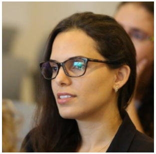
Evidenced Frames:

A Unifying Framework Broadening Realizability Models

Marseille, Journées du GT Scalp 2023

Liron Cohen
Ben-Gurion University



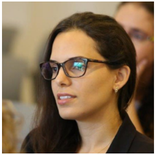
Étienne Miquey
Aix-Marseille Université

Ross Tate
Cornell University

Evidenced Frames are super cool!

Marseille, Journées du GT Scalp 2023

Liron Cohen
Ben-Gurion University



Étienne Miquey
Aix-Marseille Université

Ross Tate
Cornell University

What are *realizability models*?

Realizability

1. Provides **models** for theories *(such as HA2 / HOL / ZF / ...)*

Tarski

$A \mapsto |A| \in \mathbb{B}$ ← Boolean algebra

(intuition: level of truthness)

Realizability

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

(intuition: programs whose computational behavior is guided by A)

2. a for analyzing programs computational behavior

Realizability

1. Provides **models** for theories *(such as HA2 / HOL / ZF / ...)*

Tarski

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

Boolean algebra

(intuition: level of truthness)

Realizability

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

(intuition: programs whose computational behavior is guided by A)

2. a **tool** for analyzing programs computational behavior

Realizability: a 3-steps recipe

1. formulas (*a.k.a. types*)

   ↬ *simple types, SOL, ZF, ...*

2. a computational system (*a.k.a. your favorite calculus*)

   ↬ *some λ-calculus, a combinators algebra, etc.*

3. formulas interpretation (*a.k.a. truth values*)

   ↬ $|A| = \{t \in \Lambda : t \Vdash A\}$

Key ideas:

- realizers compute
- realizers defend the validity of their formula
- truth values are saturated:

Realizability: a 3-steps recipe

1. formulas (*a.k.a. types*)

   ↬ *simple types, SOL, ZF, ...*

2. a computational system (*a.k.a. your favorite calculus*)

   ↬ *some λ-calculus, a combinators algebra, etc.*

3. formulas interpretation (*a.k.a. truth values*)

   ↬ $|A| = \{t \in \Lambda : t \Vdash A\}$

Key ideas:

- realizers compute
- realizers defend the validity of their formula
- truth values are saturated: $t \triangleright^* t' \ \wedge \ t' \in |A| \quad \Rightarrow \quad t \in |A|$

Formal definition

?

Formal definition?

iki/Realizability

Realizability

4 languages

Article Talk Read Edit View history

From Wikipedia, the free encyclopedia

In mathematical logic, **realizability** is a collection of methods in proof theory used to study constructive proofs and extract additional information from them.[1] Formulas from a formal theory are "realized" by objects, known as "realizers", in a way that knowledge of the realizer gives knowledge about the truth of the formula. There are many variations of realizability; exactly which class of formulas is studied and which objects are realizers differ from one variation to another.

Realizability can be seen as a formalization of the BHK interpretation of intuitionistic logic; in realizability the notion of "proof" (which is left undefined in the BHK interpretation) is replaced with a formal notion of "realizer". Most variants of realizability begin with a theorem that any statement that is provable in the formal system being studied is realizable. The realizer, however, usually gives more information about the formula than a formal proof would directly provide.

Beyond giving insight into intuitionistic provability, realizability can be applied to prove the disjunction and existence properties for intuitionistic theories and to extract programs from proofs, as in proof mining. It is also related to topos theory via the realizability topos.

Example: Kleene's 1945-realizability [edit]

Kleene's original version of realizability uses natural numbers as realizers for formulas in Heyting arithmetic. A few pieces of notation are required: first, an ordered pair (n,m) is treated as a single number using a fixed primitive recursive pairing function; second, for each natural number n, φ_n is the computable function with index n. The following clauses are used to define a relation "n realizes A"

Formal definition?

wiki/Boolean_algebra_(structure)

Definition [edit]

A **Boolean algebra** is a six-tuple consisting of a set A, equipped with two binary operations $\wedge$ (called "meet" or "and"), $\vee$ (called "join" or "or"), a unary operation $\neg$ (called "complement" or "not") and two elements 0 and 1 in A (called "bottom" and "top", or "least" and "greatest" element, also denoted by the symbols $\bot$ and $\top$, respectively), such that for all elements a, b and c of A, the following axioms hold:[2]

$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	associativity
$a \vee b = b \vee a$	$a \wedge b = b \wedge a$	commutativity
$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$	absorption
$a \vee 0 = a$	$a \wedge 1 = a$	identity
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	distributivity
$a \vee \neg a = 1$	$a \wedge \neg a = 0$	complements

Note, however, that the absorption law and even the associativity law can be excluded from the set of axioms as they can be derived from the other axioms (see Proven properties).

A Boolean algebra with only one element is called a **trivial Boolean algebra** or a **degenerate Boolean algebra**. (In older works, some authors required 0 and 1 to be *distinct* elements in order to exclude this case.)[*citation needed*]

It follows from the last three pairs of axioms above (identity, distributivity and complements), or from the absorption axiom, that

$$a = b \wedge a \quad \text{if and only if} \quad a \vee b = b.$$

The relation $\leq$ defined by $a \leq b$ if these equivalent conditions hold, is a partial order with least element 0 and greatest element 1. The meet $a \wedge b$ and the join $a \vee b$ of two elements coincide with their infimum and supremum, respectively, with respect to $\leq$.

The first four pairs of axioms constitute a definition of a bounded lattice.

It follows from the first five pairs of axioms that any complement is unique.

The set of axioms is self-dual in the sense that if one exchanges $\vee$ with $\wedge$ and 0 with 1 in an axiom, the result is again an axiom. Therefore, by applying this operation to a Boolean algebra (or Boolean lattice), one obtains another Boolean algebra with the same elements; it is called its **dual**.[3]

Formal definition?

tlab.org/nlab/show/realizability+topos

Home Page All Pages Latest Revisior

realizability topos

Contents

Context

Topos Theory

Constructivism, Realizability, Computability

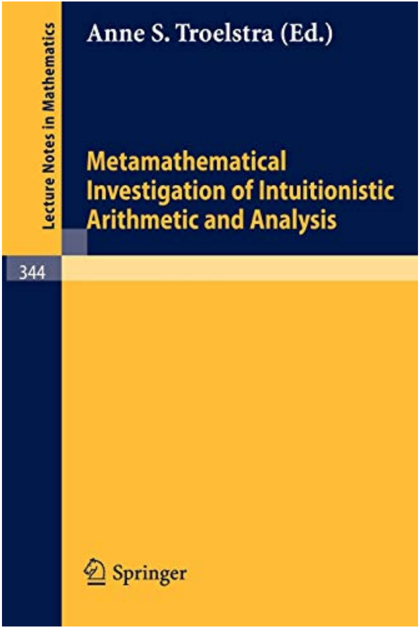
1. Idea

A realizability topos is a topos which embodies the realizability interpretation of intuitionistic number theory (due to Kleene) as part of its internal logic. Realizability toposes form an important class of elementary toposes that are not Grothendieck toposes, and don't even have a geometric morphism to Set.

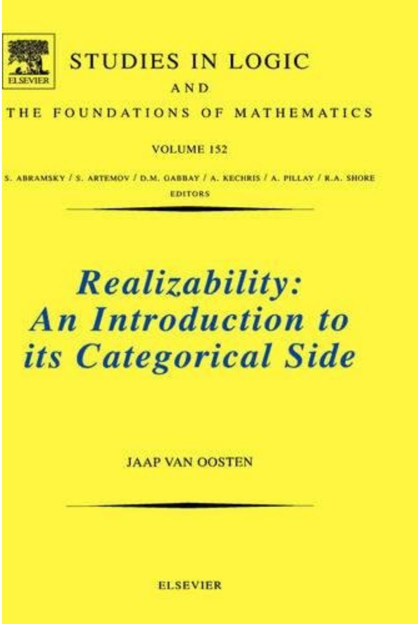
The input datum for forming a realizability topos is a partial combinatory algebra, or PCA.

- When the PCA is Kleene's first algebra $\mathcal{K}_1$, the resulting topos is called the effective topos $\mathrm{RT}(\mathcal{K}_1)$.
- When the PCA is Kleene's second algebra $\mathcal{K}_2$ then $\mathrm{RT}(\mathcal{K}_2)$ is the function realizability topos.

Formal definition?

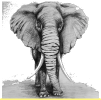


Formal definition?



Topoi, Triposes and PCAs

Pitts' PhD (80s)



PCA → Tripos realizability → Topos model

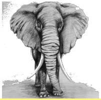
A realizability model

$$\mathcal{M} \vDash A \quad \Leftrightarrow \quad \exists t. t \Vdash A$$

Goal #1

Introduce an intermediate structure that connects the logical and computational aspects

Pitts' PhD (80s)



PCA → Tripos (realizability) → Topos (model)

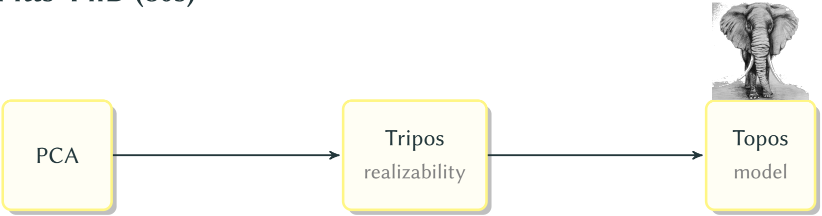
- A functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ with:
  - quantifiers
  - computability w. substitutions
  - generic predicate

Goal #1

Introduce an intermediate structure that connects

Topoi, Triposes and PCAs

Pitts' PhD (80s)



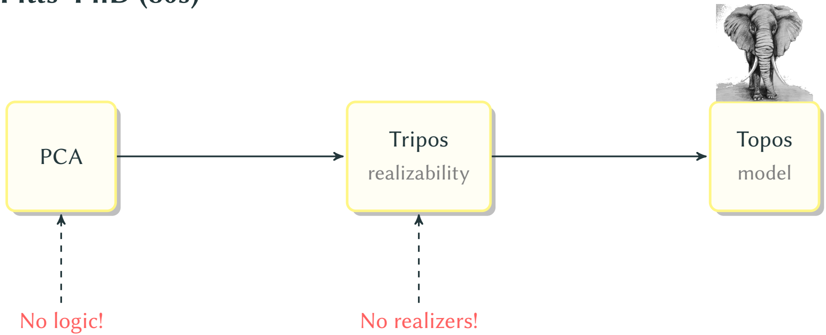
Codes: a set C
Application: a partial binary operator $\cdot : C \times C \rightharpoonup C$ with the functional completeness property.

Goal #1

Introduce an intermediate structure that connects the logical and computational aspects

Topoi, Triposes and PCAs

Pitts' PhD (80s)

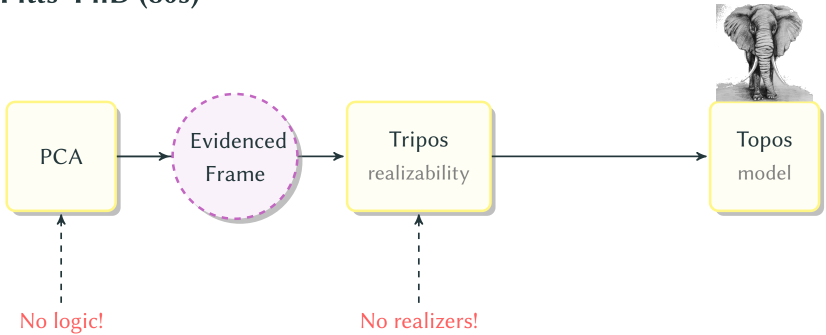


Goal #1

Introduce an intermediate structure that connects
the logical and computational aspects

This Talk: Goal #1

Pitts' PhD (80s)



Goal #1

Introduce an intermediate structure that connects the logical and computational aspects

Computational Choices Matter

With side-effects come new reasoning principles:

- exceptions $\sim$ Markov's principle
- control operators $\sim$ classical logic
- `quote` instruction $\sim$ dependent choice
- memoization $\sim$ dependent choice
- monotonic memory $\sim$ Cohen's forcing
- monotonic memory $\sim$ nonstandard analysis
- ...

But PCAs can only support non-termination!

Computational Choices Matter

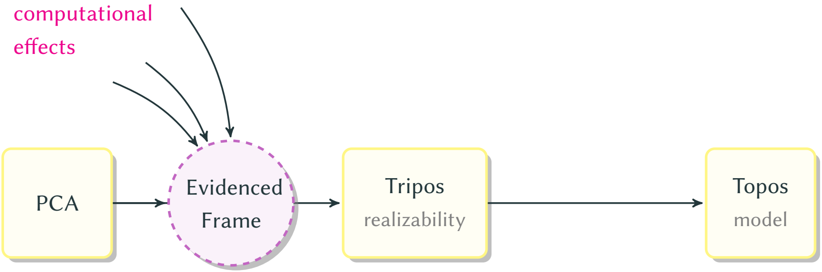
With side-effects come new reasoning principles:

- exceptions $\sim$ Markov's principle
- control operators $\sim$ classical logic
- `quote` instruction $\sim$ dependent choice
- memoization $\sim$ dependent choice
- monotonic memory $\sim$ Cohen's forcing
- monotonic memory $\sim$ nonstandard analysis
- ...

But PCAs can only support non-termination!

This Talk: Goal #2

This Talk: Goal #2



Goal #2

Smooth integration of useful computational effects

Realizability (from an algebraic perspective)

Realizability, a 3-steps recipe

1. formulas (*a.k.a. types*)

   ↬ *simple types,* $2^{nd}-order\ logic$ *, ZF, ...*

2. a computational system (*a.k.a. your favorite calculus*)

   ↬ *some* $\lambda-calculus$ *, a combinators algebra, PCF, etc.*

3. formulas interpretation

Adequacy

If $\mathfrak{p} : (\Gamma \vdash A)$ and $\sigma \Vdash \Gamma$ then $\sigma(\mathfrak{p}^*) \in |A|$.

Realizability, a 3-steps recipe

next slide

1. formulas (*a.k.a. types*)

   ↬ *simple types,* 2^{nd} *– order logic, ZF, ...*

2. a computational system (*a.k.a. your favorite calculus*)

   ↬ *some* λ *– calculus, a combinators algebra, PCF, etc.*

3. formulas interpretation (*a.k.a. truth values*)

   ↬ $|A| = \{t \in \Lambda : t \Vdash A\}$

Adequacy

If $\Gamma \vdash t : A$ and $\sigma \Vdash \Gamma$ then $\sigma(t) \in |A|$.

A simple realizability interpretation

Types & terms: (excerpt)

1st-order exp.	e	$::=$	$x \mid 0 \mid S(e) \mid f(e_1, \ldots, e_n)$
Formulas	A, B	$::=$	$\mathrm{Nat}(e) \mid X(e_1, \ldots, e_n) \mid A \to B \mid \ldots$
			$\mid \forall x.A \mid \exists x.A \mid \forall X.A \mid \exists X.A$
Terms	t, u	$::=$	$x \mid 0 \mid \mathbf{succ} \mid \mathbf{rec} \mid \lambda x.t \mid t\,u \mid \ldots$

where $f : \mathbb{N}^n \to \mathbb{N}$ is any arithmetical function.

Typing rules:

$$\frac{}{\Gamma \vdash 0 : \mathrm{Nat}(0)} \qquad \frac{}{\Gamma \vdash \mathbf{rec} : \forall Z.Z(0) \to (\forall^{\mathbb{N}} y.(Z(y) \to Z(S(y)))) \to \forall^{\mathbb{N}} x.Z(x)}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x\,.\,t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\,u : B}\,(\to_E)$$

$$\frac{\Gamma \vdash t : A[x := n]}{\Gamma \vdash t : \exists x.A} \qquad \frac{\Gamma \vdash t : A[X(x_1, \ldots, x_n) := B]}{\Gamma \vdash t : \exists X.A}$$

A simple realizability interpretation

Types & terms: (excerpt)

1st-order exp.	e	$::=$	$x \mid 0 \mid S(e) \mid f(e_1, \ldots, e_n)$
Formulas	A, B	$::=$	$\mathrm{Nat}(e) \mid X(e_1, \ldots, e_n) \mid A \to B \mid \ldots$
			$\mid \forall x.A \mid \exists x.A \mid \forall X.A \mid \exists X.A$
Terms	t, u	$::=$	$x \mid 0 \mid \mathbf{succ} \mid \mathbf{rec} \mid \lambda x.t \mid t\,u \mid \ldots$

where $f : \mathbb{N}^n \to \mathbb{N}$ is any arithmetical function.

Typing rules:

$$\ldots$$

Reductions:

$$\frac{}{(\lambda x.t)u \triangleright_\beta t[u/x]} \qquad \frac{}{\mathbf{rec}\ u_0\ u_1\ (\mathbf{succ}\ t) \triangleright_\beta u_1\ t\ (\mathbf{rec}\ u_0\ u_1\ t)} \qquad \cdots$$

A simple realizability interpretation

Realizability interpretation:

$$
\begin{array}{rcl}
|\mathrm{Nat}(e)|_\rho & \triangleq & \{t \in \Lambda : t \triangleright^* \mathbf{succ}^n 0, \text{ where } n = [\![e]\!]_\rho\} \\
|X(e_1, \ldots, e_n)|_\rho & \triangleq & \rho(X)([\![e_1]\!]_\rho, \ldots, [\![e_n]\!]_\rho) \\
|\, A \;\to\; B \,|_\rho & \triangleq & \{t \in \Lambda : \; \forall u \in |A|_\rho \,.(\; t\, u \in |B|_\rho \;)\} \\
|\, \forall x \;.A|_\rho & \triangleq & \bigcap_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\, \exists x \;.A|_\rho & \triangleq & \bigcup_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\forall X.A|_\rho & \triangleq & \bigcap_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F} \\
|\exists X.A|_\rho & \triangleq & \bigcup_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F}
\end{array}
$$

Adequacy

If $\Gamma \vdash t : A$ and $\sigma \Vdash \Gamma$ then $\sigma(t) \in |A|$.

Key ideas:

- realizers

A simple realizability interpretation

Realizability interpretation:

$$
\begin{array}{rcl}
|\mathrm{Nat}(e)|_\rho & \triangleq & \{t \in \Lambda : t \triangleright^* \mathbf{succ}^n 0, \text{ where } n = [\![e]\!]_\rho\} \\
|X(e_1, \ldots, e_n)|_\rho & \triangleq & \rho(X)([\![e_1]\!]_\rho, \ldots, [\![e_n]\!]_\rho) \\
|A \to B|_\rho & \triangleq & \{t \in \Lambda : \forall u \in |A|_\rho .(t\, u \in |B|_\rho)\} \\
|\forall x .A|_\rho & \triangleq & \bigcap_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\exists x .A|_\rho & \triangleq & \bigcup_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\forall X .A|_\rho & \triangleq & \bigcap_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F} \\
|\exists X .A|_\rho & \triangleq & \bigcup_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F}
\end{array}
$$

Key ideas:

- realizers *compute*
- realizers *defend the validity* of their formula
- truth values are *saturated*: $t \triangleright^* t' \ \wedge\ t' \in |A| \ \Longrightarrow\ t \in |A|$

A simple realizability interpretation

Realizability interpretation:

$$
\begin{array}{rcl}
|\mathrm{Nat}(e)|_\rho & \triangleq & \{t \in \Lambda : t \triangleright^* \mathbf{succ}^n 0, \text{ where } n = [\![e]\!]_\rho\} \\
|X(e_1,\dots,e_n)|_\rho & \triangleq & \rho(X)([\![e_1]\!]_\rho,\dots,[\![e_n]\!]_\rho) \\
|A \to B|_\rho & \triangleq & \{t \in \Lambda : \forall u \in |A|_\rho .(t\,u \in |B|_\rho)\} \\
|\forall x\,.A|_\rho & \triangleq & \bigcap_{n\in\mathbb{N}} |A|_{\rho,x\leftarrow n} \\
|\exists x\,.A|_\rho & \triangleq & \bigcup_{n\in\mathbb{N}} |A|_{\rho,x\leftarrow n} \\
|\forall X.A|_\rho & \triangleq & \bigcap_{F:\mathbb{N}^k\to\ \mathbf{SAT}} |A|_{\rho,X\leftarrow F} \\
|\exists X.A|_\rho & \triangleq & \bigcup_{F:\mathbb{N}^k\to\ \mathbf{SAT}} |A|_{\rho,X\leftarrow F}
\end{array}
$$

Key ideas:

- realizers *compute*
- realizers *defend the validity* of their formula
- truth values are *saturated*: $t \triangleright^* t' \ \wedge\ t' \in |A| \ \Rightarrow\ t \in |A|$

A simple realizability interpretation

Realizability interpretation:

$$
\begin{aligned}
|\mathrm{Nat}(e)|_\rho &\triangleq \{t \in \Lambda : t \triangleright^* \mathbf{succ}^n 0, \text{ where } n = [\![e]\!]_\rho\} \\
|X(e_1, \ldots, e_n)|_\rho &\triangleq \rho(X)([\![e_1]\!]_\rho, \ldots, [\![e_n]\!]_\rho) \\
|A \to B|_\rho &\triangleq \{t \in \Lambda : \forall u \in |A|_\rho .(t\, u \in |B|_\rho)\} \\
|\forall x.A|_\rho &\triangleq \bigcap_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\exists x.A|_\rho &\triangleq \bigcup_{n \in \mathbb{N}} |A|_{\rho, x \leftarrow n} \\
|\forall X.A|_\rho &\triangleq \bigcap_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F} \\
|\exists X.A|_\rho &\triangleq \bigcup_{F : \mathbb{N}^k \to \mathbf{SAT}} |A|_{\rho, X \leftarrow F}
\end{aligned}
$$

Key ideas:

- realizers *compute*
- realizers *defend the validity* of their formula
- truth values are *saturated*: $t \triangleright^* t' \wedge t' \in |A| \Rightarrow t \in |A|$

A simple realizability interpretation

Realizability interpretation:

$$
\begin{array}{rcl}
|\mathrm{Nat}(e)|_\rho & \triangleq & \{t \in \Lambda : t \triangleright^* \mathbf{succ}^n 0, \text{ where } n = [\![e]\!]_\rho\} \\
|X(e_1,\ldots,e_n)|_\rho & \triangleq & \rho(X)([\![e_1]\!]_\rho,\ldots,[\![e_n]\!]_\rho) \\
|\, A \;\rightarrow\; B\,|_\rho & \triangleq & \{t \in \Lambda :\; \forall u \in |A|_\rho .(\; t\,u \in |B|_\rho\;)\} \\
|\,\forall x\;.A|_\rho & \triangleq & \bigcap_{n\in\mathbb{N}} |A|_{\rho,x\leftarrow n} \\
|\,\exists x\;.A|_\rho & \triangleq & \bigcup_{n\in\mathbb{N}} |A|_{\rho,x\leftarrow n} \\
|\forall X.A|_\rho & \triangleq & \bigcap_{F:\mathbb{N}^k\rightarrow \mathsf{SAT}} |A|_{\rho,X\leftarrow F} \\
|\exists X.A|_\rho & \triangleq & \bigcup_{F:\mathbb{N}^k\rightarrow \mathsf{SAT}} |A|_{\rho,X\leftarrow F}
\end{array}
$$

Key ideas:

- realizers *compute*
- realizers *defend the validity* of their formula
- truth values are *saturated*: $t \triangleright^* t' \;\wedge\; t' \in |A| \;\Rightarrow\; t \in |A|$

Models and triposes

Realizability model

$$\mathcal{M} \vDash A \quad \Leftrightarrow \quad \exists t.t \Vdash A$$

Categorically speaking

a *topos*

Triposes

PCA / λ-calculus / … → Tripos realizability → Topos model

Models and triposes

Realizability model

$$\mathcal{M} \vDash A \quad \Leftrightarrow \quad \exists t.t \Vdash A$$

Categorically speaking

a *topos*

Triposes

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Intuition $\mathcal{T}(\Gamma)$: predicates $\varphi(\vec{x})$ $\quad s^*(\underbrace{\varphi}_{\in\mathcal{T}(\Gamma')}) \triangleq \varphi(\underbrace{s}_{s:\Gamma\to\Gamma'}(\vec{y}))$

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad \Leftrightarrow \quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad \Leftrightarrow \quad \varphi \leq \prod_s(\psi)$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \xrightarrow[s]{} & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Generic predicate There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t.:

for all $\phi \in \mathcal{T}(\Gamma)$, we have $\chi_\phi : \Gamma \to \Omega$ satisfying

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to$ **pHA** s.t.:

Intuition $\mathcal{T}(\Gamma)$: predicates $\varphi(\vec{x})$ $\qquad s^*(\underbrace{\varphi}_{\in\mathcal{T}(\Gamma')}) \triangleq \varphi(\underbrace{s}_{s:\Gamma\to\Gamma'}(\vec{y}))$

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad\Leftrightarrow\quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad\Leftrightarrow\quad \varphi \leq \prod_s(\psi)$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \xrightarrow[s]{} & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Generic predicate There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t.:

for all $\phi \in \mathcal{T}(\Gamma)$, we have $\chi_\phi : \Gamma \to \Omega$ satisfying

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Intuition $\mathcal{T}(\Gamma)$: predicates $\varphi(\vec{x})$ $s^*(\varphi) \triangleq \varphi(s(\vec{y}))$

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad \Leftrightarrow \quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad \Leftrightarrow \quad \varphi \leq \prod_s(\psi)$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \xrightarrow[s]{} & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Generic predicate There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t.:
for all $\phi \in \mathcal{T}(\Gamma)$, we have $\chi_\phi : \Gamma \to \Omega$ satisfying

$$\chi_\phi^*(\text{holds}) = \phi \qquad\qquad \chi_{s^*(\text{holds})} = s$$

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Intuition $\mathcal{T}(\Gamma)$: predicates $\varphi(\vec{x})$ $\quad s^*(\varphi) \triangleq \varphi(s(\vec{y}))$

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad \Leftrightarrow \quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad \Leftrightarrow \quad \varphi \leq \prod_s(\psi)$$

Intuition $\exists x^X.- : \mathcal{T}(\Gamma \times X) \to \mathcal{T}(\Gamma)$ = left adjoint to $\pi_{\Gamma,X} : \Gamma \times X \to \Gamma$;

$$\forall \vec{y}, x.\, [\varphi(\vec{y}, x) \Rightarrow \psi(\vec{y})] \quad \Leftrightarrow \quad \forall \vec{y}.\, [\exists x.\varphi(\vec{y}, x) \Rightarrow \psi(\vec{y})]$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \longrightarrow & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad \Leftrightarrow \quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad \Leftrightarrow \quad \varphi \leq \prod_s(\psi)$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \xrightarrow[s]{} & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Intuition $\qquad (\exists x.\varphi(y,x))[y := s(y')] = \exists x.\varphi(s(y'),x))$

Generic predicate There exists $\Omega \in \mathbf{Set}$ and $\mathrm{holds} \in \mathcal{T}(\Omega)$, s.t.:

for all $\phi \in \mathcal{T}(\Gamma)$, we have $\chi_\phi : \Gamma \to \Omega$ satisfying

$$\chi_\phi^*(\mathrm{holds}) = \phi \qquad\qquad \chi_{s^*(\mathrm{holds})} = s$$

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

$$\varphi \leq s^*(\psi) \quad \Leftrightarrow \quad \coprod_s(\varphi) \leq \psi \qquad\qquad s^*(\varphi) \leq \psi \quad \Leftrightarrow \quad \varphi \leq \prod_s(\psi)$$

Compatibility w. substitutions.

If $\begin{array}{ccc} \Gamma & \xrightarrow{r} & \Gamma' \\ {\scriptstyle v}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \Gamma'' & \xrightarrow[s]{} & \Gamma''' \end{array}$ then $\prod_v \circ r^* = s^* \circ \prod_u$ and $s^* \circ \coprod_u = \coprod_v \circ r^*$.

Generic predicate There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t.:

for all $\phi \in \mathcal{T}(\Gamma)$, we have $\chi_\phi : \Gamma \to \Omega$ satisfying

$$\chi_\phi^*(\text{holds}) = \phi \qquad\qquad \chi_{s^*(\text{holds})} = s$$

Intuition Ω = set of propositions $\qquad$ $\text{holds}(\chi_\varphi(\vec{x})) \equiv \varphi(\vec{x})$

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

Compatibility w. substitutions. ...

Generic predicate. There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t. ...

Standard example (with $\mathcal{H} \in \mathbf{pHA}$)

$$\mathcal{T}(I) = \mathcal{H}^I \qquad\qquad \mathcal{T}(s : I \to J) = \lambda(h : \mathcal{H}^J).\, h \circ s$$

Quantifiers. arbitrary meets/joins

Compatibility. yes.

Generic predicate. $\Omega \triangleq \mathcal{H}$ and holds $\triangleq \lambda x.x$

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{pHA}$ s.t.:

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

Compatibility w. substitutions. ...

Generic predicate. There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t. ...

So far so good but:

- It is a heavy structure
- What about terms and the realizability interpretation?

Recently

Models and triposes

A *tripos* is a functor $\mathcal{T} : \mathbf{Set}^{op} \longrightarrow \mathbf{pHA}$ s.t.:

Quantifiers. Any s^* has left/right adjoints $\coprod_s / \prod_s : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma')$:

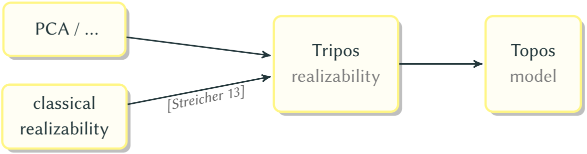
Compatibility w. substitutions. ...

Generic predicate. There exists $\Omega \in \mathbf{Set}$ and holds $\in \mathcal{T}(\Omega)$, s.t. ...

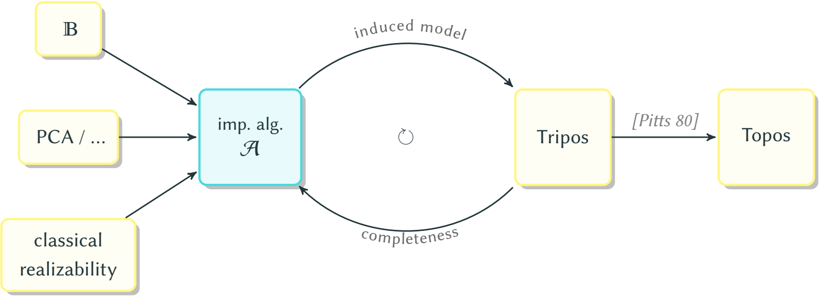
So far so good but:

- It is a heavy structure
- What about terms and the realizability interpretation?

Recently



Implicative algebra [Miquel 20]



Implicative algebra [Miquel 20]

Implicative algebra

complete lattice	$(\mathcal{A}, \preccurlyeq, \curlywedge)$	+	$\cdot \to \cdot \in \mathcal{A}^{\mathcal{A}\times\mathcal{A}}$	*"implication"*
		+	$\mathcal{S} \subseteq \mathcal{A}$	*separator*

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

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

Coq formalization available : `https://gitlab.com/emiquey/ImplicativeAlgebras`

Implicative algebra [Miquel 20]

Implicative algebra

complete lattice	$(\mathcal{A}, \preccurlyeq, \curlywedge)$	+	$\cdot \to \cdot \in \mathcal{A}^{\mathcal{A}\times\mathcal{A}}$	*"implication"*
		+	$\mathcal{S} \subseteq \mathcal{A}$	*separator*

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

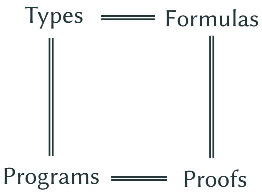
Abstraction $$\lambda f \triangleq \curlywedge_{a\in\mathcal{A}}(a \to f(a))$$

Coq formalization available : `https://gitlab.com/emiquey/ImplicativeAlgebras`

Implicative algebra [Miquel 20]

Implicative algebra

complete lattice	$(\mathcal{A}, \preccurlyeq, \curlywedge)$	+	$\cdot \to \cdot \in \mathcal{A}^{\mathcal{A}\times\mathcal{A}}$	*“implication”*
		+	$\mathcal{S} \subseteq \mathcal{A}$	*separator*



Order relation $\cdot \preccurlyeq \cdot$:

- $A \preccurlyeq B$ — A subtype of B
- $t \preccurlyeq A$ — t realizes A
- $t \preccurlyeq u$ — t is more defined than u

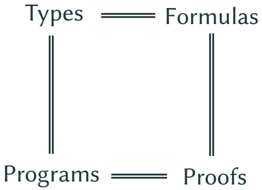
Soundness

1. If $\vdash t : A$ then $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$ *(w.r.t. typing)*
2. If $t \to_\beta u$ then $t^{\mathcal{A}} \preccurlyeq u^{\mathcal{A}}$. *(w.r.t. computation)*

Coq formalization available : `https://gitlab.com/emiquey/ImplicativeAlgebras`

Implicative algebra

complete lattice	$(\mathcal{A}, \preccurlyeq, \curlywedge)$	+	$\cdot \to \cdot \in \mathcal{A}^{\mathcal{A}\times\mathcal{A}}$	*"implication"*
		+	$\mathcal{S} \subseteq \mathcal{A}$	*separator*



Order relation $\cdot \preccurlyeq \cdot$:

- $A \preccurlyeq B$ — A subtype of B
- $t \preccurlyeq A$ — t realizes A
- $t \preccurlyeq u$ — t is more defined than u

Soundness

1. If $\vdash t : A$ then $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$ *(w.r.t. typing)*
2. If $t \to_\beta u$ then $t^{\mathcal{A}} \preccurlyeq u^{\mathcal{A}}$. *(w.r.t. computation)*

Coq formalization available : `https://gitlab.com/emiquey/ImplicativeAlgebras`

Implicative algebra [Miquel 20]

Internal logic

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

Connectives

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

↬ *similar definitions for* $+/\forall/\exists$

Adjunction

$$a \vdash_{\mathcal{S}} b \to c \qquad \text{iff} \qquad a \times b \vdash_{\mathcal{S}} c$$

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

Implicative tripos

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

Implicative algebra [Miquel 20]

Internal logic

$$a \vdash_{\mathcal{S}} b \;\triangleq\; a \rightarrow b \in \mathcal{S}$$

Connectives

$$a \times b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \rightarrow b \rightarrow c) \rightarrow c)$$

↬ *similar definitions for* $+/\forall/\exists$

Adjunction

$$a \vdash_{\mathcal{S}} b \rightarrow c \qquad \text{iff} \qquad a \times b \vdash_{\mathcal{S}} c$$

↬ $(\mathcal{A}/\mathcal{S}, \vdash_{\mathcal{S}}, \times, +, \rightarrow)$ is a Heyting algebra

Implicative tripos

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

Great, but...

- implicative algebras : a (too?) minimal structure

realizers $\in \mathcal{A} \ni$ formulas

- tripos: does not account for realizers

propositional logic	$\leftrightarrow$	Heyting prealgebra
quantifiers	$\leftrightarrow$	adjoints to substitutions
higher-order	$\leftrightarrow$	generic predicate

- lack of a smooth integration for

states / non-determinism / dependent types / ...

In short

Two general structures, *none account for realizers!*

Great, but...

- implicative algebras : a (too?) minimal structure

$$\text{realizers} \in \mathcal{A} \ni \text{formulas}$$

- tripos: does not account for realizers

propositional logic	$\leftrightarrow$	Heyting prealgebra
quantifiers	$\leftrightarrow$	adjoints to substitutions
higher-order	$\leftrightarrow$	generic predicate

- lack of a smooth integration for

states / non-determinism / dependent types / ...

In short

Two general structures, *none account for realizers!*

Great, but...

- implicative algebras : a (too?) minimal structure

$$\text{realizers} \in \mathcal{A} \ni \text{formulas}$$

- tripos: does not account for realizers

propositional logic	$\leftrightarrow$	Heyting prealgebra
quantifiers	$\leftrightarrow$	adjoints to substitutions
higher-order	$\leftrightarrow$	generic predicate

- lack of a smooth integration for

states / non-determinism / dependent types / ...

In short

Two general structures, *none account for realizers!*

Great, but...

- implicative algebras : a (too?) minimal structure

realizers $\in \mathcal{A} \ni$ formulas

- tripos: does not account for realizers

propositional logic	$\leftrightarrow$	Heyting prealgebra
quantifiers	$\leftrightarrow$	adjoints to substitutions
higher-order	$\leftrightarrow$	generic predicate

- lack of a smooth integration for

states / non-determinism / dependent types / ...

In short

Two general structures, *none account for realizers!*

Evidenced Frames

Evidenced Frame: A Unifying Framework for Realizability Models

$$(\; \Phi \; , \; E \; , \; \phi_1 \xrightarrow{e} \phi_2 \;)$$

propositions

evidences

e realizes $\phi_1 \rightarrow \phi_2$

Intuitively: a "specification" of the minimal structure

Evidenced Frame: A Unifying Framework for Realizability Models

$$(\; \Phi \; , \; E \; , \; \phi_1 \xrightarrow{e} \phi_2 \;)$$

propositions — evidences — e realizes $\phi_1 \rightarrow \phi_2$

Intuitively: a “specification” of the minimal structure

Evidenced Frame $(\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$

Reflexivity. $e_{\text{id}} \in E$ s.t.:

- $\phi \xrightarrow{e_{\text{id}}} \phi$

Transitivity. $; \in E \times E \to E$ s.t.:

- $\phi_1 \xrightarrow{e} \phi_2 \wedge \phi_2 \xrightarrow{e'} \phi_3 \implies \phi_1 \xrightarrow{e\,;e'} \phi_3$

Top. $\top \in \Phi$ and $e_\top \in E$ s.t.:

- $\phi \xrightarrow{e_\top} \top$

Conjunction. $\wedge \in \Phi \times \Phi \to \Phi$, $\langle\!|\cdot,\cdot|\!\rangle \in E \times E \to E$, and $e_{\text{fst}}, e_{\text{snd}} \in E$ s.t.:

- $\phi_1 \wedge \phi_2 \xrightarrow{e_{\text{fst}}} \phi_1$
- $\phi_1 \wedge \phi_2 \xrightarrow{e_{\text{snd}}} \phi_2$
- $\phi \xrightarrow{e_1} \phi_1 \wedge \phi \xrightarrow{e_2} \phi_2 \Longrightarrow \phi \xrightarrow{\langle\!| e_1, e_2 |\!\rangle} \phi_1 \wedge \phi_2$

$\phi_1 \wedge \phi_2 \xrightarrow{e_{\text{snd}}} \phi_2 7$

Universal implication. $\supset \in \Phi \times \mathcal{P}(\Phi) \to \Phi$, $\lambda \in E \to E$, and $e_{\text{eval}} \in E$:

- $(\forall \phi \in \vec{\phi}.\ \phi_1 \wedge \phi_2 \xrightarrow{e} \phi) \implies \phi_1 \xrightarrow{\lambda e} \phi_2 \supset \vec{\phi}$
- $\forall \phi \in \vec{\phi}.[(\phi_1 \supset \vec{\phi}) \wedge \phi_1 \xrightarrow{e_{\text{eval}}} \phi]$

Evidenced Frame $(\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$

Reflexivity. $e_{\text{id}} \in E$ s.t.:

- $\phi \xrightarrow{e_{\text{id}}} \phi$

Transitivity. $; \in E \times E \to E$ s.t.:

- $\phi_1 \xrightarrow{e} \phi_2 \wedge \phi_2 \xrightarrow{e'} \phi_3 \implies \phi_1 \xrightarrow{e\,;\,e'} \phi_3$

Top. $\top \in \Phi$ and $e_\top \in E$ s.t.:

- $\phi \xrightarrow{e_\top} \top$

Conjunction. $\wedge \in \Phi \times \Phi \to \Phi$, $\langle\!|\cdot,\cdot|\!\rangle \in E \times E \to E$, and $e_{\text{fst}}, e_{\text{snd}} \in E$ s.t.:

- $\phi_1 \wedge \phi_2 \xrightarrow{e_{\text{fst}}} \phi_1$
- $\phi_1 \wedge \phi_2 \xrightarrow{e_{\text{snd}}} \phi_2$
- $\phi \xrightarrow{e_1} \phi_1 \wedge \phi \xrightarrow{e_2} \phi_2 \implies \phi \xrightarrow{\langle\!| e_1, e_2 |\!\rangle} \phi_1 \wedge \phi_2$

Universal implication. $\supset \in \Phi \times \mathcal{P}(\Phi) \to \Phi$, $\lambda \in E \to E$, and $e_{\text{eval}} \in E$:

- $(\forall \phi \in \vec{\phi}.\ \phi_1 \wedge \phi_2 \xrightarrow{e} \phi) \implies \phi_1 \xrightarrow{\lambda e} \phi_2 \supset \vec{\phi}$
- $\forall \phi \in \vec{\phi}.[(\phi_1 \supset \vec{\phi}) \wedge \phi_1 \xrightarrow{e_{\text{eval}}} \phi]$

Goal #1: An intermediate structure

PCA to Evidenced Frame

- C of "codes"
- *partial* application $c_1 \cdot c_2$

PCA to Evidenced Frame

- C of "codes"
- *partial* application $c_1 \cdot c_2$

expressions	$e ::= i \in \mathbb{N} \mid c \in C \mid e \cdot e$
substitutions	$e[c]$
reduction	$e \downarrow c_r$

+ *functional completeness*:
assignment $e \in E_{n+1} \mapsto c_{\lambda^n.e} \in C$ s.t.

$$c_{\lambda^{n+1}.e} \cdot c_a \downarrow c_{\lambda^n.e[c_a]}$$

$$c_{\lambda^0.e} \cdot c_a \downarrow c_r \iff e[c_a] \downarrow c_r$$

The triple $(\mathcal{P}(C), C, \cdot \xrightarrow{\cdot} \cdot)$, where:

- a proposition in $\Phi = \mathcal{P}(C)$ is defined by its set of realizers
- an evidence in $E = C$ is a code
- $\phi_1 \xrightarrow{e} \phi_2$ if for all $e_1 \in \phi_1$:
  - $e \cdot e_1$ terminates
  - $e \cdot e_1 \downarrow e_2 \implies e_2 \in \phi_2$

↪ *connectives and their evidences are defined as usual in realizability models*

PCA to Evidenced Frame

- C of "codes"
- *partial* application $c_1 \cdot c_2$

expressions	$e ::= i \in \mathbb{N} \mid c \in C \mid e \cdot e$
substitutions	$e[c]$
reduction	$e \downarrow c_r$

+ *functional completeness*:
assignment $e \in E_{n+1} \mapsto c_{\lambda^n.e} \in C$ s.t.

$$c_{\lambda^{n+1}.e} \cdot c_a \downarrow c_{\lambda^n.e[c_a]}$$
$$c_{\lambda^0.e} \cdot c_a \downarrow c_r \iff e[c_a] \downarrow c_r$$

The triple $(\mathcal{P}(C), C, \cdot \xrightarrow{\cdot} \cdot)$, where:

- a proposition in $\Phi = \mathcal{P}(C)$ is defined by its set of realizers
- an evidence in $E = C$ is a code
- $\phi_1 \xrightarrow{e} \phi_2$ if for all $e_1 \in \phi_1$:
  - $e \cdot e_1$ terminates
  - $e \cdot e_1 \downarrow e_2 \implies e_2 \in \phi_2$

↬ *connectives and their evidences are defined as usual in realizability models*

Evidenced Frame to Tripos

UFam construction

Given $\mathcal{EF} = (\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$, the structure $\mathrm{UFam}(\mathcal{EF})$ is defined by:

Predicates. $\Gamma \in$ **Set** is mapped to $\Phi^{\Gamma} \in$ **pHA**

- $\phi \preccurlyeq \phi' \triangleq \exists e. \forall \gamma.\ \phi(\gamma) \xrightarrow{e} \phi'(\gamma)$ *uniform entailment*
- Heyting prealgebra: pointwise via Φ

Substitution. $s : \Gamma \to \Gamma'$ is mapped to $\mathcal{T}(s) = \lambda h.\ h \circ s$ *as usual*

Quantifiers. $\prod_u \in \Phi^I \to \Phi^J \triangleq \lambda \phi. \lambda j.\ \prod_{i \in u^{-1}(j)} \phi(i)$ *as usual*

$\coprod_u \in \Phi^I \to \Phi^J \triangleq \lambda \phi. \lambda j.\ \coprod_{i \in u^{-1}(j)} \phi(i).$

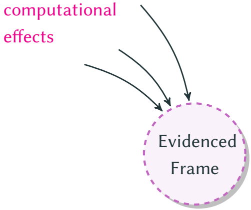
Generic predicate. $\Omega \triangleq \Phi$, holds $\triangleq \mathrm{id}_{\Omega}$, and $\chi_{\phi} \triangleq \phi$. *as usual*

Goal #2: Smooth integration of useful computational effects

More Computational Effects into Evidenced Frame

Computational System:

- Σ – inhabited set of states σ
- $\sigma \preccurlyeq \sigma'$ –"possible future" preorder
- $e \downarrow^{\sigma}_{\sigma'} c$ – reduction relation
- $e\downarrow^{\sigma}$ – termination relation



$$\frac{}{c \downarrow^{\sigma}_{\sigma} c} \qquad \frac{e_f \downarrow^{\sigma}_{\sigma'} c_f \qquad e_a \downarrow^{\sigma'}_{\sigma''} c_a \qquad c_f \cdot c_a \downarrow^{\sigma''}_{\sigma'''} c_r}{e_f \cdot e_a \downarrow^{\sigma}_{\sigma'''} c_r} \qquad \frac{}{c\downarrow^{\sigma}}$$

$$\frac{e_f\downarrow^{\sigma} \qquad \forall \sigma', c_f . e_f \downarrow^{\sigma}_{\sigma'} c_f \implies e_a\downarrow^{\sigma'} \wedge \forall \sigma'', c_a .\ e_a \downarrow^{\sigma'}_{\sigma''} c_a \Longrightarrow c_f \cdot c_a\downarrow^{\sigma''}}{e_f \cdot e_a\downarrow^{\sigma}}$$

+

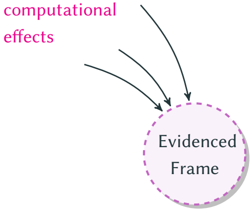
Functional completeness: assignment $e \in E_{n+1} \mapsto c_{\lambda^n.e} \in C$ s.t.:

reduction $\quad c_{\lambda^{n+1}.e} \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r \implies \sigma' = \sigma \wedge c_r = c_{\lambda^n.e[c_a]}$

More Computational Effects into Evidenced Frame

Computational System:

- Σ – inhabited set of states σ
- $\sigma \preccurlyeq \sigma'$ –"possible future" preorder
- $e \downarrow^{\sigma}_{\sigma'} c$ – reduction relation
- $e{\downarrow}^{\sigma}$ – termination relation



...

+

Functional completeness: assignment $e \in E_{n+1} \mapsto c_{\lambda^n.e} \in C$ s.t.:

$$
\begin{array}{lrcl}
\textit{reduction} & c_{\lambda^{n+1}.e} \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r & \implies & \sigma' = \sigma \wedge c_r = c_{\lambda^n.e[c_a]} \\
 & c_{\lambda^0.e} \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r & \implies & e[c_a] \downarrow^{\sigma}_{\sigma'} c_r \\
\textit{termination} & & & c_{\lambda^{n+1}.e} \cdot c_a{\downarrow}^{\sigma} \\
 & e[c_a]{\downarrow}^{\sigma} & \implies & c_{\lambda^0.e} \cdot c_a{\downarrow}^{\sigma}
\end{array}
$$

Preservation: $\forall \sigma, c_f, c_a, \sigma', c_r.\ c_f \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r \implies \sigma \preccurlyeq \sigma'$

Common Effects

★ **PCA** – $\mathscr{C}$

- states Σ : singleton

★ **Non-determinism** - $\mathscr{C}_{\mathsf{flip}}$

$$\frac{}{\mathsf{flip}\cdot c{\downarrow}^{\sigma}} \qquad \frac{}{\mathsf{flip}\cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.0}} \qquad \frac{}{\mathsf{flip}\cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.1}}$$

★ **Mutable state** - $\mathscr{C}_{\mathsf{lookup}}$

- Σ – finite partial maps from $\mathbb{N}$ to codes
- $\sigma \preccurlyeq \sigma'$ – inclusion
- codes – generated from the combinators lookup_n

$$\frac{}{\mathsf{lookup}_n\cdot c{\downarrow}^{\sigma}} \qquad \frac{n \mapsto c' \in \sigma}{\mathsf{lookup}_n\cdot c \downarrow^{\sigma}_{\sigma} c'} \qquad \frac{\neg(\exists c'.\, n \mapsto c' \in \sigma)}{\mathsf{lookup}_n\cdot c \downarrow^{\sigma}_{\sigma,n\mapsto c} c}$$

★ **Failure** - $\mathscr{C}_{\mathsf{fail}}$

$$\frac{}{\mathsf{fail}\cdot c{\downarrow}^{\sigma}}$$

Common Effects

★ **PCA** – $\mathscr{C}$

- states Σ : singleton

★ **Non-determinism** - $\mathscr{C}_{\text{flip}}$

$$\frac{}{\text{flip} \cdot c {\downarrow}^{\sigma}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.0}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.1}}$$

★ **Mutable state** - $\mathscr{C}_{\text{lookup}}$

- Σ – finite partial maps from $\mathbb{N}$ to codes
- $\sigma \preccurlyeq \sigma'$ – inclusion
- codes – generated from the combinators lookup_n

$$\frac{}{\text{lookup}_n \cdot c {\downarrow}^{\sigma}} \qquad \frac{n \mapsto c' \in \sigma}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma} c'} \qquad \frac{\neg(\exists c'.\, n \mapsto c' \in \sigma)}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma, n \mapsto c} c}$$

★ **Failure** - $\mathscr{C}_{\text{fail}}$

$$\frac{}{\text{fail} \cdot c {\downarrow}^{\sigma}}$$

Common Effects

★ **PCA** – $\mathscr{C}$

- states Σ : singleton

★ **Non-determinism** - $\mathscr{C}_{\text{flip}}$

$$\frac{}{\text{flip} \cdot c{\downarrow}^{\sigma}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.0}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.1}}$$

★ **Mutable state** - $\mathscr{C}_{\text{lookup}}$

- Σ – finite partial maps from $\mathbb{N}$ to codes
- $\sigma \preccurlyeq \sigma'$ – inclusion
- codes – generated from the combinators lookup_n

$$\frac{}{\text{lookup}_n \cdot c{\downarrow}^{\sigma}} \qquad \frac{n \mapsto c' \in \sigma}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma} c'} \qquad \frac{\neg(\exists c'.\, n \mapsto c' \in \sigma)}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma, n \mapsto c} c}$$

★ **Failure** - $\mathscr{C}_{\text{fail}}$

$$\frac{}{\text{fail} \cdot c{\downarrow}^{\sigma}}$$

Common Effects

★ **PCA** – $\mathscr{C}$

- states Σ : singleton

★ **Non-determinism** - $\mathscr{C}_{\text{flip}}$

$$\frac{}{\text{flip} \cdot c{\downarrow}^{\sigma}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.0}} \qquad \frac{}{\text{flip} \cdot c \downarrow^{\sigma}_{\sigma} c_{\lambda^1.1}}$$

★ **Mutable state** - $\mathscr{C}_{\text{lookup}}$

- Σ – finite partial maps from $\mathbb{N}$ to codes
- $\sigma \preccurlyeq \sigma'$ – inclusion
- codes – generated from the combinators lookup_n

$$\frac{}{\text{lookup}_n \cdot c{\downarrow}^{\sigma}} \qquad \frac{n \mapsto c' \in \sigma}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma} c'} \qquad \frac{\neg(\exists c'.\ n \mapsto c' \in \sigma)}{\text{lookup}_n \cdot c \downarrow^{\sigma}_{\sigma, n \mapsto c} c}$$

★ **Failure** - $\mathscr{C}_{\text{fail}}$

$$\frac{}{\text{fail} \cdot c{\downarrow}^{\sigma}}$$

The Induced Evidenced Frame

Intuition:

- propositions "*future-stable*" $\phi \in \mathcal{P}(C \times \Sigma)$

  ↬ *notation:* $\phi^\sigma(c) \triangleq (c, \sigma) \in \phi$

- evidences $E = C$ are codes,
- $\phi_1 \xrightarrow{e} \phi_2$ if for all e_1 such that ${\phi_1}^\sigma(e_1)$:
  - $e \cdot e_1 \downarrow^\sigma$ terminates
  - $e \cdot e_1 \downarrow^\sigma_{\sigma'} e_2 \quad \Rightarrow \quad {\phi_2}^{\sigma'}(e_2)$

The Induced Evidenced Frame

Intuition:

- propositions "*future-stable*" $\phi \in \mathcal{P}(C \times \Sigma)$

  ↬ *notation:* $\phi^\sigma(c) \triangleq (c, \sigma) \in \phi$

- evidences $E = C$ are valid codes,
- $\phi_1 \xrightarrow{e} \phi_2$ if for all e_1 such that ${\phi_1}^\sigma(e_1)$:
  - $e \cdot e_1 \downarrow^\sigma$ terminates
  - $e \cdot e_1 \downarrow^\sigma_{\sigma'} e_2 \quad \Rightarrow \quad {\phi_2}^{\sigma'}(e_2)$

separator $\mathcal{S}$ = functionally complete subset of C

+ closed under reduction

+ **Progress** $\forall \sigma \in \Sigma, c_f, c_a \in \mathcal{S}.\ c_f \cdot c_a \downarrow^\sigma \implies \exists \sigma', c_r.\ \ c_f \cdot c_a \downarrow^\sigma_{\sigma'} c_r$

Examples:

- $\mathcal{S}_\top$: all codes *(when progress holds for all codes)*
- $\mathcal{S}_\lambda$: generated solely from functional completeness

The Induced Evidenced Frame

Intuition:

- propositions "*future-stable*" $\phi \in \mathcal{P}(C \times \Sigma)$

  ↬ *notation:* $\phi^{\sigma}(c) \triangleq (c, \sigma) \in \phi$

- evidences $E = C$ are valid codes,
- $\phi_1 \xrightarrow{e} \phi_2$ if for all e_1 such that $\phi_1{}^{\sigma}(e_1)$:
  - $e \cdot e_1 \downarrow^{\sigma}$ terminates
  - $\{\forall/\exists\} e_2, \sigma'.\ e \cdot e_1 \downarrow^{\sigma}_{\sigma'} e_2 \quad \Rightarrow \quad \phi_2{}^{\sigma'}(e_2)$

Interpretation $\mathfrak{D}$ of non-determinism

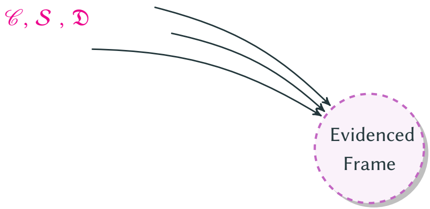
- **Demonic:** when *all* possible results of the reduction are realizers.

$$c_f \cdot c_a \Downarrow^{\sigma}_{D} \phi \triangleq c_f \cdot c_a {\downarrow}^{\sigma} \wedge \forall \sigma', c_r.\ c_f \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r \Longrightarrow \phi^{\sigma'}(c_r)$$

- **Angelic:** when *a* possible result of the reduction is a realizer.

$$c_f \cdot c_a \Downarrow^{\sigma}_{A} \phi \triangleq c_f \cdot c_a {\downarrow}^{\sigma} \wedge \exists \sigma', c_r.\ c_f \cdot c_a \downarrow^{\sigma}_{\sigma'} c_r \wedge \phi^{\sigma'}(c_r)$$

Computational Effects to Evidenced Frame



The triple $\mathcal{EF}_{\mathfrak{D}}^{\mathscr{C},\mathcal{S}} = \langle \Phi, E, \cdot \xrightarrow{\cdot} \cdot \rangle$ defines an evidenced frame where:

- propositions $\Phi \in \mathcal{P}(\Sigma \times C)$ are "future-stable" stateful predicates:

$$\forall \sigma, \sigma', c.\ \ \sigma \preccurlyeq \sigma' \wedge \phi^{\sigma}(c) \implies \phi^{\sigma'}(c)$$

- E is the set of codes in the separator $\mathcal{S}$.
- $\phi_1 \xrightarrow{e} \phi_2$ is defined as

$$\forall \sigma, c.\ \phi_1^{\sigma}(c) \Longrightarrow e \cdot c \downdownarrows_{\mathfrak{D}}^{\sigma} \phi_2$$

A Bit More...

Byproduct: Robust interpretations

Countable choice

$\mathcal{E}ff \models \forall R \in \mathbb{N} \times B.\mathrm{Tot}(R) \Rightarrow \exists S \in \mathbb{N} \times B.\mathrm{Tot}(S) \wedge \mathrm{S} \subseteq \mathrm{R} \wedge \mathrm{Det}(\mathrm{S})$

Sketch

1. Propositions $\in \mathcal{P}(\mathrm{Code})$
2. $v_{tot} \Vdash \mathrm{Tot}(R) \Rightarrow \forall n \in \mathbb{N}.\exists b \in B.\ v_{tot}\bar{n} \downarrow v_n \in R(n,b)$
3. For each n pick[†] one such b_n and define $S(n, b_n) \triangleq \{v_n\}$

Then

- $v_{tot} \Vdash \mathrm{Tot}(S)$
- $\lambda x.x \Vdash S \subseteq R$
- $\mathrm{Det}(S)$ by construction

Byproduct: Robust interpretations

Countable choice

$$\mathcal{E}ff \vDash \forall R \in \mathbb{N} \times B.\mathrm{Tot}(R) \Rightarrow \exists S \in \mathbb{N} \times B.\mathrm{Tot}(S) \land \mathrm{S} \subseteq \mathrm{R} \land \mathrm{Det}(\mathrm{S})$$

Sketch

1. Propositions $\in \mathcal{P}(\text{Code})$
2. $v_{tot} \Vdash \mathrm{Tot}(R) \Rightarrow \forall n \in \mathbb{N}.\exists b \in B.\ v_{tot}\bar{n} \downarrow v_n \in R(n, b)$
3. For each n pick[†] one such b_n and define $S(n, b_n) \triangleq \{v_n\}$

Then

- $v_{tot} \Vdash \mathrm{Tot}(S)$
- $\lambda x.x \Vdash S \subseteq R$
- $\mathrm{Det}(S)$ by construction

Byproduct: Robust interpretations

Countable choice

$$\mathcal{E}ff \vDash \forall R \in \mathbb{N} \times B.\mathrm{Tot}(R) \Rightarrow \exists S \in \mathbb{N} \times B.\mathrm{Tot}(S) \wedge \mathrm{S} \subseteq \mathrm{R} \wedge \mathrm{Det}(\mathrm{S})$$

Sketch

1. Propositions $\in \mathcal{P}(\mathrm{Code})$
2. $v_{tot} \Vdash \mathrm{Tot}(R) \Rightarrow \forall n \in \mathbb{N}.\exists b \in B.\ v_{tot}\bar{n} \downarrow v_n \in R(n, b)$
3. For each n pick[†] one such b_n and define $S(n, b_n) \triangleq \{v_n\}$

Then

- $v_{tot} \Vdash \mathrm{Tot}(S)$
- $\lambda x.x \Vdash S \subseteq R$
- $\mathrm{Det}(S)$ by construction

Byproduct: Robust interpretations

Countable choice

$$\mathcal{E}\!f\!f \vDash \forall R \in \mathbb{N} \times B.\mathrm{Tot}(R) \Rightarrow \exists S \in \mathbb{N} \times B.\mathrm{Tot}(S) \wedge \mathrm{S} \subseteq \mathrm{R} \wedge \mathrm{Det}(\mathrm{S})$$

Sketch

1. Propositions $\in \mathcal{P}(\mathrm{Code})$
2. $v_{tot} \Vdash \mathrm{Tot}(R) \Rightarrow \forall n \in \mathbb{N}.\exists b \in B.\ v_{tot}\bar{n} \downarrow v_n \in R(n, b)$
3. For each n pick† one such b_n and define $S(n, b_n) \triangleq \{v_n\}$

Then

- $v_{tot} \Vdash \mathrm{Tot}(S)$
- $\lambda x.x \Vdash S \subseteq R$
- $\mathrm{Det}(S)$ by construction

Byproduct: Robust interpretations

Countable choice

$$\mathcal{E}ff \vDash \forall R \in \mathbb{N} \times B.\mathrm{Tot}(R) \Rightarrow \exists S \in \mathbb{N} \times B.\mathrm{Tot}(S) \wedge \mathrm{S} \subseteq \mathrm{R} \wedge \mathrm{Det}(\mathrm{S})$$

Sketch

1. Propositions $\in \mathcal{P}(\mathrm{Code})$
2. $v_{tot} \Vdash \mathrm{Tot}(R) \Rightarrow \forall n \in \mathbb{N}.\exists b \in B.\ v_{tot}\bar{n} \downarrow v_n \in R(n, b)$
3. For each n pick† one such b_n and define $S(n, b_n) \triangleq \{v_n\}$

Then

- $v_{tot} \Vdash \mathrm{Tot}(S)$?
- $\lambda x.x \Vdash S \subseteq R$
- $\mathrm{Det}(S)$ by construction

Byproduct: Robust interpretations

A short story *[Cohen, Abreu, Tate 2019]*

- Thm 1. - the realizability model induced by a PCA models CC.

$$UFam(\mathcal{EF}^{\mathscr{C}}) \vDash CC$$

- Thm 2. - adding non-determinism makes it negate CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip}}) \nvDash CC$$

- Thm 3. - adding states and using memoization restores CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip,lookup}}) \vDash CC$$

Byproduct: Robust interpretations

A short story *[Cohen, Abreu, Tate 2019]*

- Thm 1. - the realizability model induced by a PCA models CC.

$$UFam(\mathcal{EF}^{\mathcal{C}}) \vDash CC$$

- Thm 2. - adding non-determinism makes it negate CC.

$$UFam(\mathcal{EF}_D^{\mathcal{C}_{flip}}) \nvDash CC$$

- Thm 3. - adding states and using memoization restores CC.

$$UFam(\mathcal{EF}_D^{\mathcal{C}_{flip,lookup}}) \vDash CC$$

$$UFam(\mathcal{EF}_D^{\mathcal{C}_{flip,lookup}}) = specification \neq implementation$$

Byproduct: Robust interpretations

A short story *[Cohen, Abreu, Tate 2019]*

- Thm 1. - the realizability model induced by a PCA models CC.

$$UFam(\mathcal{EF}^{\mathscr{C}}) \vDash CC$$

- Thm 2. - adding non-determinism makes it negate CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip}}) \nvDash CC$$

- Thm 3. - adding states and using memoization restores CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip,lookup}}) \vDash CC$$

Byproduct: Robust interpretations

A short story *[Cohen, Abreu, Tate 2019]*

- Thm 1. - the realizability model induced by a PCA models CC.

$$UFam(\mathcal{EF}^{\mathscr{C}}) \vDash CC$$

- Thm 2. - adding non-determinism makes it negate CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip}}) \nvDash CC$$

- Thm 3. - adding states and using memoization restores CC.

$$UFam(\mathcal{EF}_D^{\mathscr{C}_{flip,lookup}}) \vDash CC$$

$$\mathrm{UFam}(\mathcal{EF}_D^{\mathscr{C}_{\mathrm{flip,lookup}}}) \quad = \quad \textit{specification} \quad \neq \quad \textit{implementation}$$

From Implicative Algebras to Evidenced frames

Any implicative algebra $(\mathcal{A}, \preccurlyeq, \rightarrow, \mathcal{S})$ induces an evidenced frame

$$\mathsf{UEF}(\mathcal{A}) \triangleq (\underbrace{\mathcal{A}}_{\text{prop.}}, \underbrace{\mathcal{S}}_{\text{evidences}}, \cdot \xrightarrow{\cdot} \cdot) \qquad \text{where} \qquad a \xrightarrow{e} b \triangleq e \preccurlyeq a \rightarrow b$$

Proof. *Connectives and quantifiers from the internal logic of $\mathcal{A}$ / evidences via the expected λ-terms.*

Remark

- blurs the distinction btw. evidences & propositions
- $\mathsf{UEF}(\mathcal{A})$ is consistent if and only if $\mathcal{A}$ is.
- the implicative tripos $\mathcal{T}^{\mathcal{A}}$ and $\mathsf{UFam}(\mathsf{UEF}(\mathcal{A}))$ are equivalent.

From Implicative Algebras to Evidenced frames

Any implicative algebra $(\mathcal{A}, \preccurlyeq, \rightarrow, \mathcal{S})$ induces an evidenced frame

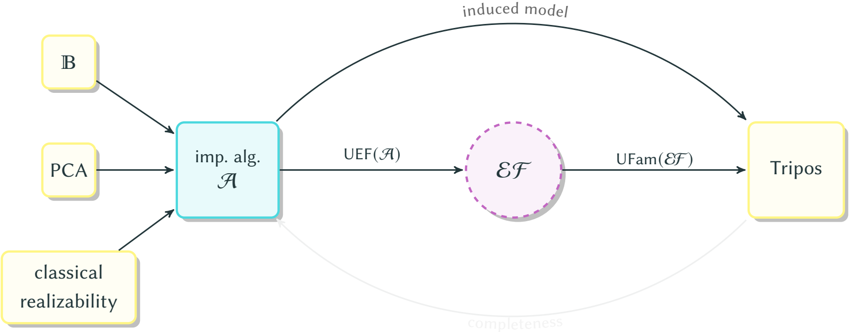
$$\mathrm{UEF}(\mathcal{A}) \triangleq (\underbrace{\mathcal{A}}_{\text{prop.}}, \underbrace{\mathcal{S}}_{\text{evidences}}, \cdot \xrightarrow{\cdot} \cdot) \qquad \text{where} \qquad a \xrightarrow{e} b \triangleq e \preccurlyeq a \rightarrow b$$

Proof. *Connectives and quantifiers from the internal logic of $\mathcal{A}$ / evidences via the expected λ-terms.*

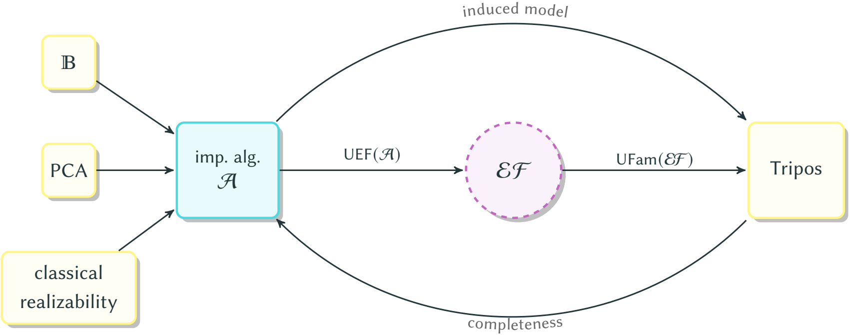
Remark

- blurs the distinction btw. evidences & propositions
- $\mathrm{UEF}(\mathcal{A})$ is consistent if and only if $\mathcal{A}$ is.
- the implicative tripos $\mathcal{T}^{\mathcal{A}}$ and $\mathrm{UFam}(\mathrm{UEF}(\mathcal{A}))$ are equivalent.

From Implicative Algebras to Evidenced frames



From Implicative Algebras to Evidenced frames



The induced implicative algebra

Implicative algebras II: completeness w.r.t. Set-based triposes
A. Miquel [2020]

Remarks

$\Phi + \{\phi \in \Phi : \exists e.\, \top \xrightarrow{e} \phi\}$ fully characterize the *logical facet* of $\mathcal{EF}$ ...

... but $\phi \preccurlyeq \phi' \triangleq \exists e.\, \phi \xrightarrow{e} \phi'$ regretably lacks the structure required by implicative algebras.

Tricks

↬ see the paper or Miquel's completeness result

Theorem

1. $\mathrm{IA}(\mathcal{EF})$ is an implicative algebra.
2. $\mathrm{UEF}(\mathrm{IA}(\mathcal{EF}))$ and $\mathcal{EF}$ are (extensionally) equivalent.

The induced implicative algebra

Implicative algebras II: completeness w.r.t. Set-based triposes
A. Miquel [2020]

Remarks

Φ + $\{\phi \in \Phi : \exists e. \top \xrightarrow{e} \phi\}$ fully characterize the *logical facet* of $\mathcal{EF}$ ...

... but $\phi \preccurlyeq \phi' \triangleq \exists e. \phi \xrightarrow{e} \phi'$ regretably lacks the structure required by implicative algebras.

Tricks

↬ see the paper or Miquel's completeness result

Theorem

1. $\mathrm{IA}(\mathcal{EF})$ is an implicative algebra.
2. $\mathrm{UEF}(\mathrm{IA}(\mathcal{EF}))$ and $\mathcal{EF}$ are (extensionally) equivalent.

The induced implicative algebra

Implicative algebras II: completeness w.r.t. Set-based triposes
A. Miquel [2020]

Remarks

$\Phi + \{\phi \in \Phi : \exists e. \top \xrightarrow{e} \phi\}$ fully characterize the *logical facet* of $\mathcal{EF}$ ...

... but $\phi \preccurlyeq \phi' \triangleq \exists e. \phi \xrightarrow{e} \phi'$ regretably lacks the structure required by implicative algebras.

Tricks

$\looparrowright$ see the paper or Miquel's completeness result

Theorem

1. $\mathrm{IA}(\mathcal{EF})$ is an implicative algebra.
2. $\mathrm{UEF}(\mathrm{IA}(\mathcal{EF})$ and $\mathcal{EF}$ are (extensionally) equivalent.

Completeness wrt. triposes

One trick

$(\!|\cdot|\!)$ in $\mathcal{T}$ reflecting external truth

Reflected axiom schema : Set-relation $R \in \mathcal{P}(\Omega \times \Omega)$ s.t.:

$$\mathcal{T} \models \phi : \Omega, \phi' : \Omega \mid (\!|\phi \; R \; \phi'|\!), \phi \vdash \phi'$$

(i.e. collection of premise-conclusion pairs that R entails)

Completeness

Given a tripos $\mathcal{T}$, the structure $\mathrm{EF}(\mathcal{T}) \triangleq (\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$ where:

- Φ is the set Ω
- E is the set of reflected axiom schemas of $\mathcal{T}$
- $\phi_1 \xrightarrow{e} \phi_2$ is defined as $\langle \phi_1, \phi_2 \rangle \in e$

defines an evidenced frame.

Completeness wrt. triposes

One trick

$(\!|\cdot|\!)$ in $\mathcal{T}$ reflecting external truth

$$(\!|\psi|\!) = (\exists i : \{\langle\rangle \in 1 \mid \psi\}.\, \top)$$

Reflected axiom schema : Set-relation $R \in \mathcal{P}(\Omega \times \Omega)$ s.t.:

$$\mathcal{T} \models \phi : \Omega, \phi' : \Omega \mid (\!|\phi\ R\ \phi'|\!), \phi \vdash \phi'$$

(i.e. collection of premise-conclusion pairs that R entails)

Completeness

Given a tripos $\mathcal{T}$, the structure $\mathrm{EF}(\mathcal{T}) \triangleq (\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$ where:

- Φ is the set Ω
- E is the set of reflected axiom schemas of $\mathcal{T}$
- $\phi_1 \xrightarrow{e} \phi_2$ is defined as $\langle \phi_1, \phi_2 \rangle \in e$

Completeness wrt. triposes

One trick

$(\!|\cdot|\!)$ in $\mathcal{T}$ reflecting external truth

Reflected axiom schema : Set-relation $R \in \mathcal{P}(\Omega \times \Omega)$ s.t.:

$$\mathcal{T} \models \phi : \Omega, \phi' : \Omega \mid (\!|\phi \; R \; \phi'|\!), \phi \vdash \phi'$$

(i.e. collection of premise-conclusion pairs that R entails)

Completeness

Given a tripos $\mathcal{T}$, the structure $\mathrm{EF}(\mathcal{T}) \triangleq (\Phi, E, \cdot \xrightarrow{\cdot} \cdot)$ where:

- Φ is the set Ω
- E is the set of reflected axiom schemas of $\mathcal{T}$
- $\phi_1 \xrightarrow{e} \phi_2$ is defined as $\langle \phi_1, \phi_2 \rangle \in e$

defines an evidenced frame.

Completeness wrt. triposes

One trick

$(\!|\cdot|\!)$ in $\mathcal{T}$ reflecting external truth

Reflected axiom schema : Set-relation $R \in \mathcal{P}(\Omega \times \Omega)$ s.t.:

$$\mathcal{T} \models \phi : \Omega, \phi' : \Omega \mid (\!|\phi \; R \; \phi'|\!), \phi \vdash \phi'$$

(i.e. collection of premise-conclusion pairs that R entails)

Completeness

Given a tripos $\mathcal{T}$, the structure $\mathrm{EF}(\mathcal{T}) \triangleq (\Phi, E, \cdot \overset{\cdot}{\longrightarrow} \cdot)$ where:

- Φ is the set Ω
- E is the set of reflected axiom schemas of $\mathcal{T}$
- $\phi_1 \overset{e}{\longrightarrow} \phi_2$ is defined as $\langle \phi_1, \phi_2 \rangle \in e$

defines an evidenced frame.

Completeness wrt. triposes

One trick

$(\!|\cdot|\!)$ in $\mathcal{T}$ reflecting external truth

Reflected axiom schema : Set-relation $R \in \mathcal{P}(\Omega \times \Omega)$ s.t.:

$$\mathcal{T} \models \phi : \Omega, \phi' : \Omega \mid (\!|\phi \; R \; \phi'|\!), \phi \vdash \phi'$$

Completeness

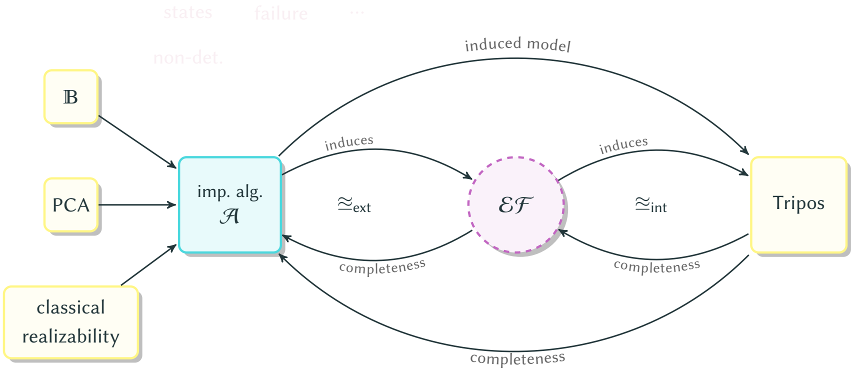
Given a tripos $\mathcal{T}$, the structure $\mathrm{EF}(\mathcal{T}) \triangleq (\Phi, E, \cdot \dot{\rightarrow} \cdot)$ where:

- Φ is the set Ω
- E is the set of reflected axiom schemas of $\mathcal{T}$
- $\phi_1 \xrightarrow{e} \phi_2$ is defined as $\langle \phi_1, \phi_2 \rangle \in e$

defines an evidenced frame.

Bonus: UFam and EF extend to a biadjoint biequivalence bw $\mathbf{EF_{int}}$ **and** $\mathbf{Trip_{int}}$

Final picture

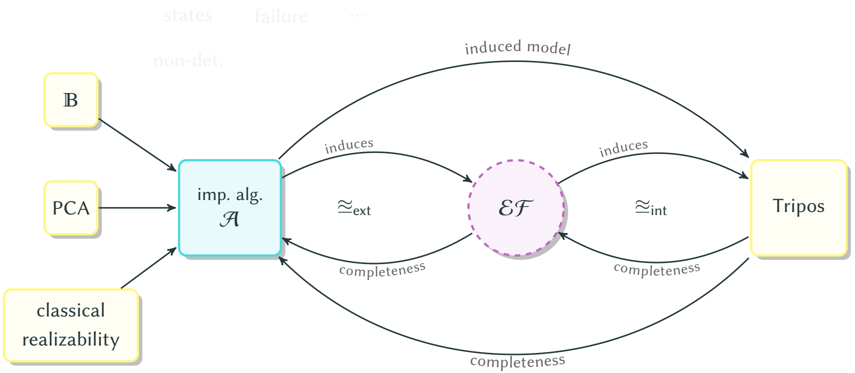


Slogan

Tripos = evidenced frame that has forgotten its evidence.

Coq formalization available : `https://www.i2m.univ-amu.fr/perso/etienne.miquey/`

Final picture

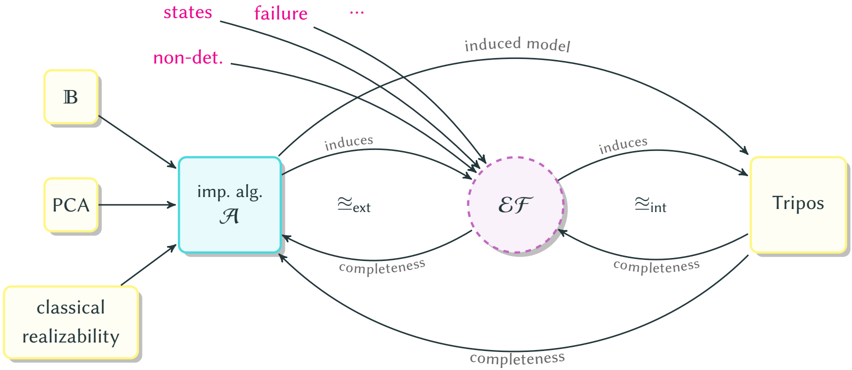


Slogan

Tripos = evidenced frame that has forgotten its evidence.

Coq formalization available : `https://www.i2m.univ-amu.fr/perso/etienne.miquey/`

Final picture



Slogan

Tripos = evidenced frame that has forgotten its evidence.

Coq formalization available : `https://www.i2m.univ-amu.fr/perso/etienne.miquey/`

Bonus

Reasoning on models

Non-determinism & forcing In $\mathcal{EF}_A^{\mathscr{C}_{\mathrm{flip}}}$, $e_{\mathrm{flip}}(e_1, e_2)$:

- conducts a coin flip
- reduces to e_1 or e_2 depending on the result
- angelisms $\sim$ it can explore both options concurrently.

Moral:
$e_{\mathrm{flip}}(e_{\mathrm{fst}}, e_{\mathrm{snd}})$ evidences that $\phi_1 \wedge \phi_2$ entails ϕ_1 **and** $\phi_1 \wedge \phi_2$ entails ϕ_2.

Finitely forced

$$\mathcal{EF}_A^{\mathscr{C}_{\mathrm{flip}}} \vDash \exists e. \forall \phi_1, \phi_2.\ \phi_1 \wedge \phi_2 \xrightarrow{e} \prod_{i \in \{1,2\}} \phi_i.$$

Proposition

finitely forced + E finitely generated = forcing tripos

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

Reasoning on models

Non-determinism & forcing In $\mathcal{EF}_A^{\mathscr{C}_{\text{flip}}}$, $e_{\text{flip}}(e_1, e_2)$:

- conducts a coin flip
- reduces to e_1 or e_2 depending on the result
- angelisms $\sim$ it can explore both options concurrently.

Finitely forced

$$\mathcal{EF}_A^{\mathscr{C}_{\text{flip}}} \vDash \exists e. \forall \phi_1, \phi_2.\ \phi_1 \wedge \phi_2 \xrightarrow{e} \textstyle\prod_{i \in \{1,2\}} \phi_i.$$

Proposition

finitely forced + E finitely generated = forcing tripos

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

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

Reasoning on models

Non-determinism & forcing In $\mathcal{EF}_A^{\mathscr{C}_{\text{flip}}}$, $e_{\text{flip}}(e_1, e_2)$:

- conducts a coin flip
- reduces to e_1 or e_2 depending on the result
- angelisms $\sim$ it can explore both options concurrently.

Finitely forced

$$\mathcal{EF}_A^{\mathscr{C}_{\text{flip}}} \vDash \exists e. \forall \phi_1, \phi_2.\ \phi_1 \wedge \phi_2 \xrightarrow{e} \prod_{i \in \{1,2\}} \phi_i.$$

Proposition

finitely forced + E finitely generated = forcing tripos

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

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

Reasoning on models

Models with failure

In traditional realizability triposes: *(∼ when $\mathcal{S}_\top$ is a valid separator)*

- many predicates for $\top$,
- only one for $\bot$: the predicate with no realizers

No longer the case *when computations can fail*:

$$\mathrm{UFam}(\mathcal{EF}_D^{\mathscr{C}_{\mathsf{fail}},\mathcal{S}_\lambda}) \text{ has many predicates that model } \bot$$

Example

$$(\exists n.\, \mathrm{Nat}(n)) \wedge (\forall n.\, \mathrm{Nat}(n) \supset \bot)$$

Reasoning on models

Models with failure

In traditional realizability triposes: *(∼ when $\mathcal{S}_\top$ is a valid separator)*

- many predicates for $\top$,
- only one for $\bot$: the predicate with no realizers

No longer the case *when computations can fail*:

$$\mathrm{UFam}(\mathcal{EF}_D^{\mathscr{C}_{\mathsf{fail}},\mathcal{S}_\lambda}) \text{ has many predicates that model } \bot$$

Example

$$(\exists n.\, \mathrm{Nat}(n)) \wedge (\forall n.\, \mathrm{Nat}(n) \supset \bot)$$

Let's use this! (1/2)

Krivine is Kleene after a CPS [Oliva-Streicher'08]

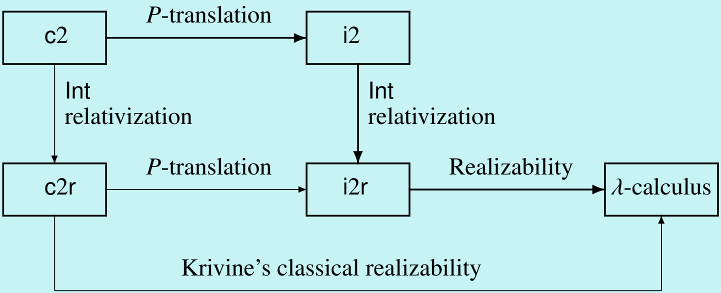
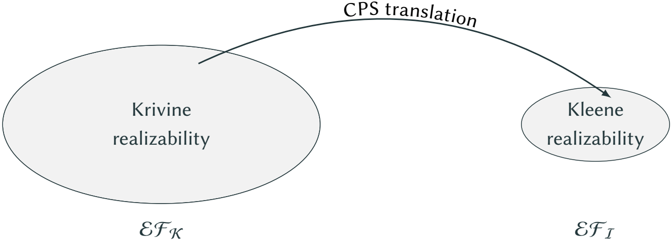


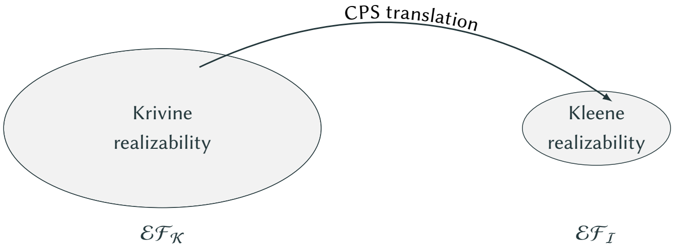
Figure 5: Alternative interpretation of c2

https://www.eecs.qmul.ac.uk/~pbo/papers/paper016.pdf

Let's use this! (1/2)



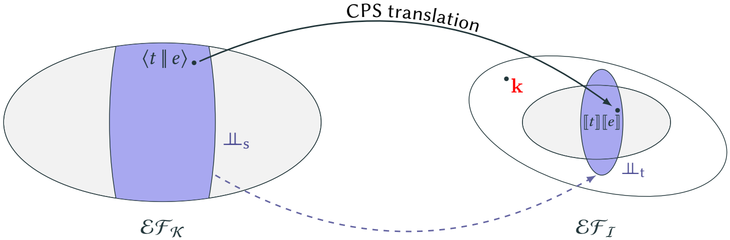
Let's use this! (1/2)



Question - Does the CPS define an EF morphism?

Let's use this! (1/2)

[Gardelle,M.'23]



Question - Does the CPS define an EF morphism?

Bad news

The CPS does not, in general, define an EF morphism.

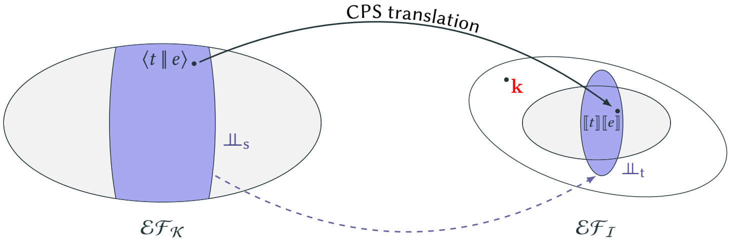
Proof :

We can exhibit $t \Vdash_{\mathcal{K}} \mathbb{B}$ such that $[t] \nVdash_{\mathcal{I}} (\mathbb{B} \to \mathcal{R}) \to \mathcal{R}$.

https://hal.inria.fr/hal-03910311

Let's use this! (1/2)

[Gardelle,M.'23]



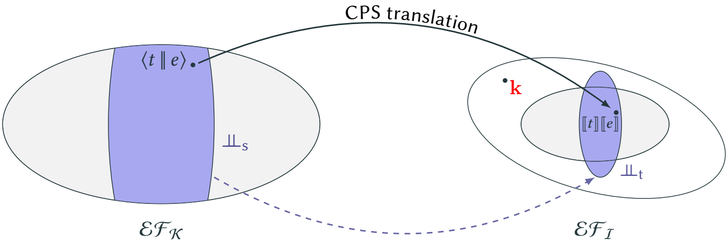
Question - Can we choose $\perp\!\!\!\perp_s$ and $\perp\!\!\!\perp_t = |\mathcal{R}|$ so that it works?

---

https://hal.inria.fr/hal-03910311

Let's use this! (1/2)

[Gardelle,M.'23]



Question - Can we choose $⫫_s$ and $⫫_t = |\mathcal{R}|$ so that it works?

Theorem

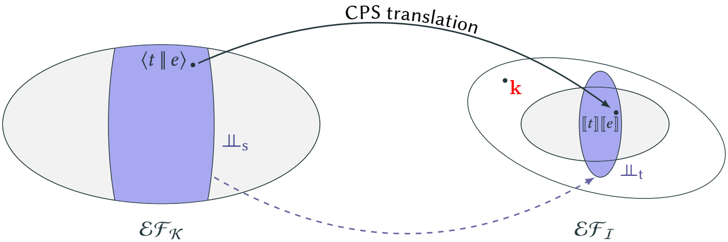
1. **Forward EF** - Given $⫫_s$, we can pick

$$⫫_t \triangleq \{t : \exists c \in ⫫_s . t \rightarrow_\beta [c]\}$$

then $\mathcal{EF}_{fw} = (\Phi_{fw}, E_{fw}, \cdot \dot{\rightarrow}_{fw} \cdot)$ defines an evidenced frame, and $[.]$ is a morphism from $\mathcal{EF}_\mathcal{K}$ to $\mathcal{EF}_{fw}$.

https://hal.inria.fr/hal-03910311

Let's use this! (1/2) [Gardelle,M.'23]



Question - Can we choose $\perp\!\!\!\perp_s$ and $\perp\!\!\!\perp_t = |\mathcal{R}|$ so that it works?

Theorem

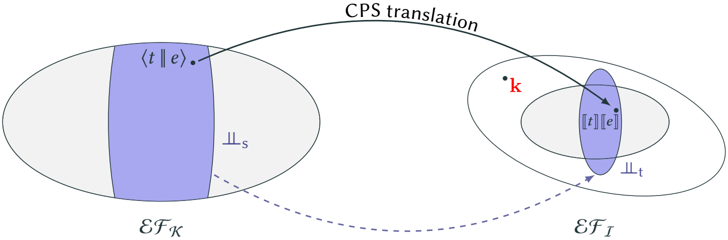
❷ **Backward EF** - Given $\perp\!\!\!\perp_t$, we can pick

$$\perp\!\!\!\perp_s = \{c : [c] \in \perp\!\!\!\perp_t\}$$

then $\mathcal{EF}_{bw} = (\Phi_{bw}, E_{bw}, \cdot \dot{\rightarrow}_{bw} \cdot)$ defines an evidenced frame and [.] is a morphism from $\mathcal{EF}_{\mathcal{K}}$ to $\mathcal{EF}_{fw}$.

https://hal.inria.fr/hal-03910311

Let's use this! (1/2) [Gardelle,M.'23]



Question - Can we choose $\perp\!\!\!\perp_s$ and $\perp\!\!\!\perp_t = |\mathcal{R}|$ so that it works?

Conclusion

Krivine is Kleene after a CPS... ***if restricted to the CPS image!***

Let's use this! (1/2)

[Gardelle,M.'23]

Open question #1

Are they realizers which are *always* compatible with CPS translations?

↬ *universal realizers?*

↬ *what about other syntactic translation/effects?*

Let's use this! (2/2) [Brede-Herbelin'21]

TABLE VII
TREE-BASED DEPENDENT CHOICE AND BAR INDUCTION DUAL PRINCIPLES

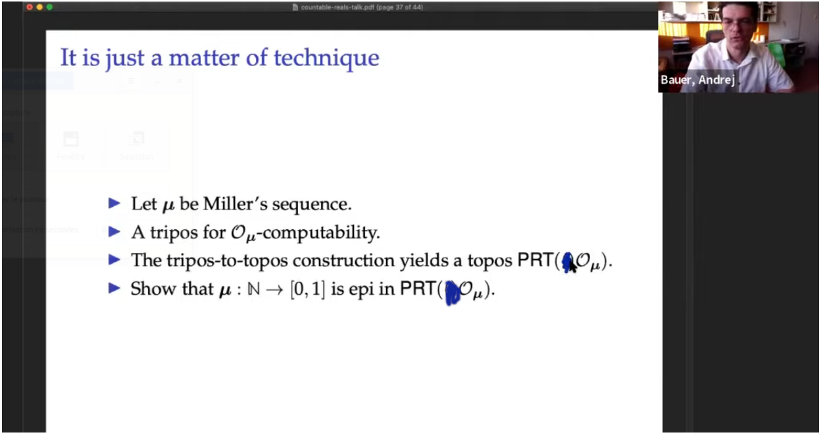
ill-foundedness-style	*well-foundedness-style*
T branching over arbitrary B	
Tree-based Dependent Choice (DC_{BT}^{spread}) T spread $\Rightarrow$ T has an infinite branch	Alternative Bar Induction ($BI_{BT}^{barricaded}$) T barred $\Rightarrow$ T is barricaded
Alternative Tree-based Dependent Choice ($DC_{BT}^{productive}$) T productive $\Rightarrow$ T has an infinite branch	Bar Induction (BI_{BT}^{ind}) T barred $\Rightarrow$ T inductively barred
T branching over non-empty finite B	
$KL_{BT}^{spread} \triangleq DC_{BT}^{spread}$ (finite B)	$FT_{BT}^{barricaded} \triangleq BI_{BT}^{barric.}$ (fin. B)
$KL_{BT}^{productive} \triangleq DC_{BT}^{prod.}$ (fin. B)	$FT_{BT}^{ind} \triangleq BI_{BT}^{ind}$ (finite B)
Alternative Kőnig's Lemma ($KL_{BT}^{unbounded}$) T with unbounded paths $\Rightarrow$ T has an infinite branch	Fan Theorem ($FT_{BT}^{uniform}$) T barred $\Rightarrow$ T uniform bar
Kőnig's Lemma (KL_{BT}^{staged}) T staged-infinite tree $\Rightarrow$ T has an infinite branch	Staged Fan Theorem (FT_{BT}^{staged}) T barred and monotone $\Rightarrow$ T staged barred

Open question #2

How to capture the *exact* computational content of these principles?

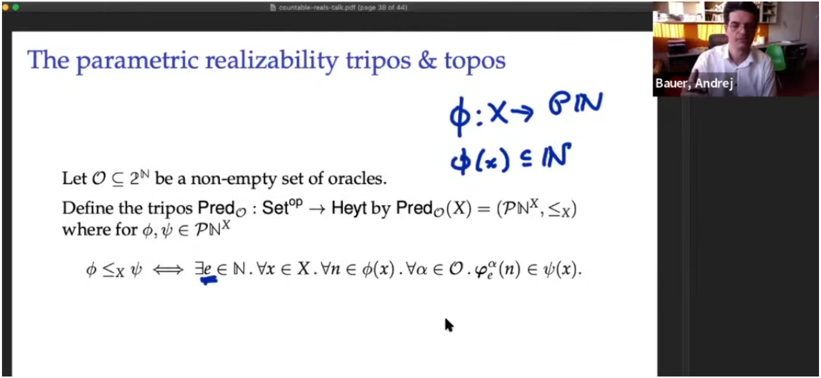
https://hal.inria.fr/hal-03144849v5

The countable reals



Andrej Bauer, *The countable reals* - `youtu.be/4CBFUojXoq4`.

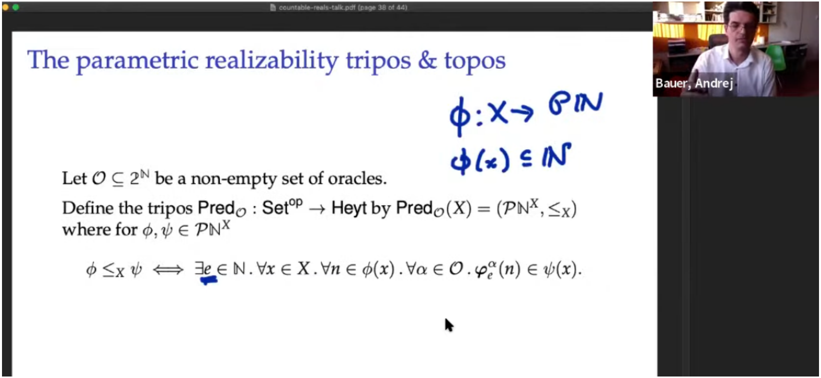
The countable reals



"we spent five days to verify that this is a tripos"

Andrej Bauer, *The countable reals* - `youtu.be/4CBFUojXoq4`.

The countable reals



"we spent five days to verify that this is a tripos"

Andrej Bauer, *The countable reals* - `youtu.be/4CBFUojXoq4`.

The countable reals

Open question #3

What can be said about the induced topos already within the EF?

Andrej Bauer, *The countable reals* - `youtu.be/4CBFUojXoq4`.

Conclusion

Summary

Evidenced Frame

An algebraic structure accounting for realizability models.

What's next?

- Tackle the questions #1, #2, #3, ...
- Study the notion of morphism for evidenced frames
- Consequences of effects on the resulting model
- ...

Summary

Evidenced Frame

A | *flexible* / *uniform* / *complete* | algebraic structure accounting for realizability models.

What's next?

- Tackle the questions #1, #2, #3, …
- Study the notion of morphism for evidenced frames
- Consequences of effects on the resulting model
- …

Summary

Evidenced Frame

A *flexible* / *uniform* / *complete* algebraic structure accounting for realizability models.

Reasoning collectively about models

Any evidenced frame satisfying … models … .

structure / meta-theory / …

What's next?

- Tackle the questions #1, #2, #3, …
- Study the notion of morphism for evidenced frames
- Consequences of effects on the resulting model

Summary

Evidenced Frame

A | *flexible* / *uniform* / *complete* | algebraic structure accounting for realizability models.

Reasoning collectively about models

Any evidenced frame satisfying ... models ... .

structure / meta-theory / ...

What's next?

- Tackle the questions #1, #2, #3, ...
- Study the notion of morphism for evidenced frames
- Consequences of effects on the resulting model

The end

Thank you for your attention!

Bibliography i

L. Cohen, É. Miquey, and R. Tate.
Evidenced frames: A unifying framework broadening realizability models.
In *Proceeding of LICS 2021*.
URL: https://hal.archives-ouvertes.fr/hal-03422961.

L. Cohen, S. A. Faro, and R. Tate.
The effects of effects on constructivism.
Proceedings of MFPS XXXV, 2019.
doi:10.1016/j.entcs.2019.09.006.

A. Miquel.
Implicative algebras: a new foundation for realizability and forcing.
Mathematical Structures in Computer Science, 30(5):458–510, 2020.
doi:10.1017/S0960129520000079.

A. M. Pitts.
Tripos theory in retrospect.
Mathematical Structures in Computer Science, 12(3):265–279, 2002.
doi:10.1017/S096012950200364X.

P. Oliva and T. Streicher.
On Krivine's realizability interpretation of classical second-order arithmetic.
Fundam. Inform., 84(2):207–220, 2008.
URL: https://www.eecs.qmul.ac.uk/~pbo/papers/paper016.pdf.

N. Brede and H. Herbelin.
On the logical structure of choice and bar induction principles.
In *Proceeding of LICS 2021* .
URL: https://hal.inria.fr/hal-03144849v5.

S. Gardelle and É. Miquey.
Do CPS translations also translate realizers?
In *JFLA 2023* .
URL: https://hal.inria.fr/hal-03910311.