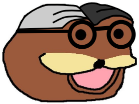
>implying implications can go another direction

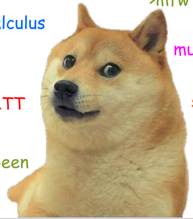
Pierre-Marie Pédrot

INRIA

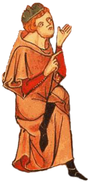
CIRM
09/25



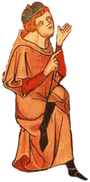
:DDDDialegtiga



On The Menu



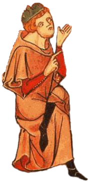
SYNTHETIC MATHEMATICS, LOGIC-AFFINE COMPUTATION
AND EFFICIENT PROOF SYSTEMS



Synthetic mathematics, logic-affine computation
and efficient proof systems

« Vaste programme ! »

On The Menu



SYNTHETIC MATHEMATICS, LOGIC-AFFINE COMPUTATION AND EFFICIENT PROOF SYSTEMS

« Vaste programme ! »

***Synthetic** frameworks have proved to be pivotal tools at the interface of mathematics and informatics, especially enabling concise formalizations and custom proof systems. Noteworthy achievements include homotopy type theory, synthetic **computability** theory, and synthetic algebraic geometry. Very similar paradigms characterize the related areas of logic-driven computational algebra and geometry, sheaf models and **modern realizability** theory, and **strong negation** for constructive reasoning with negative information. Contrasting yet complementary approaches are about to converge, emphasizing the imperative of unifying theoretical underpinnings with practical implementation. With the proposed seminar we aim to extend and deepen the convergence across disciplinary boundaries by fostering exchange and collaboration among experts and practitioners.*

Following The Gradient

Then it dawned upon me...

Following The Gradient

Then it dawned upon me...

The Good Old Gödel Dialectica Talkeroo!

Following The Gradient

Then it dawned upon me...

The Good Old Gödel Dialectica Talkeroo!

... in a somewhat freshened up version.

- Synthetic ✓
- Modern realizability ✓
- Logic-affine ✓
- Strong negation ✓
- ~~Computability~~ Complexity (close enough)

Following The Gradient

Then it dawned upon me...

The Good Old Gödel Dialectica Talkeroo!

... in a somewhat freshened up version.

- Synthetic ✓
- Modern realizability ✓
- Logic-affine ✓
- Strong negation ✓
- ~~Computability~~ Complexity (close enough)

This is a theory talk, no proof systems in sight

But still: `https://github.com/ppedrot/vitef/blob/master/dialectica/cwf.v`

Dialectica

- Shameless recycling: it was the topic of my PhD

In Case of Doubt, Reboot

Dialectica

- Shameless recycling: it was the topic of my PhD
- Advertisement-worthy: it is not taught around enough to my taste

In Case of Doubt, Reboot

Dialectica

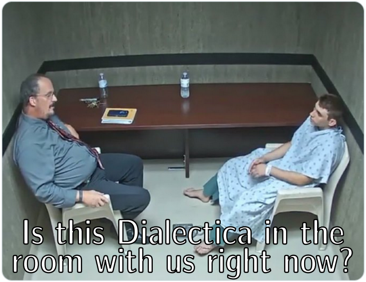
- Shameless recycling: it was the topic of my PhD
- Advertisement-worthy: it is not taught around enough to my taste
- Latent schizophrenia: it keeps appearing everywhere (and nobody sees it)

In Case of Doubt, Reboot

Dialectica

- Shameless recycling: it was the topic of my PhD
- Advertisement-worthy: it is not taught around enough to my taste
- Latent schizophrenia: it keeps appearing everywhere (and nobody sees it)

I am definitely *not* obsessed



In This Talk

A Dialectica model of MLTT

- A syntactic approach
- No crazy categories
- Almost implementable as a strict CwF in your favourite assistant

In This Talk

A Dialectica model of MLTT

- A syntactic approach
- No crazy categories
- Almost implementable as a strict CwF in your favourite assistant

Towards synthetic complexity theory

- Dialectica, the ultimate LL model
- An effectful account of ressources
- Graded types ascended through proof-relevant function space

Part 0

The Dark Ages

A Short, Mostly Wrong History of Dialectica

Dialectica in a nutshell

- Designed by Gödel in the 30's but published in 1958
- The great ancestor of realizability
- The name comes from the journal it was published in
- Basically a fancy realizability model of HA^ω into System T
- *Extremely* kludgy and antiquated

A Short, Mostly Wrong History of Dialectica

Dialectica in a nutshell

- Designed by Gödel in the 30's but published in 1958
- The great ancestor of realizability
- The name comes from the journal it was published in
- Basically a fancy realizability model of HA^ω into System T
- *Extremely* kludgy and antiquated

The More You Know

In *Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes*, Gödel passes more time discussing α-conversion than the actual Dialectica.

A Short, Mostly Wrong History of Dialectica

Dialectica in a nutshell

- Designed by Gödel in the 30's but published in 1958
- The great ancestor of realizability
- The name comes from the journal it was published in
- Basically a fancy realizability model of HA^ω into System T
- *Extremely* kludgy and antiquated

The More You Know

In *Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes*, Gödel passes more time discussing α-conversion than the actual Dialectica.

(You have been warned.)

Anatomy of a Dusty Realizability

If HA$^\omega \vdash A$ then

$$\exists \vec{u}.\, \forall \vec{x}.\, A_D[\vec{u}, \vec{x}]$$

$\mathbb{W}(A)$	$\mathbb{C}(A)$	$\perp_A$
Sequence of types	Sequence of types	$\mathbb{W}(A) \to \mathbb{C}(A) \to \texttt{Prop}$
Witness	Counter	Orthogonality

We Have Come to Realize

The $\exists \vec{u} : \mathbb{W}(A)$ is the realizability part

We Have Come to Realize

The $\exists \vec{u} : \mathbb{W}(A)$ is the realizability part

- On most connectives, Dialectica follows the BHK interpretation.

$$\begin{array}{rcl} \mathbb{W}(A \wedge B) & \sim & \mathbb{W}(A) \times \mathbb{W}(B) \\ \mathbb{W}(A \vee B) & \sim & \mathbb{W}(A) + \mathbb{W}(B) \end{array}$$

We Have Come to Realize

The $\exists \vec{u} : \mathbb{W}(A)$ is the realizability part

- On most connectives, Dialectica follows the BHK interpretation.

$$\begin{array}{rcl} \mathbb{W}(A \wedge B) & \sim & \mathbb{W}(A) \times \mathbb{W}(B) \\ \mathbb{W}(A \vee B) & \sim & \mathbb{W}(A) + \mathbb{W}(B) \end{array}$$

- That is, it is morally the same as Kreisel realizability / MLTT
- i.e. boring intuitionistic semantics
- $\mathbb{C}(A)$ is like $A^{\perp}$ / stack in the KAM / strong negation

We Have Come to Realize

The $\exists \vec{u} : \mathbb{W}(A)$ is the realizability part

- On most connectives, Dialectica follows the BHK interpretation.

$$\begin{array}{rcl} \mathbb{W}(A \wedge B) & \sim & \mathbb{W}(A) \times \mathbb{W}(B) \\ \mathbb{W}(A \vee B) & \sim & \mathbb{W}(A) + \mathbb{W}(B) \end{array}$$

- That is, it is morally the same as Kreisel realizability / MLTT
- i.e. boring intuitionistic semantics
- $\mathbb{C}(A)$ is like $A^{\perp}$ / stack in the KAM / strong negation

... except the one important connective

We Have Come to Realize

The $\exists \vec{u} : \mathbb{W}(A)$ is the realizability part

- On most connectives, Dialectica follows the BHK interpretation.

$$\begin{array}{rcl} \mathbb{W}(A \wedge B) & \sim & \mathbb{W}(A) \times \mathbb{W}(B) \\ \mathbb{W}(A \vee B) & \sim & \mathbb{W}(A) + \mathbb{W}(B) \end{array}$$

- That is, it is morally the same as Kreisel realizability / MLTT
- i.e. boring intuitionistic semantics
- $\mathbb{C}(A)$ is like $A^{\perp}$ / stack in the KAM / strong negation

... except the one important connective

$$A \rightarrow B$$

Implying this Makes Sense

There is more to arrows than just functions!

$$\mathbb{W}(A \to B) \quad = \quad \left\{ \begin{array}{l} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{array} \right.$$

$$\mathbb{C}(A \to B) \quad = \quad \mathbb{W}(A) \times \mathbb{C}(B)$$

Functions have both a **forward** and a **backward** component.

Implying this Makes Sense

There is more to arrows than just functions!

$$\mathbb{W}(A \to B) \quad = \quad \left\{ \begin{array}{l} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{array} \right.$$

$$\mathbb{C}(A \to B) \quad = \quad \mathbb{W}(A) \times \mathbb{C}(B)$$

Functions have both a **forward** and a **backward** component.

$\leadsto$ Ominous Nugget of Wisdom

backward component $=$ **intensional behaviour** of the forward component

Can Dialectica Break Bricks?

Since De Paiva, it is well-known that Dialectica factorizes through LL

Can Dialectica Break Bricks?

Since De Paiva, it is well-known that Dialectica factorizes through LL

Actually, most models of LL are Dialectica-like

- Chu spaces
- Dialectica categories
- Double-glueing

Can Dialectica Break Bricks?

Since De Paiva, it is well-known that Dialectica factorizes through LL

Actually, most models of LL are Dialectica-like

- Chu spaces
- Dialectica categories
- Double-glueing

Nothing new here, nonetheless, it does not hurt to stress it.

Can Dialectica Break Bricks?

Since De Paiva, it is well-known that Dialectica factorizes through LL

Actually, most models of LL are Dialectica-like

- Chu spaces
- Dialectica categories
- Double-glueing

Nothing new here, nonetheless, it does not hurt to stress it.

Empirical Observation

Dialectica is the most natural way to linearize an intuitionistic calculus

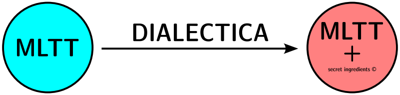
Part I

Dialectica Done Right

Lambda Akbar

Let us present Dialectica as a program translation!

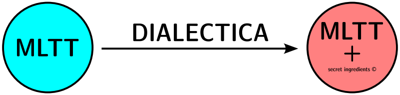
- Replace HA^ω in the source by MLTT
- Replace System T + HA^ω in the target by MLTT (some cheating involved here)
- A variant of Moss-von Glehn model (a.k.a the Google Drive note by A. Bauer and myself)
- Reminiscent of Shulman's model of affine FOL in Chu spaces



Lambda Akbar

Let us present Dialectica as a program translation!

- Replace HA^ω in the source by MLTT
- Replace System T + HA^ω in the target by MLTT (some cheating involved here)
- A variant of Moss-von Glehn model (a.k.a the Google Drive note by A. Bauer and myself)
- Reminiscent of Shulman's model of affine FOL in Chu spaces

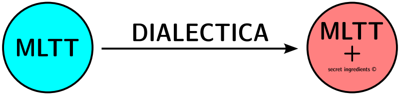


What could possibly go wrong?

Lambda Akbar

Let us present Dialectica as a program translation!

- Replace HA^ω in the source by MLTT
- Replace System T + HA^ω in the target by MLTT (some cheating involved here)
- A variant of Moss-von Glehn model (a.k.a the Google Drive note by A. Bauer and myself)
- Reminiscent of Shulman's model of affine FOL in Chu spaces



What could possibly go wrong?

Exponentially More Complex

Problem #1: MLTT has a notion of definitional equality of proofs!

$\rightsquigarrow$ No such thing in sight for HA^ω...

Exponentially More Complex

Problem #1: MLTT has a notion of definitional equality of proofs!

$\rightsquigarrow$ No such thing in sight for HA^ω...

Basically my PhD

What if we treat the propositional fragment of HA^ω as a λ-calculus?

Exponentially More Complex

Problem #1: MLTT has a notion of definitional equality of proofs!

$\rightsquigarrow$ No such thing in sight for HA^ω...

Basically my PhD

What if we treat the propositional fragment of HA^ω as a λ-calculus?

$$(\lambda x : A.\, M)\ N \qquad \not\equiv \qquad M\{x := N\}$$

Oopsie

Dialectica breaks the equational theory immediately

Exponentially More Complex

Problem #1: MLTT has a notion of definitional equality of proofs!

$\rightsquigarrow$ No such thing in sight for HA^ω...

Basically my PhD

What if we treat the propositional fragment of HA^ω as a λ-calculus?

$$(\lambda x : A.\, M)\ N \qquad \not\equiv \qquad M\{x := N\}$$

Oopsie

Dialectica breaks the equational theory immediately

😱 The original Dialectica is **not** a program translation 😱

One Semiring To Bind Them All

We must introduce an additional structure $\mathfrak{M}$ called **abstract multisets**

- A generalization of finite multisets
- Clearly there to handle LL exponentials

One Semiring To Bind Them All

We must introduce an additional structure $\mathfrak{M}$ called **abstract multisets**

- A generalization of finite multisets
- Clearly there to handle LL exponentials

Behold the Diller-Nahm interpretation

$$\mathbb{W}(A \to B) \quad = \quad \left\{ \begin{array}{l} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathfrak{M}\ \mathbb{C}(A) \end{array} \right.$$

Before: $\mathbb{W}(A \to B) \quad = \quad \left\{ \begin{array}{l} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{array} \right.$

Categorical Nonsense

An abstract multiset structure is just a **semiringoid**!

- A monad $\mathfrak{M}$ with return and bind:

$$\{\cdot\} : A \to \mathfrak{M}\ A \qquad \gg\!= : \mathfrak{M}\ A \to (A \to \mathfrak{M}\ B) \to \mathfrak{M}\ B$$

- that further has a commutative monoid structure:

$$\varnothing : \mathfrak{M}\ A \qquad \oplus : \mathfrak{M}\ A \to \mathfrak{M}\ A \to \mathfrak{M}\ A$$

Categorical Nonsense

An abstract multiset structure is just a **semiringoid**!

- A monad $\mathfrak{M}$ with return and bind:

$$\{\cdot\} : A \to \mathfrak{M}\ A \qquad \gg\!= : \mathfrak{M}\ A \to (A \to \mathfrak{M}\ B) \to \mathfrak{M}\ B$$

- that further has a commutative monoid structure:

$$\varnothing : \mathfrak{M}\ A \qquad \oplus : \mathfrak{M}\ A \to \mathfrak{M}\ A \to \mathfrak{M}\ A$$

Trigger Warning

If you forget the parameter, equations are those of a semiring

Categorical Nonsense

An abstract multiset structure is just a **semiringoid**!

- A monad $\mathfrak{M}$ with return and bind:

$$\{\cdot\} : A \to \mathfrak{M}\ A \qquad \gg\!= : \mathfrak{M}\ A \to (A \to \mathfrak{M}\ B) \to \mathfrak{M}\ B$$

- that further has a commutative monoid structure:

$$\varnothing : \mathfrak{M}\ A \qquad \oplus : \mathfrak{M}\ A \to \mathfrak{M}\ A \to \mathfrak{M}\ A$$

Trigger Warning

If you forget the parameter, equations are those of a semiring

$$\{M\} \gg\!= F \equiv F\ M \qquad M \gg\!= (\lambda x.\ \{x\}) \equiv M$$

$$(M \gg\!= F) \gg\!= G \equiv M \gg\!= (\lambda x.\ F\ x \gg\!= G)$$

$$\varnothing \oplus M \equiv M \qquad M \oplus \varnothing \equiv M \qquad (M \oplus N) \oplus P \equiv M \oplus (N \oplus P)$$

$$\varnothing \gg\!= F \equiv \varnothing \qquad M \gg\!= (\lambda x.\ \varnothing) \equiv \varnothing$$

$$(M \oplus N) \gg\!= F \equiv (M \gg\!= F) \oplus (N \gg\!= F) \quad M \gg\!= (\lambda x.\ F \oplus G) \equiv (M \gg\!= F) \oplus (M \gg\!= G)$$

Categorical Nonsense

An abstract multiset structure is just a **semiringoid**!

- A monad $\mathfrak{M}$ with return and bind:

$$\{\cdot\} : A \to \mathfrak{M}\ A \qquad \gg\!= : \mathfrak{M}\ A \to (A \to \mathfrak{M}\ B) \to \mathfrak{M}\ B$$

- that further has a commutative monoid structure:

$$\varnothing : \mathfrak{M}\ A \qquad \oplus : \mathfrak{M}\ A \to \mathfrak{M}\ A \to \mathfrak{M}\ A$$

Trigger Warning

If you forget the parameter, equations are those of a semiring

$$\{M\} \gg\!= F \equiv F\ M \qquad M \gg\!= (\lambda x.\ \{x\}) \equiv M$$

$$(M \gg\!= F) \gg\!= G \equiv M \gg\!= (\lambda x.\ F\ x \gg\!= G)$$

$$\varnothing \oplus M \equiv M \qquad M \oplus \varnothing \equiv M \qquad (M \oplus N) \oplus P \equiv M \oplus (N \oplus P)$$

$$\varnothing \gg\!= F \equiv \varnothing \qquad M \gg\!= (\lambda x.\ \varnothing) \equiv \varnothing$$

$$(M \oplus N) \gg\!= F \equiv (M \gg\!= F) \oplus (N \gg\!= F) \quad M \gg\!= (\lambda x.\ F \oplus G) \equiv (M \gg\!= F) \oplus (M \gg\!= G)$$

Prototypical example: finite multisets

Simpler? It Depends

Problem #2: We must handle dependency somehow

Simpler? It Depends

Problem #2: We must handle dependency somehow

The YOLO principle

Just make everything dependent!

Simpler? It Depends

Problem #2: We must handle dependency somehow

The YOLO principle

Just make everything dependent!

It is actually **simpler** in MLTT than in System T

A clear case where more expressivity helps

If There is No Problem there is No Solution

Remember that in standard Dialectica, a type A is converted to

$$\text{a type } \mathbb{W}(A) \quad \text{a type } \mathbb{C}(A) \quad \text{a relation } \perp_A \subseteq \mathbb{W}(A) \times \mathbb{C}(A)$$

If There is No Problem there is No Solution

Remember that in standard Dialectica, a type A is converted to

$$\text{a type } \mathbb{W}(A) \quad \text{a type } \mathbb{C}(A) \quad \text{a relation } \perp_A \subseteq \mathbb{W}(A) \times \mathbb{C}(A)$$

Thanks to the added expressivity, we can conflate $\mathbb{C}(A)$ and $\perp_A$

An MLTT type A will be converted to

- a type $\mathbb{W}(A)$
- a dependent type $\mathbb{C}(A) : \mathbb{W}(A) \to \mathtt{Type}$

No need for orthogonality, it is built into $\mathbb{C}$!

If There is No Problem there is No Solution

Remember that in standard Dialectica, a type A is converted to

$$\text{a type } \mathbb{W}(A) \quad \text{a type } \mathbb{C}(A) \quad \text{a relation } \perp_A \subseteq \mathbb{W}(A) \times \mathbb{C}(A)$$

Thanks to the added expressivity, we can conflate $\mathbb{C}(A)$ and $\perp_A$

An MLTT type A will be converted to

- a type $\mathbb{W}(A)$
- a dependent type $\mathbb{C}(A) : \mathbb{W}(A) \rightarrow \texttt{Type}$

No need for orthogonality, it is built into $\mathbb{C}$!

REMARK: for readability I will write $\mathbb{C}(A)\langle M \rangle$ for $\mathbb{C}(A)\ M$.

The Translation, At Last

Assume $\Gamma \vdash M : A$ in MLTT.

The Translation, At Last

Assume $\Gamma \vdash M : A$ in MLTT.

The Dialectica translation will produce **two** kinds of objects.

The Translation, At Last

Assume $\Gamma \vdash M : A$ in MLTT.

The Dialectica translation will produce **two** kinds of objects.

The **forward** translation:

$$\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$$

The Translation, At Last

Assume $\Gamma \vdash M : A$ in MLTT.

The Dialectica translation will produce **two** kinds of objects.

The **forward** translation:

$$\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$$

For **each** $x : X \in \Gamma$, a **reverse** translation:

$$\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle [M] \rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x \rangle$$

Reminder:

$$\mathfrak{M} : \mathtt{Type} \to \mathtt{Type} \qquad \mathbb{W}(A) : \mathtt{Type} \qquad \mathbb{C}(A) : \mathbb{W}(A) \to \mathtt{Type}$$

The Translation, At Last

Assume $\Gamma \vdash M : A$ in MLTT.

The Dialectica translation will produce **two** kinds of objects.

The **forward** translation:

$$\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$$

For **each** $x : X \in \Gamma$, a **reverse** translation:

$$\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle [M] \rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x \rangle$$

Reminder:

$$\mathfrak{M} : \mathtt{Type} \to \mathtt{Type} \qquad \mathbb{W}(A) : \mathtt{Type} \qquad \mathbb{C}(A) : \mathbb{W}(A) \to \mathtt{Type}$$

If $\|\Gamma\| = n$, this means I have $n + 1$ objects!

Motto

$[M]_x$ tracks the uses of x in M

$$\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$$

$[M]_x$ tracks the uses of x in M

$$\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$$

In particular,

- $[x]_x := \lambda\pi.\,\{\pi\}$ (dereliction)
- $[y]_x := \lambda\pi.\,\varnothing$ if $x \neq y$ (weakening)

noting that

$$[x] := x$$

(Reminder: $\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$)

$[M]_x$ tracks the uses of x in M

$$\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \rightarrow \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$$

In particular,

- $[x]_x := \lambda\pi.\,\{\pi\}$ (dereliction)
- $[y]_x := \lambda\pi.\,\varnothing$ if $x \neq y$ (weakening)

noting that

$$[x] := x$$

(Reminder: $\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$)

Substitution lemma (contraction + promotion)

$$[M\{x := N\}]_y\ \pi \quad\equiv\quad ([M]_y\{x := [N]\}\ \pi) \ \oplus\ ([M]_x\{x := [N]\}\ \pi \ggg [N]_y)$$

So I Herd U Liek CwF

This also makes sense algebraically for CwF substitutions

So I Herd U Liek CwF

This also makes sense algebraically for CwF substitutions

A substitution $\sigma : \Gamma \vdash \Delta$ has both

- A forward map $\sigma^+ : \mathbb{W}(\Gamma) \vdash \mathbb{W}(\Delta)$
- A reverse map $\sigma^- : \Pi(\gamma : \mathbb{W}(\Gamma)).\, \mathbb{C}(\Delta)\langle \sigma^+ \cdot \gamma \rangle \vdash \mathbb{C}(\Gamma)\langle \gamma \rangle$

where $\mathbb{C}(\Gamma), \mathbb{C}(\Delta)$ are $\mathfrak{M}$-algebras and σ^- an algebra morphism.

So I Herd U Liek CwF

This also makes sense algebraically for CwF substitutions

A substitution $\sigma : \Gamma \vdash \Delta$ has both

- A forward map $\sigma^+ : \mathbb{W}(\Gamma) \vdash \mathbb{W}(\Delta)$
- A reverse map $\sigma^- : \Pi(\gamma : \mathbb{W}(\Gamma)).\, \mathbb{C}(\Delta)\langle\sigma^+ \cdot \gamma\rangle \vdash \mathbb{C}(\Gamma)\langle\gamma\rangle$

where $\mathbb{C}(\Gamma), \mathbb{C}(\Delta)$ are $\mathfrak{M}$-algebras and σ^- an algebra morphism.

The $\mathfrak{M}$ operations are reflected straightforwardly

- Weakening $w : \Gamma, A \vdash \Gamma$ is $\varnothing$
- Duplication $d : \Gamma \vdash \Gamma, \Gamma$ is $\oplus$

So I Herd U Liek CwF

This also makes sense algebraically for CwF substitutions

A substitution $\sigma : \Gamma \vdash \Delta$ has both

- A forward map $\sigma^+ : \mathbb{W}(\Gamma) \vdash \mathbb{W}(\Delta)$
- A reverse map $\sigma^- : \Pi(\gamma : \mathbb{W}(\Gamma)).\, \mathbb{C}(\Delta)\langle \sigma^+ \cdot \gamma \rangle \vdash \mathbb{C}(\Gamma)\langle \gamma \rangle$

where $\mathbb{C}(\Gamma), \mathbb{C}(\Delta)$ are $\mathfrak{M}$-algebras and σ^- an algebra morphism.

The $\mathfrak{M}$ operations are reflected straightforwardly

- Weakening $w : \Gamma, A \vdash \Gamma$ is $\varnothing$
- Duplication $d : \Gamma \vdash \Gamma, \Gamma$ is $\oplus$

With Dialectica, substitution is **the** effect!

A Functional Functional Interpretation

There is no structure in our theory yet, let's look at functions!

$$\mathbb{W}(\Pi(x : A).\, B) \quad := \left\{ \begin{array}{l} f \;: \Pi(x : \mathbb{W}(A)).\, \mathbb{W}(B) \\ \varphi : \Pi(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle \to \mathfrak{M}\ \mathbb{C}(A)\langle x\rangle \end{array} \right\}$$

$$\mathbb{C}(\Pi(x : A).\, B)\langle f, \varphi\rangle := \Sigma(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle$$

A Functional Functional Interpretation

There is no structure in our theory yet, let's look at functions!

$$\mathbb{W}(\Pi(x : A).\, B) \quad := \left\{ \begin{array}{l} f \;: \Pi(x : \mathbb{W}(A)).\, \mathbb{W}(B) \\ \varphi : \Pi(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle \to \mathfrak{M}\ \mathbb{C}(A)\langle x\rangle \end{array} \right\}$$

$$\mathbb{C}(\Pi(x : A).\, B)\langle f, \varphi\rangle := \Sigma(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle$$

Up to dependent noise, this is the same as the simply typed variant!

A Functional Functional Interpretation

There is no structure in our theory yet, let's look at functions!

$$\mathbb{W}(\Pi(x : A).\, B) \quad := \left\{ \begin{array}{l} f \;: \Pi(x : \mathbb{W}(A)).\, \mathbb{W}(B) \\ \varphi : \Pi(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle \to \mathfrak{M}\ \mathbb{C}(A)\langle x\rangle \end{array} \right\}$$

$$\mathbb{C}(\Pi(x : A).\, B)\langle f, \varphi\rangle := \Sigma(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x\rangle$$

Up to dependent noise, this is the same as the simply typed variant!

Forward translations are almost the identity.

$$\begin{array}{lcl} [\lambda x.\, M] & := & (\lambda x.\, [M]), (\lambda x.\, [M]_x) \\ [M\ N] & := & [M].1\ [N] \end{array}$$

Recall: if $\Gamma \vdash M : A$ then $\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle [M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$ provided $x : X \in \Gamma$

Full Reverse

Reverse translations are quite a mouthful.

Full Reverse

Reverse translations are quite a mouthful.

$[M]_x$ tracks the uses of x in M

Full Reverse

Reverse translations are quite a mouthful.

$[M]_x$ tracks the uses of x in M

$$[\lambda y.\, M]_x := \lambda(y, \pi).\, [M]_x\ \pi$$

$$[M\ N]_x \ := \lambda\pi. \quad ([M]_x\ ([N], \pi)) \ \oplus\ ([M].2\ [N]\ \pi \ggg [N]_x)$$

Cheat Sheet

1. If $\Gamma \vdash M : A$ then $\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle [M] \rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x \rangle$ provided $x : X \in \Gamma$.
2. $\mathbb{W}(\Pi(x : A).\, B) \qquad := \left\{ \begin{array}{l} f\ : \Pi(x : \mathbb{W}(A)).\, \mathbb{W}(B) \\ \varphi : \Pi(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x \rangle \to \mathfrak{M}\ \mathbb{C}(A)\langle x \rangle \end{array} \right\}$
3. $\mathbb{C}(\Pi(x : A).\, B)\langle f, \varphi \rangle := \Sigma(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x \rangle$

Full Reverse

Reverse translations are quite a mouthful.

$[M]_x$ tracks the uses of x in M

$$[\lambda y.\, M]_x := \lambda(y, \pi).\, [M]_x\ \pi$$

$$[M\ N]_x \ := \lambda \pi. \quad ([M]_x\ ([N], \pi)) \ \oplus \ ([M].2\ [N]\ \pi \ggg [N]_x)$$

Cheat Sheet

1. If $\Gamma \vdash M : A$ then $\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle [M] \rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x \rangle$ provided $x : X \in \Gamma$.
2. $\mathbb{W}(\Pi(x : A).\, B) \quad := \left\{ \begin{array}{l} f \ : \Pi(x : \mathbb{W}(A)).\, \mathbb{W}(B) \\ \varphi : \Pi(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x \rangle \to \mathfrak{M}\ \mathbb{C}(A)\langle x \rangle \end{array} \right\}$
3. $\mathbb{C}(\Pi(x : A).\, B)\langle f, \varphi \rangle := \Sigma(x : \mathbb{W}(A)).\, \mathbb{C}(B)\langle f\ x \rangle$

Application basically follows the substitution lemma.

Positive Attitude

I have been bluffing you: a dependent $\mathbb{C}$ is only needed for positive types.

$\rightsquigarrow$ most of the dependent noise so far has been useless.

Positive Attitude

I have been bluffing you: a dependent $\mathbb{C}$ is only needed for positive types.

$\rightsquigarrow$ most of the dependent noise so far has been useless.

I don't want to enter details, here is a quick sketch of sum types:

$$\mathbb{W}(A + B) \quad := \quad \mathbb{W}(A) + \mathbb{W}(B)$$

Positive Attitude

I have been bluffing you: a dependent $\mathbb{C}$ is only needed for positive types.

$\rightsquigarrow$ most of the dependent noise so far has been useless.

I don't want to enter details, here is a quick sketch of sum types:

$$
\begin{array}{lcl}
\mathbb{W}(A + B) & := & \mathbb{W}(A) + \mathbb{W}(B) \\
\mathbb{C}(A + B)\langle \texttt{inl}\ M \rangle & := & \mathbb{C}(A)\langle M \rangle \\
\mathbb{C}(A + B)\langle \texttt{inr}\ N \rangle & := & \mathbb{C}(B)\langle N \rangle
\end{array}
$$

Positive Attitude

I have been bluffing you: a dependent $\mathbb{C}$ is only needed for positive types.

$\rightsquigarrow$ most of the dependent noise so far has been useless.

I don't want to enter details, here is a quick sketch of sum types:

$$\begin{array}{lcl} \mathbb{W}(A+B) & := & \mathbb{W}(A) + \mathbb{W}(B) \\ \mathbb{C}(A+B)\langle \mathtt{inl}\ M\rangle & := & \mathbb{C}(A)\langle M\rangle \\ \mathbb{C}(A+B)\langle \mathtt{inr}\ N\rangle & := & \mathbb{C}(B)\langle N\rangle \end{array}$$

$\rightsquigarrow$ term translation follows the leading MOTTO

$[M]_x$ tracks the uses of x in M

Positive Attitude

I have been bluffing you: a dependent $\mathbb{C}$ is only needed for positive types.

$\rightsquigarrow$ most of the dependent noise so far has been useless.

I don't want to enter details, here is a quick sketch of sum types:

$$\begin{array}{lcl} \mathbb{W}(A+B) & := & \mathbb{W}(A) + \mathbb{W}(B) \\ \mathbb{C}(A+B)\langle \texttt{inl}\ M\rangle & := & \mathbb{C}(A)\langle M\rangle \\ \mathbb{C}(A+B)\langle \texttt{inr}\ N\rangle & := & \mathbb{C}(B)\langle N\rangle \end{array}$$

$\rightsquigarrow$ term translation follows the leading **Motto**

$[M]_x$ tracks the uses of x in M

We get inductive types with large elimination

- … and also indices!
- The model preserves canonicity

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

$$\begin{array}{lcl} \mathbb{W}(\mathtt{Type}) & := & \{\ \mathbf{W} : \mathtt{Type};\ \mathbf{C} : \mathbf{W} \to \mathtt{Type}\ \} \\ \mathbb{C}(\mathtt{Type})\langle \mathbf{W}, \mathbf{C} \rangle & := & \text{whatever} \end{array}$$

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

$$\begin{array}{lcl}\mathbb{W}(\mathtt{Type}) & := & \{\ \mathbf{W} : \mathtt{Type};\ \ \mathbf{C} : \mathbf{W} \rightarrow \mathtt{Type}\ \} \\ \mathbb{C}(\mathtt{Type})\langle \mathbf{W}, \mathbf{C} \rangle & := & \text{whatever}\end{array}$$

$$\mathbb{W}(A) := [A].\mathbf{W} \qquad \mathbb{C}(A) := [A].\mathbf{C}$$

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

$$\begin{array}{lcl} \mathbb{W}(\mathtt{Type}) & := & \{\ \mathbf{W} : \mathtt{Type};\ \mathbf{C} : \mathbf{W} \rightarrow \mathtt{Type}\ \} \\ \mathbb{C}(\mathtt{Type})\langle \mathbf{W}, \mathbf{C} \rangle & := & \text{whatever} \end{array}$$

$$\mathbb{W}(A) := [A].\mathbf{W} \qquad \mathbb{C}(A) := [A].\mathbf{C}$$

$\rightsquigarrow$ for the forward translation of a type A we have little choice

$$[A] := (\mathbb{W}(A), \mathbb{C}(A))$$

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

$$\begin{array}{lcl} \mathbb{W}(\texttt{Type}) & := & \{\ \mathbf{W} : \texttt{Type};\ \mathbf{C} : \mathbf{W} \to \texttt{Type}\ \} \\ \mathbb{C}(\texttt{Type})\langle \mathbf{W}, \mathbf{C} \rangle & := & \text{whatever} \end{array}$$

$$\mathbb{W}(A) := [A].\mathbf{W} \qquad \mathbb{C}(A) := [A].\mathbf{C}$$

$\rightsquigarrow$ for the forward translation of a type A we have little choice

$$[A] := (\mathbb{W}(A), \mathbb{C}(A))$$

$\rightsquigarrow$ for the reverse translation, no constraints (“types do not use resources”)

$$[A]_x := \lambda \pi . \varnothing$$

The Universal Property of Universes

As usual in effectful models of MLTT, types-as-terms are underspecified

$$\begin{array}{lcl} \mathbb{W}(\mathtt{Type}) & := & \{\ \mathbf{W} : \mathtt{Type};\ \mathbf{C} : \mathbf{W} \to \mathtt{Type}\ \} \\ \mathbb{C}(\mathtt{Type})\langle \mathbf{W}, \mathbf{C} \rangle & := & \text{whatever} \end{array}$$

$$\mathbb{W}(A) := [A].\mathbf{W} \qquad \mathbb{C}(A) := [A].\mathbf{C}$$

$\rightsquigarrow$ for the forward translation of a type A we have little choice

$$[A] := (\mathbb{W}(A), \mathbb{C}(A))$$

$\rightsquigarrow$ for the reverse translation, no constraints (“types do not use resources”)

$$[A]_x := \lambda \pi . \varnothing$$

A lot of somewhat arbitrary choices

Trust me

We have defined a model of MLTT.

If $\Gamma \vdash M : A$,

- $\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$
- $\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$ for all $x : X \in \Gamma$

and similarly for conversion.

Trust me

We have defined a model of MLTT.

If $\Gamma \vdash M : A$,

- $\mathbb{W}(\Gamma) \vdash [M] : \mathbb{W}(A)$
- $\mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$ for all $x : X \in \Gamma$

and similarly for conversion.

This can be presented as an almost strict CwF

Part II
What Have I Got In My Pocket?

Synthetic A Posteriori

As expected, some features of LL

$A \otimes B$ v.s. $A \,\&\, B$

Synthetic A Posteriori

As expected, some features of LL

$$A \otimes B \text{ v.s. } A \,\&\, B$$

Some kind of internal exponential $!A$

Essentially $\quad \texttt{Inductive}\ !A := \texttt{box} : A \to !A$

- $\Pi(x : !A).\, \texttt{box}\ (\texttt{prj}\ x) = x$ and $\Pi(x : A).\, \texttt{prj}\ (\texttt{box}\ x) = x$
- $\texttt{prj} \circ \texttt{box} \equiv \texttt{id}$

Synthetic A Posteriori

As expected, some features of LL

$$A \otimes B \text{ v.s. } A \,\&\, B$$

Some kind of internal exponential $!A$

Essentially $\quad \texttt{Inductive } !A := \texttt{box} : A \to !A$

- $\Pi(x : !A).\, \texttt{box}\ (\texttt{prj}\ x) = x \quad$ and $\quad \Pi(x : A).\, \texttt{prj}\ (\texttt{box}\ x) = x$
- $\texttt{prj} \circ \texttt{box} \equiv \texttt{id}$
- but $\texttt{box} \circ \texttt{prj} \neq \texttt{id}$

Synthetic A Posteriori

As expected, some features of LL

$$A \otimes B \text{ v.s. } A \mathbin{\&} B$$

Some kind of internal exponential $!A$

$$\text{Essentially} \quad \texttt{Inductive}\ !A := \texttt{box} : A \rightarrow\ !A$$

- $\Pi(x : !A).\, \texttt{box}\ (\texttt{prj}\ x) = x$ and $\Pi(x : A).\, \texttt{prj}\ (\texttt{box}\ x) = x$
- $\texttt{prj} \circ \texttt{box} \equiv \texttt{id}$
- but $\texttt{box} \circ \texttt{prj} \neq \texttt{id}$

The model breaks `funext`!

$$\vdash 1 \rightarrow 1 \quad \cong \quad \mathfrak{M}\, 1 \quad \cong \quad \mathbb{N}$$

Synthetic A Posteriori

As expected, some features of LL

$$A \otimes B \text{ v.s. } A \mathbin{\&} B$$

Some kind of internal exponential $!A$

Essentially $\quad \texttt{Inductive}\ !A := \texttt{box} : A \to\ !A$

- $\Pi(x :\ !A).\, \texttt{box}\ (\texttt{prj}\ x) = x$ and $\Pi(x : A).\, \texttt{prj}\ (\texttt{box}\ x) = x$
- $\texttt{prj} \circ \texttt{box} \equiv \texttt{id}$
- but $\texttt{box} \circ \texttt{prj} \not\equiv \texttt{id}$

The model breaks `funext`!

$$\vdash 1 \to 1 \quad \cong \quad \mathfrak{M}\, 1 \quad \cong \quad \mathbb{N}$$

Can this be of any use?

And Now For Something Completely Different

People in type theory want to track resources

- for efficiency
- for security / privacy
- for the love of LL

And Now For Something Completely Different

People in type theory want to track resources

- for efficiency
- for security / privacy
- for the love of LL

Various names for somewhat similar systems

- Graded type theories (à la Orchard)
- Quantitative type theories (à la McBride-Atkey)

And Now For Something Completely Different

People in type theory want to track resources

- for efficiency
- for security / privacy
- for the love of LL

Various names for somewhat similar systems

- Graded type theories (à la Orchard)
- Quantitative type theories (à la McBride-Atkey)

As a first-order approximation I will lump them together here

- I will gloss over the differences
- and deliberately ignore some technicalities

Get in the Ring

Typically, fix a **resource** semi-ring $(\mathbb{M}, +, \times)$.

$\rightsquigarrow$ used to assign a measure to variable use

- 0 means not used
- 1 is exactly once

Get in the Ring

Typically, fix a **resource** semi-ring $(\mathbb{M}, +, \times)$.

$\rightsquigarrow$ used to assign a measure to variable use

- 0 means not used
- 1 is exactly once

$\rightsquigarrow$ you tend to expect a little bit more structure

- an order $\leq$ compatible with the semi-ring operations
- often a maximal absorbing element ω
- sometimes a max operation $\sqcup$

Get in the Ring

Typically, fix a **resource** semi-ring $(\mathbb{M}, +, \times)$.

$\rightsquigarrow$ used to assign a measure to variable use

- 0 means not used
- 1 is exactly once

$\rightsquigarrow$ you tend to expect a little bit more structure

- an order $\leq$ compatible with the semi-ring operations
- often a maximal absorbing element ω
- sometimes a max operation $\sqcup$

Simplest example is $0 \leq 1 \leq \omega$.

Varying Variables

Variables are annotated by the semi-ring

$$x_1 :_{\alpha_1} X_1, \ldots x :_{\alpha_n} X_n \vdash M : A$$

where $\alpha_i : \mathbb{M}$

Varying Variables

Variables are annotated by the semi-ring

$$x_1 :_{\alpha_1} X_1, \ldots x :_{\alpha_n} X_n \vdash M : A$$

where $\alpha_i : \mathbb{M}$

$\rightsquigarrow$ the semi-ring structure lifts to contexts naturally.

Varying Variables

Variables are annotated by the semi-ring

$$x_1 :_{\alpha_1} X_1, \ldots x :_{\alpha_n} X_n \vdash M : A$$

where $\alpha_i : \mathbb{M}$

$\rightsquigarrow$ the semi-ring structure lifts to contexts naturally.

Types and terms are annotated accordingly

$$A, B ::= \ldots \mid A \to_\alpha B \mid \ldots$$

Apply Copiously

Some typing rules are expected

$$\frac{}{0\Gamma, x :_1 A \vdash x : A} \qquad \frac{\Gamma, x :_\alpha A \vdash M : B}{\Gamma \vdash \lambda x.\, M : A \to_\alpha B}$$

Apply Copiously

Some typing rules are expected

$$\frac{}{0\Gamma, x :_1 A \vdash x : A} \qquad \frac{\Gamma, x :_\alpha A \vdash M : B}{\Gamma \vdash \lambda x.\, M : A \to_\alpha B}$$

More interesting is the case of application!

$$\frac{\Gamma \vdash M : A \to_\alpha B \qquad \Delta \vdash N : A}{\Gamma + (\alpha \times \Delta) \vdash M\, N : B}$$

This is really where the semi-ring structure shines!

The Sad Truth

Why are graded types too limited?

The Sad Truth

Why are graded types too limited?

The semi-ring gives very little expressivity!

- Poor language of annotations
- Despairingly simple-typed
- Often needs to overapproximate
- Not very modular

The Sad Truth

Why are graded types too limited?

The semi-ring gives very little expressivity!

- Poor language of annotations
- Despairingly simple-typed
- Often needs to overapproximate
- Not very modular

Let me give some examples of increasing annoyance

There is no Alternative

Problem (easy)

How to type: `if` M `then` N_1 `else` N_2

There is no Alternative

Problem (easy)

How to type: `if` M `then` N_1 `else` N_2

Solution use max $\{N_1, N_2\}$ and hope that subtyping will save you

There is no Alternative

Problem (easy)

How to type: $\texttt{if } M \texttt{ then } N_1 \texttt{ else } N_2$

Solution use max $\{N_1, N_2\}$ and hope that subtyping will save you

Overapproximation: it actually depends on M

We would like an element $\alpha \in \mathbb{M} := \texttt{if } M \texttt{ then } \alpha_1 \texttt{ else } \alpha_2$

There is no Alternative

Problem (easy)

How to type: `if` M `then` N_1 `else` N_2

Solution use $\max \{N_1, N_2\}$ and hope that subtyping will save you

Overapproximation: it actually depends on M

We would like an element $\alpha \in \mathbb{M} :=$ `if` M `then` α_1 `else` α_2

Problem (hard)

Same question with a recursion on $\mathbb{N}$ rather than $\mathbb{B}$

There is no Alternative

Problem (easy)

How to type: $\texttt{if}\ M\ \texttt{then}\ N_1\ \texttt{else}\ N_2$

Solution use max $\{N_1, N_2\}$ and hope that subtyping will save you

Overapproximation: it actually depends on M

We would like an element $\alpha \in \mathbb{M} := \texttt{if}\ M\ \texttt{then}\ \alpha_1\ \texttt{else}\ \alpha_2$

Problem (hard)

Same question with a recursion on $\mathbb{N}$ rather than $\mathbb{B}$

Solution require the semiring to be a countably complete lattice ☹

I Want My Functions Back

Problem (unsolvable)

It is actually worse: HO functions are basically broken

$$(A \to_\alpha B) \to_\beta C$$

I Want My Functions Back

Problem (unsolvable)

It is actually worse: HO functions are basically broken

$$(A \to_\alpha B) \to_\beta C$$

The number of uses α is fixed once and for all

I Want My Functions Back

Problem (unsolvable)

It is actually worse: HO functions are basically broken

$$(A \to_\alpha B) \to_\beta C$$

The number of uses α is fixed once and for all

- This completely prevents a fine analysis of its resources
- What happens when the number of calls is not static?

I Want My Functions Back

Problem (unsolvable)

It is actually worse: HO functions are basically broken

$$(A \to_\alpha B) \to_\beta C$$

The number of uses α is fixed once and for all

- This completely prevents a fine analysis of its resources
- What happens when the number of calls is not static?

What happens in practice: (a.k.a. fancy subtyping doesn't work)

- $0 \rightsquigarrow$ runtime irrelevant terms (e.g. types)
- $1 \rightsquigarrow$ extremely rare cases (e.g. linear state-passing)
- $\omega \rightsquigarrow$ all other cases

I Want My Functions Back

Problem (unsolvable)

It is actually worse: HO functions are basically broken

$$(A \to_\alpha B) \to_\beta C$$

The number of uses α is fixed once and for all

- This completely prevents a fine analysis of its resources
- What happens when the number of calls is not static?

What happens in practice: (a.k.a. fancy subtyping doesn't work)

- $0 \rightsquigarrow$ runtime irrelevant terms (e.g. types)
- $1 \rightsquigarrow$ extremely rare cases (e.g. linear state-passing)
- $\omega \rightsquigarrow$ all other cases

You have just reinvented a single runtime irrelevant modality

Obvious Progaganda is Obvious

To be fair, there have been proposals to extend the expressivity

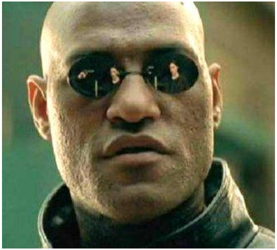
... but in my opinion these are very contrived.

Obvious Progaganda is Obvious

To be fair, there have been proposals to extend the expressivity

... but in my opinion these are very contrived.

What if I told you



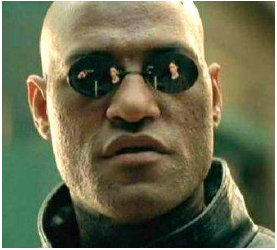
we already have a solution?

Obvious Progaganda is Obvious

To be fair, there have been proposals to extend the expressivity

… but in my opinion these are very contrived.

What if I told you



we already have a solution?

DIALECTICA

(what a surprise!)

The Main Idea

The semi-ring annotations should depend on terms

This is literally the only reason why we had to overapproximate.

The Main Idea

The semi-ring annotations should depend on terms

This is literally the only reason why we had to overapproximate.

- A function should tell you **dynamically** its argument measure
- Pattern-matching measure should depend on the scrutinee
- This clearly requires dependent types

The Main Idea

The semi-ring annotations should depend on terms

This is literally the only reason why we had to overapproximate.

- A function should tell you **dynamically** its argument measure
- Pattern-matching measure should depend on the scrutinee
- This clearly requires dependent types

For this, we need to turn $x :_\alpha X$ into something that looks like

$$\Gamma \vdash M : A \quad \text{and} \quad x : X \in \Gamma \quad \rightsquigarrow \quad \alpha_x(M) : \mathcal{O}_A^\Gamma(X) \to \mathbb{M}$$

The Main Idea

The semi-ring annotations should depend on terms

This is literally the only reason why we had to overapproximate.

- A function should tell you **dynamically** its argument measure
- Pattern-matching measure should depend on the scrutinee
- This clearly requires dependent types

For this, we need to turn $x :_\alpha X$ into something that looks like

$$\Gamma \vdash M : A \quad \text{and} \quad x : X \in \Gamma \quad \rightsquigarrow \quad \alpha_x(M) : \mathcal{O}^{\Gamma}_{A}(X) \to \mathbb{M}$$

Wait, this is a Déjà Vu

$$\Gamma \vdash M : A \text{ and } x : X \in \Gamma \;\rightsquigarrow\; \mathbb{W}(\Gamma) \vdash [M]_x : \mathbb{C}(A)\langle[M]\rangle \to \mathfrak{M}\ \mathbb{C}(X)\langle x\rangle$$

A Bug In the Semi-Ring Matrix

What has been seen cannot be unseen

$$[x]_y\ \pi := \varnothing \qquad [x]_x\ \pi := \{\pi\}$$

$$\frac{}{0\Gamma, x :_1 A \vdash x : A}$$

$$[\lambda y.\, M]_x\ (y, \pi) := [M]_x\ \pi$$

$$\frac{\Gamma, x :_\alpha A \vdash M : B}{\Gamma \vdash \lambda x.\, M : \Pi(x :_\alpha A).\, B}$$

$$[M\ N]_x\ \pi := \begin{array}{c} [M]_x\ ([N], \pi) \\ \oplus \\ ([M].2\ [N]\ \pi \gg\!= [N]_x) \end{array}$$

$$\frac{\Gamma \vdash M : \Pi(x :_\alpha A).\, B \qquad \Delta \vdash N : A}{\Gamma + (\alpha \times \Delta) \vdash M\ N : B\{x := N\}}$$

Dialectica is the HO dependent graded type of a term

- The semiring is replaced by its oidification (a monad + comm. add. monoid)
- Semiring values are now higher-order objects
- In particular they depend on the arguments

Dialectica, Dialectica Everywhere

Dialectica is the HO dependent graded type of a term

- The semiring is replaced by its oidification (a monad + comm. add. monoid)
- Semiring values are now higher-order objects
- In particular they depend on the arguments

This solves the graded expressivity issue

$\rightsquigarrow$ When pattern-matching over $M : \mathbb{B}$, $\alpha \sqcup \beta$ becomes basically

$$\texttt{if } M \texttt{ then } \alpha \texttt{ else } \beta$$

$\rightsquigarrow$ The argument of a function f has a now a highly dynamic annotation

$$\varphi : \Pi(x : \mathbb{W}(A))(\pi : \mathbb{C}(B)\langle f\ x\rangle).\,\mathfrak{M}\ \mathbb{C}(A)\langle x\rangle \qquad \text{(before: a fixed } \mathbb{M}\text{)}$$

Dialectica, Dialectica Everywhere

Dialectica is the HO dependent graded type of a term

- The semiring is replaced by its oidification (a monad + comm. add. monoid)
- Semiring values are now higher-order objects
- In particular they depend on the arguments

This solves the graded expressivity issue

$\rightsquigarrow$ When pattern-matching over $M : \mathbb{B}$, $\alpha \sqcup \beta$ becomes basically

$$\texttt{if } M \texttt{ then } \alpha \texttt{ else } \beta$$

$\rightsquigarrow$ The argument of a function f has a now a highly dynamic annotation

$$\varphi : \Pi(x : \mathbb{W}(A))(\pi : \mathbb{C}(B)\langle f\ x\rangle).\, \mathfrak{M}\ \mathbb{C}(A)\langle x\rangle$$ (before: a fixed $\mathbb{M}$)

Furthermore grading can be stated internally!

So What?

I don't know what to do of this

- Is it known? Is it useful? It is practical?
- Can we get a decidable type system?
- Towards a full-blown synthetic complexity theory?

So What?

I don't know what to do of this

- Is it known? Is it useful? It is practical?
- Can we get a decidable type system?
- Towards a full-blown synthetic complexity theory?

What do you think about all this?

$\rightsquigarrow$ I managed to attract the attention of M. D. at TLLA'24, the more the merrier.

Thanks for your attention.