

Implementing a Modal Dependent Type Theory

Daniel Gratzer⁰ Jonathan Sterling¹ Lars Birkedal⁰ August 21, 2019 ICFP '19

⁰Aarhus University ¹Carnegie Mellon University Modalities

We want to add a single modality MLTT, **D**.

$\Gamma \vdash M: \Box A$

$\leftrightarrow \rightarrow$

M ":" A and *M* only mentions variables of the shape $\square B$

- In staged programming, $\square A$ represents precomputed values.
- In modal FRP, 🗆 A represents stable types.
- In distributed programming, $\square A$ represents globally available values.



We contribute $\mathsf{MLTT}_{{\scriptscriptstyle \!\!\! \text{ B}}},$ a dependent type theory with...

- the box modality, $\Box A$
- dependent sums, $\Sigma(A, B)$
- dependent products, $\Pi(A, B)$
- natural numbers, nat
- intensional identity types, Id(A, M, N)
- a cumulative hierarchy of universes, U₀, U₁...

We have constructed a precise syntactic account of ${\rm MLTT}_{\underline{a}},$ and proved the decidability of type-checking for it.

 $\left. \right\}$ With both eta and η

We could imagine just dropping all local variables when constructing $\square A$:

 $\frac{\text{TM}/\text{LOCK}?!}{\Box\Gamma \vdash M : A}$ $\frac{\Box\Gamma, \Delta \vdash \text{box}(M) : \Box A}{\Box \Gamma, \Delta \vdash \text{box}(M) : \Box A}$

In this case box(M) cannot commute with substitution:

box(M)[N/x] could be well-typed while box(M[N/x]) is ill-typed!

We can try versions of this rule,¹ but we'll opt for another approach.

¹Prawitz 1967

We'll incorporate Fitch-style judgmental structure² to handle $\square A$:

(Contexts) $\Gamma ::= \cdot | \Gamma, x : A | \Gamma. \square$

Instead of dropping part of the context we can lock it away:

TM/LOCK	TM/VAR	
$\Gamma. \square \vdash M : A$	$\Gamma = \Gamma_0, x : A, \Gamma_1$	$\mathbf{A} \notin \Gamma_1$
$\Gamma \vdash [M]_{\triangleq} : \Box A$	$\Gamma \vdash x : A$	

We'll incorporate Fitch-style judgmental structure² to handle $\Box A$:

(Contexts) $\Gamma ::= \cdot | \Gamma, x : A | \Gamma. \square$

Instead of dropping part of the context we can lock it away:

тм/lock	TM/VAR	
$\Gamma. \square \vdash M : A$	$\Gamma = \Gamma_0, x : A, \Gamma_1$	$\mathbf{\widehat{=}} \notin \Gamma_1$
$\Gamma \vdash [M]_{\blacksquare} : \Box A$	$\Gamma \vdash x : A$	

Crucially, later on we are able to *unlock* the context:

TM/UNLOCK $\Gamma^{\bullet} \vdash M : \Box A$

 $\Gamma \vdash [M]_{\bullet}: A$

²Clouston 2018

We'll incorporate Fitch-style judgmental structure² to handle $\square A$:

(Contexts) $\Gamma ::= \cdot | \Gamma, x : A | \Gamma. \square$

Instead of dropping part of the context we can lock it away:

TM/LOCK	TM/VAR	
Γ . $\square \vdash M : A$	$\Gamma = \Gamma_0, x : A, \Gamma_1$	$\mathbf{A} \notin \Gamma_1$
$\Gamma \vdash [M]_{\textcircled{a}} : \Box A$	$\Gamma \vdash x : A$	

Crucially, later on we are able to *unlock* the context:

TM/UNLOCK $\Gamma^{\bullet} \vdash M : \Box A$

 $\Gamma = \vdash M : \Box A$

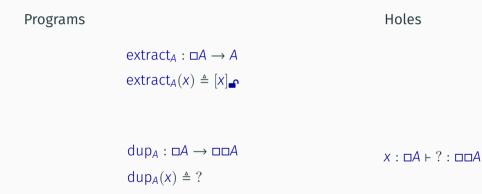
 $\Gamma \vdash [M]_{\bullet}: A$

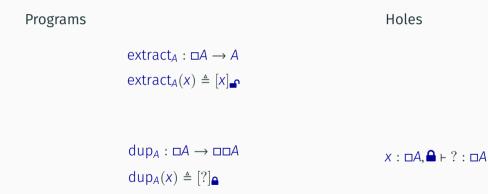
Not obvious, but these rules respect substitution!

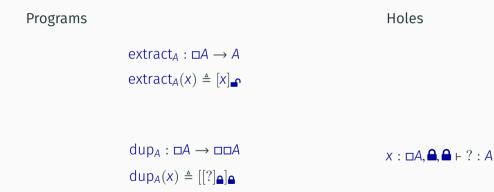
²Clouston 2018

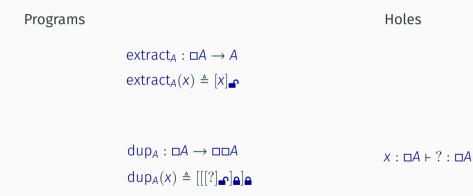
Programs

 $extract_{A} : \Box A \to A$ $extract_{A}(x) \triangleq [x]_{\bullet}$









ProgramsHoles $extract_A : \Box A \rightarrow A$ $extract_A(x) \triangleq [x]$

 $dup_{A}: \Box A \to \Box \Box A$ $dup_{A}(x) \triangleq [[[x]]_{\bullet}]_{\bullet}]_{\bullet}$

Making Hard Choices: Definitional Equalities for MLTT.

We are able to equip $\Box A$ with both a β and η rule in MLTT_a:

TM/UNLOCK-LOCK	TM/LOCK-UNLOCK
$\Gamma^{\bullet}. \square \vdash M : A$	$\Gamma \vdash M : \Box A$
$\Gamma \vdash [[M]_{\blacksquare}]_{\blacksquare} = M : A$	$\Gamma \vdash M = [[M]_{\bullet}]_{\bullet} : \Box A$

Notice, no commutating conversions, this is a win from the Fitch style.³

³Clouston 2018 and Birkedal, Clouston, Mannaa, Møgelberg, Pitts, Spitters 2019

Making Hard Choices: Definitional Equalities for MLTT.

We are able to equip $\Box A$ with both a β and η rule in MLTT_a:

TM/UNLOCK-LOCK	TM/LOCK-UNLOCK
$\Gamma^{\bullet}. \square \vdash M : A$	$\Gamma \vdash M : \Box A$
$\Gamma \vdash [[M]_{\blacksquare}]_{\blacksquare} = M : A$	$\Gamma \vdash M = [[M]_{\bullet}]_{\bullet} : \Box A$

Notice, no commutating conversions, this is a win from the Fitch style.³

The premises of these rules are subtle and important!

 $\Gamma^{\bullet} . \textcircled{\bullet} \vdash M : A \implies \Gamma \vdash M : A$ $\Gamma \vdash [[M]_{\bullet}]_{\textcircled{\bullet}} : \Box A \implies \Gamma \vdash M : \Box A$

³Clouston 2018 and Birkedal, Clouston, Mannaa, Møgelberg, Pitts, Spitters 2019

What do we have at this point?

- MLTT_a: a declarative modal dependent type theory.
- We can prove the expected admissibilities: substitution, presupposition, ...
- As well as modal admissibilities: lock contraction, strengthening...

These are important checks to ensure that MLTT_e behaves well.

What do we have at this point?

Complication: *non-local* and sensitive to extensions.

- MLTT_a: a declarative modal dependent type theory.
- We can prove the expected admissibilities: substitution, presupposition, ...
- As well as modal admissibilities: lock contraction, strengthening...

These are important checks to ensure that MLTT_e behaves well.

What do we have at this point?

- MLTT_a: a declarative modal dependent type theory.
- We can prove the expected admissibilities: substitution, presupposition, ...
- As well as modal admissibilities: lock contraction, strengthening...

These are important checks to ensure that MLTT_{\bullet} behaves well.

Big remaining question: can we implement this?

- 1. Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{\leq} .
- 2. Prove that T admits a *normalization* theorem.
- 3. Conclude that ${\mathbb T}$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\leftrightarrows}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that \mathbb{T}^{r} presents \mathbb{T} and is implementable.

- **1.** Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{\leq} .
- 2. Prove that $\mathbb T$ admits a *normalization* theorem.
- 3. Conclude that ${\mathbb T}$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\leftrightarrows}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that \mathbb{T}^{r} presents \mathbb{T} and is implementable.

- **1.** Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{\leq} .
- 2. Prove that T admits a *normalization* theorem.
- 3. Conclude that $\mathbb T$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\hookrightarrow}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that $\mathbb{T}^{\leftrightarrows}$ presents $\mathbb T$ and is implementable.

- **1.** Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{r} .
- 2. Prove that T admits a *normalization* theorem.
- 3. Conclude that ${\mathbb T}$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\leftrightarrows}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that \mathbb{T}^{r} presents \mathbb{T} and is implementable.

- **1.** Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{\leq} .
- 2. Prove that T admits a *normalization* theorem.
- 3. Conclude that ${\mathbb T}$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\leftrightarrows}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that \mathbb{T}^{r} presents \mathbb{T} and is implementable.

- **1.** Construct a bidirectional syntax for \mathbb{T} : \mathbb{T}^{\leq} .
- 2. Prove that T admits a *normalization* theorem.
- 3. Conclude that ${\mathbb T}$ enjoys decidable conversion.
- 4. Prove that $\mathbb{T}^{\leftrightarrows}$ enjoys decidable type-checking.
- 5. Prove that every term of \mathbb{T} is convertible with a term from $\mathbb{T}^{\leftrightarrows}$.
- 6. Conclude that $\mathbb{T}^{\leftrightarrows}$ presents \mathbb{T} and is implementable.

MLTT_a is a simple enough that we can extend a bidirectional presentation of MLTT:

• Terms are split into two categories:

• We split the judgments along these lines as well:

CHECK SYNTH $\Gamma \vdash M \Leftarrow A \qquad \Gamma \vdash S \Rightarrow A$

• We can extend the standard rules with the new rules for $\square A$:

 $\frac{\Gamma . \textcircled{P} \vdash M \Leftarrow A}{\Gamma \vdash [M]_{\textcircled{P}} \Leftarrow \Box A} \qquad \qquad \frac{\Gamma^{\textcircled{P}} \vdash M \Rightarrow \Box A}{\Gamma \vdash [M]_{\textcircled{P}} \Rightarrow A}$

By restricting MLTT_{\bullet} to MLTT_{\bullet} we can obtain the following result:

Theorem If we can $\Gamma \vdash A = B$ type is decidable^{*} then so are $\Gamma \vdash M \Leftarrow A$ and $\Gamma \vdash M \Rightarrow A$. We've restricted MLTT $\stackrel{\leftarrow}{\Rightarrow}$ so that most one rule applies in each case.

 $^{^{*}}$ We also need whnfs, but this will follow from how we prove the decidability of conversion.

Normalization is a common way to decide equality.

Definition The normalization function has the following type:

 $\underline{\mathbf{norm}}_{\Gamma}^{\mathcal{A}}:\mathbf{Term}_{\Gamma,\mathcal{A}}\to\mathbf{Term}_{\Gamma,\mathcal{A}},$

Completeness:

If
$$\Gamma \vdash M_1 = M_2$$
: A then $\underline{\mathbf{norm}}^{\mathcal{A}}_{\Gamma}(M_1) = \underline{\mathbf{norm}}^{\mathcal{A}}_{\Gamma}(M_2)$.

Soundness:

If $\Gamma \vdash M : A$ then $\Gamma \vdash M = \underline{\mathbf{norm}}^{A}_{\Gamma}(M) : A$

Normalization is a common way to decide equality.

Definition The normalization function has the following type:

 $\underline{\mathbf{norm}}_{\Gamma}^{\mathcal{A}}:\mathbf{Term}_{\Gamma,\mathcal{A}}\to\mathbf{Term}_{\Gamma,\mathcal{A}},$

Completeness:

If
$$\Gamma \vdash M_1 = M_2$$
: A then $\underline{\mathbf{norm}}^{\mathcal{A}}_{\Gamma}(M_1) = \underline{\mathbf{norm}}^{\mathcal{A}}_{\Gamma}(M_2)$.

Soundness:

If $\Gamma \vdash M : A$ then $\Gamma \vdash M = \underline{\mathbf{norm}}_{\Gamma}^{A}(M) : A$

Corollary $\Gamma \vdash M = N : A \iff \underline{\mathbf{norm}}_{\Gamma}^{A}(M) \text{ and } \underline{\mathbf{norm}}_{\Gamma}^{A}(N) \text{ are } \underline{identical.}$

In order to actually define $\underline{\mathbf{norm}}_{\Gamma}^{A}$ we use *normalization-by-evaluation*.⁴

Slogan: evaluate syntax to a computational domain, quote it to a normal form.

- Evaluation performs β -reduction.
- Quotation is *type-directed* and handles η -expansion.
- The algorithm scales to support $\Box A$, even with η .

⁴Martin-Löf 1975, see Abel 2013 for an overview.

Lots of details to balance here, since we also support a full dependent type theory!

Completeness:

Construct a PER model on the computational domain.

Soundness:

Construct a Kripke cross-language logical relation between the computational domain and syntax.

The main sources of complexity are the modality and universes.

Lots of details to balance here, since we also support a full dependent type theory!

Completeness:

Construct a Kripke PER model on the computational domain.

Soundness:

Construct a double Kripke cross-language logical relation between the computational domain and syntax.

The main sources of complexity are the modality and universes.

After normalization we end up with normal forms, but what do these look like?

 $\frac{\Gamma \vdash^{\text{ne}} R : \Pi(A, x.B) \qquad \Gamma \vdash^{\text{nf}} M : A}{\Gamma \vdash^{\text{ne}} R(M) : B[M/x]} \qquad \qquad \frac{\Gamma \vdash^{\text{ne}} S : U_i}{\Gamma \vdash^{\text{nf}} S : U_i}$ $\frac{\Gamma \overset{\bullet}{\bullet} \vdash^{\text{nf}} M : A}{\Gamma \vdash^{\text{nf}} [M]_{\bullet} : \Box A} \qquad \qquad \frac{\Gamma \overset{\bullet}{\bullet} \vdash^{\text{ne}} M : \Box A}{\Gamma \vdash^{\text{ne}} [M]_{\bullet} : A}$

Corollary *There is no term* $\cdot \vdash$ bad : $\Pi(A, \Box A)$

Observe that neutral terms are synthesizable, normal forms are checkable.⁵

⁵Coquand 1996

Theorem (Decidability of Type-Checking)

- Both $\Gamma \vdash M \Rightarrow A$ and $\Gamma \vdash M \Leftarrow A$ are decidable.
- If $\Gamma \vdash M \Rightarrow A \text{ or } \Gamma \vdash M \Leftarrow A \text{ then } \Gamma \vdash M : A.$
- If $\Gamma \vdash M : A$, there exists^{*} some N such that $\Gamma \vdash M = N : A$ and $\Gamma \vdash N \leftarrow A$.

This theorem provides the foundation for our implementation of $MLTT_{\bullet}$. To our knowledge, this is the first such result for MLTT with $\Box A$.

^{*}In particular, $N \triangleq \underline{\mathbf{norm}}_{\Gamma}^{A}(M)$.

We contribute $\mathsf{MLTT}_{{\scriptscriptstyle \!\!\!\! \text{ a}}}$, a dependent type theory with...

- the box modality, $\Box A$
- dependent sums, $\Sigma(A, B)$ } With both β and η
- dependent products, $\Pi(A, B)$
- natural numbers, nat
- intensional identity types, Id(A, M, N)
- a cumulative hierarchy of universes, U₀, U₁...

We have proved the decidability of typechecking for MLTT_a, and implemented it.

http://github.com/jozefg/blott