An Introduction to Dialectica Realizability

Pierre-Marie Pédrot

INRIA

22th July 2016

Once upon a time...

- Cataclysm: Gödel's incompleteness theorem (1931)

Once upon a time...

- Cataclysm: Gödel's incompleteness theorem (1931)

> We do not fight alienation with an alienated logic.

Once upon a time...

- Cataclysm: Gödel's incompleteness theorem (1931)

> We do not fight alienation with an alienated logic.

- Justifying arithmetic differently
- ... Intuitionistic logic!
  - Double-negation translation (1933)
  - **Dialectica** (30's, published in 1958)

Once upon a time...

- The name comes from the journal it was published in
- Also known as Gödel's functional interpretation
- Strange beast typical of German-style logic

Part I

Undusting Dialectica

What is Dialectica?

What is Dialectica?

- A **realizability** interpretation of HA
- That **preserves** intuitionistic content ($\vee$, $\exists$)

What is Dialectica?

- A **realizability** interpretation of HA
- That **preserves** intuitionistic content ($\vee$, $\exists$)
- But **interprets** two semi-classical principles:
  - Markov's principle
  - Independence of premises

What is Dialectica?

- A **realizability** interpretation of HA
- That **preserves** intuitionistic content ($\vee$, $\exists$)
- But **interprets** two semi-classical principles:
  - Markov's principle
  - Independence of premises

Let us discuss each point in more detail.

Realizability?

- Consider some target programming language
- Define a **meta** notion of a program p realizing a formula A

$$p \Vdash A$$

- Extract proofs to programs

$$\pi \vdash A \qquad \rightsquigarrow \qquad \pi^\bullet \Vdash A$$

- Preserve soundness

$$\text{there is no } p \text{ s.t.} \quad p \Vdash \bot$$

- Hope it realizes more

$$\text{there are } p, A \text{ s.t.} \quad p \Vdash A \quad \text{but no} \quad \pi \vdash A$$

Realizability!

Some interesting remarks:

- Dialectica somehow predates the notion of realizability
  - Kleene $\sim$ 1945
  - Kreisel $\sim$ 1959
- Actually Kreisel realizability is a byproduct of Dialectica

  Kreisel: "Dialectica too complicated, let us do simpler!"
- You've been warned...

Preserves intuitionistic content?

This means that:

- A realizer of $A \vee B$ provides you with

  a realizer of A **or** a realizer of B

- A realizer of $\exists x : \mathbb{N}. A$ provides you with

  an **integer** n and realizer of $A[x := n]$

Markov's principle?

$$\mathsf{MP}\ \frac{\neg(\forall n \in \mathbb{N}.\, \neg P\, n)}{\exists n \in \mathbb{N}.\, P\, n}$$

- Requires P decidable
- Naive computational justification: unbounded loop

Independence of Premises?

$$\frac{P \to \exists m \in \mathbb{N}.\, Q\, m}{\exists m \in \mathbb{N}.\, P \to Q\, m}\ \mathsf{IP}$$

- Requires P computationally irrelevant (e.g. $P := \neg P'$)
- Naive computational justification: dummy argument

Warning

- I'll present the historical version of Gödel
- Features a lot of horrible kludges, hacks and tricks

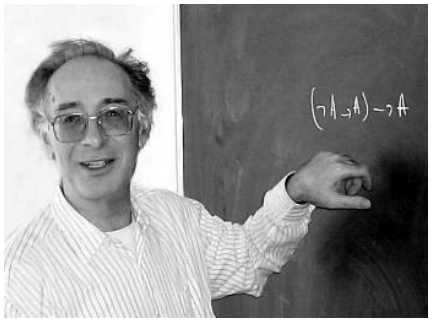
Warning

- I'll present the historical version of Gödel
- Features a lot of horrible kludges, hacks and tricks
- ... yet in a modernized fashion
  - $\rightsquigarrow$ in particular, stick to classical realizability Zeitgeist
  - $\rightsquigarrow$ terms, types, stacks, orthogonality...
- So that it is readable by a computer scientist!
- Cleaned-up version afterwards

Warning

- I'll present the historical version of Gödel
- Features a lot of horrible kludges, hacks and tricks
- ... yet in a modernized fashion
  - $\rightsquigarrow$ in particular, stick to classical realizability Zeitgeist
  - $\rightsquigarrow$ terms, types, stacks, orthogonality...
- So that it is readable by a computer scientist!
- Cleaned-up version afterwards

Wunderbar...



Ça valide ça?

Source theory

We will look at the translation of Heyting arithmetic.

- Intuitionistic
- First-order, one-sorted over integers
- Usual natural deduction on sequents $\Gamma \vdash A$
- Usual axioms for integers

$$t, u ::= x \mid \mathbf{0} \mid \mathbf{S}\ t \mid t + u \mid t \times u$$

$$A, B ::= \bot \mid \top \mid A \vee B \mid A \wedge B \mid A \to B \mid \forall x.\, A \mid \exists x.\, A \mid t = u$$

Source theory

We will look at the translation of Heyting arithmetic.

- Intuitionistic
- First-order, one-sorted over integers
- Usual natural deduction on sequents $\Gamma \vdash A$
- Usual axioms for integers

$$t, u ::= x \mid \mathbf{0} \mid \mathbf{S}\ t \mid t + u \mid t \times u$$

$$A, B ::= \bot \mid \top \mid A \vee B \mid A \wedge B \mid A \to B \mid \forall x.\, A \mid \exists x.\, A \mid t = u$$

I'll not present the rules for it is annoying...

Target language

- We'll use Gödel's famous System **T**.
- Pedantic name for simply-typed λ-calculus + integers
- Very limited, this will require a lot of hacks

$$\sigma, \tau ::= \mathbb{N} \mid \sigma \Rightarrow \tau$$

$$M, N ::= x \mid \lambda x.\, M \mid M\ N \mid \mathtt{0} \mid \mathtt{S}\ M \mid \mathtt{rec}\ M\ N\ P$$

Target language (typing)

We'll use usual simple types, together with

$$\frac{}{\Gamma \vdash \mathtt{0} : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathtt{S}\ M : \mathbb{N}}$$

$$\frac{\Gamma \vdash N_0 : \sigma \qquad \Gamma \vdash N_S : \mathbb{N} \Rightarrow \sigma \Rightarrow \sigma \qquad \Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathtt{rec}\ N_0\ N_S\ M : \sigma}$$

Target language (reduction)

Again, this is the usual λ-calculus plus integer recursion.

$$
\begin{array}{lll}
(\lambda x.\, M)\ N & \to_\beta & M[x := N] \\
\texttt{rec}\ N_0\ N_S\ \texttt{0} & \to_\beta & N_0 \\
\texttt{rec}\ N_0\ N_S\ (\texttt{S}\ M) & \to_\beta & N_S\ M\ (\texttt{rec}\ N_0\ N_S\ M)
\end{array}
$$

Meta-theory

I don't want to dwell too much on this, but

- We'll reason intuitionistically all of the time
- Mostly about term equivalence generated by $\rightarrow_\beta$
- No fancy stuff

Hacking in System **T**

There are many things we're lacking in System **T**!

- booleans
- pairs
- algebraic datatypes
- ...

Hacking in System **T**

There are many things we're lacking in System **T**!

- booleans
- pairs
- algebraic datatypes
- ...

We actually need them for the Dialectica interpretation...

Let us do a bit of assembly to emulate them.

Booleans

That one is easy!

- $\mathbb{B} := \mathbb{N}$
- $\texttt{tt} := \texttt{0}$
- $\texttt{ff} := \texttt{S 0}$
- $\texttt{if } M \texttt{ then } N \texttt{ else } P := \texttt{rec } N\ (\lambda\_\_.\, P)\ M$

Everything works as expected.

Sequences

That one is hairy... I'll consider sequences of

- System **T** types: $\vec{\sigma}$
- System **T** terms: $\vec{M}$
- Variables: $\vec{x}$

and I'll write

- empty sequence $\emptyset$
- concatenation $\vec{M}; \vec{N}$
- implicit lifting from an object to a singleton

and forget about the $\vec{\cdot}$ when I'm fed up with it.

Sequences (moar)

Due to associativity, there are natural notations for sequences:

$$\vec{\sigma} \Rightarrow \tau \quad := \quad \sigma_1 \Rightarrow \ldots \Rightarrow \sigma_n \Rightarrow \tau \qquad \text{(a type)}$$

$$\sigma \Rightarrow \vec{\tau} \quad := \quad (\sigma \Rightarrow \tau_1) \; ; \ldots ; (\sigma \Rightarrow \tau_n) \qquad \text{(a sequence of types)}$$

Sequences (moar)

Due to associativity, there are natural notations for sequences:

$$\begin{array}{lcll} \vec{\sigma} \Rightarrow \tau & := & \sigma_1 \Rightarrow \ldots \Rightarrow \sigma_n \Rightarrow \tau & \text{(a type)} \\ \sigma \Rightarrow \vec{\tau} & := & (\sigma \Rightarrow \tau_1) \; ; \ldots ; \; (\sigma \Rightarrow \tau_n) & \text{(a sequence of types)} \end{array}$$

This induces a similar structure on abstraction and application:

$$\begin{array}{lcll} \lambda \vec{x}.\, M & := & \lambda x_1 \; \ldots \; x_n.\, M & \text{(a term)} \\ \lambda x.\, \vec{M} & := & (\lambda x.\, M_1) \; ; \ldots ; \; (\lambda x.\, M_n) & \text{(a sequence of terms)} \\ M \; \vec{N} & := & M \; N_1 \; \ldots \; N_n & \text{(a term)} \\ \vec{M} \; N & := & (M_1 \; N) \; ; \ldots ; \; (M_n \; N) & \text{(a sequence of terms)} \end{array}$$

Sequences (moar)

Due to associativity, there are natural notations for sequences:

$$\begin{array}{lcll} \vec{\sigma} \Rightarrow \tau & := & \sigma_1 \Rightarrow \ldots \Rightarrow \sigma_n \Rightarrow \tau & \text{(a type)} \\ \sigma \Rightarrow \vec{\tau} & := & (\sigma \Rightarrow \tau_1) \; ; \ldots ; \; (\sigma \Rightarrow \tau_n) & \text{(a sequence of types)} \end{array}$$

This induces a similar structure on abstraction and application:

$$\begin{array}{lcll} \lambda \vec{x}.\, M & := & \lambda x_1 \; \ldots \; x_n.\, M & \text{(a term)} \\ \lambda x.\, \vec{M} & := & (\lambda x.\, M_1) \; ; \ldots ; \; (\lambda x.\, M_n) & \text{(a sequence of terms)} \\ M \; \vec{N} & := & M \; N_1 \; \ldots \; N_n & \text{(a term)} \\ \vec{M} \; N & := & (M_1 \; N) \; ; \ldots ; \; (M_n \; N) & \text{(a sequence of terms)} \end{array}$$

Typing and reduction are pointwise compatible with this sequencification.

Sequences (moar)

In particular,

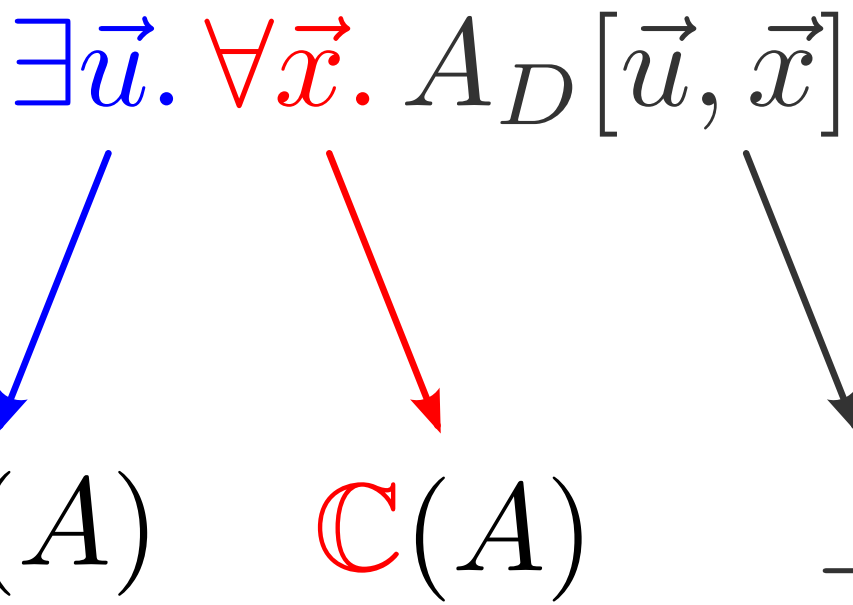
$$\begin{array}{lcll} \emptyset \Rightarrow \tau & := & \tau & \text{(a type)} \\ \sigma \Rightarrow \emptyset & := & \emptyset & \text{(a sequence of types)} \end{array}$$

and

$$\begin{array}{lcll} \lambda\emptyset.\, M & := & M & \text{(a term)} \\ \lambda x.\, \emptyset & := & \emptyset & \text{(a sequence of terms)} \\ M\ \emptyset & := & M & \text{(a term)} \\ \emptyset\ N & := & \emptyset & \text{(a sequence of terms)} \end{array}$$

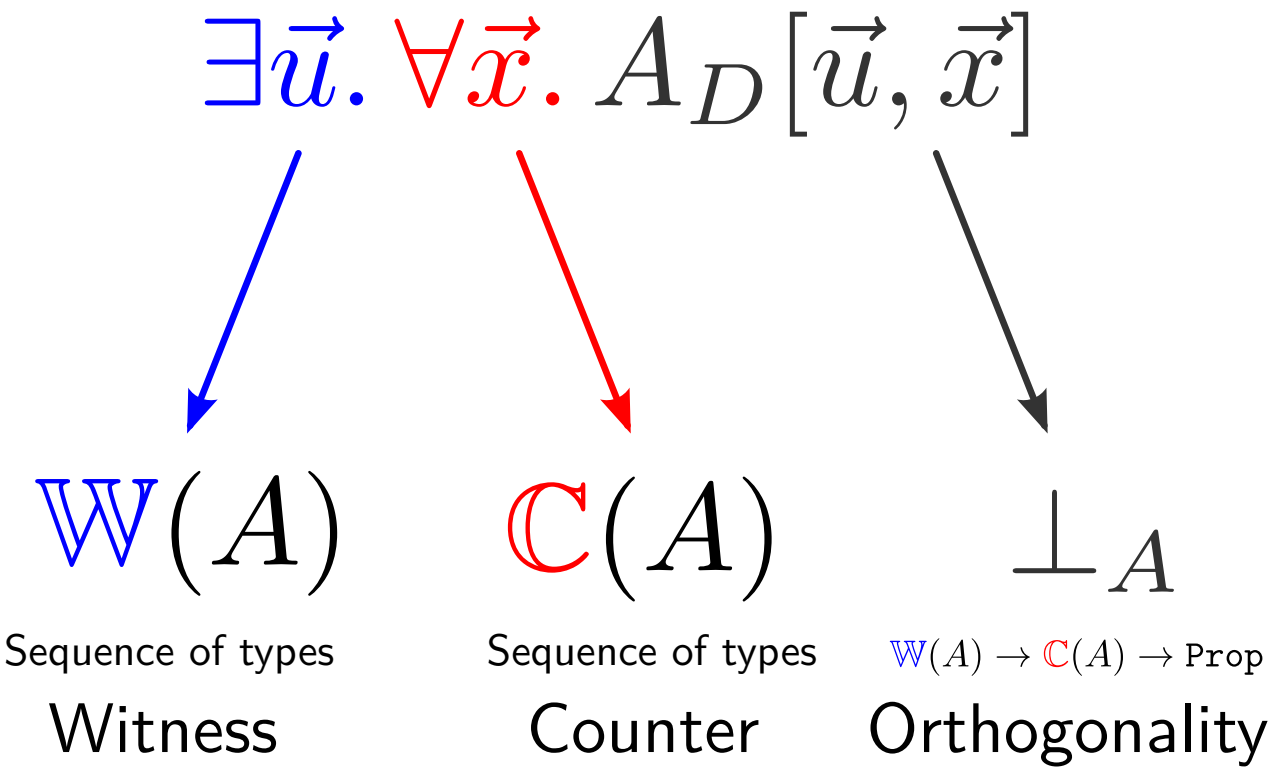
Gödel's anatomy

To any $\mathbf{HA}$ formula A, associate a meta formula



Gödel's anatomy

To any **HA** formula A, associate a meta formula



To any **HA** proof $\pi \vdash A$, associate a meta proof of

$$\forall \vec{x} : \mathbb{C}(A).\ A_D[\pi^\bullet, \vec{x}] \qquad \text{where } \pi^\bullet : \mathbb{W}(A)$$

Building bridges

Intuitively, seen through Krivine's realizability,

- $\mathbb{W}(A) \equiv |A|$, typed truth values
- $\mathbb{C}(A) \equiv ||A||$, typed falsity values
- $M \Vdash A \equiv \forall \vec{x} : \mathbb{C}(A).\, M \perp_A \vec{x}$ (assumes implicitly $M : \mathbb{W}(A)$)

Yet, actually in Krivine realizability

- $|A| \subseteq \Lambda$ and $||A|| \subseteq \Pi$ are untyped
- Orthogonality untyped as well: $M \perp \pi \equiv \langle M \mid \pi \rangle \subseteq \perp\!\!\!\perp$
- $M \Vdash A \equiv \forall \pi \in ||A||.\, M \perp \pi$

(Another alternative explanation as logical games.)

Roadmap

We're going to do the following in the next slides:

- Define $\mathbb{W}(\cdot), \mathbb{C}(\cdot)$ and $\perp_{(\cdot)}$ by induction on the type
- State the soundness theorem abstractly
- Look at a few key cases from the proof and get an idea of the realizers

Then we'll have a look at what we got new.

Dialectica translation (easy cases)

$$\mathbb{W}(A \wedge B) := \mathbb{W}(A)\ ;\ \mathbb{W}(B) \qquad \mathbb{C}(A \wedge B) := \mathbb{C}(A)\ ;\ \mathbb{C}(B)$$

$$\frac{M_A \perp_A \Pi_A \qquad M_B \perp_B \Pi_B}{M_A\ ;\ M_B \perp_{A \wedge B} \Pi_A\ ;\ \Pi_B}$$

$$\mathbb{W}(A \vee B) := \mathbb{B}\ ;\ \mathbb{W}(A)\ ;\ \mathbb{W}(B) \qquad \mathbb{C}(A \vee B) := \mathbb{C}(A)\ ;\ \mathbb{C}(B)$$

$$\frac{M_A \perp_A \Pi_A}{\mathtt{tt}\ ;\ M_A\ ;\ M_B \perp_{A \vee B} \Pi_A\ ;\ \Pi_B} \qquad \frac{M_B \perp_B \Pi_B}{\mathtt{ff}\ ;\ M_A\ ;\ M_B \perp_{A \vee B} \Pi_A\ ;\ \Pi_B}$$

Dialectica translation (easy cases)

$$\mathbb{W}(\top) := \emptyset \qquad \mathbb{C}(\top) := \emptyset \qquad \frac{}{\emptyset \perp_{\top} \emptyset}$$

$$\mathbb{W}(\bot) := \emptyset \qquad \mathbb{C}(\bot) := \emptyset \qquad \text{(no orthogonality)}$$

$$\mathbb{W}(t = u) := \emptyset \qquad \mathbb{C}(t = u) := \emptyset \qquad \frac{t = u}{\emptyset \perp_{t=u} \emptyset}$$

Dialectica translation (first-order)

$$\mathbb{W}(\forall x.\, A) := \mathbb{N} \Rightarrow \mathbb{W}(A) \qquad \mathbb{C}(\forall x.\, A) := \mathbb{N}\ ;\ \mathbb{C}(A)$$

$$\frac{M\ N \perp_{A[x:=N]} \Pi}{M \perp_{\forall x.\, A} N\ ;\ \Pi}$$

$$\mathbb{W}(\exists x.\, A) := \mathbb{N}\ ;\ \mathbb{W}(A) \qquad \mathbb{C}(\exists x.\, A) := \mathbb{N} \Rightarrow \mathbb{C}(A)$$

$$\frac{M \perp_{A[x:=N]} \Pi\ N}{N\ ;\ M \perp_{\exists x.\, A} \Pi}$$

Dialectica translation (first-order)

$$\mathbb{W}(\forall x.\, A) := \mathbb{N} \Rightarrow \mathbb{W}(A) \qquad \mathbb{C}(\forall x.\, A) := \mathbb{N}\,;\, \mathbb{C}(A)$$

$$\frac{M\ N \perp_{A[x:=N]} \Pi}{M \perp_{\forall x.\, A} N\,;\, \Pi}$$

$$\mathbb{W}(\exists x.\, A) := \mathbb{N}\,;\, \mathbb{W}(A) \qquad \mathbb{C}(\exists x.\, A) := \mathbb{N} \Rightarrow \mathbb{C}(A)$$

$$\frac{M \perp_{A[x:=N]} \Pi\ N}{N\,;\, M \perp_{\exists x.\, A} \Pi}$$

(I'm a bit cheating here for pedagogical purposes.)

Dialectica translation: the wondrous intuitionistic arrow

The mysterious part of the Dialectica translation comes from the arrow...

$$\begin{aligned}\mathbb{W}(A \to B) &:= (\mathbb{W}(A) \Rightarrow \mathbb{W}(B)) \,;\, (\mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A)) \\ \mathbb{C}(A \to B) &:= \mathbb{W}(A) \,;\, \mathbb{C}(B)\end{aligned}$$

$$\frac{\text{if} \quad N \perp_A \Phi \; N \; \Pi \quad \text{then} \quad M \; N \perp_B \Pi}{M \,;\, \Phi \perp_{A \to B} N \,;\, \Pi}$$

The witness of an arrow has a second, fancy component!

Remarks

- The $\mathbb{W}(\cdot)$ translation corresponds essentially to Kreisel extraction.
- ... except for the arrow.

Remarks

- The $\mathbb{W}(\cdot)$ translation corresponds essentially to Kreisel extraction.
- ... except for the arrow.
- Most of the logical content is pushed into the meta
- ... thanks to the orthogonality.

Remarks

- The $\mathbb{W}(\cdot)$ translation corresponds essentially to Kreisel extraction.
- ... except for the arrow.
- Most of the logical content is pushed into the meta
- ... thanks to the orthogonality.
- Hacks are quite clear, e.g.

$$A + B \sim \mathbb{B} \times A \times B$$

Easy result

Our realizability interpretation is trivially consistent.

Theorem (Consistency)

There is no System $\mathbf{T}$ sequence of terms $M \Vdash \bot$.

Easy result

Our realizability interpretation is trivially consistent.

Theorem (Consistency)

There is no System $\mathbf{T}$ sequence of terms $M \Vdash \bot$.

Proof.

Assume such sequence M, then there should not be $\vec{x} : \mathbb{C}(\bot)$ by definition of $\bot_\bot$. But $\mathbb{C}(\bot) \equiv \emptyset$ and the empty sequence is trivially inhabited, which contradicts the above statement.

□

Note that we do have **paraproofs** of falsity...

Soundness (overview)

We will now sketch the proof of the following statement.

Theorem (Soundness)

If $\vdash_{\mathbf{HA}} A$, then there is a sequence of System $\mathbf{T}$ terms $M \Vdash A$.

Soundness (overview)

We will now sketch the proof of the following statement.

Theorem (Soundness)

If $\vdash_{\mathbf{HA}} A$, then there is a sequence of System $\mathbf{T}$ terms $M \Vdash A$.

We will prove a generalized statement by induction on the proof.

Theorem (Generalized soundness)

If $\pi : \Gamma_1, \ldots, \Gamma_n \vdash_{\mathbf{HA}} A$, then there is:

- a sequence of terms $\pi^\bullet : \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A)$
- n sequences of terms $\pi_i^\circ : \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_i)$

s.t. for all $M : \mathbb{W}(\Gamma)$ and $\Pi : \mathbb{C}(A)$,

$$\text{if for all } i \leq n, \quad M_i \perp_{\Gamma_i} \pi_i^\circ\ M\ \Pi \quad \text{then} \quad \pi^\bullet M \perp_A \Pi$$

Enters technique

The theorem relies on two essential facts of the translation.

- All System $\mathbf{T}$ types are inhabited.
- The $\bot_A$ relation is decidable in System $\mathbf{T}$ for all A, i.e. there is a term of type $\mathbb{W}(A) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{B}$ which internalizes $\bot_A$.

Enters technique

The theorem relies on two essential facts of the translation.

- All System $\mathbf{T}$ types are inhabited.
- The $\bot_A$ relation is decidable in System $\mathbf{T}$ for all A, i.e. there is a term of type $\mathbb{W}(A) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{B}$ which internalizes $\bot_A$.

Proof.

By induction on the considered type. □

Hack, hack, hack your bloat

This allows to define two families of System $\mathbf{T}$ terms:

- The dummy terms $\maltese_A : \mathbb{C}(A)$
- The merge terms $\oplus_A : \mathbb{C}(A) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(A)$ s.t. for all $\Pi, \Xi : \mathbb{C}(A)$ and $M : \mathbb{W}(A)$,

$$M \perp_A (\Pi \oplus_A^M \Xi) \quad \text{iff} \quad M \perp_A \Pi \quad \text{and} \quad M \perp_A \Xi.$$

Hack, hack, hack your bloat

This allows to define two families of System $\mathbf{T}$ terms:

- The dummy terms $\maltese_A : \mathbb{C}(A)$
- The merge terms $\oplus_A : \mathbb{C}(A) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(A)$ s.t. for all $\Pi, \Xi : \mathbb{C}(A)$ and $M : \mathbb{W}(A)$,

$$M \perp_A (\Pi \oplus_A^M \Xi) \quad \text{iff} \quad M \perp_A \Pi \quad \text{and} \quad M \perp_A \Xi.$$

Proof.

- Take an arbitrary inhabitant of $\mathbb{C}(A)$ for $\maltese_A$
- Define

$$\Pi \oplus_A^M \Xi := \texttt{if } M \perp_A \Pi \texttt{ then } \Xi \texttt{ else } \Pi$$

□

Soundness (Axiom)

Let us give the interpretation of the rule $\dfrac{}{\Gamma_1, \ldots, \Gamma_n \vdash \Gamma_k}\,(\mathrm{axm})$:

$$
\begin{array}{lcll}
\mathrm{axm}^{\bullet} & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(\Gamma_k) & \\
\mathrm{axm}^{\bullet} & := & \lambda x_1 \ldots x_n . x_k & \\
\mathrm{axm}^{\circ}_k & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(\Gamma_k) \Rightarrow \mathbb{C}(\Gamma_k) & \\
\mathrm{axm}^{\circ}_k & := & \lambda x_1 \ldots x_n . \lambda \pi . \pi & \\
\mathrm{axm}^{\circ}_i & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(\Gamma_k) \Rightarrow \mathbb{C}(\Gamma_i) & (i \neq k) \\
\mathrm{axm}^{\circ}_i & := & \lambda x_1 \ldots x_n . \lambda \pi . \maltese_{\Gamma_i} &
\end{array}
$$

Soundness (Axiom)

Let us give the interpretation of the rule $\dfrac{}{\Gamma_1, \ldots, \Gamma_n \vdash \Gamma_k}\,(\mathrm{axm})$:

$$
\begin{array}{lcll}
\mathrm{axm}^{\bullet} & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(\Gamma_k) & \\
\mathrm{axm}^{\bullet} & := & \lambda x_1 \ldots x_n . x_k & \\
\mathrm{axm}^{\circ}_k & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(\Gamma_k) \Rightarrow \mathbb{C}(\Gamma_k) & \\
\mathrm{axm}^{\circ}_k & := & \lambda x_1 \ldots x_n . \lambda \pi . \pi & \\
\mathrm{axm}^{\circ}_i & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(\Gamma_k) \Rightarrow \mathbb{C}(\Gamma_i) & (i \neq k) \\
\mathrm{axm}^{\circ}_i & := & \lambda x_1 \ldots x_n . \lambda \pi . \maltese_{\Gamma_i} &
\end{array}
$$

We easily show for all $\gamma : \mathbb{W}(\Gamma)$ and $\pi : \mathbb{C}(\Gamma_k)$:

$$
\bigwedge_i \gamma_i \perp_{\Gamma_i} \mathrm{axm}^{\circ}_i \ \gamma \ \pi \quad \longrightarrow \quad \mathrm{axm}^{\bullet} \ \gamma \perp_{\Gamma_k} \pi
$$

Soundness (Axiom)

Everything has been done implicitly with sequences in the previous slide!

The actual demacroified sequence of terms is less palatable...

Soundness (λ-abstraction)

Rule $\dfrac{q : \Gamma, A \vdash B}{p : \Gamma \vdash A \to B} :$

$$
p^\bullet \begin{cases}
p^\bullet_+ & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\
p^\bullet_+ & := & \lambda\gamma.\, \lambda x.\, q^\bullet\ \gamma\ x \\
p^\bullet_- & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \\
p^\bullet_- & := & \lambda\gamma.\, \lambda x.\, \lambda\pi.\, q^\circ_A\ \gamma\ x\ \pi
\end{cases}
$$

$$
\begin{array}{lcl}
p^\circ_i & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(\Gamma_i) \\
p^\circ_i & := & \lambda\gamma.\, \lambda x.\, \lambda\pi.\, q^\circ_{\Gamma_i}\ \gamma\ x\ \pi
\end{array}
$$

Soundness (λ-abstraction)

Rule $\dfrac{q : \Gamma, A \vdash B}{p : \Gamma \vdash A \to B}$:

$$p^\bullet \begin{cases} p^\bullet_+ & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ p^\bullet_+ & := & \lambda\gamma.\, \lambda x.\, q^\bullet\ \gamma\ x \\ p^\bullet_- & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \\ p^\bullet_- & := & \lambda\gamma.\, \lambda x.\, \lambda\pi.\, q^\circ_A\ \gamma\ x\ \pi \\ p^\circ_i & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(\Gamma_i) \\ p^\circ_i & := & \lambda\gamma.\, \lambda x.\, \lambda\pi.\, q^\circ_{\Gamma_i}\ \gamma\ x\ \pi \end{cases}$$

Need to show for all $\gamma : \mathbb{W}(\Gamma)$, $x : \mathbb{W}(A)$ and $\pi : \mathbb{C}(B)$:

$$\bigwedge_i \gamma_i \perp_{\Gamma_i} p^\circ_i\ \gamma\ x\ \pi \quad \longrightarrow \quad x \perp_A p^\bullet_-\ \gamma\ x\ \pi \quad \longrightarrow \quad p^\bullet_+\ \gamma\ x \perp_B \pi$$

which comes directly from hypothesis on q.

Soundness (application)

Rule $\dfrac{q : \Gamma \vdash A \to B \qquad r : \Gamma \vdash A}{p : \Gamma \vdash B}$:

$$
\begin{array}{lll}
p^\bullet & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{W}(B) \\
p^\bullet & := & \lambda\gamma.\, q^\bullet_+\ \gamma\ (r^\bullet\ \gamma) \\
p^\circ_i & : & \mathbb{W}(\Gamma) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(\Gamma_i) \\
p^\circ_i & := & \lambda\gamma.\, \lambda\pi.\, (q^\circ_i\ \gamma\ (r^\bullet\ \gamma)\ \pi) \oplus^{\gamma_i}_{\Gamma_i} (r^\circ_i\ \gamma\ (q^\bullet_-\ \gamma\ (r^\bullet\ \gamma)\ \pi))
\end{array}
$$

Need to show for all $\gamma : \mathbb{W}(\Gamma)$ and $\pi : \mathbb{C}(B)$:

$$\bigwedge_i \gamma_i \perp_{\Gamma_i} p^\circ_i\ \gamma\ \pi \quad \longrightarrow \quad p^\bullet\ \gamma \perp_B \pi$$

which implies a bit of work...

Soundness (other cases)

Essentially,

- Nullary connectives are trivial
- The one atomic formula $t = u$ is pushed back in the meta
- The $(-)^\bullet$ translation is more or less identity on the underlying proof-term
- The $(-)_i^\circ$ translation merges all available counters

Stepping back

We now see where the technical apparatus is needed.

- The $✠_A$ terms are used to implement weakening
- The $\oplus_A$ terms are used to implement contraction
- Linear logicians should be jumping on their seats by now

Stepping back

We now see where the technical apparatus is needed.

- The $✠_A$ terms are used to implement weakening
- The $\oplus_A$ terms are used to implement contraction
- Linear logicians should be jumping on their seats by now

Note that this is not the standard presentation of Dialectica.

- Usually, realizers are not explicitly written
- Here, we can already get a rough idea of what is going on

Stepping back

We now see where the technical apparatus is needed.

- The $✠_A$ terms are used to implement weakening
- The $\oplus_A$ terms are used to implement contraction
- Linear logicians should be jumping on their seats by now

Note that this is not the standard presentation of Dialectica.

- Usually, realizers are not explicitly written
- Here, we can already get a rough idea of what is going on

Later!

You are Here

We've proved the soudness theorem, therefore

Dialectica is a realizability interpretation of **HA**.

You are Here

We've proved the soudness theorem, therefore

Dialectica is a realizability interpretation of **HA**.

We preserve good properties...

Dialectica preserves positives.

Yet, as for Kreisel and Krivine realizability, we get more!

Dialectica interprets $\mathbf{MP}$ and $\mathbf{IP}$.

i.e. there are term sequences $\mathtt{MP} \Vdash \mathbf{MP}$ and $\mathtt{IP} \Vdash \mathbf{IP}$.

Constructivity

It is obvious to check that proofs of positive connectives are preserved!

- If b ; M_1 ; $M_2 \Vdash A \vee B$ then b tells you which realizer is the good one.
- If n ; $M \Vdash \exists x.\, A$ then n is a correct witness and M realizes that.

(Remember:)

$$\mathbb{W}(A \vee B) := \mathbb{B} \,;\, \mathbb{W}(A) \,;\, \mathbb{W}(B)$$

$$\mathbb{W}(\exists x.\, A) := \mathbb{N} \,;\, \mathbb{W}(A)$$

Premises, what are they needed for?

$$\frac{P \to \exists m.\, Q\, m}{\exists m.\, P \to Q\, m}\ \mathbf{IP}$$

This is only true for **irrelevant** P!

Premises, what are they needed for?

$$\frac{P \to \exists m.\, Q\, m}{\exists m.\, P \to Q\, m}\ \mathbf{IP}$$

This is only true for **irrelevant** P!

Luckily, we can define this through the Dialectica.

Definition (Irrelevance)

A formula A is irrelevant whenever both $\mathbb{W}(A) = \emptyset$ and $\mathbb{C}(A) = \emptyset$.

Premises, what are they needed for?

$$\frac{P \to \exists m.\, Q\, m}{\exists m.\, P \to Q\, m}\ \mathbf{IP}$$

This is only true for **irrelevant** P!

Luckily, we can define this through the Dialectica.

Definition (Irrelevance)

A formula A is irrelevant whenever both $\mathbb{W}(A) = \emptyset$ and $\mathbb{C}(A) = \emptyset$.

Theorem (Irrelevance)

The negative propositional fragment A^- is irrelevant.

$$A^-, B^- ::= \bot \mid \top \mid t = u \mid A^- \wedge B^- \mid A^- \to B^-$$

Address **IP**

Assume P irrelevant, we have

$$\begin{aligned}
& \mathbb{W}((P \to \exists m.\, Q\, m) \to \exists m.\, (P \to Q\, m)) \\
= & \begin{cases}
\mathbb{W}(P \to \exists m.\, Q\, m) \Rightarrow \mathbb{N} \\
\mathbb{W}(P \to \exists m.\, Q\, m) \Rightarrow \mathbb{W}(P \to Q\, m) \\
\mathbb{W}(P \to \exists m.\, Q\, m) \Rightarrow \mathbb{C}(\exists m.\, P \to Q\, m) \Rightarrow \mathbb{W}(P) \\
\mathbb{W}(P \to \exists m.\, Q\, m) \Rightarrow \mathbb{C}(\exists m.\, P \to Q\, m) \Rightarrow \mathbb{C}(\exists m.\, Q\, m)
\end{cases} \\
= & \begin{cases}
\mathbb{N} \Rightarrow \mathbb{W}(Q\, m) \Rightarrow \mathbb{N} \\
\mathbb{N} \Rightarrow \mathbb{W}(Q\, m) \Rightarrow \mathbb{W}(Q\, m) \\
\mathbb{N} \Rightarrow \mathbb{W}(Q\, m) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{C}(Q\, m)) \Rightarrow \mathbb{N} \Rightarrow \mathbb{C}(Q\, m)
\end{cases}
\end{aligned}$$

as $\mathbb{W}(P \to A) = \mathbb{W}(A)$

$\mathbf{IP}$, at last

Therefore, just take

$$
\begin{array}{lcl}
\mathbf{IP}^{\bullet}_{\mathbb{N}} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow \mathbb{N} \\
\mathbf{IP}^{\bullet}_{\mathbb{N}} & := & \lambda n\ \_ .\, n \\
\mathbf{IP}^{\bullet}_{Q\ m} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow \mathbb{W}(Q\,m) \\
\mathbf{IP}^{\bullet}_{Q\ m} & := & \lambda \_\ p .\, p \\
\mathbf{IP}^{\bullet}_{\_} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{C}(Q\,m)) \Rightarrow \mathbb{N} \Rightarrow \mathbb{C}(Q\,m) \\
\mathbf{IP}^{\bullet}_{\_} & := & \lambda \_\ \_\ k .\, k
\end{array}
$$

$\mathbf{IP}$, at last

Therefore, just take

$$
\begin{array}{lcl}
\mathbf{IP}^{\bullet}_{\mathbb{N}} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow \mathbb{N} \\
\mathbf{IP}^{\bullet}_{\mathbb{N}} & := & \lambda n\ \_ .\, n \\
\mathbf{IP}^{\bullet}_{Q\ m} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow \mathbb{W}(Q\,m) \\
\mathbf{IP}^{\bullet}_{Q\ m} & := & \lambda \_\ p.\, p \\
\mathbf{IP}^{\bullet}_{\_} & : & \mathbb{N} \Rightarrow \mathbb{W}(Q\,m) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{C}(Q\,m)) \Rightarrow \mathbb{N} \Rightarrow \mathbb{C}(Q\,m) \\
\mathbf{IP}^{\bullet}_{\_} & := & \lambda \_\ \_\ k.\, k
\end{array}
$$

Proving that this is a proper realizer is trivial, this is mostly a projection.

$$\mathbf{IP}^{\bullet} \Vdash (P \to \exists m.\, Q\,m) \to \exists m.\, (P \to Q\,m)$$

Reflexions on $\mathbf{IP}$

Actually, $\mathbf{IP}$ is mostly an accident

- It works for about the same reasons as in usual realizability
- Namely, irrelevant stuff is erased by the translation
- The realizer does not take advantage of the added structure

"$\mathbf{IP}$ *is merely a consequence of realizability.*"

Towards $\mathbf{MP}$

$$\mathbf{MP}\ \frac{\neg(\forall n.\, \neg P\, n)}{\exists n.\, P\, n}$$

Let us try to realize this assuming P is decidable!

A quick introductory remark: our target language is strongly normalizing...
We can't write the naive loop algorithm from intuitionistic realizability!

Towards $\mathbf{MP}$

$$\mathbf{MP}\ \frac{\neg(\forall n.\, \neg P\, n)}{\exists n.\, P\, n}$$

Let us try to realize this assuming P is decidable!

A quick introductory remark: our target language is strongly normalizing...
We can't write the naive loop algorithm from intuitionistic realizability!

We'll do much better.

Nein

First, let's scrutinize the interpretation of negation.

$$\neg A := A \to \bot$$

We have:

$$\begin{array}{lcl} \mathbb{W}(\neg A) & = & \mathbb{W}(A) \Rightarrow \mathbb{C}(A) \\ \mathbb{C}(\neg A) & = & \mathbb{W}(A) \end{array}$$

and

$$\frac{M \not\perp_A \Phi \; M}{\Phi \perp_{\neg A} M}$$

Nein

First, let's scrutinize the interpretation of negation.

$$\neg A := A \to \bot$$

We have:

$$\begin{array}{lcl} \mathbb{W}(\neg A) & = & \mathbb{W}(A) \Rightarrow \mathbb{C}(A) \\ \mathbb{C}(\neg A) & = & \mathbb{W}(A) \end{array}$$

and

$$\frac{M \not\perp_A \Phi \; M}{\Phi \perp_{\neg A} M}$$

Negation highly asymmetrical, not degenerate!

Unfolding...

We have then:

$$
\begin{aligned}
& \mathbb{W}(\neg\forall n.\, \neg P\ n \to \exists n.\, P\ n) \\
= & \begin{cases} \mathbb{W}(\neg\Psi) \Rightarrow \mathbb{N} \\ \mathbb{W}(\neg\Psi) \Rightarrow \mathbb{W}(P\ n) \\ \mathbb{W}(\neg\Psi) \Rightarrow \mathbb{C}(\exists n.\, P\ n) \Rightarrow \mathbb{C}(\neg\Psi) \end{cases} \\
= & \begin{cases} (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{N} \\ (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{W}(P\ n) \\ (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{C}(\exists n.\, P\ n) \Rightarrow \mathbb{W}(\Psi) \end{cases}
\end{aligned}
$$

where $\Psi := \forall n.\, \neg P\ n.$

Simplifying things drastically

But P is decidable, so wlog we can assume it is irrelevant, thus:

$$\begin{array}{lcl} \mathbb{W}(P\ n) & = & \emptyset \\ \mathbb{W}(\neg\forall n.\, \neg P\ n) & = & \emptyset \end{array}$$

and then

$$\begin{array}{rl} & \mathbb{W}(\neg\forall n.\, \neg P\ n \to \exists n.\, P\ n) \\ = & \left\{ \begin{array}{l} (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{N} \\ (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{W}(P\ n) \\ (\mathbb{W}(\Psi) \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{W}(\Psi) \Rightarrow \mathbb{W}(P\ n)) \Rightarrow \mathbb{C}(\exists n.\, P\ n) \Rightarrow \mathbb{W}(\Psi) \end{array} \right. \\ = & \left\{ \begin{array}{l} \mathbb{N} \Rightarrow \mathbb{N} \\ \emptyset \\ \emptyset \end{array} \right. \end{array}$$

Markov's principle

It's pretty clear how to provide a term of the expected type:

$$\begin{array}{lcl} \mathbf{MP}^{\bullet} & : & \mathbb{W}(\neg\forall n.\, \neg P\ n \to \exists n.\, P\ n) \\ \mathbf{MP}^{\bullet} & := & \lambda n.\, n \end{array}$$

Markov's principle

It's pretty clear how to provide a term of the expected type:

$$\begin{array}{lcl} \mathbf{MP}^{\bullet} & : & \mathbb{W}(\neg\forall n.\, \neg P\ n \to \exists n.\, P\ n) \\ \mathbf{MP}^{\bullet} & := & \lambda n.\, n \end{array}$$

The realizibility condition amounts to prove for all $n : \mathbb{N}$:

$$\emptyset \not\perp_{\forall n.\, \neg P\ n}\ n \quad \longrightarrow \quad n \perp_{\exists n.\, P\ n}\ \emptyset$$

$$\neg\neg\emptyset \perp_{P\ n}\ \emptyset \quad \longrightarrow \quad \emptyset \perp_{P\ n}\ \emptyset$$

This is true intuitionistically thanks to decidability...

Markov's principle?

Something fishy going on here.

- The realizer is suspiciously simple...
- The realizability condition makes appear classical-looking stuff
- The integer is provided by the reverse part of the negation

"**MP** *is using the Dialectica translation in-depth.*"

End of Part I

Part II

A Functional Functional Interpretation

Can I haz Curry-Howard?

Let us forget the 50's, and rather jump directly to the 90's.

- Take seriously the **computational** content
- Dialectica as a **typed** object
- Works of De Paiva, Hyland, etc.

Get rid of Gödel's hacks:

- Proper datatypes
- No more sequences!
- Stop ugly encodings

A systematic approach

"Realizability interpretations tend to hide a programming translation."

Logic	Programming
Kreisel modified realizability	Identity translation
Krivine classical realizability	Lafont-Reus-Streicher CPS
Gödel Dialectica realizability	?

A systematic approach

"Realizability interpretations tend to hide a programming translation."

Logic	Programming
Kreisel modified realizability	Identity translation
Krivine classical realizability	Lafont-Reus-Streicher CPS
Gödel Dialectica realizability	**A fancy one!**

A systematic approach

"Realizability interpretations tend to hide a programming translation."

Logic	Programming
Kreisel modified realizability	Identity translation
Krivine classical realizability	Lafont-Reus-Streicher CPS
Gödel Dialectica realizability	**A fancy one!**

- Gives first-class status to stacks
- Features a computationally relevant substitution
- Mix of LRS with delimited continuations
- Requires computational (finite) multisets $\mathfrak{M}$

Target programming language

Instead of System $\mathbf{T}$, we take a simply-typed λ-calculus with datatypes.

$$\sigma, \tau ::= \ldots \mid \sigma \times \tau \mid \sigma + \tau \mid 0 \mid 1$$

and add the proper pattern-matchings and constructors.

We interpret everything directly into System $\mathbf{T}$ (no sequences!).

The same, with types

If we wish to put more types in there:

	$\mathbb{W}$	$\mathbb{C}$
$\top$	1	1
$\bot$	1	1
$A \wedge B$	$\mathbb{W}(A) \times \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$
$A \vee B$	$\mathbb{W}(A) + \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$
$A \to B$	$\left(\begin{array}{c} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \times \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{array}\right)$	$\mathbb{W}(A) \times \mathbb{C}(B)$

The same, with types

If we wish to put more types in there:

	$\mathbb{W}$	$\mathbb{C}$
$\top$	1	1
$\bot$	1	1
$A \wedge B$	$\mathbb{W}(A) \times \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$
$A \vee B$	$\mathbb{W}(A) + \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$
$A \to B$	$\left(\begin{array}{c} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \times \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{array}\right)$	$\mathbb{W}(A) \times \mathbb{C}(B)$

- Orthogonality is adapted in a direct way (sequences $\mapsto$ pairs).
- Observe how $\mathbb{W}(A)$ and $\mathbb{C}(A)$ are always inhabited

Not too hastily

- We could give a computational content right now

Not too hastily

- We could give a computational content right now
- But it would be a special case, taking advantage of some encodings
- Let us use our our favorite tool: **Linear Logic**.
  - It factorizes Dialectica!
  - A genuine exponential!
  - With real chunks of sum types!

Not too hastily

- We could give a computational content right now
- But it would be a special case, taking advantage of some encodings
- Let us use our our favorite tool: **Linear Logic**.
  - It factorizes Dialectica!
  - A genuine exponential!
  - With real chunks of sum types!

(Do not worry too much if you know nothing about $\mathbf{LL}$, this is mainly for general culture purposes.)

Linearized Dialectica

As forecasted on the previous slide, we essentially apply the following modifications:

- Introduction of duality with sum types
- Call-by-name decomposition of the arrow:

$$A \to B \quad \equiv \quad !A \multimap B$$

Linearized Dialectica

As forecasted on the previous slide, we essentially apply the following modifications:

- Introduction of duality with sum types
- Call-by-name decomposition of the arrow:

$$A \to B \quad \equiv \quad !A \multimap B$$

Now we will be translating $\mathbf{LL}$ into $\mathbf{LJ}$.

Requirements

We will be interpreting the formulæ of linear logic:

$$A, B ::= A \otimes B \mid A \parr B \mid A \oplus B \mid A \,\&\, B \mid !A \mid ?A$$

It is therefore sufficient to define $\mathbb{W}(A)$, $\mathbb{C}(A)$ and $\bot_A$ for each A.

Requirements

We will be interpreting the formulæ of linear logic:

$$A, B ::= A \otimes B \mid A \parr B \mid A \oplus B \mid A \,\&\, B \mid !A \mid ?A$$

It is therefore sufficient to define $\mathbb{W}(A)$, $\mathbb{C}(A)$ and $\bot_A$ for each A.

Taking inspiration from the double-orthogonality models, we require:

- $\mathbb{W}(A^\bot) \equiv \mathbb{C}(A)$ and conversely;
- thus $\bot_A \subseteq \mathbb{W}(A) \times \mathbb{C}(A) \equiv \mathbb{W}(A) \times \mathbb{W}(A^\bot)$

Forget the dual

- $\rightsquigarrow$ It is sufficient to define our structures on positive types
- $\rightsquigarrow$ We will get them for dual connectives... by duality.

We define therefore:

$$\frac{u \not\perp_A x}{x \perp_{A^\perp} u}$$

Sum types

	$\mathbb{W}$	$\mathbb{C}$
$A \,\&\, B$	$\mathbb{W}(A) \times \mathbb{W}(B)$	$\mathbb{C}(A) + \mathbb{C}(B)$
$A \oplus B$	$\mathbb{W}(A) + \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$

Sum types

	$\mathbb{W}$	$\mathbb{C}$
$A \,\&\, B$	$\mathbb{W}(A) \times \mathbb{W}(B)$	$\mathbb{C}(A) + \mathbb{C}(B)$
$A \oplus B$	$\mathbb{W}(A) + \mathbb{W}(B)$	$\mathbb{C}(A) \times \mathbb{C}(B)$

$$\frac{v \perp_A z_2}{\mathtt{inr}\, v \perp_{A \oplus B} (z_1, z_2)} \qquad \frac{u \perp_A z_1}{\mathtt{inl}\, u \perp_{A \oplus B} (z_1, z_2)}$$

Linear decomposition

	$\mathbb{W}$	$\mathbb{C}$
$A \to B$	$\begin{cases} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{cases}$	$\mathbb{W}(A) \times \mathbb{C}(B)$
$A \multimap B$	$\begin{cases} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{cases}$	$\mathbb{W}(A) \times \mathbb{C}(B)$
$!A$	$\mathbb{W}(A)$	$\mathbb{W}(A) \Rightarrow \mathbb{C}(A)$

Linear decomposition

	$\mathbb{W}$	$\mathbb{C}$
$A \to B$	$\begin{cases} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{cases}$	$\mathbb{W}(A) \times \mathbb{C}(B)$
$A \multimap B$	$\begin{cases} \mathbb{W}(A) \Rightarrow \mathbb{W}(B) \\ \mathbb{C}(B) \Rightarrow \mathbb{C}(A) \end{cases}$	$\mathbb{W}(A) \times \mathbb{C}(B)$
$!A$	$\mathbb{W}(A)$	$\mathbb{W}(A) \Rightarrow \mathbb{C}(A)$

$$\frac{u \perp_A \psi\, y \quad \to \quad \varphi\, u \perp_B y}{(\varphi, \psi) \perp_{A \multimap B} (u, y)} \qquad \frac{u \perp_A z\, u}{u \perp_{!A} z}$$

Handwaving justification

- The interpretation of arrow forces its reversibility:

$$A \multimap B \cong B^{\perp} \multimap A^{\perp}$$

  $\rightsquigarrow$ Like the two-way proofnet wires

Handwaving justification

- The interpretation of arrow forces its reversibility:

$$A \multimap B \cong B^{\perp} \multimap A^{\perp}$$

  $\rightsquigarrow$ Like the two-way proofnet wires

- The bang connective is a *shift* :

  $\rightsquigarrow$ Opponent may wait for the player to play and inspect its answer

- Duality is rôle swapping

About linearity

We're not linear by chance.

[1] Assuming we've defined 1.

[2] May contain nuts.

About linearity

We're not linear by chance.

Indeed, in Dialectica, we do not realize the following morphisms:

$$\vdash A \multimap 1^{1}$$

$$\vdash A \multimap A \otimes A$$

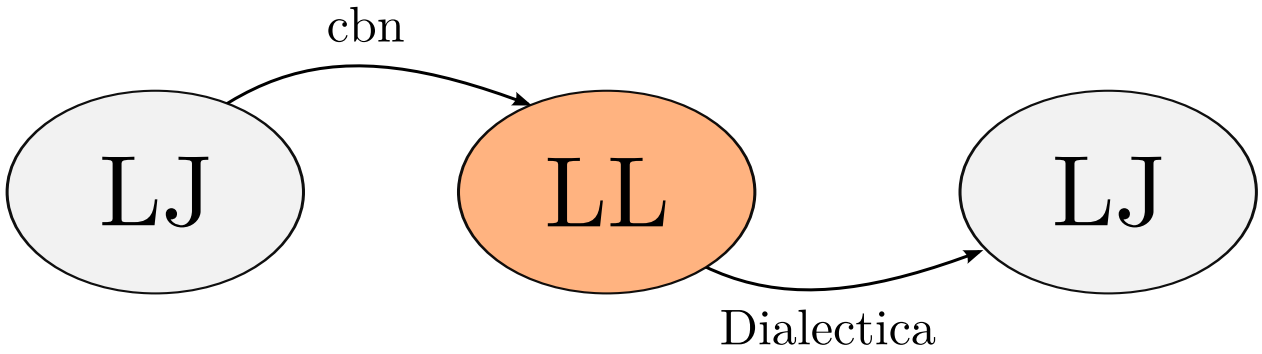
Hence we have true linear constraints![2]

[1] Assuming we've defined 1.

[2] May contain nuts.

Intepretation of the call-by-name λ-calculus

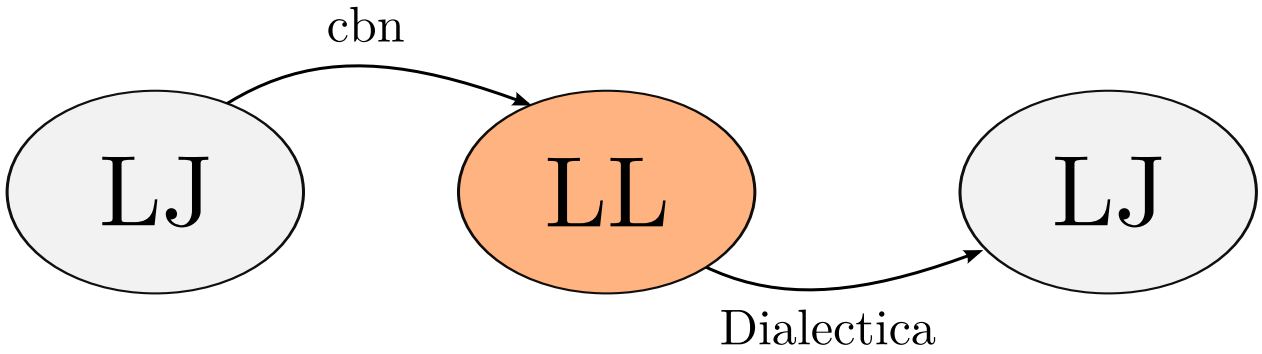
Let us now try to translate the good old λ-calculus through Dialectica.



- First through the call-by-name linear decomposition into $\mathbf{LL}$;
- Then into $\mathbf{LJ}$ with the linear Dialectica.

Intepretation of the call-by-name λ-calculus

Let us now try to translate the good old λ-calculus through Dialectica.



- First through the call-by-name linear decomposition into $\mathbf{LL}$;
- Then into $\mathbf{LJ}$ with the linear Dialectica.

We already did that when translating $\mathbf{HA}$. I'm just expliciting the proof terms we were translating!

Brief reminder

We recall here the call-by-name translation of the λ-calculus into $\mathbf{LL}$:

$$[\![A \Rightarrow B]\!] \equiv ![\![A]\!] \multimap [\![B]\!]$$

$$[\![\Gamma \vdash A]\!] \equiv \bigotimes ![\![\Gamma]\!] \vdash [\![A]\!]$$

Indeed, we recover the same translation as before:

$$\begin{array}{lcl} \mathbb{W}(A \Rightarrow B) & \cong & (\mathbb{W}(A) \Rightarrow \mathbb{W}(B)) \times (\mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A)) \\ \mathbb{C}(A \Rightarrow B) & \cong & \mathbb{W}(A) \times \mathbb{C}(A) \end{array}$$

with the same orthogonality.

$$\frac{\text{if} \quad u \perp_A \varphi \; u \; \pi \quad \text{then} \quad f \; u \perp_B \pi}{(f, \varphi) \perp_{A \Rightarrow B} (u, \pi)}$$

Say it again?

In order to interpret the λ-calculus, we need the same structure as in the intepretation of $\mathbf{HA}$.

Dummy term

For all type A, there exists $\vdash \maltese_A : \mathbb{W}(A)$.

Merge term

The $\perp_A$ relation is decidable. In particular, there exists some λ-term

$$\oplus_A : \mathbb{C}(A) \Rightarrow \mathbb{C}(A) \Rightarrow \mathbb{W}(A) \Rightarrow \mathbb{C}(A)$$

with the following behaviour:

$$\pi_1 \oplus_A^x \pi_2 \cong \texttt{if } x \perp_A \pi_1 \texttt{ then } \pi_2 \texttt{ else } \pi_1$$

Did you solve the organization issue?

If we were to use the translation as is, we would bump up into an unbearable bureaucracy. Instead, we are going to use the following isomorphism.

$$[\![x_1 : \Gamma_1, \ldots x_n : \Gamma_n \vdash t : A]\!] \cong \mathbb{W}(\Gamma) \Rightarrow \begin{cases} \mathbb{W}(A) \\ \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_1) \\ \vdots \\ \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_n) \end{cases}$$

Did you solve the organization issue?

If we were to use the translation as is, we would bump up into an unbearable bureaucracy. Instead, we are going to use the following isomorphism.

$$[\![x_1 : \Gamma_1, \dots x_n : \Gamma_n \vdash t : A]\!] \cong \mathbb{W}(\Gamma) \Rightarrow \begin{cases} \mathbb{W}(A) \\ \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_1) \\ \vdots \\ \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_n) \end{cases}$$

Which results in the following translations:

$$[\![\vec{x} : \Gamma \vdash t : A]\!] \equiv \begin{cases} \vec{x} : \mathbb{W}(\Gamma) \vdash t^\bullet : \mathbb{W}(A) \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^\circ_{x_1} : \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_1) \\ \vdots \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^\circ_{x_n} : \mathbb{C}(A) \Rightarrow \mathbb{C}(\Gamma_n) \end{cases}$$

Translation

For $(-)^\bullet$:

$$
\begin{array}{lcl}
x^\bullet & \equiv & x \\
(\lambda x.\, t)^\bullet & \equiv & \left(\begin{array}{l} \lambda x.\, t^\bullet, \\ \lambda x\, \pi.\, t^\circ_x\ \pi \end{array} \right) \\
(t\ u)^\bullet & \equiv & (\mathtt{fst}\ t^\bullet)\ u^\bullet
\end{array}
$$

Translation II

For t°_x :

$$
\begin{array}{lcl}
x^\circ_x & \equiv & \lambda\pi.\,\pi \\
 & : & \mathbb{C}(A) \to \mathbb{C}(A) \\
y^\circ_x & \equiv & \lambda\pi.\,\maltese_{\Gamma_i} \\
 & : & \mathbb{C}(A) \to \mathbb{C}(\Gamma_i) \\
(\lambda y.\,t)^\circ_x & \equiv & \lambda(y,\pi).\,t^\circ_x\ \pi \\
 & : & \mathbb{W}(A) \times \mathbb{C}(B) \to \mathbb{C}(\Gamma_i) \\
(t\ u)^\circ_x & \equiv & \lambda\pi.\,u^\circ_x\ ((\mathtt{snd}\ t^\bullet)\ \pi\ u^\bullet) \oplus^x_{\Gamma_i}\ t^\circ_x\ (u^\bullet, \pi) \\
 & : & \mathbb{C}(B) \to \mathbb{C}(\Gamma_i)
\end{array}
$$

It just works... Does it?

Soundness

If $\vdash t : A$, then $\vdash t^\bullet : \mathbb{W}(A)$, and in addition, for all $\pi : \mathbb{C}(A)$, $t^\bullet \perp_A \pi$.

It just works... Does it?

Soundness

If $\vdash t : A$, then $\vdash t^\bullet : \mathbb{W}(A)$, and in addition, for all $\pi : \mathbb{C}(A)$, $t^\bullet \perp_A \pi$.

Sadness

The translation is not stable by β-reduction.

The Dialectica translation is **not** a program translation.

Almost there

Using ✠ and ⊕ is another hack by Gödel.

- They rely on typing
- They are non-canonical
- They have no algebraic properties

Almost there

Using ✠ and $\oplus$ is another hack by Gödel.

- They rely on typing
- They are non-canonical
- They have no algebraic properties

We need finite multisets $\mathfrak{M}$!

Minor revision

- We just change $\mathbb{C}(!A) \equiv \mathbb{W}(A) \to \mathfrak{M}\ \mathbb{C}(A)$
- This gives:

$$\begin{array}{lcl} \mathbb{W}(A \Rightarrow B) & := & (\mathbb{W}(A) \Rightarrow \mathbb{W}(B)) \times (\mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathfrak{M}\ \mathbb{C}(A)) \\ \mathbb{C}(A \Rightarrow B) & := & \mathbb{W}(A) \times \mathbb{C}(A) \end{array}$$

$$\frac{\text{if for all } \rho \in \varphi\ u\ \pi, \quad u \perp_A \rho \quad \text{then} \quad f\ u \perp_B \pi}{(f, \varphi) \perp_{A \Rightarrow B} (u, \pi)}$$

- Term interpretation is almost unchanged:
  - $\rightsquigarrow$ ✠ is the empty multiset; $\oplus$ is the union

Minor revision (II)

This variant is actually well-known.

The previous translation is essentially the Diller-Nahm translation.

Minor revision (II)

This variant is actually well-known.

The previous translation is essentially the Diller-Nahm translation.

- ... for totally different reasons
- It does not require the decidability of atoms
- Not at all motivated by proof-as-program considerations

What about the computational content?

This gives us the following types for the translation:

$$[\![\vec{x} : \Gamma \vdash t : A]\!] \equiv \begin{cases} \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\bullet} : \mathbb{W}(A) \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_1} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_1) \\ \vdots \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_n} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_n) \end{cases}$$

What about the computational content?

This gives us the following types for the translation:

$$[\![\vec{x} : \Gamma \vdash t : A]\!] \equiv \begin{cases} \vec{x} : \mathbb{W}(\Gamma) \vdash t^\bullet : \mathbb{W}(A) \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^\circ_{x_1} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_1) \\ \vdots \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^\circ_{x_n} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_n) \end{cases}$$

- $t^\bullet$ is clearly the lifting of t;

What about the computational content?

This gives us the following types for the translation:

$$[\![\vec{x} : \Gamma \vdash t : A]\!] \equiv \begin{cases} \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\bullet} : \mathbb{W}(A) \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_1} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_1) \\ \vdots \\ \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_n} : \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_n) \end{cases}$$

- $t^{\bullet}$ is clearly the lifting of t;
- What on earth is $t^{\circ}_{x_i}$?

An unbearable suspense

A small interlude of ~~advertisement~~ **definitions** to reintroduce you to the KAM.

An unbearable suspense

A small interlude of ~~advertisement~~ **definitions** to reintroduce you to the KAM.

	Closures	c	$::=$	(t, σ)
	Environments	σ	$::=$	$\emptyset \mid \sigma + (x := c)$
	Stacks	π	$::=$	$\varepsilon \mid c \cdot \pi$
	Processes	p	$::=$	$\langle c \mid \pi \rangle$
Push	$\langle (t\, u, \sigma) \mid \pi \rangle$		$\to$	$\langle (t, \sigma) \mid (u, \sigma) \cdot \pi \rangle$
Pop	$\langle (\lambda x.\, t, \sigma) \mid c \cdot \pi \rangle$		$\to$	$\langle (t, \sigma + (x := c)) \mid \pi \rangle$
Grab	$\langle (x, \sigma + (x := c)) \mid \pi \rangle$		$\to$	$\langle c \mid \pi \rangle$
Garbage	$\langle (x, \sigma + (y := c)) \mid \pi \rangle$		$\to$	$\langle (x, \sigma) \mid \pi \rangle$

The Krivine Machine™

This variant has explicit substitutions...

Duality, duality!

It is easy to observe the following similarity:

$$\begin{array}{lrcl} \text{Dialectica realizability} & \mathbb{C}(A \Rightarrow B) & := & \mathbb{W}(A) \times \mathbb{C}(B) \\ \text{Krivine realizability} & ||A \Rightarrow B|| & := & |A| \cdot ||B|| \end{array}$$

Duality, duality!

It is easy to observe the following similarity:

$$\begin{array}{lrcl} \text{Dialectica realizability} & \mathbb{C}(A \Rightarrow B) & := & \mathbb{W}(A) \times \mathbb{C}(B) \\ \text{Krivine realizability} & ||A \Rightarrow B|| & := & |A| \cdot ||B|| \end{array}$$

Thus Dialectica gives a first-class status to Krivine stacks (like LRS).

In particular, the $(-)^\bullet$ translation naturally extends to stacks.

Closures all the way down

Let:

- a term $\vec{x} : \Gamma \vdash t : A \quad \rightsquigarrow \quad \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_i} : \mathbb{C}(A) \Rightarrow \mathfrak{M}\ \mathbb{C}(\Gamma_i)$
- a closure $\sigma \vdash \Gamma \quad \rightsquigarrow \quad \sigma^{\bullet} : \mathbb{W}(\Gamma)$
- a stack $\vdash \pi : A^{\perp} \quad \rightsquigarrow \quad \pi^{\bullet} : \mathbb{C}(A)$

Closures all the way down

Let:

- a term $\vec{x} : \Gamma \vdash t : A \quad \rightsquigarrow \quad \vec{x} : \mathbb{W}(\Gamma) \vdash t^{\circ}_{x_i} : \mathbb{C}(A) \Rightarrow \mathfrak{M}\ \mathbb{C}(\Gamma_i)$
- a closure $\sigma \vdash \Gamma \quad \rightsquigarrow \quad \sigma^{\bullet} : \mathbb{W}(\Gamma)$
- a stack $\vdash \pi : A^{\perp} \quad \rightsquigarrow \quad \pi^{\bullet} : \mathbb{C}(A)$

Then

$$(t^{\circ}_{x_i}\{\vec{x} := \sigma^{\bullet}\})\,\pi^{\bullet} : \mathfrak{M}\ \mathbb{C}(\Gamma_i)$$

is made of **the stacks encountered by** x_i while evaluating $\langle (t, \sigma) \mid \pi \rangle$, i.e.

$$(t^{\circ}_{x_i}\{\vec{x} := \sigma^{\bullet}\})\,\pi^{\bullet} = [\rho_1; \dots \rho_m]$$

$$\begin{array}{lll} \langle (t, \sigma) \mid \pi \rangle & \longrightarrow^* & \langle (x_i, \sigma_1) \mid \rho_1 \rangle \\ & \vdots & \vdots \\ & \longrightarrow^* & \langle (x_i, \sigma_m) \mid \rho_m \rangle \end{array}$$

Look

$$
\begin{array}{lcl}
x^{\circ}_{x} & \equiv & \lambda\pi.\,[\pi] \\
 & : & \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(A) \\
y^{\circ}_{x} & \equiv & \lambda\pi.\,[\,] \\
 & : & \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i) \\
(\lambda y.\,t)^{\circ}_{x} & \equiv & \lambda(y,\pi).\,t^{\circ}_{x}\,\pi \\
 & : & \mathbb{W}(A) \times \mathbb{C}(B) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i) \\
(t\,u)^{\circ}_{x} & \equiv & \lambda\pi.\,(((\mathtt{snd}\ t^{\bullet})\ \pi\ u^{\bullet})\ \gg\!=\ u^{\circ}_{x}) \oplus t^{\circ}_{x}\,(u^{\bullet},\pi) \\
 & : & \mathbb{C}(B) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i)
\end{array}
$$

Look

$$
\begin{array}{lcl}
x^{\circ}_{x} & \equiv & \lambda\pi.\,[\pi] \\
 & : & \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(A) \\
y^{\circ}_{x} & \equiv & \lambda\pi.\,[\,] \\
 & : & \mathbb{C}(A) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i) \\
(\lambda y.\,t)^{\circ}_{x} & \equiv & \lambda(y,\pi).\,t^{\circ}_{x}\,\pi \\
 & : & \mathbb{W}(A) \times \mathbb{C}(B) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i) \\
(t\,u)^{\circ}_{x} & \equiv & \lambda\pi.\,(((\mathtt{snd}\ t^{\bullet})\ \pi\ u^{\bullet}) \gg\!= u^{\circ}_{x}) \oplus t^{\circ}_{x}\,(u^{\bullet},\pi) \\
 & : & \mathbb{C}(B) \to \mathfrak{M}\ \mathbb{C}(\Gamma_i)
\end{array}
$$

(We can generalize to algebraic datatypes directly.)

Implementation-wise

For computer scientists:

Dialectica instruments dereferencing by:

1. capturing the current stack
2. storing it somewhere
3. returning it when the function returns

This is the only effect I know of which is sensitive to substitution. Hence the linearish nature...

Dialectica Reloaded

- The standard Dialectica only returns one stack
  - $\rightsquigarrow$ the first non-dummy stack, dynamically tested

Dialectica Reloaded

- The standard Dialectica only returns one stack
  - $\rightsquigarrow$ the first non-dummy stack, dynamically tested
- This is somehow a weak form of delimited control
  - $\rightsquigarrow$ Inspectable stacks: $\sim A$ vs. $\neg A$
  - $\rightsquigarrow$ First class access to those stacks with $(-)^{\circ}_{x}$

Dialectica Reloaded

- The standard Dialectica only returns one stack
  - $\rightsquigarrow$ the first non-dummy stack, dynamically tested
- This is somehow a weak form of delimited control
  - $\rightsquigarrow$ Inspectable stacks: $\sim A$ vs. $\neg A$
  - $\rightsquigarrow$ First class access to those stacks with $(-)^\circ_x$
- We can do the same thing with other calling conventions
  - $\rightsquigarrow$ The protohistoric Dialectica was call-by-name
  - $\rightsquigarrow$ Choose your favorite translation into LL!

Simulating the KAM simulation

Actually, there is something wrong.

Simulating the KAM simulation

Actually, there is something wrong.

- Produced stacks are the right ones...

Simulating the KAM simulation

Actually, there is something wrong.

- Produced stacks are the right ones...
- They have the right multiplicity...

Simulating the KAM simulation

Actually, there is something wrong.

- Produced stacks are the right ones...
- They have the right multiplicity...
- But they are not respecting the KAM order!
- This is because of **finite multisets**

Simulating the KAM simulation

Actually, there is something wrong.

- Produced stacks are the right ones...
- They have the right multiplicity...
- But they are not respecting the KAM order!
- This is because of **finite multisets**

The faulty one is the application case (more generally duplication).

$$(t\,u)_x \equiv \lambda\pi.\,(((\mathtt{snd}\ t^\bullet)\ \pi\ u^\bullet) \gg\!= u_x^\circ) \oplus t_x^\circ\ (u^\bullet, \pi)$$

A deep issue

- The KAM imposes us sequentiality
- We want to reflect it into the translation

A deep issue

- The KAM imposes us sequentiality
- We want to reflect it into the translation
- Alas, no way to do that
- The $\mathcal{Y}$ translation is far too symmetrical
  - $\rightsquigarrow$ We want *interleaving*
  - $\rightsquigarrow$ Dialectica can't achieve it as is
  - $\rightsquigarrow$ Polarization? Tensorial logic? Dump Dialectica?

What about realized principles?

We do not reach the historic Dialectica.

- **IP** comes from the realizability part (no **IP**)
- In our setting we only realize a weak version of **MP**

$$\widetilde{\mathbf{MP}} : \sim (\forall x. \sim P\ x) \to \exists x. P\ x$$

where $\sim$ is a weak negation:

$$\sim A := A \Rightarrow \perp\!\!\!\perp$$

akin to $\perp$ from **LL**:

$$\begin{aligned} \mathbb{W}(\perp\!\!\!\perp) &:= 1 \\ \mathbb{C}(\perp\!\!\!\perp) &:= 1 \end{aligned}$$

(no proof of orthogonality)

i.e. a type with paraproofs but no proofs!

Exploiting weak negations

In particular,

$$\begin{array}{lcl} \mathbb{W}(\sim A) & \cong & \mathbb{W}(A) \Rightarrow \mathfrak{M}\ \mathbb{C}(A) \\ \mathbb{C}(\sim A) & \cong & \mathbb{W}(A) \end{array}$$

$$\frac{\text{not for all } \rho \in f\ u, \quad u \perp_A \rho}{f \perp_{\sim A} u}$$

so that

$$\mathbb{W}(\sim \forall x.\ \sim P) \cong (\mathbb{N} \Rightarrow \mathbb{W}(P) \Rightarrow \mathfrak{M}\ \mathbb{C}(P)) \Rightarrow \mathfrak{M}\ (\mathbb{N} \times \mathbb{W}(P))$$

$$\mathbb{C}(\sim \forall x.\ \sim P) \cong \mathbb{N} \Rightarrow \mathbb{W}(P) \Rightarrow \mathfrak{M}\ \mathbb{C}(P)$$

and

$$t \Vdash \sim \forall x.\ \sim P := \forall \pi. \neg(\forall (x, u) \in t\ \pi. \neg(\forall \rho \in \pi\ x\ u. u \perp_P \rho))$$

Pfu!!

In the end, we're able to recover a witness from the fact that

- P is decidable
- the multisets are finite

$$t \Vdash \sim \forall x.\ \sim P \cong \forall \pi. \exists (x, u) \in t\ \pi. \forall \rho \in \pi\ x\ u. u \perp_P \rho$$

This requires crawling through the various multisets to do so.

Instantiate π by a function producing an empty multiset and you're done!

Towards CC^ω

- What about more expressive systems?
- We follow the computation intuition we presented
- ... and we apply Dialectica to dependent types
  - $\rightsquigarrow$ subsuming first-order logic;
  - $\rightsquigarrow$ a proof-relevant $\forall$;
  - $\rightsquigarrow$ towards CC^ω and further!

Main lines

- We keep the CBN λ-calculus
  - $\rightsquigarrow$ it can be lifted readily to dependent types
  - $\rightsquigarrow$ nothing special to do!

Main lines

- We keep the CBN λ-calculus
  - $\rightsquigarrow$ it can be lifted readily to dependent types
  - $\rightsquigarrow$ nothing special to do!
- Design choice: types have no computational content (effect-free):
  - $\rightsquigarrow$ a bit disappointing;
  - $\rightsquigarrow$ but it works...
  - $\rightsquigarrow$ and the usual CC presentation does not help much!

Type translation

Idea: if A is a type,

$$
\begin{aligned}
A^\bullet &\equiv \quad (\mathbb{W}(A), \mathbb{C}(A)) : \texttt{Type} \times \texttt{Type} \\
A_x &\equiv \qquad\qquad \lambda\pi.\,[] \qquad\qquad \text{(effect-free)}
\end{aligned}
$$

Type translation

Idea: if A is a type,

$$\begin{array}{lll} A^\bullet \equiv & (\mathbb{W}(A), \mathbb{C}(A)) : \mathtt{Type} \times \mathtt{Type} & \\ A_x \equiv & \lambda\pi.\,[] & \text{(effect-free)} \end{array}$$

We get:

$$\begin{array}{lcl} \mathtt{Type}^\bullet & \equiv & (\mathtt{Type} \times \mathtt{Type}, 1) \\ \mathtt{Type}_x & \equiv & \lambda\pi.\,[\,] \\ (\Pi y : A.\,B)^\bullet & \equiv & \left(\begin{array}{c} (\Pi y : \mathbb{W}(A).\,\mathbb{W}(B)) \\ \times \\ (\Pi y : \mathbb{W}(A).\,\mathbb{C}(B) \to \mathfrak{M}\ \mathbb{C}(A)) \end{array} , \Sigma y : \mathbb{W}(A).\,\mathbb{C}(B) \right) \\ (\Pi y : A.\,B)_x & \equiv & \lambda\pi.\,[\,] \end{array}$$

Soundness

The translation is sound, but it's not really pure CIC.

Soundness

The translation is sound, but it's not really pure CIC.

- We need finite multisets
  - HITs, HITs, HITs!
- We need some commutative cut rules
  - First class (read: negative) records may do the trick
- Or extensionality hammer
  - Maybe Oury-like tricks

Inductives, please

- We can obtain dependent destruction quite easily
- Just tweak the linear decomposition and there you go!

Conclusion

- Actually, Dialectica is quite simple.
  - $\rightsquigarrow$ ... at least once we removed encoding artifacts

Conclusion

- Actually, Dialectica is quite simple.
  - $\rightsquigarrow$ ... at least once we removed encoding artifacts
- It is an approximation of two side-effects:
  - $\rightsquigarrow$ A bit of delimited control (the $(-)_x$ part)
  - $\rightsquigarrow$ A form of exceptions (with $\varnothing$)

Conclusion

- Actually, Dialectica is quite simple.
  - $\rightsquigarrow$ ... at least once we removed encoding artifacts
- It is an approximation of two side-effects:
  - $\rightsquigarrow$ A bit of delimited control (the $(-)_x$ part)
  - $\rightsquigarrow$ A form of exceptions (with $\varnothing$)
- But is is partially wrong:
  - $\rightsquigarrow$ it is oblivious of sequentiality
  - $\rightsquigarrow$ how can we fix it?

Conclusion

- Actually, Dialectica is quite simple.
  - $\rightsquigarrow$ ... at least once we removed encoding artifacts
- It is an approximation of two side-effects:
  - $\rightsquigarrow$ A bit of delimited control (the $(-)_x$ part)
  - $\rightsquigarrow$ A form of exceptions (with $\varnothing$)
- But is is partially wrong:
  - $\rightsquigarrow$ it is oblivious of sequentiality
  - $\rightsquigarrow$ how can we fix it?
- The delimited control part can be lifted seamlessly to CC^{ω}
  - $\rightsquigarrow$ as soon as we have a little bit more than CC
  - $\rightsquigarrow$ we need a more computation-relevant presentation of CC

Thanks for your attention.