# Concrete Models for Classical Realizability

Jaap van Oosten (joint work with Tinxiang Zou)

Department of Mathematics, Utrecht University

Utrecht Workshop on Proof Theory, April 16, 2015

Classical Realizability was developed in the middle of the 1990s by Jean-Louis Krivine.
Its aim is twofold:

- Give new models for classical theories (in particular, set theory)
- Understand classical truth in terms which have computational meaning

In this talk, we concentrate on the first aspect.

Outline of the talk:

1) Description of Krivine's classical realizability
2) Krivine realizability as a tripos/topos construction
3) A connection with relative realizability

Sources:

1) Papers by Jean-Louis Krivine, see Krivine's home page: `http://www.pps.jussieu.fr/∼krivine/`
2) Paper *Krivine's Classical Realizability from a Categorical Perspective* by Thomas Streicher (to appear in MSCS); available at Streicher's home page: `http://www.mathematik.tu-darmstadt.de/∼streicher/`
3) Paper *All realizability is relative* by Pieter Hofstra (Math. Proc. Camb. Phil. Soc. **141** (2006), 239–264
4) Some ideas of Wouter Stekelenburg
5) Tingxiang Zou's MSc Thesis (in preparation)

There are two kinds of objects: *terms* (denoted $t, t', s, u, \ldots$) and *stacks* (denoted $\pi, \pi'$).
We may have *stack constants* (basic stacks) from a set $\Pi_0$; we think of a stack as a sequence of closed terms ended by a stack constant. Given a closed term $t$ and a stack $\pi$, we have a new stack $t.\pi$.
The terms come from a $\lambda$-calculus enriched with extra constants. In this talk, we shall only consider the following extra constants:

> For every stack $\pi$ there is a constant $k_\pi$ (sometimes called *continuation constants*)
>
> There is a constant $\text{æ}$ (*call/cc*)

If we denote the set of stacks by $\Pi$ and the set of terms by $\Lambda$, we have therefore the following formal syntax:

$$\Pi ::= \alpha | t.\pi \ (\alpha \in \Pi_0,\ t \in \Lambda,\ t \text{ closed})$$

$$\Lambda ::= x | \lambda x.t | tu | \text{æ} | k_\pi \ (\pi \in \Pi)$$

An element of $\Lambda \times \Pi$ (typically written as $t * \pi$) is called a *process*. There is a *reduction relation* on processes, generated by the following one-step reductions:

$$\begin{array}{ll} \text{Push} & tu * \pi \succ t * u.\pi \\ \text{Grab} & \lambda x.t * u.\pi \succ t[u/x] * \pi \\ \text{Save} & \text{cc} * u.\pi \succ u * k_\pi.\pi \\ \text{Restore} & k_\pi * u.\pi' \succ u * \pi \end{array}$$

This is called *Krivine's Abstract Machine*. Note that the first two rules implement *weak head reduction*:

$$(\lambda x_1 \cdots x_n.t)M_1 \cdots M_n * \pi \succ\succ t[M_1/x_1, \ldots, M_n/x_n] * \pi$$

A set of $\mathcal{U}$ processes is *saturated* if $t * \pi \in \mathcal{U}$ whenever $t * \pi \succ t' * \pi'$ and $t' * \pi' \in \mathcal{U}$.
We fix a saturated set of processes: a *pole* $\perp\!\!\!\perp$.
We also fix a set of terms: the set PL of *proof-like* terms. Krivine stipulates: PL is the set of closed terms which don't contain a continuation constant $k_\pi$ (this may be too strict).

## Logic

Consider a language in second-order logic: we have certain first-order constants, function symbols and relation symbols; first-order varables $x, y, \ldots$, second-order variables $X, Y, \ldots$ (of each arity $\geq 0$), and the logical symbols $\rightarrow, \forall x, \forall X$.

We have the usual definitions:

$$\begin{array}{rcl}
\bot & \equiv & \forall X.X \\
\neg A & \equiv & A \rightarrow \bot \\
A \wedge B & \equiv & \forall X.(A \rightarrow (B \rightarrow X)) \rightarrow X \\
A \vee B & \equiv & \forall X.(A \rightarrow X) \rightarrow ((B \rightarrow X) \rightarrow X) \\
\exists x A & \equiv & \forall X.(\forall x(A \rightarrow X) \rightarrow X) \\
\text{etc.} & &
\end{array}$$

## Curry Howard for Classical second-order logic

Define a derivation system of typing judgements $\Gamma \vdash t : A$ where $\Gamma$ is a variable declaration $x_1 : A_1, \ldots, x_n : A_n$, the $A_i$ are second-order formulas and $t$ is a term:

$$\frac{}{\Gamma \vdash x : A}\ (x : A) \in \Gamma$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A[e/x]}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X.A}\ X \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[P/X]}$$

And one classical rule (Peirce's Law):

$$\frac{}{\Gamma \vdash \text{œ} : ((A \to B) \to A) \to A}$$

Examples of derivable judgements:

$$\begin{array}{rcllcl} \mathbf{pair} & \equiv & \lambda xyz.zxy & : & \forall XY.X \to (Y \to X \wedge Y) \\ \mathbf{fst} & \equiv & \lambda z.z(\lambda xy.x) & : & \forall XY.X \wedge Y \to X \\ \mathbf{left} & \equiv & \lambda xuv.ux & : & \forall XY.X \to X \vee Y \\ \mathbf{right} & \equiv & \lambda yuv.vy & : & \forall XY.Y \to X \vee Y \end{array}$$

and also

$$\mathbf{EM} \equiv \text{œ}(\lambda k.\mathbf{right}(\lambda x.k(\mathbf{left}x))) : \forall X.X \vee \neg X$$

We should have: whenever $\Gamma \vdash t : A$ is derivable, $t$ is a proof-like term.

Realizability

Suppose we are given a set $U$ of individuals. Relative to an assignment of variables, where elements of $U$ are assigned to first-order variables and functions $U^k \to \mathcal{P}(\Pi)$ are assigned to $k$-ary predicate variables, we now assign to any formula $A$ a set of stacks $\|A\|$, a set of "witnesses against $A$". The set of *realizers of* $A$, written $|A|$, is defined as

$$|A| \;=\; \{t \in \Lambda \,|\, \forall \pi \in \|A\| \; t * \pi \in \perp\!\!\!\perp\}$$

The definition is simple:

$$\begin{array}{rcl} \|A \to B\| & = & |A|.\|B\| \;=\; \{t.\pi \,|\, t \in |A|, \pi \in \|B\|\} \\ \|\forall x A\| & = & \bigcup_{u \in U} \|A(u)\| \\ \|\forall X.A\| & = & \bigcup_{F:U^k \to \mathcal{P}(\Pi)} \|A(F)\| \end{array}$$

Then $|\forall x A| = \bigcap_{u \in U} |A(u)|$, etc.

A complication: if the pole $\perp\!\!\!\perp$ is empty, we always have: $|A| = \emptyset$ or $|A| = \Lambda$. We have classical, two-valued semantics.
On the other hand, if the pole contains one process, say $t * \pi$, then by the rule (Restore) we have $k_\pi * t.\pi' \in\perp\!\!\!\perp$ for any $\pi'$; whence by (Push), $k_\pi t * \pi' \in\perp\!\!\!\perp$ for any $\pi'$; which means that $k_\pi t \in |A|$ for any $A$, in particular for $A \equiv \forall X.X$.
Therefore we say: a closed formula $A$ is *true* under this realizability, if its set $|A|$ of realizers contains an element of PL, the set of proof-like terms.
**Strong Soundness Theorem** Suppose the typing judgement $x_1 : A_1, \ldots, x_n : A_n \vdash t : B$ is derivable; suppose that relative to an assignment $\rho$ we have $u_1 \in |A_1[\rho]|, \ldots, u_n \in |A_n[\rho]|$. Then

$$t[u_1/x_1, \ldots, u_n/x_n] \in |B[\rho]|$$

Note that the hypothesis implies that $t$ is proof-like; so if $u_1, \ldots, u_n$ are proof-like, so is $t(u_1, \ldots, u_n)$.

Examples

1. For any $A, B$ and term $t$:

$$t \in |A \to B| \;\Rightarrow\; \forall u(u \in |A| \Rightarrow tu \in |B|)$$

For, suppose $\pi \in \|B\|$, $u \in |A|$. Then $u.\pi \in \|A \to B\|$ so $t * u.\pi \in \perp\!\!\!\perp$; by (Push), $tu * \pi \in \perp\!\!\!\perp$.

2. For any $A$ and $B$: if $\pi \in \|A\|$ then $k_\pi \in |A \to B|$. For, suppose $\pi \in \|A\|$, $u.\rho \in \|A \to B\|$ so $u \in |A|$, $\rho \in \|B\|$. Then $u * \pi \in \perp\!\!\!\perp$ whence by (Restore), $k_\pi * u.\rho \in \perp\!\!\!\perp$.

3. Let us see that $\text{œ}$ realizes Peirce's Law: suppose $t.\pi \in \|((A \to B) \to A) \to A\|$, so $t \in |(A \to B) \to A|$, $\pi \in \|A\|$. Then $k_\pi \in |A \to B|$, so $k_\pi.\pi \in \|(A \to B) \to A\|$. Hence $t * k_\pi.\pi \in \perp\!\!\!\perp$. By (Save), $\text{œ} * t.\pi \in \perp\!\!\!\perp$. we conclude that $\text{œ} \in |((A \to B) \to A) \to A|$

So far the treatment of Krivine/Miquel. Can we understand this interpretation in terms of categorical logic?

Definition. A tripos on $\mathrm{Set}$ is a pseudofunctor $P : \mathrm{Set}^{\mathrm{op}} \to \mathrm{Preord}$, satisfying:

a) For each set $X$ the preorder $PX$ is endowed with a binary operation $(\cdot) \to (\cdot)$ which obeys the laws of intuitionistic implicational logic (e.g., $\phi \leq \psi \to \phi$, $\theta \to (\phi \to \psi) \leq (\theta \to \phi) \to (\theta \to \psi)$);

b) For every function $f : X \to Y$ of sets, the map $Pf : PY \to PX$ preserves $\to$ up to isomorphism. Moreover, $Pf$ has a right adjoint $\forall f$, which satisfies the Beck condition and the condition that for $\phi \in PX$, $\psi \in PY$,

$$\forall f(Pf(\psi) \to \phi) \simeq \psi \to \forall f(\phi)$$

c) There is a *generic predicate*: a set $\Sigma$ and an element $\sigma \in P\Sigma$ with the property that for every $\phi \in PX$ there is a function $\{\phi\} : X \to \Sigma$ such that $P\{\phi\}(\sigma) \simeq \phi$.

Every tripos on $\mathrm{Set}$ gives rise to a model of second-order logic. Formulas with parameters from a set $X$ are interpreted as elements of $PX$

Second-order (unary) predicates are interpreted as elements of $\Sigma^X$ (where $\Sigma$ is the carrier of a chosen generic predicate)

The element relation must be an element of $P(\Sigma^X \times X)$: it can be taken as $P(\mathrm{ev})(\sigma)$ where $\mathrm{ev} : \Sigma^X \times X \to \Sigma$ is the evaluation map.

A closed formula is interpreted as an alement of $P1$ (1 a fixed one-element set); it is *true* if its interpretation is the top element in this preorder.

Krivine's realizability defines a Boolean tripos $\mathcal{K}$ on Set: for a set $X$, let $\mathcal{K}X$ be the set of functions $X \to \mathcal{P}(\Pi)$. Given such a function $\phi$, we define $|\phi(x)|$ by

$$|\phi(x)| \ = \ \{t \in \Lambda \,|\, \forall \pi \in \phi(x)\, t * \pi \in \perp\!\!\!\perp\}$$

Define $\to$ on $\mathcal{K}X$ by

$$(\phi \to \psi)(x) \ = \ \{t.\pi \,|\, t \in |\phi(x)|, \ \pi \in \psi(x)\}$$

The order is given by: $\phi \leq \psi$ if and only if $\bigcap_x |(\phi \to \psi)(x)|$ contains a proof-like term.
For $f : X \to Y$, $\mathcal{K}f : \mathcal{K}Y \to \mathcal{K}X$ is given by composition with $f$.
So $\mathcal{K}f$ preserves $\to$ and is order-preserving. Its right adjoint $\forall f$ is given by

$$\forall f(\phi)(y) \ = \ \|\forall x(f(x) \asymp y \to \phi(x)\|$$

Here $\|\forall x(f(x) \asymp y \to \phi(x)\| =$
$\bigcup_{x \in X}\{t.\pi \,|\, t \in |f(x) \asymp y|, \ \pi \in \phi(x)\}$

Thomas Streicher has given a reformulation of Krivine's realizability in terms reminiscent of combinatory logic.
An *abstract Krivine structure* consists of:

a set $\Lambda$ of "terms", with elements $K$, $S$ and $\text{\ae}$

an application operation $t, s \mapsto ts : \Lambda \times \Lambda \to \Lambda$

a subset QP of $\Lambda$: the 'quasi-proofs'; QP is closed under application, and contains the elements $K$, $S$ and $\text{\ae}$

a set $\Pi$ of "stacks"

an operation $t, \pi \mapsto t.\pi : \Lambda \times \Pi \to \Pi$

an operation $k_{(-)} : \Pi \to \Lambda$

and a 'pole', a saturated subset $\perp\!\!\!\perp$ of $\Lambda \times \Pi$

As usual, we write elements of $\Lambda \times \Pi$ as $t * \pi$

The saturatedness of $\perp\!\!\!\perp$ means that the following axioms are satisfied:

$$\begin{array}{ll}
\text{(S1)} & \text{if } t * s.\pi \in \perp\!\!\!\perp \text{ then } ts * \pi \in \perp\!\!\!\perp \\
\text{(S2)} & \text{if } t * \pi \in \perp\!\!\!\perp \text{ then } K * t.s.\pi \in \perp\!\!\!\perp \\
\text{(S3)} & \text{if } tu(su) * \pi \in \perp\!\!\!\perp \text{ then } S * t.s.u.\pi \in \perp\!\!\!\perp \\
\text{(S4)} & \text{if } t * k_\pi.\pi \in \perp\!\!\!\perp \text{ then } \text{œ} * t.\pi \in \perp\!\!\!\perp \\
\text{(S5)} & \text{if } t * \pi \in \perp\!\!\!\perp \text{ then } k_\pi * t.\pi' \in \perp\!\!\!\perp
\end{array}$$

Again, we have a tripos: $PX = \mathcal{P}(\Pi)^X$
$\phi \leq \psi$ if and only if $\bigcap_{x \in X} |\phi(x) \to \psi(x)|$ contains a proof-like element, where:
$|\chi(x)| \; = \; \{t \in \Lambda \,|\, \forall \pi \in \chi(x)\, t * \pi \in \perp\!\!\!\perp\}$
$\phi(x) \to \psi(x) \; = \; \{t.\pi \,|\, t \in |\phi(x)|, \pi \in \psi(x)\}$

Streicher's formulation facilitates drawing a parallel with 'relative realizability'.

An *order-pca* (opca) is a poset $A$ with a partial application $a, b \mapsto ab$ on $A$ which satisfies:

> if $ab$ is defined, $a' \leq a$ and $b' \leq b$ then $a'b'$ is defined and $a'b' \leq ab$
>
> there are elements k and s in $A$ such that $\mathsf{k}ab \leq a$, $\mathsf{s}ab$ is defined, and whenever $ac(bc)$ is defined then so is $\mathsf{s}abc$, and $\mathsf{s}abc \leq ac(bc)$

A *filter* $\Phi$ on an opca $A$ is a subset which contains some choice for k and s, and is closed under application.

Relative realizability triposes:
Given an opca $A$ and a filter $\Phi$ we have a tripos $P_{A,\Phi}$. Let $\mathcal{D}(A)$ be the set of all downwards closed subsets of $A$.
Let $P_{A,\Phi}(X)$ the set of all functions $X \rightarrow \mathcal{D}(A)$.
Define $\phi \leq \psi$ iff for some element $c$ of the filter $\Phi$ we have: for all $x \in X$ and $a \in \phi(x)$, $ca \in \psi(x)$

Prime example of an opca with filter: let $A$ the set of all functions $\mathbb{N} \to \mathbb{N}$, and $\Phi$ the set of all total recursive functions.
The application on $A$ is as follows: for $\alpha, \beta, \gamma \in \mathbb{N}^{\mathbb{N}}$, $\alpha\beta = \gamma$ if for every $n \in \mathbb{N}$ there is a $k \in \mathbb{N}$ such that

$$\alpha(\langle n, \beta(0), \ldots, \beta(k-1)\rangle) = \gamma(n) + 1$$

$$\alpha(\langle n, \beta(0), \ldots, \beta(l-1)\rangle) = 0 \text{ for } l < k$$

Given an opca $A$ with filter $\Phi$, fix a subset $\mathcal{U}$ of $A$ which is disjoint from $\Phi$.
Consider a standard coding of finite sequences in $A$. We define an abstract Krivine structure as follows:

Let $\Pi$ be the set of coded sequences of $A$

Put $\Lambda = A$

Let $a.\pi$ be the code of the sequence $\pi$ with $a$ appended at the front.

Define a pole $\perp\!\!\!\perp$ by:

$$\perp\!\!\!\perp \;=\; \{t * \pi \mid t\pi \text{ is defined and an element of } \mathcal{U}\}$$

define a new, total, application on $A$ by:

$$a{\cdot}b \;=\; \lambda\pi.a(b.\pi)$$

Our set QP of quasi-proofs is $\Phi$.
For the rest of the structure, let $\pi_{\geq k}$ be a code of the sequence $\pi_k, \pi_{k+1}, \ldots$, if $\pi$ is code of the sequence $\pi_0, \pi_1, \ldots$. Then define:

$$\begin{array}{rcl} K & = & \lambda\pi.\pi_0(\pi_{\geq 2}) \\ S & = & \lambda\rho.\rho_0(\rho_2.[\lambda\nu.\rho_1(\rho_2.\nu)].\rho_{\geq 3}) \\ k_\pi & = & \lambda\rho.\rho_0\pi \\ \text{\oe} & = & \lambda\rho.\rho_0(k_{\rho_{\geq 1}}.\rho_{\geq 1}) \end{array}$$

We have: $(\mathrm{S1})$ if $t * s.\pi \in \perp\!\!\!\perp$, then $t(s.\pi) \in \mathcal{U}$, so $(t{\cdot}s)\pi \in \mathcal{U}$, therefore $t{\cdot}s * \pi \in \perp\!\!\!\perp$, etc.

The tripos obtained from this abstract Krivine structure can equivalently be described as follows:
define a new preorder on the sets $P_{A,\Phi}(X)$, by putting:
$\phi \leq \psi$ iff the set $\bigcap_x \phi(x) \to [(\psi(x) \to \mathcal{U}) \to \mathcal{U}]$ contains an element of $A^\sharp$.
The topos one constructs from this tripos is the Booleanization of a closed subtopos of $\mathrm{Set}[P_{A^\sharp,A}]$

Thomas Streicher shows that every abstract Krivine structure gives rise to an opca with a filter, but he does not compare the Krivine tripos with the standard relative realizability tripos $P_{A,\Phi}$.

**Theorem** (Zou) Every abstract Krivine structure is equivalent to one formed from an opca $A$, a filter $\Phi$ and a subset $\mathcal{U} \subset A - \Phi$.

**Theorem** (Zou) For $A = \mathbb{N}^{\mathbb{N}}$, $\Phi$ the set of total recursive functions, and $\mathcal{U} = \{\tau\}$ for some non-recursive $\tau$, the tripos obtained from the abstract Krivine structure as above, is not equivalent to a tripos of the form $[-, \mathcal{B}]$ for some complete Boolean algebra $\mathcal{B}$.