Revisiting the duality of computation

An algebraic analysis of classical realizability models

Étienne Miquey

Équipe Gallinette (INRIA), LS2N, Université de Nantes

Workshop Realizability - 18/09/2019

Classical Curry-Howard

Classical logic = Intuitionistic logic + $A \vee \neg A$

1990: Griffin discovered that `call/cc` can be typed by Peirce's law
(well-known fact: Peirce's law $\Rightarrow A \vee \neg A$)

Classical Curry-Howard:

λ-calculus + `call/cc`

With side effects come new reasoning principles:

- quote instruction $\sim$ dependent choice
- monotonic memory $\sim$ Cohen's forcing
- ...

The duality of computation

Curien-Herbelin (2000):

ABSTRACT

We present the $\overline{\lambda}\mu\tilde{\mu}$-calculus, a syntax for λ-calculus + control operators exhibiting symmetries such as program/context and call-by-name/call-by-value. This calculus is derived from implicational Gentzen's sequent calculus LK, a key classical logical system in proof theory. Under the Curry-Howard correspondence between proofs and programs, we can see LK, or more precisely a formulation called $LK_{\mu\tilde{\mu}}$, as a syntax-directed system of simple types for $\overline{\lambda}\mu\tilde{\mu}$-calculus.

Krivine realizability as a tool

Many applications:

- Specification problem

$$t \text{ realizes } \forall X.X \to X \quad \text{iff} \quad t \star u \cdot \pi \succ u \star \pi$$

- Normalization proofs

If $\vdash t : A$ then t normalizes.

- Soundness proofs

My calculus doesn't allow any proof term of $\bot$.

- Witness extraction

If $\vdash t : \exists x^{\mathbb{N}}.f(x) = 0$ then we can use t to compute n s.t. $f(n) = 0$.

Krivine realizability as a model

Krivine realizability:

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

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

Tarski

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

(intuition: level of truthness)

Great news

Classical realizability semantics gives surprisingly new models!

(in particular, provides us with a direct construction of $\mathcal{M} \vDash ZF_\varepsilon + \neg CH + \neg AC$)

This talk

Question #1

What is the algebraic structure of Krivine realizability models?

Question #2

Are call-by-name and call-by-value inducing the same models?

This talk

Question #1

What is the algebraic structure of Krivine realizability models?

Question #2

Are call-by-name and call-by-value inducing the same models?

Outline

1. **Krivine classical realizability**
   Realizability models, AKS, $^{\mathcal{K}}$OCA
2. **Implicative algebras**
   Implicative struct., λ_c-calculus, models & completeness properties
3. **Disjunctive algebras**
   Disjunctive struct., $L^{\parr}$ & connection with implicative struct.
4. **Conjunctive algebras**
   Conjunctive struct., $L^{\otimes}$ & connection with disjunctive struct.

(Bonus:  *)*

Krivine classical realizability

(from an algebraic perspective)

Krivine realizability, a 3-steps recipe

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

Krivine realizability, a 3-steps recipe

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

$$\begin{array}{lrcl} \textsc{Push}: & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\ \textsc{Grab}: & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\ \textsc{Save}: & \boldsymbol{cc} \star t \cdot \pi & \succ_1 & t \star \boldsymbol{k}_\pi \cdot \pi \\ \textsc{Restore}: & \boldsymbol{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi \end{array}$$

2. a logical language
3. formulas interpretation

Krivine realizability, a 3-steps recipe

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

$$
\begin{array}{lrcl}
\textsc{Push}: & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab}: & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save}: & \mathbf{cc} \star t \cdot \pi & \succ_1 & t \star \mathbf{k}_\pi \cdot \pi \\
\textsc{Restore}: & \mathbf{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}
$$

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

Krivine realizability, a 3-steps recipe

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

$$\begin{array}{lrcl}
\textsc{Push}: & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab}: & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save}: & \mathbf{cc} \star t \cdot \pi & \succ_1 & t \star \mathbf{k}_\pi \cdot \pi \\
\textsc{Restore}: & \mathbf{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $⫫$: processes, referee
- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: terms, player of A

Krivine realizability, a 3-steps recipe

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

$$\begin{array}{rcrcl}
\textsc{Push} & : & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab} & : & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save} & : & \mathbf{cc} \star t \cdot \pi & \succ_1 & t \star \mathbf{k}_\pi \cdot \pi \\
\textsc{Restore} & : & \mathbf{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $⫫$:

$$t \star \pi \succ p_0 \succ \cdots \succ p_n \in ⫫?$$

- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: terms, player of A

Krivine realizability, a 3-steps recipe

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

$$\begin{array}{llcl}
\textsc{Push} : & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab} : & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save} : & \boldsymbol{cc} \star t \cdot \pi & \succ_1 & t \star \boldsymbol{k}_\pi \cdot \pi \\
\textsc{Restore} : & \boldsymbol{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $⫫$: $⫫ \subset \Lambda \star \Pi$ closed by anti-reduction
- falsity value $\|A\|$: stacks, opponent to A
- truth value $|A|$: terms, player of A

Krivine realizability, a 3-steps recipe

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

$$\begin{array}{rcrcl}
\textsc{Push} & : & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab} & : & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save} & : & \boldsymbol{cc} \star t \cdot \pi & \succ_1 & t \star \boldsymbol{k}_\pi \cdot \pi \\
\textsc{Restore} & : & \boldsymbol{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $⫫$: $⫫ \subset \Lambda \star \Pi$ closed by anti-reduction
- falsity value $\|A\|$: by structural induction
  - $\|A \Rightarrow B\| = \{t \cdot \pi : t \in |A| \wedge \pi \in \|B\|\}$
  - $\|\forall x.A\| = \bigcup_{n \in \mathbb{N}} \|A\{x := n\}\|$
- truth value $|A|$: terms, player of A

Krivine realizability, a 3-steps recipe

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

$$
\begin{array}{rcrcl}
\textsc{Push} & : & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab} & : & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save} & : & \mathbf{cc} \star t \cdot \pi & \succ_1 & t \star \mathbf{k}_\pi \cdot \pi \\
\textsc{Restore} & : & \mathbf{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}
$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $\perp\!\!\!\perp$: $\perp\!\!\!\perp \subset \Lambda \star \Pi$ closed by anti-reduction
- falsity value $\|A\|$: by structural induction
  - $\|A \Rightarrow B\| = \{t \cdot \pi : t \in |A| \wedge \pi \in \|B\|\}$
  - $\|\forall x.A\| = \bigcup_{n\in\mathbb{N}} \|A\{x := n\}\|$
- truth value $|A|$: $|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall \pi \in \|A\|, t \star \pi \in \perp\!\!\!\perp\}$

Krivine realizability, a 3-steps recipe

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

$$
\begin{array}{lrcl}
\textsc{Push}: & (t)u \star \pi & \succ_1 & t \star u \cdot \pi \\
\textsc{Grab}: & \lambda x.t \star u \cdot \pi & \succ_1 & t\{x := u\} \star \pi \\
\textsc{Save}: & \boldsymbol{cc} \star t \cdot \pi & \succ_1 & t \star \boldsymbol{k}_\pi \cdot \pi \\
\textsc{Restore}: & \boldsymbol{k}_\pi \star t \cdot \rho & \succ_1 & t \star \pi
\end{array}
$$

2. a logical language (*a.k.a. a type system*)
3. formulas interpretation

- pole $⫫$: $⫫ \subset \Lambda \star \Pi$ closed by anti-reduction
- falsity value $\|A\|$: by structural induction
- truth value $|A|$: $|A| = \|A\|^{⫫} = \{t \in \Lambda : \forall \pi \in \|A\|, t \star \pi \in ⫫\}$

Adequacy

Typed terms are realizers.

Realizability models

Given the previous ingredients:

1. a calculus
2. its type system
3. an adequate interpretation of formula

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

Realizability model

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

(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 the previous ingredients:

1. a calculus
2. its type system
3. an adequate interpretation of formula

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

Realizability model

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

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

In other words:

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

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. app : $t, u \mapsto tu$ is from $\Lambda \times \Lambda$ to Λ (*application*)
3. 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*)

$$\begin{array}{rcl} t \star u \cdot \pi \in \perp\!\!\!\perp & \Rightarrow & tu \star \pi \in \perp\!\!\!\perp \\ t \star \pi \in \perp\!\!\!\perp & \Rightarrow & \boldsymbol{k} \star t \cdot u \cdot \pi \in \perp\!\!\!\perp \\ tv(uv) \star \pi \in \perp\!\!\!\perp & \Rightarrow & \boldsymbol{s} \star t \cdot u \cdot v \cdot \pi \in \perp\!\!\!\perp \end{array}$$

$$\begin{array}{rcl} t \star \mathrm{k}_\pi \cdot \pi \in \perp\!\!\!\perp & \Rightarrow & \boldsymbol{cc} \star t \cdot \pi \in \perp\!\!\!\perp \\ t \star \pi \in \perp\!\!\!\perp & \Rightarrow & \mathrm{k}_\pi \star t \cdot \pi' \in \perp\!\!\!\perp \end{array}$$

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 $(\Lambda, \Pi, \mathrm{app}, \mathrm{push}, \mathrm{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, ⫫)$ where:

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

⋮

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

$$\begin{array}{rcl} t \star u \cdot \pi \in ⫫ & \Rightarrow & tu \star \pi \in ⫫ \\ t \star \pi \in ⫫ & \Rightarrow & \boldsymbol{k} \star t \cdot u \cdot \pi \in ⫫ \\ tv(uv) \star \pi \in ⫫ & \Rightarrow & \boldsymbol{s} \star t \cdot u \cdot v \cdot \pi \in ⫫ \end{array} \quad \Bigg| \quad \begin{array}{rcl} t \star \mathrm{k}_\pi \cdot \pi \in ⫫ & \Rightarrow & \boldsymbol{cc} \star t \cdot \pi \in ⫫ \\ t \star \pi \in ⫫ & \Rightarrow & \mathrm{k}_\pi \star t \cdot \pi' \in ⫫ \end{array}$$

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:* $X^{⫫} \triangleq \{t \in \Lambda : \forall \pi \in X, t \star \pi \in ⫫\}$

Streicher's *Abstract Krivine Structures*

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

Abstract Krivine Structures

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

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

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:* $X^{\perp\!\!\!\perp} \triangleq \{t \in \Lambda : \forall \pi \in X, t \star \pi \in \perp\!\!\!\perp\}$

↬ *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 $(\mathcal{A}, \leq, \mathrm{app}, \boldsymbol{k}, \boldsymbol{s})$ where:

- $(\mathcal{A}, \leq)$ is a poset
- $\mathrm{app} : (a, b) \mapsto ab$ 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 $(\mathcal{A}, \leq, \mathrm{app}, \mathtt{imp}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{c}, \Phi)$ where:

- $(\mathcal{A}, \leq, \Phi)$ is a filtered OCA
- $\boldsymbol{e}, \boldsymbol{c} \in \Phi$
- $\mathtt{imp} : (a, b) \mapsto a \to b$ is monotonic from $\mathcal{A}^{op} \times \mathcal{A} \to \mathcal{A}$
- $\boldsymbol{c} \leq ((a \to b) \to a) \to a$
- $a \leq b \to c \quad \Rightarrow \quad ab \leq c \quad$ 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 $(\mathcal{A}, \leq, \text{app}, \boldsymbol{k}, \boldsymbol{s})$ 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 $(\mathcal{A}, \leq, \text{app}, \texttt{imp}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{c}, \Phi)$ where:

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

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

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

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

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

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

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

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

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

Observations

- Everything lays in the order:

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

- In particular, $t \perp\!\!\!\perp A \to B \triangleq t \leq A \to B$ implies that $tA \leq B$ but the converse implication requires e.

- Closure required when defining the AKS:

$$\mathcal{P}_{\perp\!\!\!\perp}(X) \triangleq \{Y \subset X : Y = Y^{\perp\!\!\!\perp\perp\!\!\!\perp}\}$$

- From a filtered OCA, one can define a tripos

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

endowed with the following *entailment* relation:

Observations

- Everything lays in the order:

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

- In particular, $t \perp\!\!\!\perp A \rightarrow B \triangleq t \leq A \rightarrow B$ implies that $tA \leq B$ but the converse implication requires $\boldsymbol{e}$.

- Closure required when defining the AKS:

$$\mathcal{P}_{\perp\!\!\!\perp}(X) \triangleq \{Y \subset X : Y = Y^{\perp\!\!\!\perp\perp\!\!\!\perp}\}$$

- From a filtered OCA, one can define a tripos

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

endowed with the following *entailment* relation:

Observations

- Everything lays in the order:

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

- In particular, $t \perp\!\!\!\perp A \to B \triangleq t \leq A \to B$ implies that $tA \leq B$ but the converse implication requires $\boldsymbol{e}$.

- Closure required when defining the AKS:

$$\mathcal{P}_{\perp\!\!\!\perp}(X) \triangleq \{Y \subset X : Y = Y^{\perp\!\!\!\perp\perp\!\!\!\perp}\}$$

- From a filtered OCA, one can define a tripos

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

endowed with the following *entailment* relation:

Observations

- Everything lays in the order:

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

- In particular, $t \perp\!\!\!\perp A \to B \triangleq t \leq A \to B$ implies that $tA \leq B$ but the converse implication requires $\boldsymbol{e}$.

- Closure required when defining the AKS:

$$\mathcal{P}_{\perp\!\!\!\perp}(X) \triangleq \{Y \subset X : Y = Y^{\perp\!\!\!\perp\perp\!\!\!\perp}\}$$

- From a filtered OCA, one can define a tripos

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

endowed with the following *entailment* relation:

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

Implicative algebras

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:

if $A <: B$, for any pole, if $t \Vdash A$ then $t \Vdash B$.

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:

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

Classical realizability:

if $A <: B$, for any pole, if $t \Vdash A$ then $t \Vdash B$.

In terms of truth values:

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

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:

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

Classical realizability:

if $A <: B$, for any pole, if $t \Vdash A$ then $t \Vdash B$.

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:

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

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

$$\begin{tikzcd} \text{Types} \arrow[r, equal] \arrow[d, dash, dashed] & \text{Formulas} \arrow[d, dash, dashed] \\ \lambda\text{-terms} \arrow[r, equal] & \text{Proofs} \end{tikzcd}$$

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

$$\begin{tikzcd}
\text{Types} \arrow[r, equal] \arrow[d, dashed, dash, "\rotatebox{90}{$\subseteq$}"'] & \text{Formulas} \arrow[d, dashed, dash] \\
\lambda\text{-terms} \arrow[r, equal] & \text{Proofs}
\end{tikzcd}$$

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

$$\begin{tikzcd} \text{Types} \arrow[r, equal] \arrow[d, equal] & \text{Formulas} \arrow[d, equal] \\ \lambda\text{-terms} \arrow[r, equal] & \text{Proofs} \end{tikzcd}$$

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 $(\mathcal{A}, \preccurlyeq, \rightarrow)$ 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)*

Implicative Structures

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

Definition:

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

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

Examples:

- complete Heyting/Boolean algebras

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

- 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:

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

Interpretation of λ-terms

Application:

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

Abstraction:

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

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:

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

Embedding of closed formulas with parameters:

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

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

In particular:

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

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. $\kappa^{\mathcal{A}} \in \mathcal{S}$, $s^{\mathcal{A}} \in \mathcal{S}$,($\boldsymbol{cc} \in \mathcal{S}$) *(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

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

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

Internal logic

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

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 $\vdash_{\mathcal{S}}$)

Adjunction

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

Internal logic

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

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 $\vdash_{\mathcal{S}}$)

Quantifiers:

$$\bigforall_{i \in I} a_i \triangleq \bigcurlywedge_{i \in I} a_i \qquad \bigexists_{i \in I} a_i \triangleq \bigcurlywedge_{c \in A}\Big(\bigcurlywedge_{i \in I}(a_i \to c) \to c\Big)$$

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

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 $\vdash_{\mathcal{S}}$)

Connectives:

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

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

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 $\vdash_{\mathcal{S}}$)

Connectives:

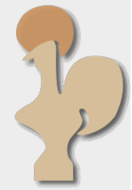
$$a \times b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \to b \to c) \to c) \qquad a + b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \to c) \to (b \to c) \to c)$$

Adjunction

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

Advertisement

Up to this point, everything you saw has been formalized in Coq.



Advertisement

Implicative Structures:

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)*
- $\bigcurlywedge_{b \in B}(a \rightarrow b) = a \rightarrow \bigcurlywedge_{b \in B} b$ *(Distributivity)*

```
Class ImplicativeStructure `{CL:CompleteLattice} := {
   ↦ : X → X → X;
   arrow_mon_l : ∀ a a' b, a ≼ a' → a' ↦ b ≼ a ↦ b;
   arrow_mon_r : ∀ a b b', b ≼ b' → a ↦ b ≼ a ↦ b';
   arrow_meet : ∀ a B, ⋏_{b∈B}(a ↦ b) = a ↦ ⋏_{b∈B} b
 }.
```

Advertisement

Application:

$$a@b \triangleq \bigcurlywedge\{c \in \mathcal{A} : a \preccurlyeq b \mapsto c\}$$

```
Definition app a b := ⋏ (fun c ⇒ a ≼ b ↦ c).
```

Abstraction:

$$\lambda f \triangleq \bigcurlywedge_{a \in \mathcal{A}}(a \mapsto f(a))$$

```
Definition abs f := ⋏ (fun x ⇒ ∃ a, x = a ↦ f a).
```

Advertisement

Adequacy:

$$\vdash t : T \quad \Rightarrow \quad \mathsf{t}^{\mathcal{A}} = \mathsf{x} \quad \Rightarrow \quad \mathsf{T}^{\mathcal{A}} = \mathsf{a} \quad \Rightarrow \quad \mathsf{x} \preccurlyeq \mathsf{a}$$

```
Theorem adequacy_empty:
 ∀ t T x a, fv_typ T = {} → typing_trm empty t T →
              translated t x → translated_typ T a → x ≼ a.
```

Advertisement

Try it!

`https://gitlab.com/emiquey/ImplicativeAlgebras/`

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:*

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

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

$$(\lambda xy.yx)@t \preccurlyeq \Big(\curlywedge_{d\in\mathcal{A}}(a \mapsto b \mapsto d) \mapsto d \Big) \mapsto c$$

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

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

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

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

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

□

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

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

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

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

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

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

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

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

Within Coq:

```
Proof. intros a b c; split;intro H.
- ...   . realizer (λ+ λ+ [$0] $1).
- ...   . realizer (λ+ λ+ λ+ [$2] (λ+ ([[$0] $2] $1))).
Qed. ☺
```

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 $\pitchfork \in \mathcal{S}$.

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.

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

  ↬ *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:

$$\begin{tikzcd} \mathcal{AKS} \arrow[r, bend left, "\mathsf{A}"] & \mathcal{IA} \arrow[l, bend left, "\mathsf{K}"] \end{tikzcd}$$

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 $r \in \mathcal{S}_{\mathcal{B}}$ such that:

1. $f(\mathcal{S}_{\mathcal{A}}) \subseteq \mathcal{S}_{\mathcal{B}}$
2. If $a \vdash a'$, then $r \preccurlyeq f(a \to a') \to f(a) \to f(a')$ $\qquad (\forall a, a' \in \mathcal{A})$
3. $f(\curlywedge P) = \curlywedge\{f(x) : x \in P\}$ $\qquad (\forall P \subseteq \mathcal{A})$

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 $A : \mathcal{AKS} \to \mathcal{IA}$ and $K : \mathcal{IA} \to \mathcal{AKS}$ extend to functors:

$$\begin{tikzcd} \mathbf{AKS} \arrow[r, bend left, "A"] & \mathbf{IA} \arrow[l, bend left, "K"] \end{tikzcd} \qquad \begin{tikzcd} \mathbf{AKSc} \arrow[r, bend left, "A"] & \mathbf{IAc} \arrow[l, bend left, "K"] \end{tikzcd}$$

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:

$$\begin{tikzcd} \mathbf{AKS} \arrow[r, bend left, "A"] \arrow[r, phantom, "\perp"] & \mathbf{IA} \arrow[l, bend left, "K"] \end{tikzcd} \qquad \begin{tikzcd} \mathbf{AKSc} \arrow[r, bend left, "A"] \arrow[r, phantom, "\perp"] & \mathbf{IAc} \arrow[l, bend left, "K"] \end{tikzcd}$$

Theorem

These functors form an adjoint pair.

The end?

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

Does the quest of algebraic foundations for classical realizability stop here?

Questions:

- logic = $\forall, \rightarrow$?
- call-by-name λ_c-calculus ?

The end?

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

Does the quest of algebraic foundations for classical realizability stop here?

Questions:

- logic = $\forall, \rightarrow$?
- call-by-name λ_c-calculus ?

The end?

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

Does the quest of algebraic foundations
for classical realizability stop here?

Questions:

- logic = $\forall, \rightarrow$?
- call-by-name λ_c-calculus ?

The end?

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

Does the quest of algebraic foundations for classical realizability stop here?

Questions:

- logic = $\forall, \rightarrow$?
- call-by-name λ_c-calculus ?

Disjunctive algebras

(fast-tracked)

Decomposing the arrow

Logic:

$$A \longrightarrow B \quad \triangleq \quad \neg A \vee B$$

Different axiomatic:

$$\begin{array}{lcl} S1 & : & (A \vee A) \longrightarrow A \\ S2 & : & A \longrightarrow (A \vee B) \end{array} \qquad \begin{array}{lcl} S3 & : & (A \vee B) \longrightarrow (B \vee A) \\ S4 & : & (A \longrightarrow B) \longrightarrow ((C \vee A) \longrightarrow (C \vee B)) \end{array}$$

λ-calculus:

$$\lambda x.t \triangleq \tilde{\mu}([x],\beta).\langle t \,\|\, \beta\rangle : \neg A \parr B$$

- $L^{\parr}$ fragment of Munch-Maccagnoni's system L
- embedding of the *call-by-name* λ-calculus

Decomposing the arrow

Logic:

$$A \longrightarrow B \quad \triangleq \quad \neg A \vee B$$

Different axiomatic:

$$\begin{array}{lllllll} S1 & : & (A \vee A) \longrightarrow A & \qquad & S3 & : & (A \vee B) \longrightarrow (B \vee A) \\ S2 & : & A \longrightarrow (A \vee B) & & S4 & : & (A \longrightarrow B) \longrightarrow ((C \vee A) \longrightarrow (C \vee B)) \end{array}$$

λ-calculus:

$$\lambda x.t \triangleq \tilde{\mu}([x], \beta).\langle t \parallel \beta \rangle : \neg A \parr B$$

- $L^{\parr}$ fragment of Munch-Maccagnoni's system L
- embedding of the *call-by-name* λ-calculus

Focusing on disjunction: $L^{\parr}$

Focalisation and Classical Realisability
Guillaume Munch-Maccagnoni [2009]

Syntax:

Contexts	e	$::=$	$\alpha \mid (e_1, e_2) \mid [t] \mid \mu x.c$
Terms	t	$::=$	$x \mid \mu(\alpha_1, \alpha_2).c \mid \mu[x].c \mid \mu\alpha.c$
Commands	c	$::=$	$\langle t \,\|\, e \rangle$

Types $A, B ::= X \mid A \parr B \mid \neg A \mid \forall X.A$

Type system:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \,\|\, e \rangle : \Gamma \vdash \Delta}\ (\textsc{Cut})$$

$$\frac{\Gamma \mid e_1 : A \vdash \Delta \quad \Gamma \mid e_2 : B \vdash \Delta}{\Gamma \mid (e_1, e_2) : A \parr B \vdash \Delta}\ (\parr\vdash) \qquad \frac{c : \Gamma \vdash \Delta, \alpha_1 : A, \alpha_2 : B}{\Gamma \vdash \mu(\alpha_1, \alpha_2).c : A \parr B \mid \Delta}\ (\vdash\parr)$$

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \mid [t] : \neg A \vdash \Delta}\ (\neg\vdash) \qquad \frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta}\ (\vdash\neg)$$

Disjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \parr, \neg)$:

1. $\neg$ is anti-monotonic
2. $\parr$ is monotonic
3. $\curlywedge_{b\in B}(a \parr b) = a \parr (\curlywedge_{b\in B} b)$ and $\curlywedge_{b\in B}(b \parr a) = (\curlywedge_{b\in B} b) \parr a$
4. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{\parr}$

Induced implication

$(\mathcal{A}, \preccurlyeq, \overset{\parr}{\mapsto})$ with $a \overset{\parr}{\mapsto} b \triangleq \neg a \parr b$ is an implicative structure

Disjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \parr, \neg)$:

1. $\neg$ is anti-monotonic
2. $\parr$ is monotonic
3. $\bigcurlywedge_{b\in B}(a \parr b) = a \parr (\bigcurlywedge_{b\in B} b)$ and $\bigcurlywedge_{b\in B}(b \parr a) = (\bigcurlywedge_{b\in B} b) \parr a$
4. $\neg \bigcurlywedge_{a\in A} a = \bigcurlyvee_{a\in A} \neg a$

Examples:

- complete Boolean algebras

$$a \parr b \triangleq a \vee b \qquad\qquad \neg a \triangleq \neg a$$

- classical realizability in $L^{\parr}$

Induced implication

$(\mathcal{A}, \preccurlyeq, \overset{\parr}{\mapsto})$ with $a \overset{\parr}{\mapsto} b \triangleq \neg a \parr b$ is an implicative structure

Disjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \parr, \neg)$:

1. $\neg$ is anti-monotonic
2. $\parr$ is monotonic
3. $\curlywedge_{b\in B}(a \parr b) = a \parr (\curlywedge_{b\in B} b)$ and $\curlywedge_{b\in B}(b \parr a) = (\curlywedge_{b\in B} b) \parr a$
4. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{\parr}$

- $\mathcal{A} \triangleq \mathcal{P}(\Pi)$
- $a \preccurlyeq b \triangleq a \supseteq b$
- $a \parr b \triangleq (a, b)$
- $\neg a \triangleq [a^{\perp\!\!\perp}]$

Induced implication

$(\mathcal{A}, \preccurlyeq, \xrightarrow{\parr})$ with $a \xrightarrow{\parr} b \triangleq \neg a \parr b$ is an implicative structure

Disjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, ⅋, \neg)$:

1. $\neg$ is anti-monotonic
2. $⅋$ is monotonic
3. $\curlywedge_{b\in B}(a ⅋ b) = a ⅋ (\curlywedge_{b\in B} b)$ and $\curlywedge_{b\in B}(b ⅋ a) = (\curlywedge_{b\in B} b) ⅋ a$
4. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{⅋}$

Induced implication

$(\mathcal{A}, \preccurlyeq, \overset{⅋}{\mapsto})$ with $a \overset{⅋}{\mapsto} b \triangleq \neg a ⅋ b$ is an implicative structure

Interpreting $\mathrm{L}^{\parr}$

$$\perp\!\!\!\perp \;\triangleq\; ?$$

Intuition:

$$t \Vdash A \quad \text{if} \quad t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}} \qquad \text{and} \qquad t \perp\!\!\!\perp \pi \quad \text{if} \quad t^{\mathcal{A}} \preccurlyeq \pi^{\mathcal{A}}$$

Terms:

- $\mu^{-}.c \triangleq \curlywedge_{a\in\mathcal{A}}\{a : c(a) \in \perp\!\!\!\perp\}$
- $\mu^{()}.c \triangleq \curlywedge_{a,b\in\mathcal{A}}\{a \parr b : c(a,b) \in \perp\!\!\!\perp\}$
- $\mu^{[]}.c \triangleq \curlywedge_{a\in\mathcal{A}}\{\neg a : c(a) \in \perp\!\!\!\perp\}$

Contexts:

- $(a,b) \triangleq a \parr b$
- $[a] \triangleq \neg a$
- $\mu^{+}.c \triangleq \curlyvee_{a\in\mathcal{A}}\{a : c(a) \in \perp\!\!\!\perp\}$

Properties:

Interpreting $L^{⅋}$

$$\perp\!\!\!\perp \triangleq \{(t,e) : t \preccurlyeq e\}$$

Terms:

- $\mu^{-}.c \triangleq \bigcurlywedge_{a\in\mathcal{A}}\{a : c(a) \in \perp\!\!\!\perp\}$
- $\mu^{()}.c \triangleq \bigcurlywedge_{a,b\in\mathcal{A}}\{a ⅋ b : c(a,b) \in \perp\!\!\!\perp\}$
- $\mu^{[]}.c \triangleq \bigcurlywedge_{a\in\mathcal{A}}\{\neg a : c(a) \in \perp\!\!\!\perp\}$

Contexts:

- $(a,b) \triangleq a ⅋ b$
- $[a] \triangleq \neg a$
- $\mu^{+}.c \triangleq \bigcurlyvee_{a\in\mathcal{A}}\{a : c(a) \in \perp\!\!\!\perp\}$

Properties:

Interpreting $L^{\parr}$

$$⫫ \triangleq \{(t,e) : t \preccurlyeq e\}$$

Terms:

- $\mu^{-}.c \triangleq \curlywedge_{a\in\mathcal{A}}\{a : c(a) \in ⫫\}$
- $\mu^{()}.c \triangleq \curlywedge_{a,b\in\mathcal{A}}\{a \parr b : c(a,b) \in ⫫\}$
- $\mu^{[]}.c \triangleq \curlywedge_{a\in\mathcal{A}}\{\neg a : c(a) \in ⫫\}$

Contexts:

- $(a,b) \triangleq a \parr b$
- $[a] \triangleq \neg a$
- $\mu^{+}.c \triangleq \curlyvee_{a\in\mathcal{A}}\{a : c(a) \in ⫫\}$

Properties:

If $c_1 \rightarrow_\beta c_2$ then $c_1^{\mathcal{A}} \trianglelefteq c_2^{\mathcal{A}}$. *($\beta$-reduction)*

Adequacy

1. for any term t, if $\Gamma \vdash t : A \mid \Delta$, then $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$;
2. for any context e, if $\Gamma \mid e : A \vdash \Delta$, then $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$;
3. for any command c, if $c : (\Gamma \vdash \Delta)$, then $(c[\sigma])^{\mathcal{A}} \in ⫫$.

Interpreting $L^{\parr}$

$$⫫ \triangleq \{(t,e) : t \preccurlyeq e\}$$

Terms:

- $\mu^{-}.c \triangleq \bigcurlywedge_{a\in\mathcal{A}}\{a : c(a) \in ⫫\}$
- $\mu^{()}.c \triangleq \bigcurlywedge_{a,b\in\mathcal{A}}\{a \parr b : c(a,b) \in ⫫\}$
- $\mu^{[]}.c \triangleq \bigcurlywedge_{a\in\mathcal{A}}\{\neg a : c(a) \in ⫫\}$

Contexts:

- $(a,b) \triangleq a \parr b$
- $[a] \triangleq \neg a$
- $\mu^{+}.c \triangleq \bigcurlyvee_{a\in\mathcal{A}}\{a : c(a) \in ⫫\}$

Properties:

$$\begin{tikzcd}
\lambda\text{-calculus} \arrow[r, "{[\![\,]\!]}"] \arrow[d, "{[\,]^{\mathcal{A}^{\to}}}"'] & L^{\parr} \arrow[d, "{[\,]^{\mathcal{A}^{\parr}}}"] \\
(\mathcal{A}^{\to}, \preccurlyeq, \to) \arrow[r, "\iota"] & (\mathcal{A}^{\parr}, \preccurlyeq, \parr, \neg)
\end{tikzcd}$$

(Soundness)

Disjunctive algebras

Russel's axioms:

$$
\begin{array}{lcll}
s_1^{⅋} & \triangleq & \bigcurlywedge_{a\in\mathcal{A}} & [(a ⅋ a) \mapsto a] \\
s_2^{⅋} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [a \mapsto (a ⅋ b)] \\
s_3^{⅋} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [(a ⅋ b) \mapsto (b ⅋ a)] \\
s_4^{⅋} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a \mapsto b) \mapsto (c ⅋ a) \mapsto (c ⅋ b)] \\
s_5^{⅋} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a ⅋ (b ⅋ c)) \mapsto ((a ⅋ b) ⅋ c)]
\end{array}
$$

Separator $\mathcal{S}$:

1. If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$ (upward closure)
2. s_1, s_2, s_3, s_4 and s_5 are in $\mathcal{S}$ (combinators)
3. If $a \overset{⅋}{\mapsto} b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (modus ponens)

Disjunctive algebras

$$
\begin{array}{lcll}
s_1^{\parr} & \triangleq & \bigcurlywedge_{a\in\mathcal{A}} & [(a \parr a) \mapsto a] \\
s_2^{\parr} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [a \mapsto (a \parr b)] \\
s_3^{\parr} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [(a \parr b) \mapsto (b \parr a)] \\
s_4^{\parr} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a \mapsto b) \mapsto (c \parr a) \mapsto (c \parr b)] \\
s_5^{\parr} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a \parr (b \parr c)) \mapsto ((a \parr b) \parr c)]
\end{array}
$$

Separator $\mathcal{S}$:

1. If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$ (upward closure)
2. s_1, s_2, s_3, s_4 and s_5 are in $\mathcal{S}$ (combinators)
3. If $a \overset{\parr}{\mapsto} b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (modus ponens)

Examples:

- Complete Boolean algebras

All combinators verify $(s_i)^{\mathcal{B}} = \top$, thus $\top$ or any filter define separators.

- realizability models in $L^{!\parr}$

Disjunctive algebras

$$\begin{array}{lcll}
s_1^{\parr} & \triangleq & \bigcurlywedge_{a\in\mathcal{A}} & [(a \parr a) \mapsto a] \\
s_2^{\parr} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [a \mapsto (a \parr b)] \\
s_3^{\parr} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} & [(a \parr b) \mapsto (b \parr a)] \\
s_4^{\parr} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a \mapsto b) \mapsto (c \parr a) \mapsto (c \parr b)] \\
s_5^{\parr} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} & [(a \parr (b \parr c)) \mapsto ((a \parr b) \parr c)]
\end{array}$$

Separator $\mathcal{S}$:

1. If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$ (upward closure)
2. s_1, s_2, s_3, s_4 and s_5 are in $\mathcal{S}$ (combinators)
3. If $a \overset{\parr}{\mapsto} b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (modus ponens)

Examples:

- Complete Boolean algebras
- realizability models in $L^{!\parr}$

The set of realized falsity values is again a separator.

Internal logic

Recall:

$$a \vdash_{\mathcal{S}} b \;\triangleq\; a \overset{\parr}{\mapsto} b \in \mathcal{S}$$

Sum type:

- $a \parr b \vdash_{\mathcal{S}} a + b$
- $a + b \vdash_{\mathcal{S}} a \parr b$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\parr}{\mapsto} \bot$
- $a \overset{\parr}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

In an implicative structure, we don't have in general:

Internal logic

Recall:

$$a \vdash_{\mathcal{S}} b \;\triangleq\; a \overset{\parr}{\mapsto} b \in \mathcal{S}$$

Sum type:

- $a \parr b \vdash_{\mathcal{S}} a + b$
- $a + b \vdash_{\mathcal{S}} a \parr b$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\parr}{\mapsto} \bot$
- $a \overset{\parr}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

Combinators:

$$\mathbf{K}, \mathbf{S}, \mathbf{cc} \in \mathcal{S}$$

Theorem

$\mathcal{S}$ is an implicative separator

Internal logic

Recall:

$$a \vdash_{\mathcal{S}} b \;\triangleq\; a \overset{\invamp}{\mapsto} b \in \mathcal{S}$$

Sum type:

- $a \invamp b \vdash_{\mathcal{S}} a + b$
- $a + b \vdash_{\mathcal{S}} a \invamp b$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\invamp}{\mapsto} \bot$
- $a \overset{\invamp}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

In an implicative structure, we don't have in general:

$$\bigcurlywedge_{b \in B} (a + b) \overset{?}{=} a + (\bigcurlywedge_{b \in B} b) \qquad \bigcurlywedge_{b \in B} (b + a) \overset{?}{=} (\bigcurlywedge_{b \in B} b) + a \qquad (\bigcurlywedge_{a \in A} a) \to \bot \overset{?}{=} \bigcurlyvee_{a \in A} (a \to \bot)$$

↬ *Implicative algebras are more general !*

Recap

Disjunctive structures:

- induced by classical realizability
- allow to adequately embed $\mathrm{L}^{⅋}$
- are implicative structures

Disjunctive algebras:

- are intrinsically classical
- are implicative algebras
- do not necessarily collapse to a forcing situation

Conclusion

Implicative algebras are more general.

Recap

Disjunctive structures:

- induced by classical realizability
- allow to adequately embed $\mathrm{L}^{\parr}$
- are implicative structures

Disjunctive algebras:

- are intrinsically classical
- are implicative algebras
- do not necessarily collapse to a forcing situation

Conclusion

Implicative algebras are more general.

Recap

Disjunctive structures:

- induced by classical realizability
- allow to adequately embed $L^{\parr}$
- are implicative structures

Disjunctive algebras:

- are intrinsically classical
- are implicative algebras
- do not necessarily collapse to a forcing situation

Conclusion

Implicative algebras are more general.

Conjunctive algebras

(entering call-by-value)

Conjunctive algebras

Logic:

$$A \longrightarrow B \quad \triangleq \quad \neg(A \wedge \neg B)$$

λ-calculus:

$$\lambda x.t \triangleq [\mu(x,[\alpha]).\langle t \parallel \alpha\rangle] : \neg(A \otimes \neg B)$$

↪ *$L^{\otimes}$ fragment* ↪ *call-by-value λ-calculus*

Same process:

1. Conjunctive structures $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$
2. Adequate embedding of $L^{\otimes}$
3. Conjunctive algebras

Conjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$:

1. $\neg$ is anti-monotonic
2. $\otimes$ is monotonic
3. $\curlyvee_{b\in B}(a \otimes b) = a \otimes (\curlyvee_{b\in B} b)$ and $\curlyvee_{b\in B}(b \otimes a) = (\curlyvee_{b\in B} b) \otimes a$
4. $\neg \curlyvee_{a\in A} a = \curlywedge_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{\otimes}$

Duality

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, ⅋, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, ⅋, \neg)$ conjunctive str.

Conjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$:

1. $\neg$ is anti-monotonic
2. $\otimes$ is monotonic
3. $\curlyvee_{b\in B}(a \otimes b) = a \otimes (\curlyvee_{b\in B} b)$ and $\curlyvee_{b\in B}(b \otimes a) = (\curlyvee_{b\in B} b) \otimes a$
4. $\neg \curlyvee_{a\in A} a = \curlywedge_{a\in A} \neg a$

Examples:

- complete Boolean algebras

$$a \otimes b \triangleq a \wedge b \qquad\qquad \neg a \triangleq \neg a$$

- classical realizability in $L^{\otimes}$

Duality

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \parr, \neg)$ conjunctive str.

Conjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$:

1. $\neg$ is anti-monotonic
2. $\otimes$ is monotonic
3. $\curlyvee_{b\in B}(a \otimes b) = a \otimes (\curlyvee_{b\in B} b)$ and $\curlyvee_{b\in B}(b \otimes a) = (\curlyvee_{b\in B} b) \otimes a$
4. $\neg \curlyvee_{a\in A} a = \curlywedge_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{\otimes}$

- $\mathcal{A} \triangleq \mathcal{P}(\mathcal{V})$
- $a \preccurlyeq b \triangleq a \subseteq b$
- $a \otimes b \triangleq (a, b)$
- $\neg a \triangleq [a^{\perp\!\!\perp}]$

Duality

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.

Conjunctive structures

Complete meet-semilattice $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$:

1. $\neg$ is anti-monotonic
2. $\otimes$ is monotonic
3. $\curlyvee_{b\in B}(a \otimes b) = a \otimes (\curlyvee_{b\in B} b)$ and $\curlyvee_{b\in B}(b \otimes a) = (\curlyvee_{b\in B} b) \otimes a$
4. $\neg \curlyvee_{a\in A} a = \curlywedge_{a\in A} \neg a$

Examples:

- complete Boolean algebras
- classical realizability in $L^{\otimes}$

Duality

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \parr, \neg)$ conjunctive str.

Conjunctive algebras ?

Russel's axioms:

$$
\begin{array}{lcl}
\mathbf{s}_1^{\otimes} & \triangleq & \bigwedge_{a\in\mathcal{A}} [a \overset{\otimes}{\mapsto} (a\otimes a)] \\
\mathbf{s}_2^{\otimes} & \triangleq & \bigwedge_{a,b\in\mathcal{A}} [(a\otimes b) \overset{\otimes}{\mapsto} a] \\
\mathbf{s}_3^{\otimes} & \triangleq & \bigwedge_{a,b\in\mathcal{A}} [(a\otimes b) \overset{\otimes}{\mapsto} (b\otimes a)] \\
\mathbf{s}_4^{\otimes} & \triangleq & \bigwedge_{a,b,c\in\mathcal{A}} [(a \overset{\otimes}{\mapsto} b) \overset{\otimes}{\mapsto} (c\otimes a) \overset{\otimes}{\mapsto} (c\otimes b)] \\
\mathbf{s}_5^{\otimes} & \triangleq & \bigwedge_{a,b,c\in\mathcal{A}} [(a\otimes(b\otimes c)) \overset{\otimes}{\mapsto} ((a\otimes b)\otimes c))]
\end{array}
$$

Separator $\mathcal{S}$:

(1) If $a\in\mathcal{S}$ and $a\preccurlyeq b$ then $b\in\mathcal{S}$ (upward closure)

(2) $\mathbf{s}_1^{\otimes}, \mathbf{s}_2^{\otimes}, \mathbf{s}_3^{\otimes}, \mathbf{s}_4^{\otimes}$ and $\mathbf{s}_5^{\otimes}$ are in $\mathcal{S}$ (combinators)

(3) If $a \overset{\otimes}{\mapsto} b\in\mathcal{S}$ and $a\in\mathcal{S}$ then $b\in\mathcal{S}$ (modus ponens)

Reversed disjunctive algebra

If $(\mathcal{A},\preccurlyeq,⅋,\neg,\mathcal{S})$ is a disjunctive algebra, then $(\mathcal{A},\succcurlyeq,⅋,\neg,\neg^{-1}(\mathcal{S}))$ is a conjunctive algebra.

Conjunctive algebras ?

Russel's axioms:

$$
\begin{array}{rcl}
\mathbf{s}_1^{\otimes} & \triangleq & \curlywedge_{a\in\mathcal{A}} [a \overset{\otimes}{\mapsto} (a \otimes a)] \\
\mathbf{s}_2^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} [(a \otimes b) \overset{\otimes}{\mapsto} a] \\
\mathbf{s}_3^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} [(a \otimes b) \overset{\otimes}{\mapsto} (b \otimes a)] \\
\mathbf{s}_4^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} [(a \overset{\otimes}{\mapsto} b) \overset{\otimes}{\mapsto} (c \otimes a) \overset{\otimes}{\mapsto} (c \otimes b)] \\
\mathbf{s}_5^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} [(a \otimes (b \otimes c)) \overset{\otimes}{\mapsto} ((a \otimes b) \otimes c))]
\end{array}
$$

Separator $\mathcal{S}$**:**

(1) If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$ (upward closure)

(2) $\mathbf{s}_1^{\otimes}, \mathbf{s}_2^{\otimes}, \mathbf{s}_3^{\otimes}, \mathbf{s}_4^{\otimes}$ and $\mathbf{s}_5^{\otimes}$ are in $\mathcal{S}$ (combinators)

(3) If $a \overset{\otimes}{\mapsto} b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (modus ponens)

Reversed disjunctive algebra

If $(\mathcal{A}, \preccurlyeq, ⅋, \neg, \mathcal{S})$ is a disjunctive algebra, then $(\mathcal{A}, \succcurlyeq, ⅋, \neg, \neg^{-1}(\mathcal{S}))$ is a conjunctive algebra.

Conjunctive algebras ✗

Reversed disjunctive algebra

If $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$ is a disjunctive algebra, then $(\mathcal{A}, \succcurlyeq, \parr, \neg, \neg^{-1}(\mathcal{S}))$ is a conjunctive algebra.

C'était trop beau pour être vrai

The converse seems impossible to prove...

Technically, we are missing the adjunction:

$$a \preccurlyeq b \overset{\otimes}{\mapsto} c \quad \not\Leftrightarrow \quad a@^{\otimes} b \preccurlyeq c$$

Indeed we have:

$$\bigcurlywedge (a \overset{\otimes}{\mapsto} b) \overset{?}{\neq} a \overset{\otimes}{\mapsto} \bigcurlywedge b \qquad\qquad \bigcurlywedge (a \overset{\otimes}{\mapsto} b) = (\bigcurlyvee a) \overset{\otimes}{\mapsto} b$$

Intuitions:

- conjunctive $\leftrightsquigarrow$ positive $\leftrightsquigarrow$ joins $\bigcurlyvee$, yet axioms with $\forall/\bigcurlywedge$
- values are not closed by application (MP flawed)

Inspecting L$^\otimes$ deduction system

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \,\|\, e\rangle : \Gamma \vdash \Delta}\,(\textsc{Cut})$$

$$\frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}\,(ax\vdash) \qquad \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A \mid \Delta}\,(\vdash ax)$$

$$\frac{c : \Gamma \vdash \Delta, x : A}{\Gamma \mid \mu x.c : A \vdash \Delta}\,(\mu\vdash) \qquad \frac{c : \Gamma, \alpha : A \vdash \Delta}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\,(\vdash\mu)$$

$$\frac{c : (\Gamma, x : A, x' : B \vdash \Delta)}{\Gamma \mid \mu(x, x').c : A \otimes B \vdash \Delta}\,(\otimes\vdash) \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \vdash u : B \mid \Delta}{\Gamma \vdash (t, u) : A \otimes B \mid \Delta}\,(\vdash\otimes)$$

$$\frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \mu[\alpha].c : \neg A \vdash \Delta}\,(\neg\vdash) \qquad \frac{\Gamma \mid e : A \vdash \Delta}{\Gamma \vdash [e] : \neg A \vdash \Delta}\,(\vdash\neg)$$

Inspecting $L^{\otimes}$ deduction system

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\ (\textsc{Cut})$$

$$\frac{A \in \Delta}{\Gamma, A \vdash \Delta}(ax\vdash) \qquad \frac{A \in \Gamma}{\Gamma \vdash A, \Delta}(\vdash ax)$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}(\otimes\vdash) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \otimes B, \Delta}(\vdash\otimes)$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}(\neg\vdash) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}(\vdash\neg)$$

Inspecting L$^{\otimes}$deduction system

One-sided sequents, inlining cuts and contexts

$$\frac{}{\Gamma, A \vdash A}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \otimes B}$$

$$\frac{\Gamma, A, B \vdash C \quad \Gamma, A, B \vdash \neg C}{\Gamma \vdash \neg(A \otimes B)}$$

$$\frac{\Gamma, A \vdash C \quad \Gamma, A \vdash \neg C}{\Gamma \vdash \neg A}$$

Inspecting L$^{\otimes}$deduction system

One-sided sequents, inlining cuts and contexts

$$\frac{}{\Gamma, A \vdash A}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \otimes B}$$

$$\boxed{\frac{\Gamma, A, B \vdash C \quad \Gamma, A, B \vdash \neg C}{\Gamma \vdash \neg(A \otimes B)}}$$

$$\boxed{\frac{\Gamma, A \vdash C \quad \Gamma, A \vdash \neg C}{\Gamma \vdash \neg A}}$$

$\looparrowright$ *this is a calculus of* ***contradiction***

Conjunctive algebras

Axioms through contradictions:

$$
\begin{array}{lcl}
\mathbf{s}_1^{\otimes} & \triangleq & \bigcurlywedge_{a\in\mathcal{A}} \neg[\neg(a\otimes a)\otimes a] \\
\mathbf{s}_2^{\otimes} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} \neg[\neg a\otimes(a\otimes b)] \\
\mathbf{s}_3^{\otimes} & \triangleq & \bigcurlywedge_{a,b\in\mathcal{A}} \neg[\neg(a\otimes b)\otimes(b\otimes a)] \\
\mathbf{s}_4^{\otimes} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(\neg a\otimes b)\otimes(\neg(c\otimes a)\otimes(c\otimes b))] \\
\mathbf{s}_5^{\otimes} & \triangleq & \bigcurlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(a\otimes(b\otimes c))\otimes((a\otimes b)\otimes c)]
\end{array}
$$

Separator $\mathcal{S}$:

1. If $a\in\mathcal{S}$ and $a\preccurlyeq b$ then $b\in\mathcal{S}$. *(upward closure)*
2. $\mathbf{s}_1^{\otimes},\mathbf{s}_2^{\otimes},\mathbf{s}_3^{\otimes},\mathbf{s}_4^{\otimes}$ and $\mathbf{s}_5^{\otimes}$ are in $\mathcal{S}$. *(combinators)*
3. $\boxed{\text{If } \neg(a\otimes b)\in\mathcal{S} \text{ and } a\in\mathcal{S} \text{ then } \neg b\in\mathcal{S}.}$ *(deduction)*
4. If $a\in\mathcal{S}$ and $b\in\mathcal{S}$ then $a\otimes b\in\mathcal{S}$. *(pairs)*

Classical: If $\neg\neg a\in\mathcal{S}$ then $a\in\mathcal{S}$.

Conjunctive algebras

$$
\begin{array}{lcl}
s_1^{\otimes} & \triangleq & \curlywedge_{a\in\mathcal{A}} \neg[\neg(a\otimes a)\otimes a] \\
s_2^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} \neg[\neg a\otimes(a\otimes b)] \\
s_3^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} \neg[\neg(a\otimes b)\otimes(b\otimes a)] \\
s_4^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(\neg a\otimes b)\otimes(\neg(c\otimes a)\otimes(c\otimes b))] \\
s_5^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(a\otimes(b\otimes c))\otimes((a\otimes b)\otimes c)]
\end{array}
$$

Separator $\mathcal{S}$:

1. If $a\in\mathcal{S}$ and $a\preccurlyeq b$ then $b\in\mathcal{S}$. *(upward closure)*
2. $s_1^{\otimes}, s_2^{\otimes}, s_3^{\otimes}, s_4^{\otimes}$ and $s_5^{\otimes}$ are in $\mathcal{S}$. *(combinators)*
3. If $\neg(a\otimes b)\in\mathcal{S}$ and $a\in\mathcal{S}$ then $\neg b\in\mathcal{S}$. *(deduction)*
4. If $a\in\mathcal{S}$ and $b\in\mathcal{S}$ then $a\otimes b\in\mathcal{S}$. *(pairs)*

Classical: If $\neg\neg a\in\mathcal{S}$ then $a\in\mathcal{S}$.

Examples:

- Complete Boolean algebras
- realizability models in $L^{\otimes}$

Conjunctive algebras

$$\begin{array}{lcl}
\boldsymbol{s}_1^{\otimes} & \triangleq & \curlywedge_{a\in\mathcal{A}} \neg[\neg(a\otimes a)\otimes a] \\
\boldsymbol{s}_2^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} \neg[\neg a\otimes(a\otimes b)] \\
\boldsymbol{s}_3^{\otimes} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} \neg[\neg(a\otimes b)\otimes(b\otimes a)] \\
\boldsymbol{s}_4^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(\neg a\otimes b)\otimes(\neg(c\otimes a)\otimes(c\otimes b))] \\
\boldsymbol{s}_5^{\otimes} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(a\otimes(b\otimes c))\otimes((a\otimes b)\otimes c)]
\end{array}$$

Separator $\mathcal{S}$:

1. If $a\in\mathcal{S}$ and $a\preccurlyeq b$ then $b\in\mathcal{S}$. *(upward closure)*
2. $\boldsymbol{s}_1^{\otimes}, \boldsymbol{s}_2^{\otimes}, \boldsymbol{s}_3^{\otimes}, \boldsymbol{s}_4^{\otimes}$ and $\boldsymbol{s}_5^{\otimes}$ are in $\mathcal{S}$. *(combinators)*
3. If $\neg(a\otimes b)\in\mathcal{S}$ and $a\in\mathcal{S}$ then $\neg b\in\mathcal{S}$. *(deduction)*
4. If $a\in\mathcal{S}$ and $b\in\mathcal{S}$ then $a\otimes b\in\mathcal{S}$. *(pairs)*

Classical: If $\neg\neg a\in\mathcal{S}$ then $a\in\mathcal{S}$.

Examples:

- Complete Boolean algebras
- realizability models in $\mathrm{L}^{\otimes}$

Internal logic

Remark:

- in general, we only have: $\dfrac{a \vdash_{\mathcal{S}} b \quad a \in \mathcal{S}}{\neg\neg b \in \mathcal{S}}$
- $a \vdash_{\mathcal{S}} b$ can be composed:

$$a \vdash_{\mathcal{S}} b \quad \textit{and} \quad b \vdash_{\mathcal{S}} c \quad \textit{implies} \quad a \vdash_{\mathcal{S}} c$$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\otimes}{\mapsto} \bot$
- $a \overset{\otimes}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

Heyting Algebra $a \times b \triangleq a \otimes b$ and $a + b \triangleq \neg(\neg a \otimes \neg b)$

1. $a \times b \vdash_{\mathcal{S}} a$
2. $a \times b \vdash_{\mathcal{S}} b$
3. $a \vdash_{\mathcal{S}} a + b$
4. $b \vdash_{\mathcal{S}} a + b$
5. $a \vdash_{\mathcal{S}} b \overset{\otimes}{\mapsto} c$ iff $a \times b \vdash_{\mathcal{S}} c$

Internal logic

Remark:

- in general, we only have: $\dfrac{a \vdash_{\mathcal{S}} b \quad a \in \mathcal{S}}{\neg\neg b \in \mathcal{S}}$
- $a \vdash_{\mathcal{S}} b$ can be composed:

$$a \vdash_{\mathcal{S}} b \quad \textit{and} \quad b \vdash_{\mathcal{S}} c \quad \textit{implies} \quad a \vdash_{\mathcal{S}} c$$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\otimes}{\mapsto} \bot$
- $a \overset{\otimes}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

Heyting Algebra $\quad a \times b \triangleq a \otimes b$ and $a + b \triangleq \neg(\neg a \otimes \neg b)$

1. $a \times b \vdash_{\mathcal{S}} a$
2. $a \times b \vdash_{\mathcal{S}} b$
3. $a \vdash_{\mathcal{S}} a + b$
4. $b \vdash_{\mathcal{S}} a + b$
5. $a \vdash_{\mathcal{S}} b \overset{\otimes}{\mapsto} c$ iff $a \times b \vdash_{\mathcal{S}} c$

Internal logic

Remark:

- in general, we only have: $\dfrac{a \vdash_{\mathcal{S}} b \quad a \in \mathcal{S}}{\neg\neg b \in \mathcal{S}}$
- $a \vdash_{\mathcal{S}} b$ can be composed:

$$a \vdash_{\mathcal{S}} b \quad \textit{and} \quad b \vdash_{\mathcal{S}} c \quad \textit{implies} \quad a \vdash_{\mathcal{S}} c$$

Negation:

- $\neg a \vdash_{\mathcal{S}} a \overset{\otimes}{\mapsto} \bot$
- $a \overset{\otimes}{\mapsto} \bot \vdash_{\mathcal{S}} \neg a$
- $a \vdash_{\mathcal{S}} \neg\neg a$
- $\neg\neg a \vdash_{\mathcal{S}} a$

Heyting Algebra $\qquad a \times b \triangleq a \otimes b$ and $a + b \triangleq \neg(\neg a \otimes \neg b)$

1. $a \times b \vdash_{\mathcal{S}} a$
2. $a \times b \vdash_{\mathcal{S}} b$
3. $a \vdash_{\mathcal{S}} a + b$
4. $b \vdash_{\mathcal{S}} a + b$
5. $a \vdash_{\mathcal{S}} b \overset{\otimes}{\mapsto} c$ iff $a \times b \vdash_{\mathcal{S}} c$

λ-calculus

λ-calculus

$$\lambda f \triangleq \bigcurlywedge_{a \in \mathcal{A}} (a \overset{\otimes}{\mapsto} f(a)) \qquad ab \triangleq \bigcurlywedge \{\neg\neg c : a \preccurlyeq b \overset{\otimes}{\mapsto} c\}$$

Properties

- If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$.
- $(\lambda f)a \preccurlyeq \neg\neg f(a)$

Combinators

1. $\mathbf{s} \in \mathcal{S}$
2. $\mathbf{k} \in \mathcal{S}$

λ-calculus

If $\mathcal{S}$ is classical and t is a closed λ-term, then $t^{\mathcal{A}} \in \mathcal{S}$.

λ-calculus

λ-calculus

$$\lambda f \triangleq \bigcurlywedge_{a \in \mathcal{A}} (a \overset{\otimes}{\mapsto} f(a)) \qquad\qquad ab \triangleq \bigcurlywedge \{\neg\neg c : a \preccurlyeq b \overset{\otimes}{\mapsto} c\}$$

Properties

- If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$.
- $(\lambda f)a \preccurlyeq \neg\neg f(a)$

Combinators

1. $\mathbf{s} \in \mathcal{S}$
2. $\mathbf{k} \in \mathcal{S}$

λ-calculus

If $\mathcal{S}$ is classical and t is a closed λ-term, then $t^{\mathcal{A}} \in \mathcal{S}$.

Conjunctive tripos

Great news:

Theorem

If $(\mathcal{A}, \preccurlyeq, \rightarrow, \mathcal{S})$ is a classical conjunctive algebra, the functor:

$$\mathcal{T} : I \mapsto \mathcal{A}^I/\mathcal{S}[I] \qquad \mathcal{T}(f) : \begin{cases} \mathcal{A}^I/\mathcal{S}[I] & \rightarrow & \mathcal{A}^J/\mathcal{S}[J] \\ [(a_i)_{i\in I}] & \mapsto & [(a_{f(j)})_{j\in J}] \end{cases}$$

(where $f : J \rightarrow I$) defines a tripos.

Duality, the come-back

Structures

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \parr, \neg)$ conjunctive str.

Algebras

1. If $(\mathcal{A}^{\parr}, \mathcal{S}^{\parr})$ is a $\parr$-algebra, then $\neg^{-1}(\mathcal{S}^{\parr}) = \{a : \neg a \in \mathcal{S}^{\parr}\}$ is a valid separator for the dual $\otimes$-structure.
2. If $(\mathcal{A}^{\otimes}, \mathcal{S}^{\otimes})$ is a $\otimes$-algebra, then $\neg^{-1}(\mathcal{S}^{\otimes}) = \{a : \neg a \in \mathcal{S}^{\otimes}\}$ is a valid separator for the dual $\parr$-structure.

Triposes

Let $(\mathcal{A}, \mathcal{S})$ be a $\parr$-algebra and $(\bar{\mathcal{A}}, \bar{\mathcal{S}})$ its dual $\otimes$-algebra. The family:

$$\varphi_I : \left\{ \begin{array}{ccc} \bar{\mathcal{A}}/\bar{\mathcal{S}}[I] & \rightarrow & \mathcal{A}/\mathcal{S}[I] \\ [a_i] & \mapsto & [\neg a_i] \end{array} \right.$$

defines a tripos isomorphism.

Duality, the come-back

Structures

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \parr, \neg)$ conjunctive str.

Algebras

1. If $(\mathcal{A}^{\parr}, \mathcal{S}^{\parr})$ is a $\parr$-algebra, then $\neg^{-1}(\mathcal{S}^{\parr}) = \{a : \neg a \in \mathcal{S}^{\parr}\}$ is a valid separator for the dual $\otimes$-structure.
2. If $(\mathcal{A}^{\otimes}, \mathcal{S}^{\otimes})$ is a $\otimes$-algebra, then $\neg^{-1}(\mathcal{S}^{\otimes}) = \{a : \neg a \in \mathcal{S}^{\otimes}\}$ is a valid separator for the dual $\parr$-structure.

Triposes

Let $(\mathcal{A}, \mathcal{S})$ be a $\parr$-algebra and $(\bar{\mathcal{A}}, \bar{\mathcal{S}})$ its dual $\otimes$-algebra. The family:

$$\varphi_I : \left\{ \begin{array}{ccc} \bar{\mathcal{A}}/\bar{\mathcal{S}}[I] & \to & \mathcal{A}/\mathcal{S}[I] \\ [a_i] & \mapsto & [\neg a_i] \end{array} \right.$$

defines a tripos isomorphism.

Duality, the come-back

Structures

1. $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ conjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \otimes, \neg)$ disjunctive str.
2. $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ disjunctive str. $\Rightarrow$ $(\mathcal{A}, \succcurlyeq, \parr, \neg)$ conjunctive str.

Algebras

1. If $(\mathcal{A}^{\parr}, \mathcal{S}^{\parr})$ is a $\parr$-algebra, then $\neg^{-1}(\mathcal{S}^{\parr}) = \{a : \neg a \in \mathcal{S}^{\parr}\}$ is a valid separator for the dual $\otimes$-structure.
2. If $(\mathcal{A}^{\otimes}, \mathcal{S}^{\otimes})$ is a $\otimes$-algebra, then $\neg^{-1}(\mathcal{S}^{\otimes}) = \{a : \neg a \in \mathcal{S}^{\otimes}\}$ is a valid separator for the dual $\parr$-structure.

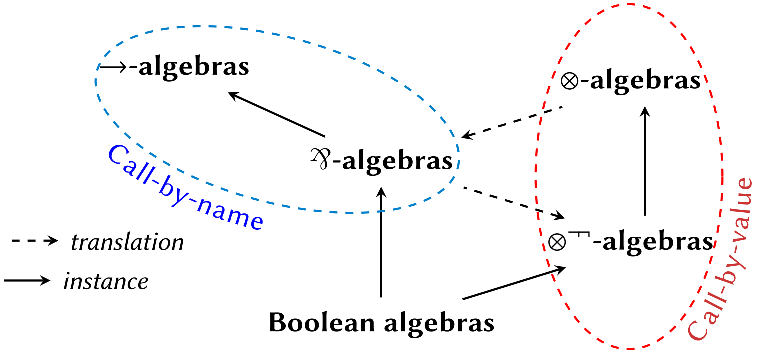
Triposes

Let $(\mathcal{A}, \mathcal{S})$ be a $\parr$-algebra and $(\bar{\mathcal{A}}, \bar{\mathcal{S}})$ its dual $\otimes$-algebra. The family:

$$\varphi_I : \left\{ \begin{array}{ccc} \bar{\mathcal{A}}/\bar{\mathcal{S}}[I] & \to & \mathcal{A}/\mathcal{S}[I] \\ [a_i] & \mapsto & [\neg a_i] \end{array} \right.$$

defines a tripos isomorphism.

Final picture



Fun fact

Oliva & Streicher (2008)

Krivine = Kleene ∘ Friedman

Definition 8.1 (Definition of the negative translation)**.** The formula $A^\perp$ is defined by induction on A by the equations

$$\begin{aligned} (X(e_1,\dots,e_k))^\perp &\equiv X(e_1,\dots,e_k) \\ (A \Rightarrow B)^\perp &\equiv A^{\neg\neg} \wedge B^\perp \\ (\{e\} \Rightarrow B)^\perp &\equiv \mathrm{nat}(e) \wedge B^\perp \end{aligned} \qquad \begin{aligned} (\mathrm{null}(e))^\perp &\equiv \mathrm{null}(\mathrm{neg}(e)) \\ (\forall x\, A)^\perp &\equiv \exists x\, A^\perp \\ (\forall X\, A)^\perp &\equiv \exists X\, A^\perp \end{aligned}$$

(using the unary function 'neg' defined in section 2.1), whereas the formula $A^{\neg\neg}$ is defined as $A^{\neg\neg} \equiv \neg_R A^\perp \equiv A^\perp \Rightarrow R$.

In our setting:

Fun fact

Oliva & Streicher (2008)

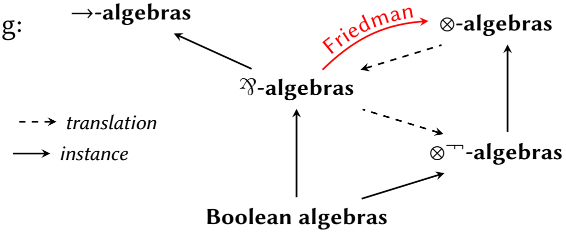
Krivine = Kleene ∘ Friedman

Definition 8.1 (Definition of the negative translation)**.** The formula $A^\perp$ is defined by induction on A by the equations

$$\begin{aligned}(X(e_1,\dots,e_k))^\perp &\equiv X(e_1,\dots,e_k) & (\mathrm{null}(e))^\perp &\equiv \mathrm{null}(\mathrm{neg}(e))\\ (A\Rightarrow B)^\perp &\equiv A^{\neg\neg}\wedge B^\perp & (\forall x\ A)^\perp &\equiv \exists x\ A^\perp\\ (\{e\}\Rightarrow B)^\perp &\equiv \mathrm{nat}(e)\wedge B^\perp & (\forall X\ A)^\perp &\equiv \exists X\ A^\perp\end{aligned}$$

(using the unary function 'neg' defined in section 2.1), whereas the formula $A^{\neg\neg}$ is defined as $A^{\neg\neg}\ \equiv\ \neg_R A^\perp\ \equiv\ A^\perp\Rightarrow R$.

In our setting:



Conclusion

What we have:

- **Disjunctive algebras**, particular cases of implicative ones
- **Conjunctive algebras**, harder to handle
- **Duality** between conjunctive and disjunctive algebras

What is left:

1. By-value implicative algebras:
    - Does it exist?
    - Relation to (by-name) implicative algebras?
    - Tripos equivalence ?
2. Combination of disjunctive and conjunctive algebras:
    - Would it collapse to a forcing situation?
    - Any chance to get call-by-push-value algebras?
3. Algebraic counterpart of strategy/side-effects ? (lazy? memory?)
4. What about toposes? (remark from Van Oosten/Zou's paper)

Conclusion

What we have:

- **Disjunctive algebras**, particular cases of implicative ones
- **Conjunctive algebras**, harder to handle
- **Duality** between conjunctive and disjunctive algebras

What is left:

1. By-value implicative algebras:
   - Does it exist?
   - Relation to (by-name) implicative algebras?
   - Tripos equivalence ?
2. Combination of disjunctive and conjunctive algebras:
   - Would it collapse to a forcing situation?
   - Any chance to get call-by-push-value algebras?
3. Algebraic counterpart of strategy/side-effects ? (lazy? memory?)
4. What about toposes? (remark from Van Oosten/Zou's paper)

Conclusion

What we have:

- **Disjunctive algebras**, particular cases of implicative ones
- **Conjunctive algebras**, harder to handle
- **Duality** between conjunctive and disjunctive algebras

What is left:

1. By-value implicative algebras:
   - Does it exist?
   - Relation to (by-name) implicative algebras?
   - Tripos equivalence ?
2. Combination of disjunctive and conjunctive algebras:
   - Would it collapse to a forcing situation?
   - Any chance to get call-by-push-value algebras?
3. Algebraic counterpart of strategy/side-effects ? (lazy? memory?)
4. What about toposes? (remark from Van Oosten/Zou's paper)

Conclusion

What we have:

- **Disjunctive algebras**, particular cases of implicative ones
- **Conjunctive algebras**, harder to handle
- **Duality** between conjunctive and disjunctive algebras

What is left:

1. By-value implicative algebras:
   - Does it exist?
   - Relation to (by-name) implicative algebras?
   - Tripos equivalence ?
2. Combination of disjunctive and conjunctive algebras:
   - Would it collapse to a forcing situation?
   - Any chance to get call-by-push-value algebras?
3. Algebraic counterpart of strategy/side-effects ? (lazy? memory?)
4. What about toposes? (remark from Van Oosten/Zou's paper)

Conclusion

What we have:

- **Disjunctive algebras**, particular cases of implicative ones
- **Conjunctive algebras**, harder to handle
- **Duality** between conjunctive and disjunctive algebras

What is left:

1. By-value implicative algebras:
   - Does it exist?
   - Relation to (by-name) implicative algebras?
   - Tripos equivalence ?
2. Combination of disjunctive and conjunctive algebras:
   - Would it collapse to a forcing situation?
   - Any chance to get call-by-push-value algebras?
3. Algebraic counterpart of strategy/side-effects ? (lazy? memory?)
4. What about toposes? (remark from Van Oosten/Zou's paper)

References

Implicative algebras

- A. Miquel, *Implicative algebras: a new foundation for realizability and forcing*, 2018, arXiv
- M., *Formalizing Implicative Algebras in Coq* , ITP 18

  ↬ *Coq development:* https://gitlab.com/emiquey/ImplicativeAlgebras

AKS, OCA and categorical stuffs

- W. Ferrer, J. Frey, M. Guillermo, O. Malherbe and A. Miquel, *Ordered combinatory algebras and realizability*, MSCS 2017
- W. Ferrer and O. Malherbe, *The category of implicative algebras and realizability*, 2018, arXiv
- J. Van Oosten and T. Zou, *Classical and relative realizability*, TAC 2016.

Disjunctive & conjunctive algebras

- M., *Revisiting the duality of computation: an algebraic analysis of classical realizability models*, draft