A journey through in Krivine realizability

PART II

Étienne Miquey

ÉNS de Lyon, LIP

Seminário de Lógica Matemática
23/11/2020

ENS DE LYON

UNIVERSITÉ DE LYON

Introduction

A short recap on last week's talk

Krivine realizability, from above

- A **complete reformulation** of intuitionistic realizability.

  Necessary reformulation:

  $$\forall x.(H(x) \vee \neg H(x)) \textit{ not realized}$$

- Computational **classical logic**:
  - duality between terms / contexts
  - interaction player / opponent

- **Powerful tool** to:
  - prove normalization/soundness properties
  - analyze computational behaviours of programs
  - build new models

  (today's talk)

Krivine realizability, from inside

A 3-steps recipe

1. an operational semantics
2. a logical language
3. formulas interpretation

Krivine realizability, from inside

A 3-steps recipe

1. an operational semantics (*a.k.a. the abstract Krivine machine*)

$$\begin{array}{lrcl} \textsc{Push} : & \langle tu \parallel \pi \rangle & \succ & \langle t \parallel u \cdot \pi \rangle \\ \textsc{Grab} : & \langle \lambda x.t \parallel u \cdot \pi \rangle & \succ & \langle t\{x := u\} \parallel \pi \rangle \\ \textsc{Save} : & \langle \mathbf{cc} \parallel t \cdot \pi \rangle & \succ & \langle t \parallel \mathbf{k}_\pi \cdot \pi \rangle \\ \textsc{Restore} : & \langle \mathbf{k}_\pi \parallel t \cdot \rho \rangle & \succ & \langle t \parallel \pi \rangle \end{array}$$

2. a logical language (*a.k.a. a type system*)

1st-order terms	e	$::=$	$x \mid f(e_1, \ldots, e_k)$
Formulas	A, B	$::=$	$X(e_1, \ldots, e_k) \mid A \Rightarrow B \mid \forall x.A \mid \forall X.A$

3. formulas interpretation

Realizability interpretation

Intuition

- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: proofs, player of A
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| \pi \rangle \succ p_0 \succ \cdots \succ p_n \in \perp\!\!\!\perp ?$$

$\leadsto$ $\perp\!\!\!\perp \subset \Lambda \times \Pi$ *closed by anti-reduction*

Falsity value (tests):

$$\begin{array}{rcl} \|A \to B\| & = & \{u \cdot \pi : \quad u \in |A| \wedge \pi \in \|B\|\} \\ \|\forall x.A\| & = & \bigcup_{n \in \mathbb{N}} \|A[n/x]\| \end{array}$$

Truth value by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall \pi \in \|A\|, \langle t \| \pi \rangle \in \perp\!\!\!\perp\}$$

Realizability interpretation

Intuition

- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: proofs, player of A
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \parallel \pi \rangle \succ p_0 \succ \cdots \succ p_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subset \Lambda \times \Pi$ *closed by anti-reduction*

Falsity value (tests):

$$\begin{array}{rcl} \|A \to B\| & = & \{u \cdot \pi : \quad u \in |A| \wedge \pi \in \|B\|\} \\ \|\forall x.A\| & = & \bigcup_{n \in \mathbb{N}} \|A[n/x]\| \end{array}$$

Truth value by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall \pi \in \|A\|, \langle t \parallel \pi \rangle \in \perp\!\!\!\perp\}$$

Realizability interpretation

Intuition

- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: proofs, player of A
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \parallel \pi \rangle \succ p_0 \succ \cdots \succ p_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subset \Lambda \times \Pi$ *closed by anti-reduction*

Falsity value (tests):

$$\begin{array}{rcl} \|A \to B\| & = & \{u \cdot \pi : \ u \in |A| \wedge \pi \in \|B\|\} \\ \|\forall x.A\| & = & \bigcup_{n \in \mathbb{N}} \|A[n/x]\| \end{array}$$

Truth value by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall \pi \in \|A\|, \langle t \parallel \pi \rangle \in \perp\!\!\!\perp\}$$

Realizability interpretation

Intuition

- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: proofs, player of A
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \parallel \pi \rangle \succ p_0 \succ \cdots \succ p_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subset \Lambda \times \Pi$ *closed by anti-reduction*

Falsity value (tests):

$$\begin{array}{rcl} \|A \to B\| & = & \{u \cdot \pi : \; u \in |A| \wedge \pi \in \|B\|\} \\ \|\forall x.A\| & = & \bigcup_{n \in \mathbb{N}} \|A[n/x]\| \end{array}$$

Truth value by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall \pi \in \|A\|, \langle t \parallel \pi \rangle \in \perp\!\!\!\perp\}$$

Results

Adequacy

If $\vdash t : A$ then $t \in |A|$ for any pole.

(intuition: the proof proceeds by normalization)

Consequences:

- Normalization

Typed terms normalize.

- Soundness

There is no term t such that $\vdash t : \bot$.

Results

Adequacy

If $\vdash t : A$ then $t \in |A|$ for any pole.

(intuition: the proof proceeds by normalization)

Consequences:

- Normalization

Typed terms normalize.

- Soundness

There is no term t such that $\vdash t : \bot$.

This talk

Today, we shall dwell on:

- **specification problem**

"Who are the realizers of A?"

- **witness extraction**

Spoiler: *it works for Σ^0_1-formulas.*

- connexion with **forcing**

Spoiler: *realizability generalizes forcing!*

- the **algebraic structure** of realizability models

The wonderland of implicative algebras

This talk

Today, we shall dwell on:

- **specification problem**

"Who are the realizers of A?"

- **witness extraction**

Spoiler: *it works for Σ_1^0-formulas.*

- connexion with **forcing**

Spoiler: *realizability generalizes forcing!*

- the **algebraic structure** of realizability models

The wonderland of implicative algebras

This talk

Today, we shall dwell on:

- **specification problem**

"Who are the realizers of A?"

- **witness extraction**

Spoiler: *it works for Σ_1^0-formulas.*

- connexion with **forcing**

Spoiler: *realizability generalizes forcing!*

- the **algebraic structure** of realizability models

The wonderland of implicative algebras

This talk

Today, we shall dwell on:

- **specification problem**

"Who are the realizers of A?"

- **witness extraction**

Spoiler: *it works for Σ_1^0-formulas.*

- connexion with **forcing**

Spoiler: *realizability generalizes forcing!*

- the **algebraic structure** of realizability models

The wonderland of implicative algebras

Specification

Who are the (universal) realizers of A?

Building poles.

Two ways of building poles from any set P of processes.

- goal-oriented :

$$\perp\!\!\!\perp \equiv \{p \in \Lambda_c \times \Pi : \exists p' \in P,\ p > p'\}$$

- thread-oriented :

Definition

Thread of a process p : $\quad \text{th}_p = \{p' \in \Lambda_c \times \Pi : p > p'\}$

$$\perp\!\!\!\perp \equiv (\bigcup_{p \in P} \text{th}_p)^c \equiv \bigcap_{p \in P} \text{th}_p^c$$

Building poles.

Two ways of building poles from any set P of processes.

- goal-oriented :

$$⫫ \equiv \{p \in \Lambda_c \times \Pi : \exists p' \in P,\ p \succ p'\}$$

- thread-oriented :

Definition

Thread of a process p : $\quad \text{th}_p = \{p' \in \Lambda_c \times \Pi : p \succ p'\}$

$$⫫ \equiv \Big(\bigcup_{p \in P} \text{th}_p\Big)^c \equiv \bigcap_{p \in P} \text{th}_p^c$$

Building poles.

Two ways of building poles from any set P of processes.

- goal-oriented :

$$\perp\!\!\!\perp \equiv \{p \in \Lambda_c \times \Pi : \exists p' \in P,\ p \succ p'\}$$

- thread-oriented :

Definition

Thread of a process p : $\quad \text{th}_p = \{p' \in \Lambda_c \times \Pi : p \succ p'\}$

$$\perp\!\!\!\perp \equiv \Big(\bigcup_{p \in P} \text{th}_p\Big)^c \equiv \bigcap_{p \in P} \text{th}_p^c$$

Building poles.

Two ways of building poles from any set P of processes.

- goal-oriented :

$$\perp\!\!\!\perp \equiv \{p \in \Lambda_c \times \Pi : \exists p' \in P,\ p \succ p'\}$$

- thread-oriented :

Definition

Thread of a process p : $\quad \mathrm{th}_p = \{p' \in \Lambda_c \times \Pi : p \succ p'\}$

$$\perp\!\!\!\perp \equiv (\bigcup_{p \in P} \mathrm{th}_p)^c \equiv \bigcap_{p \in P} \mathrm{th}_p^c$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi\rangle \succ \langle u \parallel \pi\rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\perp\!\!\!\perp := \{p : p \succ \langle u \parallel \pi\rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad\qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\| = \bigcup_{\mathbb{X}\in\mathcal{P}(\Pi)}\{u \cdot \pi : u \in |\mathbb{X}| \wedge \pi \in \mathbb{X}\}$$

Therefore,

$$\langle t \parallel u \cdot \pi\rangle \in \perp\!\!\!\perp$$

i.e.

$$\langle t \parallel u \cdot \pi\rangle \succ \langle u \parallel \pi\rangle \qquad ✓$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \| u \cdot \pi \rangle \succ \langle u \| \pi \rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\perp\!\!\!\perp := \{p : p \succ \langle u \| \pi \rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad\qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\| = \bigcup_{\mathbb{X} \in \mathcal{P}(\Pi)} \{u \cdot \pi : u \in |\mathbb{X}| \wedge \pi \in \mathbb{X}\}$$

Therefore,

$$\langle t \| u \cdot \pi \rangle \in \perp\!\!\!\perp$$

i.e.

$$\langle t \| u \cdot \pi \rangle \succ \langle u \| \pi \rangle \qquad ✓$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \| u \cdot \pi\rangle \succ \langle u \| \pi\rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\bot\!\!\!\bot := \{p : p \succ \langle u \| \pi\rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\|$$

$$= \bigcup_{X \in \mathcal{P}(\Pi)} \{u \cdot \pi : u \in |X| \wedge \pi \in X\}$$

Therefore,

$$\langle t \| u \cdot \pi\rangle \in \bot\!\!\!\bot$$

i.e.

$$\langle t \| u \cdot \pi\rangle \succ \langle u \| \pi\rangle \qquad \checkmark$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \| u \cdot \pi \rangle \succ \langle u \| \pi \rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\perp\!\!\!\perp := \{p : p \succ \langle u \| \pi \rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad\qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\|$$

$$= \bigcup_{\mathbb{X} \in \mathcal{P}(\Pi)} \{u \cdot \pi : u \in |\mathbb{X}| \wedge \pi \in \mathbb{X}\}$$

Therefore,

$$\langle t \| u \cdot \pi \rangle \in \perp\!\!\!\perp$$

i.e.

$$\langle t \| u \cdot \pi \rangle \succ \langle u \| \pi \rangle \qquad \checkmark$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi\rangle \succ \langle u \parallel \pi\rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\perp\!\!\!\perp := \{p : p \succ \langle u \parallel \pi\rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad\qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\|$$

$$= \bigcup_{\mathbb{X} \in \mathcal{P}(\Pi)} \{u \cdot \pi : u \in |\mathbb{X}| \wedge \pi \in \mathbb{X}\}$$

Therefore,

$$\langle t \parallel u \cdot \pi\rangle \in \perp\!\!\!\perp$$

i.e.

$$\langle t \parallel u \cdot \pi\rangle \succ \langle u \parallel \pi\rangle \quad ✓$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle$$

Proof :

Method 1 - Goal-oriented

Take u, π, define $\perp\!\!\!\perp := \{p : p \succ \langle u \parallel \pi \rangle\}$.

Let us pose $\mathbb{X} = \{\pi\}$. In particular, we have:

$$u \Vdash \mathbb{X} \qquad\qquad u \cdot \pi \in \|\forall X.(X \Rightarrow X)\|$$

$$= \bigcup_{\mathbb{X} \in \mathcal{P}(\Pi)} \{u \cdot \pi : u \in |\mathbb{X}| \wedge \pi \in \mathbb{X}\}$$

Therefore,

$$\langle t \parallel u \cdot \pi \rangle \in \perp\!\!\!\perp$$

i.e.

$$\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle \qquad ✓$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \,\|\, u \cdot \pi \rangle \succ \langle u \,\|\, \pi \rangle$$

Proof :

Method 2 - Using threads

Take u, π, define $\perp\!\!\!\perp \triangleq \mathrm{th}^c_{\langle t \,\|\, u\cdot\pi\rangle} = \{p : \langle t \,\|\, u \cdot \pi\rangle \not\succ p\}$.

By construction:

$$\langle t \,\|\, u \cdot \pi\rangle \notin \perp\!\!\!\perp \qquad \text{thus} \qquad u \cdot \pi \notin \|\forall X.(X \Rightarrow X)\|$$

Let us pose $\mathbb{X} \triangleq \{\pi\}$, we deduce

$$u \not\Vdash \mathbb{X} \qquad \text{i.e.} \qquad \exists \pi' \in \mathbb{X}.\langle u \,\|\, \pi'\rangle \notin \perp\!\!\!\perp$$

Necessarily, $\pi = \pi'$ and so $\langle u \,\|\, \pi\rangle \notin \perp\!\!\!\perp$, i.e.:

$$\langle t \,\|\, u \cdot \pi\rangle \succ \langle u \,\|\, \pi\rangle$$ ✓

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle$$

Proof :

Method 2 - Using threads

Take u, π, define $\perp\!\!\!\perp \triangleq \text{th}^c_{\langle t \parallel u \cdot \pi \rangle} = \{p : \langle t \parallel u \cdot \pi \rangle \not\succ p\}$.

By construction:

$$\langle t \parallel u \cdot \pi \rangle \notin \perp\!\!\!\perp \qquad \text{thus} \qquad u \cdot \pi \notin \|\forall X.(X \Rightarrow X)\|$$

Let us pose $\mathbb{X} \triangleq \{\pi\}$, we deduce

$$u \not\Vdash \mathbb{X} \qquad \text{i.e.} \qquad \exists \pi' \in \mathbb{X}.\langle u \parallel \pi' \rangle \notin \perp\!\!\!\perp$$

Necessarily, $\pi = \pi'$ and so $\langle u \parallel \pi \rangle \notin \perp\!\!\!\perp$, i.e.:

$$\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle \qquad \checkmark$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle$$

Proof :

Method 2 - Using threads

Take u, π, define $\perp\!\!\!\perp \triangleq \mathrm{th}^c_{\langle t \parallel u \cdot \pi \rangle} = \{p : \langle t \parallel u \cdot \pi \rangle \not\succ p\}$.

By construction:

$$\langle t \parallel u \cdot \pi \rangle \notin \perp\!\!\!\perp \qquad \text{thus} \qquad u \cdot \pi \notin \| \forall X.(X \Rightarrow X) \|$$

Let us pose $\mathbb{X} \triangleq \{\pi\}$, we deduce

$$u \not\Vdash \mathbb{X} \qquad \text{i.e.} \qquad \exists \pi' \in \mathbb{X}.\langle u \parallel \pi' \rangle \notin \perp\!\!\!\perp$$

Necessarily, $\pi = \pi'$ and so $\langle u \parallel \pi \rangle \notin \perp\!\!\!\perp$, i.e.:

$$\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle \qquad ✓$$

Example

Proposition

$$t \Vdash \forall X.(X \Rightarrow X) \qquad \text{iff} \qquad \forall u.\forall \pi.\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle$$

Proof :

Method 2 - Using threads

Take u, π, define $\perp\!\!\!\perp \triangleq \text{th}^c_{\langle t \parallel u \cdot \pi \rangle} = \{p : \langle t \parallel u \cdot \pi \rangle \not\succ p\}$.

By construction:

$$\langle t \parallel u \cdot \pi \rangle \notin \perp\!\!\!\perp \qquad \text{thus} \qquad u \cdot \pi \notin \|\forall X.(X \Rightarrow X)\|$$

Let us pose $\mathbb{X} \triangleq \{\pi\}$, we deduce

$$u \not\Vdash \mathbb{X} \qquad \text{i.e.} \qquad \exists \pi' \in \mathbb{X}.\langle u \parallel \pi' \rangle \notin \perp\!\!\!\perp$$

Necessarily, $\pi = \pi'$ and so $\langle u \parallel \pi \rangle \notin \perp\!\!\!\perp$, i.e.:

$$\langle t \parallel u \cdot \pi \rangle \succ \langle u \parallel \pi \rangle$$

✓

Σ_1-formulas

What about

$$t \Vdash \exists x. f(x) = 0 \qquad \text{iff} \qquad ??$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

Σ_1-formulas

$$t \Vdash \exists x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Remind that :

$$\|\forall x.A\| = \bigcup_{n \in \mathbb{N}} \|A\{x := n\}\|$$

In particular, *n does not appear on the stack*!

Fix: *relativized quantifier*

$$A, B \ ::= \ \dots \mid \{e\} \Rightarrow A$$

$$\|\{e\} \Rightarrow A\| \triangleq \{\bar{n} \cdot \pi : [\![e]\!] = n \wedge \pi \in \|A\|\}$$

$$\forall^{\mathbb{N}} x.A(x) \triangleq \forall x.(\{x\} \Rightarrow A(x))$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Remind that :

$$\|\forall x.A\| = \bigcup_{n \in \mathbb{N}} \|A\{x := n\}\|$$

In particular, *n does not appear on the stack*!

Fix: *relativized quantifier*

$$A, B \; ::= \; \ldots \mid \{e\} \Rightarrow A$$

$$\|\{e\} \Rightarrow A\| \triangleq \{\bar{n} \cdot \pi : [\![e]\!] = n \wedge \pi \in \|A\|\}$$

$$\forall^{\mathbb{N}} x.A(x) \triangleq \forall x.(\{x\} \Rightarrow A(x))$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

$$\begin{array}{lll} \langle t \parallel u \cdot \pi \rangle & \succ & \\ \langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \\ & \vdots & \\ \langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\ & \vdots & \\ \langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \end{array}$$

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

$$\begin{array}{lcl} \langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\ \langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \\ & \vdots & \\ \langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\ & \vdots & \\ \langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \end{array}$$

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \,\|\, u \cdot \pi \rangle$?

$$\begin{array}{rcl} \langle t \,\|\, u \cdot \pi \rangle & \succ & \langle u \,\|\, \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\ \langle t_0 \,\|\, u_0 \cdot \pi_0 \rangle & \succ & \\ & \vdots & \\ \langle t_i \,\|\, u_i \cdot \pi_i \rangle & \succ & \langle u \,\|\, \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\ & \vdots & \\ \langle t_k \,\|\, u_k \cdot \pi_k \rangle & \succ & \end{array}$$

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \| u \cdot \pi \rangle$?

$$\begin{array}{rcl} \langle t \| u \cdot \pi \rangle & \succ & \langle u \| \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\ \langle t_0 \| u_0 \cdot \pi_0 \rangle & \succ & \langle u \| \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\ & \vdots & \\ \langle t_i \| u_i \cdot \pi_i \rangle & \succ & \langle u \| \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\ & \vdots & \\ \langle t_k \| u_k \cdot \pi_k \rangle & \succ & \end{array}$$

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

$$\begin{array}{rcl} \langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\ \langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\ & \vdots & \\ \langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\ & \vdots & \\ \langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \end{array}$$

Σ_1-formulas

$$t \Vdash \exists^{\mathbb{N}} x. f(x) = 0 \qquad \text{iff} \qquad ??$$

Recall that:

$$\exists^{\mathbb{N}} x.(f(x) = 0) \equiv \forall X.(\forall x.(\{x\} \Rightarrow (f(x) = 0) \Rightarrow X) \Rightarrow X)$$

$$\|f(x) = 0\| = \begin{cases} \|\forall X.(X \Rightarrow X)\| & \text{if } \mathcal{M} \models e_1 = e_2 \\ \|\top \Rightarrow \bot\| & \text{if } \mathcal{M} \models e_1 \neq e_2 \end{cases}$$

What is the thread of $\langle t \parallel u \cdot \pi \rangle$?

$$\begin{array}{rcll} \langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle & \\ \langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle & \\ & \vdots & & \\ \langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle & \\ & \vdots & & \\ \langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \langle u_s \parallel \pi_s \rangle & (\mathcal{M} \models f(m_s) = 0) \end{array}$$

Witness extraction

Say we have a term :

$$t \Vdash \exists^{\mathbb{N}} x.(f(x) = 0)$$

Goal: we would like to use t to compute some $m \in \mathbb{N}$ st. $f(m) = 0$.

Witness extraction

Say we have a term :

$$t \Vdash \exists^{\mathbb{N}} x.(f(x) = 0)$$

Goal: we would like to use t to compute some $m \in \mathbb{N}$ st. $f(m) = 0$.

$$\begin{array}{rcll}
\langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle & \\
\langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle & \\
& \vdots & & \\
\langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle & \\
& \vdots & & \\
\langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \langle u_s \parallel \pi_s \rangle & (\mathcal{M} \models f(m_s) = 0)
\end{array}$$

Witness extraction

Say we have a term :

$$t \Vdash \exists^{\mathbb{N}} x.(f(x) = 0)$$

Goal: we would like to use t to compute some $m \in \mathbb{N}$ st. $f(m) = 0$.

$$\begin{array}{rcl}
\langle t \,\|\, u \cdot \pi \rangle & \succ & \langle u \,\|\, \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\
\langle t_0 \,\|\, u_0 \cdot \pi_0 \rangle & \succ & \langle u \,\|\, \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\
 & \vdots & \\
\langle t_i \,\|\, u_i \cdot \pi_i \rangle & \succ & \langle u \,\|\, \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\
 & \vdots & \\
\langle t_k \,\|\, u_k \cdot \pi_k \rangle & \succ & \langle u_s \,\|\, \pi_s \rangle \qquad (\mathcal{M} \models f(m_s) = 0)
\end{array}$$

Define $u := \lambda xy.y\,(\texttt{stop}\ x)$

(with stop *a new instruction blocking computations)*

Witness extraction

Say we have a term :

$$t \Vdash \exists^{\mathbb{N}} x.(f(x) = 0)$$

Goal: we would like to use t to compute some $m \in \mathbb{N}$ st. $f(m) = 0$.

$$\begin{array}{rlcl}
 & \langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\
\succ & \langle t_0 \parallel \mathtt{stop}\, m_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\
 & & \vdots & \\
\succ & \langle t_i \parallel \mathtt{stop}\, m_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\
 & & \vdots & \\
\succ & \langle t_k \parallel \mathtt{stop}\, m_k \cdot \pi_k \rangle & \succ & \langle \mathtt{stop}\, m_s \parallel \pi_s \rangle \qquad (\mathcal{M} \models f(m_s) = 0)
\end{array}$$

Define $u := \lambda xy.y\,(\mathtt{stop}\ x)$

(with `stop` *a new instruction blocking computations)*

Witness extraction

Witness extraction [Miquel'11]

If $t \Vdash \exists^{\mathbb{N}} x.(f(x) = 0)$ then $\forall \pi \in \Pi$ there exists $m \in \mathbb{N}$ s.t.:

$$\langle t \parallel \lambda xy.y\,(\mathtt{stop}\ x) \cdot \pi \rangle \succ \langle \mathtt{stop} \parallel \overline{m} \cdot \pi \rangle \qquad \wedge \qquad f(m) = 0$$

Preuve:

$$\begin{array}{rlcl}
 & \langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\
\succ & \langle t_0 \parallel \mathtt{stop}\, m_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\
 & & \vdots & \\
\succ & \langle t_i \parallel \mathtt{stop}\, m_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\
 & & \vdots & \\
\succ & \langle t_k \parallel \mathtt{stop}\, m_k \cdot \pi_k \rangle & \succ & \langle \mathtt{stop}\, m_s \parallel \pi_s \rangle \qquad (\mathcal{M} \models f(m_s) = 0)
\end{array}$$

Define $u := \lambda xy.y\,(\mathtt{stop}\ x)$

(with stop *a new instruction blocking computations)*

Σ_2-formulas?

If we have a term :

$$t \Vdash \exists^{\mathbb{N}} x . \forall^{\mathbb{N}} y . f(x) \leq f(y)$$

then the thread of $p := \langle t \parallel u \cdot \pi \rangle$ is as follows :

$$\begin{array}{ccll}
\langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle & \\
\langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle & \\
 & \vdots & & \\
\langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle & \\
 & \vdots & & \\
\langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \langle u_s \parallel \pi_s \rangle & (f(m_s) \leq f(n_s))
\end{array}$$

Bad news :

$f(m_s) \leq f(n_s)$ is far from implying $\forall y . f(m_s) \leq f(y)$

Σ_2-formulas?

If we have a term :

$$t \Vdash \exists^{\mathbb{N}} x.\forall^{\mathbb{N}} y. f(x) \leq f(y)$$

then the thread of $p := \langle t \parallel u \cdot \pi \rangle$ is as follows :

$$\begin{array}{ccl}
\langle t \parallel u \cdot \pi \rangle & \succ & \langle u \parallel \bar{m}_0 \cdot t_0 \cdot \pi \rangle \\
\langle t_0 \parallel u_0 \cdot \pi_0 \rangle & \succ & \langle u \parallel \bar{m}_1 \cdot t_1 \cdot \pi \rangle \\
 & \vdots & \\
\langle t_i \parallel u_i \cdot \pi_i \rangle & \succ & \langle u \parallel \bar{m}_{i+1} \cdot t_{i+1} \cdot \pi \rangle \\
 & \vdots & \\
\langle t_k \parallel u_k \cdot \pi_k \rangle & \succ & \langle u_s \parallel \pi_s \rangle \qquad (f(m_s) \leq f(n_s))
\end{array}$$

Bad news :

$f(m_s) \leq f(n_s)$ is far from implying $\forall y. f(m_s) \leq f(y)$

Coquand's games

Arithmetical formula

$$\Phi : \exists x_1 \forall y_1 \ldots \exists x_h \forall y_h f(\vec{x}_h, \vec{y}_h) = 0$$

Rules of $\mathbb{G}_\Phi$:

- **Players** : Eloise $\exists$ and Abelard $\forall$.
- **Moves** : - at his turn, each player instantiates his variable
  - Eloise allowed to backtrack
- **Final position** : evaluation of $f(\vec{m}_h, \vec{n}_h) = 0$:
  - true : Eloise wins
  - false : game continues
- Abelard wins if the game never ends

Winning strategy

Way of playing that ensures the victory, independently of the opponent moves.

Coquand's games

Arithmetical formula

$$\Phi : \exists x_1 \forall y_1 \ldots \exists x_h \forall y_h f(\vec{x}_h, \vec{y}_h) = 0$$

Rules of $\mathbb{G}_\Phi$:

- **Players** : Eloise ($\exists$) and Abelard ($\forall$) .
- **Moves** : - at his turn, each player instantiates his variable
  - Eloise allowed to backtrack
- **Final position** : evaluation of $f(\vec{m}_h, \vec{n}_h) = 0$:
  - true : Eloise wins
  - false : game continues
- Abelard wins if the game never ends

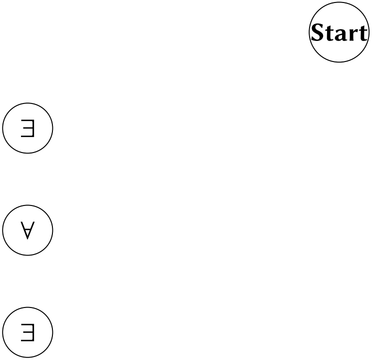
Winning strategy

Way of playing that ensures the victory, independently of the opponent moves.

Example

Formula

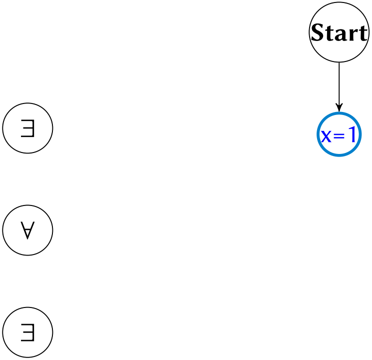
$$\exists x.\, \forall y.\, \exists z.\, x \times y = 2 \times z$$



Example

Formula

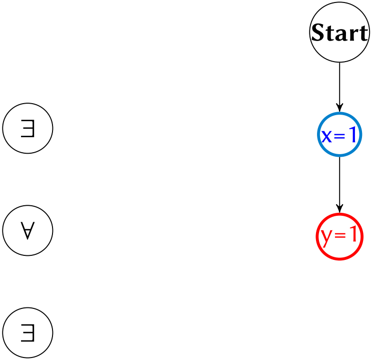
$$\forall y.\, \exists z.\, 1 \times y = 2 \times z$$



Example

Formula

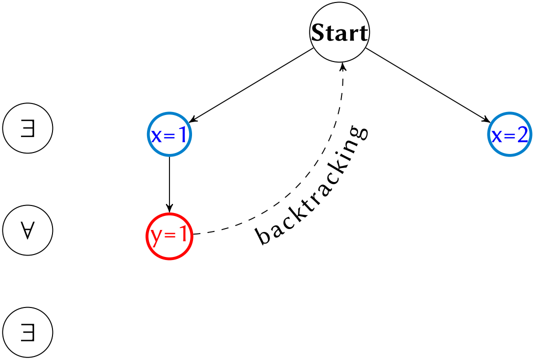
$$\exists z.\, 1 = 2 \times z$$



Example

Formula

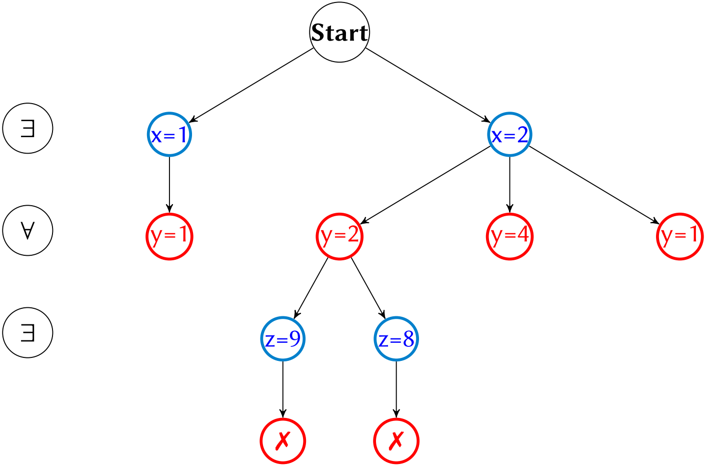
$$\forall y. \exists z. 2 \times y = 2 \times z$$



Example

Formula

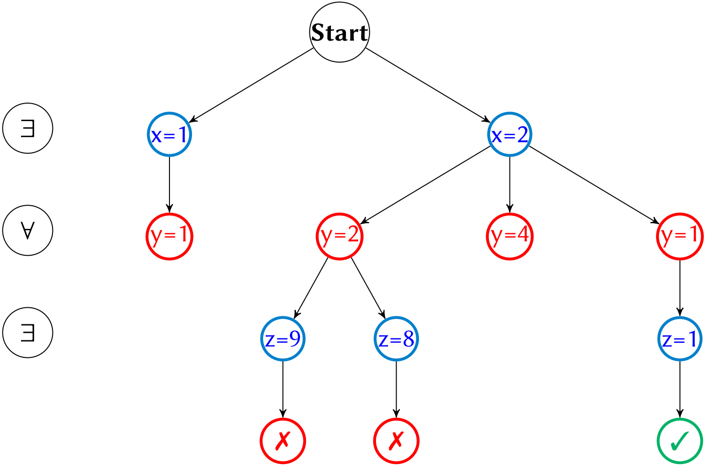
$$\exists z.\ 2 = 2 \times z$$



Example

Formula

$$2 = 2$$



Eloise wins

Arithmetical formulas

Using the threads method, we can show that for any arithmetical formula Φ:

Theorem [Guillermo, M.'15]

$t \Vdash \Phi$ iff t implements a winning strategy for the game $\mathbb{G}_\Phi$

Besides, there exists a winning strategy for $\mathbb{G}_\Phi$ iff $\mathcal{M} \models \Phi$, therefore:

Absoluteness

If Φ is an arithmetical formula, then

$$\exists t \in \Lambda_c, t \Vdash \Phi \quad \text{iff} \quad \mathcal{M} \models \Phi$$

Arithmetical formulas

Using the threads method, we can show that for any arithmetical formula Φ:

Theorem [Guillermo, M.'15]

$t \Vdash \Phi$ iff t implements a winning strategy for the game $\mathbb{G}_\Phi$

Besides, there exists a winning strategy for $\mathbb{G}_\Phi$ iff $\mathcal{M} \models \Phi$, therefore:

Absoluteness

If Φ is an arithmetical formula, then

$$\exists t \in \Lambda_c, t \Vdash \Phi \quad \text{iff} \quad \mathcal{M} \vDash \Phi$$

Realizability & model theory

Theory vs Model

What is the status of axioms (*e.g.* $A \vee \neg A$)?

↬ neither true nor false in the ambient theory

(here, *true* means *provable*)

There is another point of view:

- **Theory**: *provability* in an axiomatic representation (syntax)
- **Model**: *validity* in a particular structure (semantic)

Example:

$A \wedge B$		
A \ B	✓	✗
✓	✓	✗
✗	✗	✗

$A \vee B$		
A \ B	✓	✗
✓	✓	✓
✗	✓	✗

A	$\neg A$	$A \vee \neg A$
✓	✗	✓
✗	✓	✓

Theory vs Model

What is the status of axioms (*e.g.* $A \vee \neg A$)?

↬ neither true nor false in the ambient theory

(here, *true* means *provable*)

There is another point of view:

- **Theory**: *provability* in an axiomatic representation (syntax)
- **Model**: *validity* in a particular structure (semantic)

Example:

$A \wedge B$		
A \ B	✓	✗
✓	✓	✗
✗	✗	✗

$A \vee B$		
A \ B	✓	✗
✓	✓	✓
✗	✓	✗

A	$\neg A$	$A \vee \neg A$
✓	✗	✓
✗	✓	✓

Theory vs Model

What is the status of axioms (*e.g.* $A \vee \neg A$)?

↬ neither true nor false in the ambient theory

(here, *true* means *provable*)

There is another point of view:

- **Theory**: *provability* in an axiomatic representation (syntax)
- **Model**: *validity* in a particular structure (semantic)

Example:

$A \wedge B$		
A \ B	✓	✗
✓	✓	✗
✗	✗	✗

$A \vee B$		
A \ B	✓	✗
✓	✓	✓
✗	✓	✗

A	$\neg A$	$A \vee \neg A$
✓	✗	✓
✗	✓	✓

Theory vs Model

What is the status of axioms (*e.g.* $A \vee \neg A$)?

↬ neither true nor false in the ambient theory

(here, *true* means *provable*)

There is another point of view:

- **Theory**: *provability* in an axiomatic representation (syntax)
- **Model**: *validity* in a particular structure (semantic)

Example:

$A \wedge B$		
A \ B	✓	✗
✓	✓	✗
✗	✗	✗

$A \vee B$		
A \ B	✓	✗
✓	✓	✓
✗	✓	✗

A	$\neg A$	$A \vee \neg A$
✓	✗	✓
✗	✓	✓

Valid formula

Krivine realizability as a model

Krivine realizability:

$$A \mapsto \{t : t \Vdash A\}$$

(intuition: programs that share a common computational behavior given by A)

Tarski

$$A \mapsto |A| \in \mathbb{B}$$

(intuition: level of truthness)

Great news #1

Classical realizability semantics gives surprisingly new models!

(generalize forcing, e.g. direct construction of $\mathcal{M} \models ZF_\varepsilon + \neg CH + \neg AC$)

Great news #2

Classical realizability models have a simple algebraic structure.

(generalize Boolean algebras)

Krivine realizability as a model

Krivine realizability:

$$A \mapsto \{t : t \Vdash A\}$$

(intuition: programs that share a common computational behavior given by A)

Tarski

$$A \mapsto |A| \in \mathbb{B}$$

(intuition: level of truthness)

Great news #1

Classical realizability semantics gives surprisingly new models!

(generalize forcing, e.g. direct construction of $\mathcal{M} \vDash ZF_\varepsilon + \neg CH + \neg AC$*)*

Great news #2

Classical realizability models have a simple algebraic structure.

(generalize Boolean algebras)

Krivine realizability as a model

Krivine realizability:

$$A \mapsto \{t : t \Vdash A\}$$

(intuition: programs that share a common computational behavior given by A)

Tarski

$$A \mapsto |A| \in \mathbb{B}$$

(intuition: level of truthness)

Great news #1

Classical realizability semantics gives surprisingly new models!

(generalize forcing, e.g. direct construction of $\mathcal{M} \vDash ZF_\varepsilon + \neg CH + \neg AC$)

Great news #2

Classical realizability models have a simple algebraic structure.

(generalize Boolean algebras)

Realizability models

Given:

1. a calculus
2. its type system
3. an adequate interpretation of formula
4. a pole $⫫$

one defines a model $\mathcal{M}_{⫫}$ by:

Realizability model

$$\mathcal{M}_{⫫} \vDash A \quad \text{iff} \quad |A| \cap \mathbf{PL} \neq \emptyset$$

(where **PL** is the set of *proof-like* terms)

In other words:

$$A \text{ is satisfied} \triangleq \text{“\textit{there exists a proof-like realizer of } } A\text{”}$$

Realizability models

Given:

1. a calculus
2. its type system
3. an adequate interpretation of formula
4. a pole $\perp\!\!\!\perp$

one defines a model $\mathcal{M}_{\perp\!\!\!\perp}$ by:

Realizability model

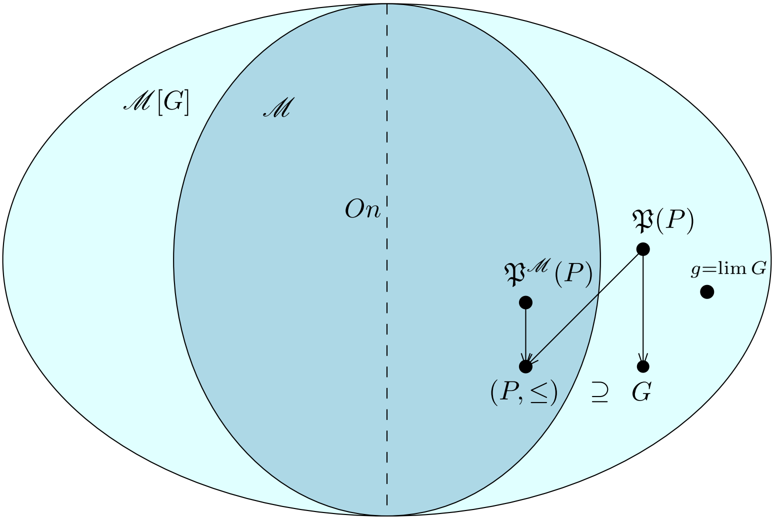
$$\mathcal{M}_{\perp\!\!\!\perp} \vDash A \quad \text{iff} \quad |A| \cap \mathbf{PL} \neq \emptyset$$

(where **PL** is the set of *proof-like* terms)

In other words:

A is satisfied $\triangleq$ "*there exists a proof-like realizer of* A"

Forcing in one picture (©Miquel)



A transformation on *formulæ* [Cohen'63]

- Definition of a *forcing notion* :

> • a poset P of *conditions*, with $\mathbb{1}$ the largest one
> • $p, q \in P$ are *compatible* when $(\exists r \in P)(r \leq p \wedge r \leq q)$

- Definition of the *forcing relation* :

$$\begin{array}{lcl} p \Vdash \neg A & \equiv & \neg(\exists q \preccurlyeq p) q \Vdash A \\ p \Vdash (A \wedge B) & \equiv & (p \Vdash A) \wedge (p \Vdash B) \\ p \Vdash A \Rightarrow B & \equiv & \forall q (q \Vdash A \Rightarrow (\forall r \leq p, q) r \Vdash B) \\ & \vdots & \end{array}$$

- Definition of $\mathcal{M}[G]$ such that $\mathcal{M}[G] \models ZFC$

Forcing Theorem

Let $(P, <)$ be a forcing notion. Then $\forall G \subset P$ generic over $\mathcal{M}$:

$$\mathcal{M}[G] \models A \quad \Leftrightarrow \quad (\exists p \in G) p \Vdash A$$

A transformation on *formulæ* [Cohen'63]

- Definition of a *forcing notion* :

> - a poset P of *conditions*, with $\mathbb{1}$ the largest one
> - $p, q \in P$ are *compatible* when $(\exists r \in P)(r \leq p \wedge r \leq q)$

- Definition of the *forcing relation* :

$$
\begin{array}{lcl}
p \Vdash \neg A & \equiv & \neg(\exists q \preccurlyeq p) q \Vdash A \\
p \Vdash (A \wedge B) & \equiv & (p \Vdash A) \wedge (p \Vdash B) \\
p \Vdash A \Rightarrow B & \equiv & \forall q (q \Vdash A \Rightarrow (\forall r \leq p, q) r \Vdash B) \\
& \vdots &
\end{array}
$$

- Definition of $\mathcal{M}[G]$ such that $\mathcal{M}[G] \models ZFC$

Forcing Theorem

Let $(P, <)$ be a forcing notion. Then $\forall G \subset P$ generic over $\mathcal{M}$:

$$\mathcal{M}[G] \models A \quad \Leftrightarrow \quad (\exists p \in G) p \Vdash A$$

A transformation on *formulæ* [Cohen'63]

- Definition of a *forcing notion* :

> • a poset P of *conditions*, with $\mathbb{1}$ the largest one
> • $p, q \in P$ are *compatible* when $(\exists r \in P)(r \leq p \wedge r \leq q)$

- Definition of the *forcing relation* :

$$
\begin{array}{lcl}
p \Vdash \neg A & \equiv & \neg(\exists q \preccurlyeq p) q \Vdash A \\
p \Vdash (A \wedge B) & \equiv & (p \Vdash A) \wedge (p \Vdash B) \\
p \Vdash A \Rightarrow B & \equiv & \forall q (q \Vdash A \Rightarrow (\forall r \leq p, q) r \Vdash B) \\
& \vdots &
\end{array}
$$

- Definition of $\mathcal{M}[G]$ such that $\mathcal{M}[G] \models ZFC$

Forcing Theorem

Let $(P, <)$ be a forcing notion. Then $\forall G \subset P$ generic over $\mathcal{M}$:

$$\mathcal{M}[G] \models A \quad \Leftrightarrow \quad (\exists p \in G) p \Vdash A$$

A transformation on *formulæ* [Cohen'63]

- Definition of a *forcing notion* :

  - a poset P of *conditions*, with $\mathbb{1}$ the largest one
  - $p, q \in P$ are *compatible* when $(\exists r \in P)(r \leq p \wedge r \leq q)$

- Definition of the *forcing relation* :

$$
\begin{array}{lcl}
p \Vdash \neg A & \equiv & \neg(\exists q \preccurlyeq p) q \Vdash A \\
p \Vdash (A \wedge B) & \equiv & (p \Vdash A) \wedge (p \Vdash B) \\
p \Vdash A \Rightarrow B & \equiv & \forall q (q \Vdash A \Rightarrow (\forall r \leq p, q) r \Vdash B) \\
& \vdots &
\end{array}
$$

- Definition of $\mathcal{M}[G]$ such that $\mathcal{M}[G] \models ZFC$

Forcing Theorem

Let $(P, <)$ be a forcing notion. Then $\forall G \subset P$ generic over $\mathcal{M}$:

$$\mathcal{M}[G] \models A \quad \Leftrightarrow \quad (\exists p \in G) p \Vdash A$$

A transformation on *programs* [Krivine'10]

- A *forcing structure* is given by :

  - a sort κ of *conditions*, with $\mathbf{1}$ the largest one
  - a predicate $C[p]$ (p is well-founded)
  - a closed term $(\cdot)$ for the product
  - a lot of combinators :

$$
\begin{aligned}
\alpha_0 &: C[\mathbf{1}] \\
\alpha_1 &: \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[p]) \\
\alpha_3 &: \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[qp]) \\
\alpha_6 &: \forall p^\kappa \forall q^\kappa \forall r^\kappa (C[p(qr)] \Rightarrow C[(pq)r])
\end{aligned}
$$

- Definition of the forcing relation :

$$
\begin{aligned}
p \Vdash A &\equiv \forall r^\kappa (C[pr] \Rightarrow A^* r) \\
(A \Rightarrow B)^* &\equiv \lambda r^\kappa . \forall q \forall r' \langle r = qr' \rangle (\forall s (C[qs] \Rightarrow A^* s) \Rightarrow B^* r')
\end{aligned}
$$

- Translation on programs:

$$
\begin{aligned}
(tu)^* &\equiv \gamma_3 t^* u^* \\
(\lambda x.t)^* &\equiv \gamma_1 (\lambda x.t^* \{x_i := \beta_3 x_i\}\{x := \beta_4 x\}) \\
\mathbf{cc}^* &\equiv \lambda cx.\mathbf{cc}(\lambda k.x(\alpha_{14} c)(\gamma_4 k))
\end{aligned}
$$

A transformation on *programs* [Krivine'10]

- A *forcing structure* is given by :

  - a sort κ of *conditions*, with $\mathbf{1}$ the largest one
  - a predicate $C[p]$ (p is well-founded)
  - a closed term $(\cdot)$ for the product
  - a lot of combinators :

$$\begin{array}{rcl} \alpha_0 & : & C[\mathbf{1}] \\ \alpha_1 & : & \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[p]) \\ \alpha_3 & : & \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[qp]) \\ \alpha_6 & : & \forall p^\kappa \forall q^\kappa \forall r^\kappa (C[p(qr)] \Rightarrow C[(pq)r]) \end{array}$$

- Definition of the forcing relation :

$$\begin{array}{rcl} p \Vdash A & \equiv & \forall r^\kappa (C[pr] \Rightarrow A^* r) \\ (A \Rightarrow B)^* & \equiv & \lambda r^\kappa . \forall q \forall r' \langle r = qr' \rangle (\forall s (C[qs] \Rightarrow A^* s) \Rightarrow B^* r') \end{array}$$

- Translation on programs:

$$\begin{array}{rcl} (tu)^* & \equiv & \gamma_3 t^* u^* \\ (\lambda x.t)^* & \equiv & \gamma_1 (\lambda x. t^* \{x_i := \beta_3 x_i\}\{x := \beta_4 x\}) \\ \mathbf{cc}^* & \equiv & \lambda c x. \mathbf{cc} (\lambda k. x(\alpha_{14} c)(\gamma_4 k)) \end{array}$$

A transformation on *programs* [Krivine'10]

- A *forcing structure* is given by :

  - a sort κ of *conditions*, with $\mathbf{1}$ the largest one
  - a predicate $C[p]$ (p is well-founded)
  - a closed term $(\cdot)$ for the product
  - a lot of combinators :

$$\begin{array}{lcl} \alpha_0 & : & C[\mathbf{1}] \\ \alpha_1 & : & \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[p]) \\ \alpha_3 & : & \forall p^\kappa \forall q^\kappa (C[pq] \Rightarrow C[qp]) \\ \alpha_6 & : & \forall p^\kappa \forall q^\kappa \forall r^\kappa (C[p(qr)] \Rightarrow C[(pq)r]) \end{array}$$

- Definition of the forcing relation :

$$\begin{array}{rcl} p \Vdash A & \equiv & \forall r^\kappa (C[pr] \Rightarrow A^* r) \\ (A \Rightarrow B)^* & \equiv & \lambda r^\kappa . \forall q \forall r' \langle r = qr' \rangle (\forall s (C[qs] \Rightarrow A^* s) \Rightarrow B^* r') \end{array}$$

- Translation on programs:

$$\begin{array}{rcl} (tu)^* & \equiv & \gamma_3 t^* u^* \\ (\lambda x.t)^* & \equiv & \gamma_1 (\lambda x.t^* \{x_i := \beta_3 x_i\}\{x := \beta_4 x\}) \\ \boldsymbol{cc}^* & \equiv & \lambda cx.\boldsymbol{cc}(\lambda k.x(\alpha_{14} c)(\gamma_4 k)) \end{array}$$

A transformation on *programs* [Krivine'10]

- A *forcing structure* is given by :

  - a sort κ of *conditions*, with $\mathbf{1}$ the largest one
  - a predicate $C[p]$ (p is well-founded)
  - a closed term $(\cdot)$ for the product
  - a lot of combinators

- Definition of the forcing relation :

$$\begin{array}{rcl} p \Vdash A & \equiv & \forall r^\kappa (C[pr] \Rightarrow A^* r) \\ (A \Rightarrow B)^* & \equiv & \lambda r^\kappa . \forall q \forall r' \langle r = qr' \rangle (\forall s (C[qs] \Rightarrow A^* s) \Rightarrow B^* r') \end{array}$$

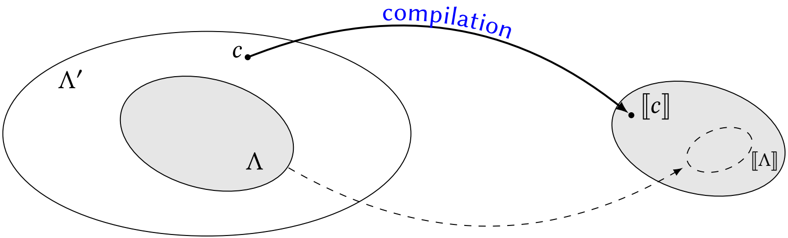
- Translation on programs:

$$\begin{array}{rcl} (tu)^* & \equiv & \gamma_3 t^* u^* \\ (\lambda x.t)^* & \equiv & \gamma_1(\lambda x.t^* \{x_i := \beta_3 x_i\}\{x := \beta_4 x\}) \\ \mathbf{cc}^* & \equiv & \lambda cx.\mathbf{cc}(\lambda k.x(\alpha_{14}c)(\gamma_4 k)) \end{array}$$

Soundness

$$\Gamma \vdash t : A \quad \Rightarrow \quad \forall p.(p \Vdash \Gamma \vdash t^* : p \Vdash A)$$

The KFAM: the transformation hard-wired [Miquel'11]



New axiom	$\sim$	Programing primitive
$\Updownarrow$		$\Updownarrow$
Logical translation	$\sim$	Program translation

The KFAM: the transformation hard-wired [Miquel'11]

Terms	t, u	$::=$	$x \mid \lambda x.t \mid tu \mid \mathbf{cc}$
Environments	e	$::=$	$\emptyset \mid e, x := c$
Closures	c	$::=$	$t[e] \mid k_\pi \mid \underbrace{t[e]^* \mid k^*_\pi}_{\text{forcing closures}}$
Stacks	π	$::=$	$\diamond \mid t \cdot \pi$

- Evaluation rules : real mode

$$\begin{array}{rcll}
\langle x[e, y := c] \parallel \pi \rangle & \succ & \langle x[e] \parallel \pi \rangle & (y \neq x) \\
\langle x[e, x := c] \parallel \pi \rangle & \succ & \langle c \parallel \pi \rangle & \\
\langle (\lambda x.t)[e] \parallel c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \pi \rangle & \\
\langle (tu)[e] \parallel \pi \rangle & \succ & \langle t[e] \parallel u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e] \parallel c \cdot \pi \rangle & \succ & \langle c \parallel k_\pi \cdot \pi \rangle & \\
\langle k_\pi \parallel c \cdot \pi' \rangle & \succ & \langle c \parallel \pi \rangle &
\end{array}$$

- Evaluation rules : forcing mode

$$\begin{array}{rcll}
\langle x[e, y := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle x[e] \parallel \alpha_9\, c_0 \cdot \pi \rangle & (y \neq x) \\
\langle x[e, x := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{10}\, c_0 \cdot \pi \rangle & \\
\langle (\lambda x.t)[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \alpha_6\, c_0 \cdot \pi \rangle & \\
\langle (tu)[e]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle t[e] \parallel \alpha_{11}\, c_0 \cdot u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{14}\, c_0 \cdot k_\pi \cdot \pi \rangle & \\
\langle k^*_\pi \parallel c_0 \cdot c \cdot \pi' \rangle & \succ & \langle c \parallel \alpha_{15}\, c_0 \cdot \pi \rangle &
\end{array}$$

The KFAM: the transformation hard-wired [Miquel'11]

Terms	t, u	$::=$	$x \mid \lambda x.t \mid tu \mid \mathbf{cc}$
Environments	e	$::=$	$\emptyset \mid e, x := c$
Closures	c	$::=$	$t[e] \mid k_\pi \mid t[e]^* \mid k_\pi^*$
Stacks	π	$::=$	$\diamond \mid t \cdot \pi$

($t[e]^* \mid k_\pi^*$: forcing closures)

- Evaluation rules : real mode

$$
\begin{array}{rcll}
\langle x[e, y := c] \parallel \pi \rangle & \succ & \langle x[e] \parallel \pi \rangle & (y \neq x) \\
\langle x[e, x := c] \parallel \pi \rangle & \succ & \langle c \parallel \pi \rangle & \\
\langle (\lambda x.t)[e] \parallel c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \pi \rangle & \\
\langle (tu)[e] \parallel \pi \rangle & \succ & \langle t[e] \parallel u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e] \parallel c \cdot \pi \rangle & \succ & \langle c \parallel k_\pi \cdot \pi \rangle & \\
\langle k_\pi \parallel c \cdot \pi' \rangle & \succ & \langle c \parallel \pi \rangle &
\end{array}
$$

- Evaluation rules : forcing mode

$$
\begin{array}{rcll}
\langle x[e, y := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle x[e] \parallel \alpha_9\, c_0 \cdot \pi \rangle & (y \neq x) \\
\langle x[e, x := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{10}\, c_0 \cdot \pi \rangle & \\
\langle (\lambda x.t)[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \alpha_6\, c_0 \cdot \pi \rangle & \\
\langle (tu)[e]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle t[e] \parallel \alpha_{11}\, c_0 \cdot u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{14}\, c_0 \cdot k_\pi \cdot \pi \rangle & \\
\langle k_\pi^* \parallel c_0 \cdot c \cdot \pi' \rangle & \succ & \langle c \parallel \alpha_{15}\, c_0 \cdot \pi \rangle &
\end{array}
$$

The KFAM: the transformation hard-wired [Miquel'11]

Terms	t, u	$::=$	$x \mid \lambda x.t \mid tu \mid \mathbf{cc}$
Environments	e	$::=$	$\emptyset \mid e, x := c$
Closures	c	$::=$	$t[e] \mid k_\pi \mid \underbrace{t[e]^* \mid k_\pi^*}_{\text{forcing closures}}$
Stacks	π	$::=$	$\diamond \mid t \cdot \pi$

- Evaluation rules : real mode

$$
\begin{array}{rcll}
\langle x[e, y := c] \parallel \pi \rangle & \succ & \langle x[e] \parallel \pi \rangle & (y \neq x) \\
\langle x[e, x := c] \parallel \pi \rangle & \succ & \langle c \parallel \pi \rangle & \\
\langle (\lambda x.t)[e] \parallel c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \pi \rangle & \\
\langle (tu)[e] \parallel \pi \rangle & \succ & \langle t[e] \parallel u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e] \parallel c \cdot \pi \rangle & \succ & \langle c \parallel k_\pi \cdot \pi \rangle & \\
\langle k_\pi \parallel c \cdot \pi' \rangle & \succ & \langle c \parallel \pi \rangle &
\end{array}
$$

- Evaluation rules : forcing mode

$$
\begin{array}{rcll}
\langle x[e, y := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle x[e] \parallel \alpha_9\, c_0 \cdot \pi \rangle & (y \neq x) \\
\langle x[e, x := c]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{10}\, c_0 \cdot \pi \rangle & \\
\langle (\lambda x.t)[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle t[e, x := c] \parallel \alpha_6\, c_0 \cdot \pi \rangle & \\
\langle (tu)[e]^* \parallel c_0 \cdot \pi \rangle & \succ & \langle t[e] \parallel \alpha_{11}\, c_0 \cdot u[e] \cdot \pi \rangle & \\
\langle \mathbf{cc}[e]^* \parallel c_0 \cdot c \cdot \pi \rangle & \succ & \langle c \parallel \alpha_{14}\, c_0 \cdot k_\pi \cdot \pi \rangle & \\
\langle k_\pi^* \parallel c_0 \cdot c \cdot \pi' \rangle & \succ & \langle c \parallel \alpha_{15}\, c_0 \cdot \pi \rangle &
\end{array}
$$

A barrier

According to the previous slides :

Motto

What forcing can, classical realizability can too.

But in fact, the same limitation appears :

Schoenfield's barrier [Krivine'14]

Σ^1_2- and Π^1_2-formulæ are absolute for realizability models.

A barrier

According to the previous slides :

Motto

What forcing can, classical realizability can too.

But in fact, the same limitation appears :

Schoenfield's barrier [Krivine'14]

Σ^1_2- and Π^1_2-formulæ are absolute for realizability models.

New models

$$\mathrm{Nat}(x) \triangleq \forall X.(X0 \Rightarrow \forall y.(Xy \Rightarrow X(sy)) \Rightarrow Xx)$$

Fact : There is no universal realizer of $\forall x.\mathrm{Nat}(x)$.

There are unnamed elements.

In fact, we can find a pole $\perp\!\!\!\perp$ s.t.

$$(\forall n \in \mathbb{N}) \quad \mathcal{M}_{\perp\!\!\!\perp} \models \mathrm{Nat}(n) \qquad \text{and} \qquad \mathcal{M}_{\perp\!\!\!\perp} \models \exists x.\neg\mathrm{Nat}(x)$$

More surprisingly, $\nabla_n \in \mathcal{P}(\mathbb{N})$ s.t.:

1. ∇_2 is not well-ordered
2. there ian injection $\nabla_n \hookrightarrow \nabla_{n+1}$
3. there is no surjection from ∇_n to ∇_{n+1}
4. $\nabla_m \times \nabla_n \simeq \nabla_{mn}$

$$\mathcal{M}_{\perp\!\!\!\perp} \models ZF_\varepsilon + \neg AC + \neg CH$$

New models

$$\mathrm{Nat}(x) \triangleq \forall X.(X0 \Rightarrow \forall y.(Xy \Rightarrow X(sy)) \Rightarrow Xx)$$

Fact : There is no universal realizer of $\forall x.\mathrm{Nat}(x)$.

There are unnamed elements.

In fact, we can find a pole $\perp\!\!\!\perp$ s.t.

$$(\forall n \in \mathbb{N}) \quad \mathcal{M}_{\perp\!\!\!\perp} \models \mathrm{Nat}(n) \qquad \text{and} \qquad \mathcal{M}_{\perp\!\!\!\perp} \models \exists x.\neg\mathrm{Nat}(x)$$

More surprisingly, $\nabla_n \in \mathcal{P}(\mathbb{N})$ s.t.:

1. ∇_2 is not well-ordered
2. there ian injection $\nabla_n \hookrightarrow \nabla_{n+1}$
3. there is no surjection from ∇_n to ∇_{n+1}
4. $\nabla_m \times \nabla_n \simeq \nabla_{mn}$

$$\mathcal{M}_{\perp\!\!\!\perp} \models ZF_\varepsilon + \neg AC + \neg CH$$

New models

$$\mathrm{Nat}(x) \triangleq \forall X.(X0 \Rightarrow \forall y.(Xy \Rightarrow X(sy)) \Rightarrow Xx)$$

In fact, we can find a pole $\perp\!\!\!\perp$ s.t.

$$(\forall n \in \mathbb{N}) \quad \mathcal{M}_{\perp\!\!\!\perp} \models \mathrm{Nat}(n) \qquad \text{and} \qquad \mathcal{M}_{\perp\!\!\!\perp} \models \exists x.\neg\mathrm{Nat}(x)$$

More surprisingly, $\nabla_n \in \mathcal{P}(\mathbb{N})$ s.t.:

1. ∇_2 is not well-ordered
2. there ian injection $\nabla_n \hookrightarrow \nabla_{n+1}$
3. there is no surjection from ∇_n to ∇_{n+1}
4. $\nabla_m \times \nabla_n \simeq \nabla_{mn}$

$$\mathcal{M}_{\perp\!\!\!\perp} \models ZF_\varepsilon + \neg AC + \neg CH$$

New models

$$\mathrm{Nat}(x) \triangleq \forall X.(X0 \Rightarrow \forall y.(Xy \Rightarrow X(sy)) \Rightarrow Xx)$$

In fact, we can find a pole $\perp\!\!\!\perp$ s.t.

$$(\forall n \in \mathbb{N}) \quad \mathcal{M}_{\perp\!\!\!\perp} \models \mathrm{Nat}(n) \qquad \text{and} \qquad \mathcal{M}_{\perp\!\!\!\perp} \models \exists x.\neg \mathrm{Nat}(x)$$

More surprisingly, $\nabla_n \in \mathcal{P}(\mathbb{N})$ s.t.:

Realizability algebras II: new models of ZF + DC
J.-L. Krivine [2014]

1. ∇_2 is not well-ordered
2. there ian injection $\nabla_n \hookrightarrow \nabla_{n+1}$
3. there is no surjection from ∇_n to ∇_{n+1}
4. $\nabla_m \times \nabla_n \simeq \nabla_{mn}$

$$\mathcal{M}_{\perp\!\!\!\perp} \models ZF_\varepsilon + \neg \mathrm{AC} + \neg \mathrm{CH}$$

New models

Great news #1

These are really new and interesting models for set theorists.

ask J.-L. Krivine or A. Karagila!

New models

What about:

Great news #2

Classical realizability models have a simple algebraic structure.

?

Krivine realizability, algebraically

Entering the wonderland of implicative algebras

Streicher's *Abstract Krivine Structures*

Krivine's classical realisability from (...)
Thomas Streicher [2013]

Abstract Krivine Structures

An AKS is given by $(\Lambda, \Pi, \mathrm{app}, \mathrm{push}, \mathrm{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ where:

1. Λ and Π are non-empty sets (*terms* and *stacks*)
2. $\mathrm{app} : t, u \mapsto tu$ is from $\Lambda \times \Lambda$ to Λ (*application*)
3. $\mathrm{push} : t, \pi \mapsto t \cdot \pi$ is from $\Lambda \times \Pi$ to Π (*push*)
4. $\mathrm{k}_{\_} : \pi \mapsto \mathrm{k}_\pi$ is from Π to Λ (*continuation*)
5. $\boldsymbol{k}$, $\boldsymbol{s}$ and $\boldsymbol{cc}$ are distinguished terms of Λ;
6. $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is a relation s.t.: (*pole*)

$$\begin{aligned} \langle t \,\|\, u \cdot \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle tu \,\|\, \pi \rangle \in \perp\!\!\!\perp \\ \langle t \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{k} \,\|\, t \cdot u \cdot \pi \rangle \in \perp\!\!\!\perp \\ \langle tv(uv) \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{s} \,\|\, t \cdot u \cdot v \cdot \pi \rangle \in \perp\!\!\!\perp \end{aligned}$$

$$\begin{aligned} \langle t \,\|\, \mathrm{k}_\pi \cdot \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{cc} \,\|\, t \cdot \pi \rangle \in \perp\!\!\!\perp \\ \langle t \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \mathrm{k}_\pi \,\|\, t \cdot \pi' \rangle \in \perp\!\!\!\perp \end{aligned}$$

7. $\mathbf{PL} \subseteq \Lambda$ contains $\boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}$ is closed under app (*proof-like*)

Streicher's *Abstract Krivine Structures*

Krivine's classical realisability from (...)
Thomas Streicher [2013]

Abstract Krivine Structures

An AKS is given by $(\Lambda, \Pi, \text{app}, \text{push}, \mathsf{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ where:

1. Λ and Π are non-empty sets (*terms* and *stacks*)

⋮

6. $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is a relation s.t.: (*pole*)

$$\begin{aligned} \langle t \,\|\, u \cdot \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle tu \,\|\, \pi \rangle \in \perp\!\!\!\perp \\ \langle t \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{k} \,\|\, t \cdot u \cdot \pi \rangle \in \perp\!\!\!\perp \\ \langle tv(uv) \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{s} \,\|\, t \cdot u \cdot v \cdot \pi \rangle \in \perp\!\!\!\perp \end{aligned}$$

$$\begin{aligned} \langle t \,\|\, \mathsf{k}_\pi \cdot \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \boldsymbol{cc} \,\|\, t \cdot \pi \rangle \in \perp\!\!\!\perp \\ \langle t \,\|\, \pi \rangle \in \perp\!\!\!\perp &\Rightarrow \langle \mathsf{k}_\pi \,\|\, t \cdot \pi' \rangle \in \perp\!\!\!\perp \end{aligned}$$

7. $\mathbf{PL} \subseteq \Lambda$ contains $\boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}$ is closed under app (*proof-like*)

Definitions:

- *Falsity value:* subset $X \subseteq \Pi$
- *Orthogonality:* $X^{\perp\!\!\!\perp} \triangleq \{t \in \Lambda : \forall \pi \in X, \langle t \,\|\, \pi \rangle \in \perp\!\!\!\perp\}$

Streicher's *Abstract Krivine Structures*

Krivine's classical realisability from (...)
Thomas Streicher [2013]

Abstract Krivine Structures

An AKS is given by $(\Lambda, \Pi, \mathrm{app}, \mathrm{push}, \mathrm{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ where:

1. Λ and Π are non-empty sets (*terms* and *stacks*)

⋮

6. $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is a relation s.t.: (*pole*)

$$\begin{array}{rcl|rcl}
\langle t \parallel u \cdot \pi \rangle \in \perp\!\!\!\perp & \Rightarrow & \langle tu \parallel \pi \rangle \in \perp\!\!\!\perp & \langle t \parallel \mathrm{k}_\pi \cdot \pi \rangle \in \perp\!\!\!\perp & \Rightarrow & \langle \boldsymbol{cc} \parallel t \cdot \pi \rangle \in \perp\!\!\!\perp \\
\langle t \parallel \pi \rangle \in \perp\!\!\!\perp & \Rightarrow & \langle \boldsymbol{k} \parallel t \cdot u \cdot \pi \rangle \in \perp\!\!\!\perp & \langle t \parallel \pi \rangle \in \perp\!\!\!\perp & \Rightarrow & \langle \mathrm{k}_\pi \parallel t \cdot \pi' \rangle \in \perp\!\!\!\perp \\
\langle tv(uv) \parallel \pi \rangle \in \perp\!\!\!\perp & \Rightarrow & \langle \boldsymbol{s} \parallel t \cdot u \cdot v \cdot \pi \rangle \in \perp\!\!\!\perp & & &
\end{array}$$

7. $\mathbf{PL} \subseteq \Lambda$ contains $\boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}$ is closed under app (*proof-like*)

Definitions:

- *Falsity value:* subset $X \subseteq \Pi$
- *Orthogonality:* $X^{\perp\!\!\!\perp} \triangleq \{t \in \Lambda : \forall \pi \in X, \langle t \parallel \pi \rangle \in \perp\!\!\!\perp\}$

↬ *you know the rest!*

Ordered combinatory algebras

Ordered combinatory algebras and realizability
Ferrer et al. [2017]

The Uruguayan approach (similar to PCA for Kleene realizability)

An OCA is given by $(\mathcal{A}, \leq, \text{app}, \boldsymbol{k}, \boldsymbol{s})$ where:

- $(\mathcal{A}, \leq)$ is a poset
- app : $(a, b) \mapsto ab$ is monotonic
- $\boldsymbol{k}ab \leq a$
- $\boldsymbol{s}abc \leq ac(bc)$

If $\mathcal{A}$ is an OCA, a *filter* over $\mathcal{A}$ is a subset $\Phi \subseteq \mathcal{A}$ s.t.:

- $\boldsymbol{k} \in \Phi$ and $\boldsymbol{s} \in \Phi$
- Φ is closed under application

Krivine Ordered Combinatory Algebra

A ${}^{\mathcal{K}}$OCA is given by $(\mathcal{A}, \leq, \text{app}, \text{imp}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{cc}, \Phi)$ where:

- $(\mathcal{A}, \leq, \Phi)$ is a filtered OCA
- $\boldsymbol{e}, \boldsymbol{cc} \in \Phi$
- imp : $(a, b) \mapsto a \to b$ is monotonic from $\mathcal{A}^{op} \times \mathcal{A} \to \mathcal{A}$
- $\boldsymbol{cc} \leq ((a \to b) \to a) \to a$
- $a \leq b \to c \quad \Rightarrow \quad ab \leq c \quad$ and $\quad ab \leq c \quad \Rightarrow \quad \boldsymbol{e}a \leq b \to c$

Ordered combinatory algebras

Ordered combinatory algebras and realizability
Ferrer et al. [2017]

The Uruguayan approach (similar to PCA for Kleene realizability)

An OCA is given by $(\mathcal{A}, \leq, \mathrm{app}, \boldsymbol{k}, \boldsymbol{s})$ where:

- $(\mathcal{A}, \leq)$ is a poset
- app : $(a, b) \mapsto ab$ is monotonic
- $\boldsymbol{k}ab \leq a$
- $\boldsymbol{s}abc \leq ac(bc)$

If $\mathcal{A}$ is an OCA, a *filter* over $\mathcal{A}$ is a subset $\Phi \subseteq \mathcal{A}$ s.t.:

- $\boldsymbol{k} \in \Phi$ and $\boldsymbol{s} \in \Phi$
- Φ is closed under application

Krivine Ordered Combinatory Algebra

A $^{\mathcal{K}}$OCA is given by $(\mathcal{A}, \leq, \mathrm{app}, \mathtt{imp}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{cc}, \Phi)$ where:

- $(\mathcal{A}, \leq, \Phi)$ is a filtered OCA
- $\boldsymbol{e}, \boldsymbol{cc} \in \Phi$
- imp : $(a, b) \mapsto a \to b$ is monotonic from $\mathcal{A}^{op} \times \mathcal{A} \to \mathcal{A}$
- $\boldsymbol{cc} \leq ((a \to b) \to a) \to a$
- $a \leq b \to c \quad \Rightarrow \quad ab \leq c$ and $ab \leq c \quad \Rightarrow \quad \boldsymbol{e}a \leq b \to c$

Connecting the dots

From AKS to $^{\mathcal{K}}$OCA

If $(\Lambda, \Pi, \mathrm{app}, \mathrm{push}, \mathrm{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS, then $(\mathcal{P}_{\perp\!\!\!\perp}(\Pi), \leq, \mathrm{app}', \mathtt{imp}', \{\boldsymbol{k}\}^{\perp\!\!\!\perp}, \{\boldsymbol{s}\}^{\perp\!\!\!\perp}, \{\boldsymbol{cc}\}^{\perp\!\!\!\perp}, \{\boldsymbol{e}\}^{\perp\!\!\!\perp}, \Phi)$ is a $^{\mathcal{K}}$OCA, with:

- $X \leq Y \quad \triangleq \quad X \supseteq Y$;
- $X \to Y \triangleq \{t \cdot \pi \in \Pi : t \in X^{\perp\!\!\!\perp} \wedge \pi \in Y\}^{\perp\!\!\!\perp\perp\!\!\!\perp}$;
- $\Phi \triangleq \{X \in \mathcal{P}_{\perp\!\!\!\perp} : \exists t \in \mathbf{PL}. t \perp\!\!\!\perp X\}$

From $^{\mathcal{K}}$OCA to AKS

If $(\mathcal{A}, \leq, \mathrm{app}_{\mathcal{A}}, \mathtt{imp}_{\mathcal{A}}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{e}, \Phi)$ is a $^{\mathcal{K}}$OCA, then $(\mathcal{A}, \mathcal{A}, \mathrm{app}, \mathrm{push}, \mathrm{k}_{\_}, \kappa, \mathrm{s}, \mathrm{c}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS where:

- $t \perp\!\!\!\perp \pi \triangleq t \leq \pi$;
- $\mathrm{app}(t, u) \triangleq \mathrm{app}_{\mathcal{A}}(t, u) = tu$;
- $\mathrm{push}(t, \pi) \triangleq t \to \pi$;
- $\mathrm{k}_\pi \triangleq \pi \to \bot$;
- $\mathbf{PL} \triangleq \Phi$;

Connecting the dots

From AKS to $^{\mathcal{K}}$OCA

If $(\Lambda, \Pi, \mathsf{app}, \mathsf{push}, \mathsf{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS, then $(\underline{\mathcal{P}_{\perp\!\!\!\perp}(\Pi)}, \leq, \mathsf{app}', \mathtt{imp}', \{\boldsymbol{k}\}^{\perp\!\!\!\perp}, \{\boldsymbol{s}\}^{\perp\!\!\!\perp}, \{\boldsymbol{cc}\}^{\perp\!\!\!\perp}, \{\boldsymbol{e}\}^{\perp\!\!\!\perp}, \Phi)$ is a $^{\mathcal{K}}$OCA, with:

- $X \leq Y \quad \triangleq \quad X \supseteq Y$;
- $X \to Y \triangleq \{t \cdot \pi \in \Pi : t \in X^{\perp\!\!\!\perp} \wedge \pi \in Y\}^{\perp\!\!\!\perp\perp\!\!\!\perp}$;
- $\underline{\Phi \triangleq \{X \in \mathcal{P}_{\perp\!\!\!\perp} : \exists t \in \mathbf{PL}.t \perp\!\!\!\perp X\}}$

From $^{\mathcal{K}}$OCA to AKS

If $(\mathcal{A}, \leq, \mathsf{app}_{\mathcal{A}}, \mathtt{imp}_{\mathcal{A}}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{e}, \Phi)$ is a $^{\mathcal{K}}$OCA, then $(\mathcal{A}, \mathcal{A}, \mathsf{app}, \mathsf{push}, \mathsf{k}_{\_}, \mathsf{K}, \mathsf{s}, \mathsf{c}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS where:

- $t \perp\!\!\!\perp \pi \triangleq t \leq \pi$;
- $\mathsf{app}(t, u) \triangleq \mathsf{app}_{\mathcal{A}}(t, u) = tu$;
- $\mathsf{push}(t, \pi) \triangleq t \to \pi$;
- $\mathsf{k}_\pi \triangleq \pi \to \bot$;
- $\mathbf{PL} \triangleq \Phi$;

Connecting the dots

From AKS to ${}^{\mathcal{K}}$OCA

If $(\Lambda, \Pi, \text{app}, \text{push}, \mathsf{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS, then
$(\underline{\mathcal{P}_{\perp\!\!\perp}(\Pi)}, \leq, \text{app}', \text{imp}', \{\boldsymbol{k}\}^{\perp\!\!\perp}, \{\boldsymbol{s}\}^{\perp\!\!\perp}, \{\boldsymbol{cc}\}^{\perp\!\!\perp}, \{\boldsymbol{e}\}^{\perp\!\!\perp}, \Phi)$ is a ${}^{\mathcal{K}}$OCA, with:

- $X \leq Y \quad \triangleq \quad X \supseteq Y;$
- $X \to Y \triangleq \{t \cdot \pi \in \Pi : t \in X^{\perp\!\!\perp} \wedge \pi \in Y\}^{\perp\!\!\perp\perp\!\!\perp};$
- $\underline{\Phi \triangleq \{X \in \mathcal{P}_{\perp\!\!\perp} : \exists t \in \mathbf{PL}. t \perp\!\!\perp X\}}$

From ${}^{\mathcal{K}}$OCA to AKS

If $(\mathcal{A}, \leq, \text{app}_{\mathcal{A}}, \text{imp}_{\mathcal{A}}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{e}, \Phi)$ is a ${}^{\mathcal{K}}$OCA,
then $(\underline{\mathcal{A}, \mathcal{A}}, \text{app}, \text{push}, \mathsf{k}_{\_}, \kappa, \mathsf{s}, \mathsf{c}, \mathbf{PL}, \perp\!\!\!\perp)$ is an AKS where:

- $\underline{t \perp\!\!\perp \pi \triangleq t \leq \pi};$
- $\text{app}(t, u) \triangleq \text{app}_{\mathcal{A}}(t, u) = tu;$
- $\text{push}(t, \pi) \triangleq t \to \pi;$
- $\mathsf{k}_\pi \triangleq \pi \to \bot;$
- $\mathbf{PL} \triangleq \Phi;$

Observations

From a filtered OCA, one can define a tripos

$$\mathcal{T} : \begin{cases} \mathbf{Set}^{op} & \longrightarrow & \mathbf{HA} \\ X & \mapsto & \mathcal{A}^X \end{cases}$$

endowed with the following *entailment* relation:

$$\varphi \vdash \psi \quad \triangleq \quad |\varphi \longrightarrow \psi| \cap \mathbf{PL} \neq \emptyset$$

Observations

Remark: everything lays in the order

$$t \perp\!\!\!\perp A \triangleq t \leq A \qquad \text{(AKS to } {}^{\mathcal{K}}\text{OCA)}$$

...there is always a lattice somewhere...

Observations

Remark: everything lays in the order

$$t \perp\!\!\!\perp A \triangleq t \leq A \qquad \text{(AKS to } {}^{\mathcal{K}}\text{OCA)}$$

...there is always a lattice somewhere...

Underlying lattice structures

Subtyping relation:

$$\frac{\Gamma \vdash p : T \quad T <: U}{\Gamma \vdash p : U} \;(\textsc{Sub}) \qquad\qquad \frac{U_1 <: T_1 \quad T_2 <: U_2}{T_1 \to T_2 <: U_1 \to U_2} \;(\textsc{S-Arr})$$

Classical realizability:

$$\text{if } A <: B \text{ then } \quad t \Vdash A \Rightarrow t \Vdash B \qquad\qquad (\text{for any } \perp\!\!\!\perp)$$

In terms of truth values:

Subtyping $\qquad A \leq_{\perp\!\!\!\perp} B \triangleq \|B\| \subseteq \|A\|$

Induces a structure of complete lattice, where $\curlywedge = \bigcup$, as in:

$$\|\forall x.A\|_\rho \triangleq \bigcup_{n \in \mathbb{N}} \|A\{x := n\}\| = \curlywedge \{\|A\{x := n\}\| : n \in \mathbb{N}\}$$

Realizability: $\quad \forall = \curlywedge \qquad \wedge = \times \qquad \exists = \curlyvee \qquad \vee = +$

Underlying lattice structures

Subtyping relation:

$$\frac{\Gamma \vdash p : T \qquad T <: U}{\Gamma \vdash p : U}\ (\text{SUB}) \qquad\qquad \frac{U_1 <: T_1 \qquad T_2 <: U_2}{T_1 \to T_2 <: U_1 \to U_2}\ (\text{S-ARR})$$

Classical realizability:

$$\text{if } A <: B \text{ then } \quad t \Vdash A \Rightarrow t \Vdash B \qquad\qquad (\text{for any } \perp\!\!\!\perp)$$

In terms of truth values:

Subtyping $\qquad A \leq_{\perp\!\!\!\perp} B \ \triangleq\ \|B\| \subseteq \|A\|$

Induces a structure of complete lattice, where $\curlywedge = \bigcup$, as in:

$$\|\forall x.A\|_\rho \triangleq \bigcup_{n\in\mathbb{N}} \|A\{x := n\}\| = \curlywedge \{\|A\{x := n\}\| : n \in \mathbb{N}\}$$

Realizability: $\qquad \forall = \curlywedge \qquad \wedge = \times \qquad \exists = \curlyvee \qquad \vee = +$

Underlying lattice structures

Subtyping relation:

$$\frac{\Gamma \vdash p : T \qquad T <: U}{\Gamma \vdash p : U}\ (\textsc{Sub}) \qquad\qquad \frac{U_1 <: T_1 \qquad T_2 <: U_2}{T_1 \to T_2 <: U_1 \to U_2}\ (\textsc{S-Arr})$$

Classical realizability:

$$\text{if } A <: B \text{ then } \quad t \Vdash A \Rightarrow t \Vdash B \qquad\qquad (\text{for any } \perp\!\!\!\perp)$$

In terms of truth values:

Subtyping $\qquad A \leq_{\perp\!\!\perp} B \triangleq \|B\| \subseteq \|A\|$

Induces a structure of complete lattice, where $\curlywedge = \bigcup$, as in:

$$\|\forall x.A\|_\rho \triangleq \bigcup_{n \in \mathbb{N}} \|A\{x := n\}\| = \curlywedge \{\|A\{x := n\}\| : n \in \mathbb{N}\}$$

Realizability: $\qquad \forall = \curlywedge \qquad \wedge = \times \qquad \exists = \curlyvee \qquad \vee = +$

Underlying lattice structures

Subtyping relation:

$$\frac{\Gamma \vdash p : T \quad T <: U}{\Gamma \vdash p : U}\ (\textsc{Sub}) \qquad \frac{U_1 <: T_1 \quad T_2 <: U_2}{T_1 \to T_2 <: U_1 \to U_2}\ (\textsc{S-Arr})$$

Classical realizability:

Subtyping $\quad A \leq_{\perp\!\!\!\perp} B \triangleq \|B\| \subseteq \|A\|$

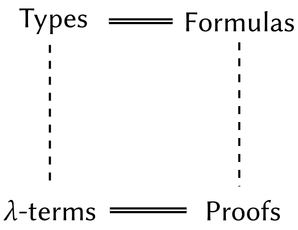
Realizability: $\quad \forall = \curlywedge \qquad \wedge = \times \qquad \exists = \curlyvee \qquad \vee = +$

Boolean algebras:
quantifiers and connectives both interpreted by meets and joins:

$$\|\forall x.A\| = \|A(0) \wedge A(1) \wedge \ldots \wedge A(n) \wedge \ldots\| = \bigcurlywedge_{n \in \mathbb{N}} \|A(n)\|$$

Forcing: $\quad \forall = \wedge = \curlywedge \qquad \exists = \vee = \curlyvee$

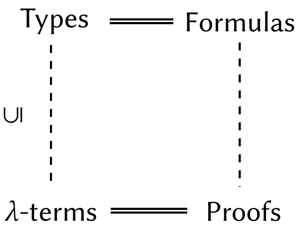
Curry-Howard, one step further



In particular, $a \preccurlyeq b$ reads:

- a is a *subtype* of b
- a is a *realizer* of b
- the realizer a is *more defined* than b

Curry-Howard, one step further



In particular, $a \preccurlyeq b$ reads:

- a is a *subtype* of b
- a is a *realizer* of b
- the realizer a is *more defined* than b

Curry-Howard, one step further

Types ═══ Formulas

‖ ‖

λ-terms ═══ Proofs

In particular, $a \preccurlyeq b$ reads:

- a is a *subtype* of b
- a is a *realizer* of b
- the realizer a is *more defined* than b

Implicative Structures

Implicative algebras: a new (...)
Alexandre Miquel [2018]

Definition:

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \longrightarrow)$ s.t.:

- if $a_0 \preccurlyeq a$ and $b \preccurlyeq b_0$ then $(a \longrightarrow b) \preccurlyeq (a_0 \longrightarrow b_0)$ *(Variance)*
- $\curlywedge_{b \in B}(a \longrightarrow b) = a \longrightarrow \curlywedge_{b \in B} b$ *(Distributivity)*

Implicative Structures

Implicative algebras: a new (...)
Alexandre Miquel [2018]

Definition:

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \rightarrow)$ s.t.:

- if $a_0 \preccurlyeq a$ and $b \preccurlyeq b_0$ then $(a \rightarrow b) \preccurlyeq (a_0 \rightarrow b_0)$ *(Variance)*
- $\curlywedge_{b \in B}(a \rightarrow b) = a \rightarrow \curlywedge_{b \in B} b$ *(Distributivity)*

Examples:

- complete Heyting/Boolean algebras

If $\mathcal{H}$ is complete, $a \mapsto b = \curlyvee\{x \in \mathcal{H} : a \wedge x \preccurlyeq b\}$.

- Ordered Combinatory Algebras

Complete lattice $\mathcal{P}(\mathcal{A})$ equipped with $A \mapsto B \triangleq \{r \in \mathcal{A} : \forall a \in A. ra \in B\}$.

- Abstract Krivine Structures

Complete lattice $\mathcal{P}(\Pi)$, equipped with:

$$a \preccurlyeq b \triangleq a \supseteq b \qquad\qquad a \mapsto b \triangleq a^{\perp\!\!\perp} \cdot b = \{t \cdot \pi : t \in a^{\perp\!\!\perp}, \pi \in b\}$$

Interpretation of λ-terms

Application:

$$a@b \triangleq \bigwedge\{c \in \mathcal{A} : a \preccurlyeq b \to c\}$$

Abstraction:

$$\lambda f \triangleq \bigwedge_{a \in \mathcal{A}}(a \to f(a))$$

Properties

1. If $t \to_\beta u$, then $t^{\mathcal{A}} \preccurlyeq u^{\mathcal{A}}$. ($\beta$-reduction)
2. If $t \to_\eta u$, then $u^{\mathcal{A}} \preccurlyeq t^{\mathcal{A}}$. ($\eta$-expansion)
3. $a@b \preccurlyeq c \quad \Leftrightarrow \quad a \preccurlyeq b \mapsto c$ (Adjunction)

Interpretation of λ-terms

Application:

$$a@b \triangleq \curlywedge\{c \in \mathcal{A} : a \preccurlyeq b \to c\}$$

Abstraction:

$$\lambda f \triangleq \curlywedge_{a\in\mathcal{A}}(a \to f(a))$$

Properties

1. If $t \to_\beta u$, then $t^{\mathcal{A}} \preccurlyeq u^{\mathcal{A}}$. ($\beta$-reduction)
2. If $t \to_\eta u$, then $u^{\mathcal{A}} \preccurlyeq t^{\mathcal{A}}$. ($\eta$-expansion)
3. $a@b \preccurlyeq c \quad\Leftrightarrow\quad a \preccurlyeq b \mapsto c$ (Adjunction)

Interpretation of formulas

Formulas with parameters:

$$A, B ::= a \mid X \mid A \Rightarrow B \mid \forall X.A \qquad (a \in \mathcal{A})$$

Embedding of closed formulas with parameters:

$$\begin{array}{rcll} a^{\mathcal{A}} & \triangleq & a & (\text{if } a \in \mathcal{A}) \\ (A \Rightarrow B)^{\mathcal{A}} & \triangleq & A^{\mathcal{A}} \to B^{\mathcal{A}} & \\ (\forall X.A)^{\mathcal{A}} & \triangleq & \curlywedge_{a \in \mathcal{A}} (A\{X := a\})^{\mathcal{A}} & \end{array}$$

Adequacy: If $\vdash t : A$ then $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$

In particular:

$$\begin{array}{rcl} \mathbf{K}^{\mathcal{A}} & = & \curlywedge_{a,b \in \mathcal{A}} (a \to b \to a) \\ \mathbf{S}^{\mathcal{A}} & = & \curlywedge_{a,b,c \in \mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \\ \mathbf{cc} & \triangleq & \curlywedge_{a,b \in \mathcal{A}} (((a \to b) \to a) \to a) \end{array}$$

Interpretation of formulas

Formulas with parameters:

$$A, B ::= a \mid X \mid A \Rightarrow B \mid \forall X.A \qquad (a \in \mathcal{A})$$

Embedding of closed formulas with parameters:

$$\begin{array}{rcll} a^{\mathcal{A}} & \triangleq & a & (\text{if } a \in \mathcal{A}) \\ (A \Rightarrow B)^{\mathcal{A}} & \triangleq & A^{\mathcal{A}} \to B^{\mathcal{A}} & \\ (\forall X.A)^{\mathcal{A}} & \triangleq & \curlywedge_{a \in \mathcal{A}} (A\{X := a\})^{\mathcal{A}} & \end{array}$$

Adequacy: If $\vdash t : A$ then $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$

In particular:

$$\begin{array}{rcl} \mathbf{K}^{\mathcal{A}} & = & \curlywedge_{a,b \in \mathcal{A}} (a \to b \to a) \\ \mathbf{S}^{\mathcal{A}} & = & \curlywedge_{a,b,c \in \mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \\ \mathbf{cc} & \triangleq & \curlywedge_{a,b \in \mathcal{A}} (((a \to b) \to a) \to a) \end{array}$$

Implicative algebras

Separator $\mathcal{S}$:

1. $\kappa^{\mathcal{A}} \in \mathcal{S}$, $s^{\mathcal{A}} \in \mathcal{S}$, $(\mathbf{cc} \in \mathcal{S})$ *(Combinators)*
2. If $a \in \mathcal{S}$ and $a \preccurlyeq b$, then $b \in \mathcal{S}$. *(Upwards closure)*
3. If $(a \to b) \in \mathcal{S}$ and $a \in \mathcal{S}$, then $b \in \mathcal{S}$. *(Modus ponens)*

Implicative algebras:

$$(\mathcal{A}, \preccurlyeq, \to) \quad + \quad \text{separator } \mathcal{S}$$

Examples:

- Complete Boolean algebras
- Abstract Krivine structures

The set $\mathcal{S} = \{a \in \mathcal{P}(\Pi) : a^{\perp\!\!\perp} \cap PL \neq \emptyset\}$ is a separator.

Implicative algebras

Separator $\mathcal{S}$:

1. $\kappa^{\mathcal{A}} \in \mathcal{S}$, $s^{\mathcal{A}} \in \mathcal{S}$,($\mathbf{cc} \in \mathcal{S}$) *(Combinators)*
2. If $a \in \mathcal{S}$ and $a \preccurlyeq b$, then $b \in \mathcal{S}$. *(Upwards closure)*
3. If $(a \to b) \in \mathcal{S}$ and $a \in \mathcal{S}$, then $b \in \mathcal{S}$. *(Modus ponens)*

Implicative algebras:

$$(\mathcal{A}, \preccurlyeq, \to) \quad + \quad \text{separator } \mathcal{S}$$

Examples:

- Complete Boolean algebras

For all λ-term t, $t^{\mathcal{B}} = \top$ and $a@b = a \curlywedge b$. Thus, $\top$ or any filter define separators.

- Abstract Krivine structures

Implicative algebras

Separator $\mathcal{S}$:

1. $\kappa^{\mathcal{A}} \in \mathcal{S}$, $s^{\mathcal{A}} \in \mathcal{S}$,($\mathbf{cc} \in \mathcal{S}$) *(Combinators)*
2. If $a \in \mathcal{S}$ and $a \preccurlyeq b$, then $b \in \mathcal{S}$. *(Upwards closure)*
3. If $(a \to b) \in \mathcal{S}$ and $a \in \mathcal{S}$, then $b \in \mathcal{S}$. *(Modus ponens)*

Implicative algebras:

$$(\mathcal{A}, \preccurlyeq, \to) \quad + \quad \text{separator } \mathcal{S}$$

Examples:

- Complete Boolean algebras

For all λ-term t, $t^{\mathcal{B}} = \top$ and $a@b = a \curlywedge b$. Thus, $\top$ or any filter define separators.

- Abstract Krivine structures

The set $\mathcal{S} = \{a \in \mathcal{P}(\Pi) : a^{\perp\!\!\perp} \cap \mathbf{PL} \neq \emptyset\}$ is a separator.

Internal logic

Entailment: $a \vdash_{\mathcal{S}} b \;\triangleq\; a \to b \in \mathcal{S}$

Properties

1. $\vdash_{\mathcal{S}}$ is a preorder
2. if $a \preccurlyeq b$ then $a \vdash_{\mathcal{S}} b$ (Subtyping)
3. if $a \vdash_{\mathcal{S}} b$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (Closure under $\vdash_{\mathcal{S}}$)

Adjunction

$a \vdash_{\mathcal{S}} b \to c$ if and only if $a \times b \vdash_{\mathcal{S}} c$

Internal logic

Entailment: $a \vdash_{\mathcal{S}} b \quad \triangleq \quad a \to b \in \mathcal{S}$

Properties

1. $\vdash_{\mathcal{S}}$ is a preorder
2. if $a \preccurlyeq b$ then $a \vdash_{\mathcal{S}} b$ (Subtyping)
3. if $a \vdash_{\mathcal{S}} b$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (Closure under $\vdash_{\mathcal{S}}$)

Quantifiers:

$$\bigvee_{i \in I} a_i \;\triangleq\; \bigcurlywedge_{i \in I} a_i \qquad\qquad \exists_{i \in I} a_i \;\triangleq\; \bigcurlywedge_{c \in A} \Big(\bigcurlywedge_{i \in I} (a_i \to c) \to c \Big)$$

Semantic rules:

$$\frac{\Gamma \vdash t : a_i \quad \text{for all } i \in I}{\Gamma \vdash t : \forall_{i \in I} a_i} \qquad \frac{\Gamma \vdash t : \forall_{i \in I} a_i \quad i_0 \in I}{\Gamma \vdash t : a_{i_0}}$$

$$\frac{\Gamma \vdash t : a_{i_0} \quad i_0 \in I}{\Gamma \vdash \lambda x.xt : \exists_{i \in I} a_i} \qquad \frac{\Gamma \vdash t : \exists_{i \in I} a_i \quad \Gamma, x : a_i \vdash u : c \quad (\text{for all } i \in I)}{\Gamma \vdash t(\lambda x.u) : c}$$

Internal logic

Entailment: $a \vdash_{\mathcal{S}} b \ \triangleq\ a \to b \in \mathcal{S}$

Properties

1. $\vdash_{\mathcal{S}}$ is a preorder
2. if $a \preccurlyeq b$ then $a \vdash_{\mathcal{S}} b$ (Subtyping)
3. if $a \vdash_{\mathcal{S}} b$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (Closure under $\vdash_{\mathcal{S}}$)

Connectives:

$$a \times b \triangleq \bigwedge_{c \in \mathcal{A}} ((a \to b \to c) \to c) \qquad a + b \triangleq \bigwedge_{c \in \mathcal{A}} ((a \to c) \to (b \to c) \to c)$$

Semantic rules:

$$\frac{\Gamma \vdash t : a \quad \Gamma \vdash u : b}{\Gamma \vdash \lambda z.ztu : a \times b} \qquad \frac{\Gamma \vdash t : a + b \quad \Gamma, x : a \vdash u : c \quad \Gamma, y : b \vdash v : c}{\Gamma \vdash t(\lambda x.u)(\lambda y.v) : c}$$

$$\frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\pi_1 : a} \qquad \frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\pi_2 : b} \qquad \frac{\Gamma \vdash t : a}{\Gamma \vdash \lambda lr.lt : a + b} \qquad \frac{\Gamma \vdash t : b}{\Gamma \vdash \lambda lr.rt : a + b}$$

Internal logic

Entailment: $a \vdash_{\mathcal{S}} b \;\triangleq\; a \to b \in \mathcal{S}$

Properties

1. $\vdash_{\mathcal{S}}$ is a preorder
2. if $a \preccurlyeq b$ then $a \vdash_{\mathcal{S}} b$ (Subtyping)
3. if $a \vdash_{\mathcal{S}} b$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (Closure under $\vdash_{\mathcal{S}}$)

Connectives:

$$a \times b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \to b \to c) \to c) \qquad a + b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \to c) \to (b \to c) \to c)$$

Adjunction

$a \vdash_{\mathcal{S}} b \to c$ if and only if $a \times b \vdash_{\mathcal{S}} c$

A incredibly nice framework

Adjunction

$$a \vdash_{\mathcal{S}} b \mapsto c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c.$$

Proof. (⇒) *Assume that* $t := a \mapsto b \mapsto c \in \mathcal{S}$. *We shall find* $?u \in \mathcal{S}$ *s.t.:*

$$?u \preccurlyeq a \times b \mapsto c$$

A incredibly nice framework

Adjunction

$$a \vdash_{\mathcal{S}} b \mapsto c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c.$$

Proof. (⇒) *Assume that* $t := a \mapsto b \mapsto c \in \mathcal{S}$. *We shall find* $?u \in \mathcal{S}$ *s.t.:*

$$?u \preccurlyeq \Big(\bigcurlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d \Big) \mapsto c$$

A incredibly nice framework

Adjunction

$$a \vdash_{\mathcal{S}} b \mapsto c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c.$$

Proof. (⇒) *Assume that* $t := a \mapsto b \mapsto c \in \mathcal{S}$. *Let us prove that:*

$$\lambda x.x@t \preccurlyeq \left(\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d \right) \mapsto c$$

A incredibly nice framework

Adjunction

$$a \vdash_{\mathcal{S}} b \mapsto c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c.$$

Proof. ($\Rightarrow$) *Assume that* $t := a \mapsto b \mapsto c \in \mathcal{S}$. *Let us prove that:*

$$\lambda x.x@t \preccurlyeq \left(\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d \right) \mapsto c$$

$$\Leftarrow \lambda x.x@(a \mapsto b \mapsto c) \preccurlyeq (\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d) \mapsto c \qquad \text{(β-reduction)}$$

$$\Leftrightarrow (\lambda x.x@(a \mapsto b \mapsto c))@(\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d) \preccurlyeq c \qquad \text{(adjunction)}$$

$$\Leftarrow (\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d)@(a \mapsto b \mapsto c) \preccurlyeq c \qquad \text{(β-reduction)}$$

$$\Leftrightarrow (\curlywedge_{d \in \mathcal{A}} (a \mapsto b \mapsto d) \mapsto d) \preccurlyeq (a \mapsto b \mapsto c) \mapsto c \qquad \text{(adjunction)}$$

$$\Leftarrow (a \mapsto b \mapsto c) \mapsto c \preccurlyeq (a \mapsto b \mapsto c) \mapsto c \qquad \text{(meet def.)}$$

□

A incredibly nice framework

Adjunction

$$a \vdash_{\mathcal{S}} b \mapsto c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c.$$

Proof. (⇒) *Assume that* $t := a \mapsto b \mapsto c \in \mathcal{S}$. *It suffices to prove that:*

$$\lambda xy.yx \preccurlyeq (a \mapsto b \mapsto c) \mapsto (a \times b) \mapsto c$$

(⇐) *Assume that* $(a \times b) \mapsto c \in \mathcal{S}$. *It suffices to prove that:*

$$\lambda fab.f(\lambda z.zab) \preccurlyeq ((a \times b) \mapsto c) \mapsto (a \mapsto b \mapsto c)$$

Implicative tripos

Adjunction

$a \vdash_{\mathcal{S}} b \to c$ if and only if $a \times b \vdash_{\mathcal{S}} c$

($\looparrowright$ $(\mathcal{A}/\mathcal{S}, \vdash_{\mathcal{S}}, \times, +, \to)$ *is a Heyting algebra*)

Tripos:

$$\mathcal{T} : \begin{cases} \mathbf{Set}^{op} & \to & \mathbf{HA} \\ I & \mapsto & \mathcal{A}^I/\mathcal{S}[I] \end{cases}$$

Collapse criteria

The following are equivalent:

1. $\mathcal{T}$ is isomorphic to a forcing tripos
2. $\mathcal{S} \subseteq \mathcal{A}$ is a principal filter of $\mathcal{A}$.
3. $\mathcal{S} \subseteq \mathcal{A}$ is finitely generated and $\pitchfork \in \mathcal{S}$.

Implicative tripos

Adjunction

$$a \vdash_{\mathcal{S}} b \to c \qquad \text{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c$$

($\looparrowright$ $(\mathcal{A}/\mathcal{S}, \vdash_{\mathcal{S}}, \times, +, \to)$ *is a Heyting algebra*)

Tripos:

$$\mathcal{T} : \begin{cases} \mathbf{Set}^{op} & \to & \mathbf{HA} \\ I & \mapsto & \mathcal{A}^I/\mathcal{S}[I] \end{cases}$$

Collapse criteria

The following are equivalent:

1. $\mathcal{T}$ is isomorphic to a forcing tripos
2. $\mathcal{S} \subseteq \mathcal{A}$ is a principal filter of $\mathcal{A}$.
3. $\mathcal{S} \subseteq \mathcal{A}$ is finitely generated and $\pitchfork \in \mathcal{S}$.

Completeness of implicative triposes

Theorem [Miquel 18]

Each **Set**-based tripos is (isomorphic to) an implicative tripos.

The proof is based on several observations:

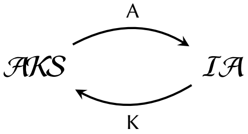
- *generic predicate*: there exists Σ and $\mathtt{tr} \in \mathcal{T}(\Sigma)$ s.t.

$$[\![-]\!]_X : \begin{cases} \Sigma^X & \rightarrow & \mathcal{T}(X) \\ \sigma & \mapsto & \mathcal{T}(\sigma)(\mathtt{tr}) \end{cases} \qquad \text{is surjective}$$

  ↬ *each predicate on X has a* **code** *in* Σ^X
- we can define codes $\dot{\wedge}, \dot{\vee}, \dot{\Rightarrow}$ for connectives
  $\dot{\forall}, \dot{\exists}$ for quantifiers
- this *almost* endows Σ with a structure of complete HA
- it "leads" to an implicative algebra
  ↬ *the corresponding tripos is* ***isomorphic*** *to the original one*

Categorifying a bit more

We have:



Questions:

1. Can we define categories for $\mathcal{IA}$ / $\mathcal{AKS}$?
2. Does this diagram have a categorical meaning?

The category of Implicative Algebras

The category of Implicative Algebras and Realizability
W. Ferrer, O. Malherbe [2018]

Assume two IAs $\mathcal{A}$ and $\mathcal{B}$

Applicative morphism

$f : \mathcal{A} \to \mathcal{B}$ with $r, u \in \mathcal{S}_{\mathcal{B}}$ such that:

1. $f(\mathcal{S}_{\mathcal{A}}) \subseteq \mathcal{S}_{\mathcal{B}}$
2. $r f(a) f(a') \preccurlyeq f(aa')$ $\qquad (\forall a, a' \in \mathcal{A})$
3. If $a \preccurlyeq a'$ then $u f(a) \preccurlyeq f(a')$

Computationally dense morphism

$f : \mathcal{A} \to \mathcal{B}$ applicative with $h : \mathcal{S}_{\mathcal{B}} \to \mathcal{S}_{\mathcal{A}}$ monotonic, $t \in \mathcal{S}_{\mathcal{B}}$ s.t.:

$$t \preccurlyeq f(h(b)) \to b \qquad (\forall b \in \mathcal{S}_{\mathcal{B}})$$

Proposition

The two notions give rise to categories **IA** / **IAc**.

The category of Implicative Algebras

> The category of Implicative Algebras and Realizability
> *W. Ferrer, O. Malherbe* [2018]

Good news:

- The two notions also give rise to categories **AKS** / **AKSc**.
- The maps $A : \mathcal{AKS} \rightarrow \mathcal{IA}$ and $K : \mathcal{IA} \rightarrow \mathcal{AKS}$ extend to functors:

Theorem

These functors form an adjoint pair.

The category of Implicative Algebras

The category of Implicative Algebras and Realizability
W. Ferrer, O. Malherbe [2018]

Good news:

- The two notions also give rise to categories **AKS** / **AKSc**.
- The maps $A : \mathcal{AKS} \to \mathcal{IA}$ and $K : \mathcal{IA} \to \mathcal{AKS}$ extend to functors:

$$\mathbf{AKS} \underset{K}{\overset{A}{\rightleftarrows}} \mathbf{IA} \quad (\perp) \qquad\qquad \mathbf{AKSc} \underset{K}{\overset{A}{\rightleftarrows}} \mathbf{IAc} \quad (\perp)$$

Theorem

These functors form an adjoint pair.

Is that it?

Implicative structures:

- simple algebraic structures
- adequate embedding of types and terms

Implicative algebras:

- encompass usual approaches to realizability
- generalize Boolean algebras and forcing
- complete w.r.t. **Set**-based triposes

Further questions:

- account for different evaluation strategies [M. '20]
- account for side effects
- IA morphisms that induce tripos isomorphisms?

Is that it?

Implicative structures:

- simple algebraic structures
- adequate embedding of types and terms

Implicative algebras:

- encompass usual approaches to realizability
- generalize Boolean algebras and forcing
- complete w.r.t. **Set**-based triposes

Further questions:

- account for different evaluation strategies [M. '20]
- account for side effects
- IA morphisms that induce tripos isomorphisms?

Is that it?

Implicative structures:

- simple algebraic structures
- adequate embedding of types and terms

Implicative algebras:

- encompass usual approaches to realizability
- generalize Boolean algebras and forcing
- complete w.r.t. **Set**-based triposes

Further questions:

- account for different evaluation strategies [M. '20]
- account for side effects
- IA morphisms that induce tripos isomorphisms?

Conclusion

Last week

We saw:

- **Classical logic**: interaction terms/contexts
- **Krivine realizability**:
  - interaction player/opponent
  - primitive falsity values + **orthogonality**
- Key property: **adequacy** w.r.t. typing

Killer features

- Normalization / soundness as corollaries
- Very modular: *With side-effects come new reasoning principles.*
- Compatible with *your* favorite calculus (probably)

Last week

We saw:

- **Classical logic**: interaction terms/contexts
- **Krivine realizability**:
  - interaction player/opponent
  - primitive falsity values + **orthogonality**
- Key property: **adequacy** w.r.t. typing

Killer features

- Normalization / soundness as corollaries
- Very modular: *With side-effects come new reasoning principles.*
- Compatible with *your* favorite calculus (probably)

Today wrapped up

We saw:

- **specification problem**

✓ solutions via the *threads method.*

- **witness extraction**

✓ works for Σ_1^0-formulas

- connexion with **forcing**

✓ realizability **generalizes** forcing!

- the **algebraic structure** of realizability models

✓ implicative algebras

Today wrapped up

We saw:

- **specification problem**

✓ solutions via the *threads method.*

- **witness extraction**

✓ works for Σ^0_1-formulas

- connexion with **forcing**

✓ realizability **generalizes** forcing!

- the **algebraic structure** of realizability models

✓ implicative algebras

Today wrapped up

We saw:

- **specification problem**

✓ solutions via the *threads method.*

- **witness extraction**

✓ works for Σ^0_1-formulas

- connexion with **forcing**

✓ realizability **generalizes** forcing!

- the **algebraic structure** of realizability models

✓ implicative algebras

Today wrapped up

We saw:

- **specification problem**

✓ solutions via the *threads method.*

- **witness extraction**

✓ works for Σ_1^0-formulas

- connexion with **forcing**

✓ realizability **generalizes** forcing!

- the **algebraic structure** of realizability models

✓ implicative algebras

Future lines of work

1. Logical counterpart of side effects
   AC via memoization, resources management, ...
2. Realizability models
   structure, properties, connexion with usual models, ...
3. Implicative algebras
   include effects, algebraic properties, ...
4. *You tell me!*

Future lines of work

1. Logical counterpart of side effects

   AC via memoization, resources management, ...

2. Realizability models

   structure, properties, connexion with usual models, ...

3. Implicative algebras

   include effects, algebraic properties, ...

4. *You tell me!*

Future lines of work

1. Logical counterpart of side effects

   AC via memoization, resources management, ...

2. Realizability models

   structure, properties, connexion with usual models, ...

3. Implicative algebras

   include effects, algebraic properties, ...

4. *You tell me!*

Future lines of work

1. Logical counterpart of side effects

   AC via memoization, resources management, ...

2. Realizability models

   structure, properties, connexion with usual models, ...

3. Implicative algebras

   include effects, algebraic properties, ...

4. *You tell me!*

Questions?