" in MLTT

Church's Church

Turns out it is a well-known principle in constructive maths.

Church's Church

Turns out it is a well-known principle in constructive maths.

The internal Church Thesis

$$\vdash \forall (f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{N}.\, \mathbf{calc}\ f\ p \quad (\mathsf{CT})$$

Church's Church

Turns out it is a well-known principle in constructive maths.

The internal Church Thesis

$$\vdash \forall (f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{N}.\, \mathbf{calc}\ f\ p \qquad \text{(CT)}$$

where $\mathbf{calc}\ f\ p$ means that f is computed by the program p, i.e.

$$\vdash \forall n : \mathbb{N}.\, \exists k : \mathbb{N}.\, \mathtt{eval}\ p\ n\ (f\ n)\ k$$

with $\mathtt{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \square$ the Kleene predicate

"$\mathtt{eval}\ p\ n\ v\ k \quad \sim \quad$ the Turing machine p run on n returns v in $\leq k$ steps."

Church's Church

Turns out it is a well-known principle in constructive maths.

The internal Church Thesis

$$\vdash \forall (f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{N}.\, \mathbf{calc}\ f\ p \quad (\mathsf{CT})$$

where $\mathbf{calc}\ f\ p$ means that f is computed by the program p, i.e.

$$\vdash \forall n : \mathbb{N}.\, \exists k : \mathbb{N}.\, \mathtt{eval}\ p\ n\ (f\ n)\ k$$

with $\mathtt{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \square$ the Kleene predicate

"$\mathtt{eval}\ p\ n\ v\ k \quad \sim \quad$ the Turing machine p run on n returns v in $\leq k$ steps."

In case of allergy to Turing machines, pick any other model.

Church's Church

Turns out it is a well-known principle in constructive maths.

The internal Church Thesis

$$\vdash \forall (f : \mathbb{N} \to \mathbb{N}).\, \exists p : \mathbb{N}.\, \mathbf{calc}\ f\ p \quad (\mathsf{CT})$$

where $\mathbf{calc}\ f\ p$ means that f is computed by the program p, i.e.

$$\vdash \forall n : \mathbb{N}.\, \exists k : \mathbb{N}.\, \mathtt{eval}\ p\ n\ (f\ n)\ k$$

with $\mathtt{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \square$ the Kleene predicate

"$\mathtt{eval}\ p\ n\ v\ k \quad \sim \quad$ the Turing machine p run on n returns v in $\leq k$ steps."

In case of allergy to Turing machines, pick any other model.

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

I am not Making this CT Up

Synthetic Computability

Never suffer with Turing machines again!

Satisfied or money back

AS SEEN ON TV!

Prove computability results (almost) pain-free in Coq!

100% GLUTEN FREE

I am not Making this CT Up

Synthetic Computability

Never suffer with Turing machines again!

Satisfied or money back

AS SEEN ON TV!

Prove computability results (almost) pain-free in Coq!

100% GLUTEN FREE

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

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

I am not Making this CT Up

Synthetic Computability

Never suffer with Turing machines again!

Satisfied or money back

AS SEEN ON TV!

Prove computability results (almost) pain-free in Coq!

100% GLUTEN FREE

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

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

"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
- non-computational, relatively innocuous
- $\mathsf{MLTT} + \mathsf{CT}_\exists$ 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
- non-computational, relatively innocuous
- $\mathsf{MLTT} + \mathsf{CT}_\exists$ 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
- non-computational, relatively innocuous
- $\mathsf{MLTT} + \mathsf{CT}_\exists$ is known to be consistent (The Effective Topos™)

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

- Weird consequences: anti-funext, anti-choice, anti-classical logic
- Intuitionistic non-choice gives a quote function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{P}$
- Consistency of $\mathsf{MLTT} + \mathsf{CT}_\Sigma$ 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
- non-computational, relatively innocuous
- MLTT + $\mathsf{CT}_{\exists}$ is known to be consistent (The Effective Topos™)

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

- Weird consequences: anti-funext, anti-choice, anti-classical logic
- Intuitionistic non-choice gives a quote function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{P}$
- 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 CT_Σ"

"I believe that MLTT cannot validate CT_Σ"

T.S. (Darmstadt)

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

Thou Shalt Not Bear False Witness

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

Thou Shalt Not Bear False Witness

But consistency of $\mathsf{MLTT} + \mathsf{CT}_\Sigma$ 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

Thou Shalt Not Bear False Witness

But consistency of $\mathsf{MLTT} + \mathsf{CT}_\Sigma$ 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”

“MLTT” is the extension of MLTT with three quoting primitives.

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

“MLTT”

“MLTT” is the extension of MLTT with three quoting primitives.

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

$$\text{(quote)}\ \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{♀}\ M : \mathbb{P}}$$

"MLTT"

"MLTT" is the extension of MLTT with three quoting primitives.

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

$$\text{(quote)}\ \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{⚲}\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{ʕ}\ M\ N : \mathbb{N}}\ \text{(count-steps)}$$

“MLTT”

“MLTT” is the extension of MLTT with three quoting primitives.

$$M, N := \ldots \mid \qoppa\ M \mid \stigma\ M\ N \mid \varrho\ M\ N$$

$$\text{(quote)}\ \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \qoppa\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \stigma\ M\ N : \mathbb{N}}\ \text{(count-steps)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \varrho\ M\ N : \mathtt{eval}\ (\qoppa\ M)\ N\ (M\ N)\ (\stigma\ M\ N)}\ \text{(reflect)}$$

“MLTT” is the extension of MLTT with three quoting primitives.

$$M, N := \ldots \mid \text{ϙ}\ M \mid \text{ϛ}\ M\ N \mid \varrho\ M\ N$$

$$\text{(quote)}\ \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N}}{\Gamma \vdash \text{ϙ}\ M : \mathbb{P}} \qquad \frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \text{ϛ}\ M\ N : \mathbb{N}}\ \text{(count-steps)}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \varrho\ M\ N : \mathtt{eval}\ (\text{ϙ}\ M)\ N\ (M\ N)\ (\text{ϛ}\ M\ N)}\ \text{(reflect)}$$

These three operations are just the Skolemization of CT_Σ!

$$\mathsf{CT}_\Sigma := \Pi(f : \mathbb{N} \to \mathbb{N}).\, \Sigma p : \mathbb{P}.\, \Pi(n : \mathbb{N}).\, \Sigma(k : \mathbb{N}).\, \mathtt{eval}\ p\ n\ (f\ n)\ k$$

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Conversion!

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Conversion!

😱 **Convertible terms must be quoted to the same code** 😱

In particular, quoting must be stable by substitution. How to do that?

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Conversion!

😱 **Convertible terms must be quoted to the same code** 😱

In particular, quoting must be stable by substitution. How to do that?

Before breaking this unbearable suspense, I need a bit more stuff.

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Conversion!

😱 **Convertible terms must be quoted to the same code** 😱

In particular, quoting must be stable by substitution. How to do that?

Before breaking this unbearable suspense, I need a bit more stuff.

"MLTT" is parameterized by a *computation model*, given by:

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

Convert Now or Face Type Shunning

Where is the type-theoretic fish?

Conversion!

😱 **Convertible terms must be quoted to the same code** 😱

In particular, quoting must be stable by substitution. How to do that?

Before breaking this unbearable suspense, I need a bit more stuff.

"MLTT" is parameterized by a *computation model*, given by:

- A meta-function $\lceil \cdot \rceil : \mathtt{term} \Rightarrow \mathbf{N}$ (your favourite Gödel numbering)
- An MLTT function $\vdash \mathtt{run} : \mathbb{P} \to \mathbb{N} \to \mathfrak{P}(\mathbb{N})$

where $\mathfrak{P}(A) := \mathbb{N} \to 1 + A$ is the partiality monad and `eval` is derived from `run`

Wake Up Sheeple!

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

Wake Up Sheeple!

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

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

Wake Up Sheeple!

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

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

The quoting primitives will only compute on **closed** deep normal terms

Wake Up Sheeple!

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

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

The quoting primitives will only compute on **closed** deep normal terms

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \color{red}{M \text{ cldnf}}}{\Gamma \vdash ♀\, M \equiv \ulcorner M \urcorner : \mathbb{P}}$$

Wake Up Sheeple!

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

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

The quoting primitives will only compute on **closed** deep normal terms

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

$$\frac{M \text{ cldnf} \qquad \{\Gamma \vdash \mathbf{run}\ \ulcorner M \urcorner\ \overline{n}\ \overline{k} \equiv \mathtt{None}\}_{k<k_0} \qquad \Gamma \vdash \mathbf{run}\ \ulcorner M \urcorner\ \overline{n}\ \overline{k}_0 \equiv \mathtt{Some}\ \overline{v}}{\Gamma \vdash \text{⚲}\ M\ \overline{n} \equiv \overline{k}_0 : \mathbb{N}}$$

Wake Up Sheeple!

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

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

The quoting primitives will only compute on **closed** deep normal terms

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad M \text{ cldnf}}{\Gamma \vdash \text{ϙ}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{M \text{ cldnf} \qquad \{\Gamma \vdash \mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k} \equiv \mathtt{None}\}_{k<k_0} \qquad \Gamma \vdash \mathtt{run}\ \lceil M \rceil\ \overline{n}\ \overline{k}_0 \equiv \mathtt{Some}\ \overline{v}}{\Gamma \vdash \rotatebox{180}{ϙ}\ M\ \overline{n} \equiv \overline{k}_0 : \mathbb{N}}$$

(Congruences trivial, similar rule for ϱ.)

Wake Up Sheeple!

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

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

The quoting primitives will only compute on **closed** deep normal terms

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{N} \qquad \color{red}{M\ \text{cldnf}}}{\Gamma \vdash \text{♀}\ M \equiv \lceil M \rceil : \mathbb{P}}$$

$$\frac{\color{red}{M\ \text{cldnf}} \qquad \{\Gamma \vdash \mathbf{run}\ \lceil M \rceil\ \overline{n}\ \overline{k} \equiv \mathtt{None}\}_{k<k_0} \qquad \Gamma \vdash \mathbf{run}\ \lceil M \rceil\ \overline{n}\ \overline{k}_0 \equiv \mathtt{Some}\ \overline{v}}{\Gamma \vdash \text{ʕ}\ M\ \overline{n} \equiv \overline{k}_0 : \mathbb{N}}$$

(Congruences trivial, similar rule for ϱ.)

This One Weird Trick

Closed terms are stable by substitution.

(Some additional technicalities to validate η-laws.)

The Basic Model

A straightforward variant of Abel's style NbE logical relation

The Basic Model

A straightforward variant of Abel's style NbE logical relation

⇝ annotate reducibility proofs with deep normalization

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

⇝ normal / neutral terms generalized into deep and weak-head variants

⇝ extend neutrals to contain quoting primitives blocked on open terms

$$\frac{\mathsf{dnf}\,(M) \qquad M \text{ not closed}}{\mathsf{wne}\,(♀\ M)} \qquad \text{(similar for ƭ, } \varrho\text{)}$$

The Basic Model

A straightforward variant of Abel's style NbE logical relation

$\rightsquigarrow$ annotate reducibility proofs with deep normalization

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

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

$\rightsquigarrow$ extend neutrals to contain quoting primitives blocked on open terms

$$\frac{\mathsf{dnf}\,(M) \qquad M \text{ not closed}}{\mathsf{wne}\,(♀\ M)} \qquad \text{(similar for } ⍑,\ \varrho\text{)}$$

... and that's about it.

Some Dust under the Rug

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

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.

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.
- Weak-head reduction of quoting primitives depends on deep reduction

$$\frac{M \Rightarrow_{\mathsf{deep}} R}{\text{♀}\ M \Rightarrow_{\mathsf{wh}} \text{♀}\ R} \qquad \frac{M \text{ closed dnf}}{\text{♀}\ M \Rightarrow_{\mathsf{wh}} \lceil M \rceil}$$

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.
- Weak-head reduction of quoting primitives depends on deep reduction

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

- Reduction of ⫰ and ϱ additionally require counting steps

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.
- Weak-head reduction of quoting primitives depends on deep reduction

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

- Reduction of ɂ and ϱ additionally require counting steps

Better presented as a step-indexed big-step reduction

$$M \Rightarrow^{*}_{\mathsf{deep}} N\ \text{dnf} \qquad \text{iff} \qquad \exists k.\ M \Downarrow^{k} N$$

Well-typed models cannot go wrong

We haven't assumed anything about the computation model so far.

Well-typed models cannot go wrong

We haven't assumed anything about the computation model so far.

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

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

Well-typed models cannot go wrong

We haven't assumed anything about the computation model so far.

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

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

Theorem

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

The Real Results

Theorem (It's written on the can)

"MLTT" proves CT_Σ*.*

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.*

Theorem (Normalization)

Well-typed "MLTT" terms are strongly normalizing.

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, $\perp$, $\mathbb{N}$, $\mathbf{Id}$

Fully formalized in Coq up to one axiom.

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, $\bot$, $\mathbb{N}$, $\mathbf{Id}$

Fully formalized in Coq up to one axiom.

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

Typical instance of "conceptually trivial but practically impossible".

We have *already* implemented an adequate model in the Coq meta.

MLTT contains PRA and the evaluator is primitive recursive.

Conclusion

In MLTT, "proofs *are* programs" in the end

- The model is a trivial adaptation of standard NbE models
- Essentially fully formalized in Coq
- Open terms do not exist, I have met them
- What kind of axioms can we cheaply internalize like this?

Conclusion

In MLTT, "proofs *are* programs" in the end

- The model is a trivial adaptation of standard NbE models
- Essentially fully formalized in Coq
- Open terms do not exist, I have met them
- What kind of axioms can we cheaply internalize like this?

A lingering doubt

Why was this considered a difficult question?

Thanks for your attention.