About realizability models

(at last a program for AC)

Jean-Louis Krivine

IRIF - University Paris-Diderot

krivine@irif.fr

Marseillle, september 2019

Introduction

Classical realisability (c.r.) is an extension of forcing which gives models of ZF_ε
(a conservative extension of ZF, ε is a non extensional symbol).

These new *realizability models (r.m.)* are ***much*** more complicated.

It is a bit like passing from commutative to non commutative groups.

By the way, A. Miquel calls c.r. : "non commutative forcing".

To understand the scale of the problem, compare the stucture of realizability algebra with that of set of forcing conditions.

I will speak about available tools in order to study these models, in the general case and in some particular ones.

They are much fewer and less powerful than for forcing.

But the theory is very young and more difficult.

Another reason to work hard is the connection with computer science.

Realizability algebras (r.a.)

It is a 3-sorted first order structure, which consists of :

- *Three sets* : Λ the set of *terms* (programs), Π the set of *stacks* (environments),
  $\Lambda \star \Pi$ the set of *processes* (executable).
- *Six distinguished terms* : B, C, I, K, W, cc (*elementary combinators*).
- *Four operations* :

  Application : $\Lambda \times \Lambda \to \Lambda$ denoted $(\xi)\eta$, (or often $\xi\eta$) where ξ, η are terms ;

  Push : $\Lambda \times \Pi \to \Pi$ denoted $\xi \,.\, \pi$, where π is a stack ;

  Continuation : $\Pi \to \Lambda$ denoted k_π ;

  Process : $\Lambda \times \Pi \to \Lambda \star \Pi$ denoted $\xi \star \pi$.
- *A preorder on processes*, denoted $\succ$ (*execution*)
- *A distinguished subset* $\perp\!\!\!\perp$ of $\Lambda \star \Pi$ such that : $p \notin \perp\!\!\!\perp, p \succ p' \Rightarrow p' \notin \perp\!\!\!\perp$.
- *A distinguished subset* PL of Λ (*proof-like terms*) such that :

  B, C, I, K, W, cc $\in$ PL ; $\xi, \eta \in \mathrm{PL} \Rightarrow \xi\eta \in \mathrm{PL}$; $(\forall \xi \in \mathrm{PL})(\exists \pi \in \Pi)(\xi \star \pi \notin \perp\!\!\!\perp)$.

Axioms of r.a.

The preorder $\succ$ represents *execution in a weak head reduction machine* :

$\xi\eta \star \pi \succ \xi \star \eta . \pi$	(push)
$\mathsf{I} \star \xi . \pi \succ \xi \star \pi$	(no operation)
$\mathsf{K} \star \xi . \eta . \pi \succ \xi \star \pi$	(delete)
$\mathsf{W} \star \xi . \eta . \pi \succ \xi \star \eta . \eta . \pi$	(duplicate)
$\mathsf{C} \star \xi . \eta . \zeta . \pi \succ \xi \star \zeta . \eta . \pi$	(swap)
$\mathsf{B} \star \xi . \eta . \zeta . \pi \succ \xi \star \eta\zeta . \pi$	(apply)
$\mathsf{cc} \star \xi . \pi \succ \xi \star \mathsf{k}_\pi . \pi$	(save the stack)
$\mathsf{k}_\pi \star \xi . \varpi \succ \xi \star \pi$	(restore the stack).

A Curry-style translation of λ-calculus

It is possible to translate λ-terms to accommodate weak head reduction.

$\lambda x_1 \ldots \lambda x_n\, t \star \xi_1 . \ldots . \xi_n . \pi \succ t[\xi_1/x_1, \ldots, \xi_n/x_n] \star \pi$.

I do not give here the precise translation.

Remark. The usual KS -translation does not work. For instance :

$\lambda x(x)xx \star \xi . \pi \equiv ((\mathsf{S})(\mathsf{S})\mathsf{I}\,\mathsf{I})\,\mathsf{I} \star \xi . \pi \succ \xi \star \mathsf{I}\xi . \mathsf{I}\xi . \pi$ instead of $(\xi)\xi\xi \star \pi$.

We use λ-calculus only as a convenient way of writing combinatory terms,

because it is much more intuitive for programming than combinatory logic

so we almost always use λ-terms.

But combinatory logic is much better for theory

because it is a first order structure, λ-calculus is not.

The theory ZF_ε

It is ZF set theory with a non extensional *well founded* symbol ε (strong membership). The usual membership $\in$ is obtained by collapsing ε.

Remark. This theory appears already naturally in forcing :

$a\,\varepsilon\,b$ is $(\exists p \in G)((a,p) \in b)$.

Each formula $F(\vec{a})$ of ZF_ε with $\vec{a} \in \mathcal{M}$, takes a "falsity value" $\|F(\vec{a})\| \in 2^\Pi$ and a "truth value" $|F(\vec{a})| \in 2^\Lambda$. They are linked by :

$$|F(\vec{a})| = \{t \in \Lambda\,;\ (\forall \pi \in \|F(\vec{a})\|)(t \star \pi \in \perp\!\!\!\perp)\}$$

Realizability models (r.m.)

They are built like forcing models, but with a r.a. in place of a set of conditions.

The ground model $(\mathcal{M}, \in)$ satisfies ZFC or even ZFL.

The realizability model $(\mathcal{N}, \varepsilon)$ satisfies ZF$_\varepsilon$.

$\mathcal{N} \supset \mathcal{M}$ ***strictly*** except in the case of forcing : there are not named objects.

We define in $\mathcal{N}$ a Boolean algebra $\gimel 2$ *which is trivial in the case of forcing*.

$\mathcal{N}$ has a structure of boolean model on $\gimel 2$, here denoted as $\mathcal{M}_{\gimel 2}$.

It is an elementary extension of the ground model $(\mathcal{M}, \in)$.

The boolean value in $\mathcal{M}_{\gimel 2}$ of a formula Φ of ZF is denoted by $\langle \Phi \rangle$.

Thus $\mathcal{M}_{\gimel 2}$ satisfies ZFC or even ZFL for the relations $\langle x \in y \rangle = 1, x = y$

and *all the functionals on $\mathcal{M}$*.

Realizability models

There are two essential equivalence relations on the model $(\mathcal{N}, \varepsilon)$:

$\simeq$ (extensional equivalence)

$=_{\mathcal{D}}$ (equivalence for the canonical ultrafilter $\mathcal{D}$ on $\daleth 2$).

The quotient models are respectively : $\mathcal{N}_{\in} \models$ ZF and $\mathcal{M}_{\mathcal{D}} \succ \mathcal{M}$.

Remark. In the forcing case, $\mathcal{N}_{\in}$ is the forcing model and $\mathcal{M}_{\mathcal{D}} = \mathcal{M}_{\daleth 2} = \mathcal{M}$.

In c.r. we consider primarily the model $(\mathcal{N}, \varepsilon)$. The essential tool is :

Every functional defined in $\mathcal{M}$ can be extended to this model.

For instance, a functional $f : \mathcal{M} \times X \times Y \to \mathcal{M} \times Z$ is extended

into a functional $f : \mathcal{N} \times \daleth X \times \daleth Y \to \mathcal{N} \times \daleth Z$ with $\daleth X = X \times \Pi$.

It is in this way that we define the Boolean algebra $\daleth 2$;

and also the value $\langle \Phi \rangle \, \varepsilon \daleth 2$ of a formula Φ of ZF in the boolean model $\mathcal{M}_{\daleth 2}$.

For every $X \in \mathcal{M}$ we define the quantifier $\forall x^{\daleth X}$ by $\| \forall x^{\daleth X} F(x) \| = \bigcup_{a \in X} \| F(a) \|$.

Realizability models

The model $\mathcal{N}$ is an *algebra* over $\beth 2$: we define the product $\beth 2 \times \mathcal{N} \to \mathcal{N}$ as an extension of the trivial functional $2 \times \mathcal{M} \to \mathcal{M}$:

$$(0, x) \mapsto 0 \; ; \; (1, x) \mapsto x.$$

Each ultrafilter U on $\beth 2$, in particular each atom, gives a model $\mathcal{M}_U \succ \mathcal{M}$ which is well founded iff $U = \mathcal{D}$ (by definition of $\mathcal{D}$).

If a is an atom, we have $\mathcal{M}_a = a\mathcal{N}$ *which is a class*.
For instance, if $\beth 2$ is finite, with the atoms a_0 (canonical), $a_1, \ldots, a_{n-1}$, we have : $\mathcal{N} = \mathcal{M}_{a_0} \times \mathcal{M}_{a_1} \times \cdots \times \mathcal{M}_{a_{n-1}}$ and $\mathcal{M}_{a_0} = \mathcal{M}_{\mathcal{D}}$ is the only well founded one.
More generally, $a\mathcal{N} \succ \mathcal{M}$ for all $a \,\varepsilon\, \beth 2$; $a\mathcal{N}$ is a $a\beth 2$-boolean model.

If $ab = 0$, the classes $a\mathcal{N}$ and $b\mathcal{N}$ are somewhat "incompatible" [Kr2] : for every functional $F : a\mathcal{N} \to b\mathcal{N}$, there is a surjection from $\beth \Lambda$ onto $\mathrm{Im}(F)$.

Realizability models

The functional $\daleth$ is very interesting : for each set X in the ground model $\mathcal{M}$,
$\daleth X = X \times \Pi$ defines *the type associated with X*.

For instance $\daleth 2$ is the type of booleans and $\daleth\mathbb{N}$ the type of integers.

If we identify the r.m. $\mathcal{N}$ with the boolean model $\mathcal{M}_{\daleth 2} \models$ ZFC
the meaning of $\daleth X$ becomes clear : $a \,\varepsilon\, \daleth X$ means $\langle a \in X \rangle = 1$
(a is always in X). Indeed, we have trivially $\|a \not\varepsilon \daleth X\| = \|\langle a \in X \rangle \neq 1\|$.

For instance $v \,\varepsilon\, \daleth\mathbb{N}$ means $\langle v \in \mathbb{N} \rangle = 1$, i.e. "$v$ is always an integer"
(not always the same, not even always standard in $\mathcal{N}$).

But $v \in \mathbb{N}$ means : "v is always *the same* integer" (necessarily standard in $\mathcal{N}$).

This is clearer in the particular case where $\daleth 2$ is finite, with $\mathcal{N} = \mathcal{M}_{a_0} \times \cdots \times \mathcal{M}_{a_{n-1}}$.

$\mathcal{M}_{a_0} = \mathcal{M}_{\mathcal{D}}$ is the only one which is well founded in $\mathcal{N}$.

We have $v = (v_0, \ldots, v_{n-1})$; $v \,\varepsilon\, \daleth\mathbb{N}$ means : v_i is an integer of $\mathcal{M}_i$ for each i.

The generic

In the case of forcing, the generic is $G = \{(p,q)\,;\, q \le p\}$. In the general case we define the generic $G = \{(t, t \bullet \pi)\,;\, t \in \Lambda, \pi \in \Pi\}$; we have $\Vdash G \subseteq \gimel\Lambda$. And also : $\|(\forall t\,\varepsilon\, G)F(t)\| = \|\forall t^{\gimel\Lambda}(\{t\} \to F(t))\| = \{t \bullet \pi\,;\, t \in \Lambda, \pi \in \|F(t)\|\}$.

Truth lemma. For every formula $F(\vec{a})$ with parameters in $\mathcal{N}$, we have :
$\mathcal{N} \models F(\vec{a}) \Leftrightarrow (\exists t\,\varepsilon\, G)\Big(\mathcal{M}_{\gimel 2} \models \langle t \Vdash F(\vec{a})\rangle = 1\Big)$.
Proof : $\|\neg F(\vec{a})\| = \{t \bullet \pi\,;\, t \Vdash F(\vec{a}), \pi \in \Pi\} = \|\forall t(\langle t \Vdash F(\vec{a})\rangle = 1 \hookrightarrow t \not\varepsilon\, G)\|$. QED

In particular, we have : $a\,\varepsilon\, b \Leftrightarrow (\exists t\,\varepsilon\, G)(\langle (a,t) \in b\rangle = 1)$.

In the particular case of forcing, the truth in $(\mathcal{N}, \varepsilon)$ is determined by means of *G and the truth in $(\mathcal{M}, \in)$*. This is no longer sufficient in the general case (the model of threads is an extreme case where G is already in $\mathcal{M}$!)
We must consider *G and the truth in the boolean model $(\mathcal{M}_{\gimel 2}, \in) \succ (\mathcal{M}, \in)$*.

The generic

The situation is therefore ***much more complicated*** than in forcing.

We have $\mathsf{I} \Vdash \forall t^{\daleth\Lambda}(\{t\} \to \langle t \Vdash \bot\rangle \neq 1)$ and

$\lambda x \lambda y \lambda z(z)xy \Vdash \forall^{\daleth\Lambda}\forall'^{\daleth\Lambda}(\{t\}, \{t'\}, \neg\{t\,t'\} \to \bot)$.

Therefore $t\,\varepsilon\, G \to \langle t \Vdash \bot\rangle \neq 1$ et $t\,\varepsilon\, G, t'\,\varepsilon\, G \to t\,t'\,\varepsilon\, G$ as in the case of forcing.

We can generalize the property " G meets all dense subsets of Λ which are in $\mathcal{M}$ " :

$D \in \mathcal{M}, D \subset \Lambda$ will be said *dense* if there is some $\theta \in$ PL s.t. :

$(\forall \xi \in \Lambda)(\theta\xi \not\Vdash \bot \Rightarrow (\exists t \in D)(\xi t \not\Vdash \bot))$. Then we have $\theta \Vdash (\exists t\,\varepsilon\, G)(\langle t \in D\rangle = 1)$.

Indeed $\theta \Vdash \forall t^{\daleth\Lambda}(\langle t \in D\rangle = 1 \hookrightarrow t \not\varepsilon\, G) \to \bot$.

The model of threads

The r.a. has two new instructions : quote and eval.

Each stack is terminated by a stack constant $\pi_n (n \in \mathbb{N})$.

PL is the set of terms which do not contain any continuation k_π.

Let $\theta_n (n \in \mathbb{N})$ be an enumeration of PL. We define $\perp\!\!\!\perp$ by :

$$\xi \star \pi \notin \perp\!\!\!\perp \Leftrightarrow \exists n(\theta_n \star \pi_n \succ \xi \star \pi).$$

Λ_n (resp. Π_n) is the set of terms (resp. stacks) which contain
the only continuation k_{π_n}. We have essentially $\Lambda = \bigcup_n \Lambda_n$ and $\Pi = \bigcup_n \Pi_n$.

For $t \in \Lambda, \pi \in \Pi$, we define $n[t], n[\pi] \in \mathbb{N}$ such that $t \in \Lambda_{n[t]}, \pi \in \Pi_{n[\pi]}$.

Execution of quote and eval : $\mathrm{quote} \star \xi \bullet t \bullet \pi \succ \xi \star \underline{n}[\pi] \bullet \underline{\nu}[t] \bullet \pi$;
where $\nu[t]$ is the number of t in an enumeration of $\Lambda_{n[\pi]}$.

$\mathrm{eval} \star \xi \bullet \underline{\nu} \bullet \pi \succ \xi \star t \bullet \pi$ where $t \in \Lambda_{n[\pi]}$ is such that $\nu[t] = \nu$.

The model of threads

Remark. We might think to list $\Lambda = (t_n)_{n\in\mathbb{N}}$ and define the execution of eval by :
eval$\star\xi\,.\,\underline{n}\,.\,\pi \succ \xi \star t_n\,.\,\pi$.
But if $t_n \Vdash \perp$ then eval I $\underline{n} \Vdash \perp$ and is proof-like ; therefore $\perp\!\!\!\perp = \emptyset$.
Let us define $\gamma = \{(\underline{n},\pi)\ ;\ n \in \mathbb{N}, \pi \in \Pi_n\}$. We have easily :
λx(K)(q)(K)x $\Vdash \exists n^{\text{int}}(n\,\varepsilon\,\gamma)$; $\lambda x\lambda y\lambda z\, z \Vdash \forall n \forall n'(n\,\varepsilon\,\gamma, n'\,\varepsilon\,\gamma \to n = n')$.
Therefore γ has exactly one element which is an integer denoted by n[g].
It is non standard : indeed, $\omega\underline{0}$ or $\omega\underline{1} \Vdash \underline{n} \not\varepsilon\,\gamma$ for each standard integer n.
The notation n[g] means that it is the number of a proof-like term g.
Thus g is a program and it has extraordinary properties.
For instance, every cooperative process is executed inside g.
It is a pity it is non standard.

The model of threads

Let us show the remarkable fact that $\mathcal{N} \models G = \Lambda_{n[\mathrm{g}]}$.

In other words, the generic is the set of terms which contain only $\mathsf{k}_{\pi_{n[\mathrm{g}]}}$.

Indeed, $G \subset \Lambda_{n[\mathrm{g}]}$ because I $\Vdash \forall t^{\daleth\Lambda}(n[t] \neq n[\mathrm{g}] \to \neg\{t\})$.

Moreover $\Lambda_{n[\mathrm{g}]} \subset G$ because eval $\Vdash \forall t^{\daleth\Lambda}(t \not\varepsilon G, t \,\varepsilon\, \Lambda \to n[t] \neq n[\mathrm{g}])$

i.e. $\forall t^{\daleth\Lambda}(\neg\{t\}, \{\underline{v}[t]\} \to n[t] \neq n[\mathrm{g}])$. QED

Therefore ***G is a recursive real !***

It is natural to call $n[\mathrm{g}]$ *the generic integer* since G is determined by it.

Realizing DC with fresh constants

Here is a new way of realizing NEC (non extensional choice) and therefore DC.

We need for this a countable realizability algebra containing :

- the λ-calculus ;
- a sequence $h_n (n \in \mathbb{N})$ of distinct term constants ;
- a new instruction κ with the following execution rule (introduction) :

$$\kappa \star \xi \cdot \pi \succ \xi \star h_n \cdot \pi$$

where h_n is a *fresh constant*, i.e. which does not appear in ξ, π, for instance, the first.

- a new instruction e with the following execution rule (elimination) :

$$\mathrm{e} \star h_m \star h_n \star \xi \cdot \eta \cdot \pi \succ \xi \star \pi \text{ if } m = n \text{ and } \eta \star \pi \text{ if } m \neq n.$$

Realizing DC with fresh constants

We will show that $\Vdash$ "$\beth\mathbb{N}$ is countable" and thus $\Vdash$ "$\beth\Pi$ is countable".

This implies, rather trivially, that NEC is realized (cf. [Kr2]).

Let h : $\mathbb{N} \to \Lambda$ defined, in $\mathscr{M}$, by h$[n] = h_n$. In $\mathscr{N}$, we have h : $\beth\mathbb{N} \to \beth\Lambda$.

Since h is injective in $\mathscr{M}$, it is the same in $\mathscr{N}$.

Let H $= \{(h_n, h_n \bullet \pi)\ ;\ n \in \mathbb{N}, \pi \in \Pi\}$.

Then $\kappa \Vdash \forall \nu^{\beth\mathbb{N}} \exists \mu^{\beth\mathbb{N}} (\exists h\, \varepsilon\, \mathrm{H})\{h = \mathrm{h}[\mu], \nu = pr_1[\mu]\}$. In this way, we get a surjection from H onto $\beth\mathbb{N}$. We finish by showing that H is countable.

This follows from : $\forall \mu^{\beth\mathbb{N}} \forall \nu^{\beth\mathbb{N}} (\mathrm{h}[\mu]\, \varepsilon\, \mathrm{H}, \mathrm{h}[\nu]\, \varepsilon\, \mathrm{H}, \langle \mu = \nu \rangle\, \varepsilon\, \mathscr{D} \to \mathrm{h}[\mu] = \mathrm{h}[\nu])$

given by e $\Vdash \forall \mu^{\beth\mathbb{N}} \forall \nu^{\beth\mathbb{N}} (\mathrm{h}[\mu]\, \varepsilon\, \mathrm{H}, \mathrm{h}[\nu]\, \varepsilon\, \mathrm{H}, \langle \mu = \nu \rangle \neq 0 \to \mathrm{h}[\mu] = \mathrm{h}[\nu])$.

The program obtained pour NEC and DC contains the instructions κ, e.

It is rather complicated, since the proof involves the ultrafilter $\mathscr{D}$.

Realizing WOC

The *well ordered axiom of choice (WOC)* is the following :

The product of a family of non empty sets indexed by an ordinal is non empty.

It implies DC (cf. [J]).

We show that this axiom is satisfied in the last realizability model considered in [Kr2].

This has two interesting consequences :

1. We can write a program which realizes WOC.

This program contains the parallel instruction γ defined below.

2. A new proof of the independence of AC from ZF + WOC (cf. [J]).

This a joint work with L. Fontanella.

Realizing WOC

The realizability algebra considered in [Kr2] is obtained as follows :

Consider first the algebra $\mathscr{A}_0$ the terms of which are

the λ-terms with two supplementary instructions : stop and γ.

Recursive definition of $\perp\!\!\!\perp$ i.e. execution of stop and γ :

- stop$\star\pi \in \perp\!\!\!\perp$;
- if two among $\xi \star \pi, \eta \star \pi, \zeta \star \pi$ are in $\perp\!\!\!\perp$, then $\gamma \star \xi . \eta . \zeta . \pi \in \perp\!\!\!\perp$.

This implies that $\beth 2$ has 4 elements at most.

Realizing WOC

Now we extend the realizability model $\mathcal{N}$ by forcing
so that $\beth\mathbb{N}$ becomes countable and therefore NEC is satisfied.
We get a new r.a. $\mathscr{A}_1$ and a new r.m. with the same $\beth\mathbb{N}$ (cf. [Kr2]).
Let a_0, a_1 be the two atoms of $\beth 2$; we have $a_i \mathcal{N} = \mathcal{M}_{a_i} \succ \mathcal{M}$ and $\mathcal{M}_{a_0} = \mathcal{M}_{\mathscr{D}}$.
And also $\mathcal{N} = \mathcal{M}_{\mathscr{D}} \times \mathcal{M}_{a_1}$.
Since $\mathcal{M}_{\mathscr{D}}$ is well founded, its class of ordinals $\mathrm{On}_{\mathscr{D}}$ is isomophic with On.
If α is an ordinal, let $\alpha_{\mathscr{D}}$ be its image in $\mathrm{On}_{\mathscr{D}}$.
The axiom NEC implies that the product of a family of non empty sets
indexed by $\alpha_{\mathscr{D}}$ is non empty. QED

Realizing AC

We will now build a realizability algebra :

1. Of the "informatic kind" i.e. the terms are real programs.
2. Every realizability model satisfies AC.

Then, there exists a program, i.e. *a proof-like term, which realizes AC*.

The forcing models satisfy 2 but not 1.

This is the first instance of an algebra, not coming from forcing (i.e. $\beth 2 \neq 2$) the models of which satisfy AC.

We start with the algebra $\mathscr{A}_1$ and the r. model $\mathscr{N}$ of the previous slides, in which $\beth\mathbb{N}$ is countable and NEC and WOC are realized [Kr2].

Let a_0, a_1 be the two atoms ; we have $a_i\mathscr{N} = \mathscr{M}_{a_i} \succ \mathscr{M}$ and $\mathscr{N} = \mathscr{M}_{a_0} \times \mathscr{M}_{a_1}$.

Let us assume that $\mathscr{M} \models$ V = L. Then $\mathscr{M}_{a_0} = \mathscr{M}_{\mathscr{D}} =$ L.

Realizing AC

We will show that the extensional model $\mathcal{N}_\in$ has the following property :

$(*)$ *There exists an X and a functional Φ which is a surjection from L×X onto $\mathcal{N}_\in$*

Remark. It matters that X be in $\mathcal{N}_\in$, not only in $\mathcal{N}$.

Now, by means of a generic extension $\mathcal{N}[G]$ of $\mathcal{N}$, *we make this set X countable*.

Then, by $(*)$, the new model $\mathcal{N}_\in[G]$ satisfies AC.

It is shown in [Kr1,Kr2] that $\mathcal{N}[G]$ is a realizability model.

for an algebra $\mathscr{A}_2 \in \mathcal{M}$ which has the same terms, stacks and PL than $\mathscr{A}_0$ or $\mathscr{A}_1$

but neither the same $\perp\!\!\!\perp$ nor the same execution.

Moreover, we have $(\mathcal{N}[G])_\in = \mathcal{N}_\in[G]$ *because X is in $\mathcal{N}_\in$* (cf. remark above).

Therefore, $(\mathcal{N}[G])_\in \models$ AC, and *there exists a proof-like term for AC*.

Realizing AC

Remark. In a recent and very useful discussion with A. Karagila, I asserted that no generic extension of $\mathscr{N}_\in$ satisfies AC. This is a (welcome) counter-example.

Now, it remains to prove property $(*)$ above.

We use the fact that $\mathscr{M}_{a_0}$ and $\mathscr{M}_{a_1}$ are "incompatible", i.e. :

For every functional $F : \mathscr{M}_{a_1} \to \mathscr{M}_{a_0}$, its image $\mathrm{Im}(F) = F(\mathscr{M}_{a_1})$ is a set.

Indeed, there exists a surjection from $\daleth\Lambda$ onto $\mathrm{Im}(F)$ (cf. [Kr2]).

Let Φ be the collapsing functional from $\mathscr{N} = \mathscr{M}_{\mathscr{D}} \times \mathscr{M}_{a_1}$ onto $\mathscr{N}_\in$.

We show there is some Y in $\mathscr{M}_{a_1}$ s.t. $\Phi(\mathscr{M}_{\mathscr{D}} \times Y) = \mathscr{N}_\in$:

If it's false, define $F : \mathscr{M}_{a_1} \to \mathscr{M}_{\mathscr{D}}$ by $F(Y) =$ the least α s.t. $V_\alpha \not\subset \Phi(\mathscr{M}_{\mathscr{D}} \times Y)$.

The image of F is a set, so it has an upper bound in On

which contradicts $\Phi(\mathscr{M}_{\mathscr{D}} \times \mathscr{M}_{a_1}) = \mathscr{N}_\in$.

Realizing AC

Therefore $\mathscr{N}_\in = \Phi(\mathscr{N}) = \Phi(\mathscr{M}_\mathscr{D} \times Y) = \bigcup_{a\,\varepsilon\,\mathscr{M}_\mathscr{D}} \Phi(\{a\} \times Y)$.

The sets $X_a = \Phi(\{a\} \times Y)$ are in $\mathscr{N}_\in$, images of a unique set Y which is in $\mathscr{N}$.

Each one is equipotent to a quotient of Y by an equivalence relation.

Now, these e.r. form a set (subset of $\mathscr{P}(Y^2)$).

By means of the collection axiom, we obtain a set X in $\mathscr{N}_\in$

which contains at least one representative X_a for each e.r.

Therefore, there is, in $\mathscr{N}_\in$, a surjection from X onto each X_a.

Using NEC, we get a surjection of $\mathscr{M}_\mathscr{D} \times X$ onto $\bigcup_a X_a = \mathscr{N}_\in$. QED

Note that the program for AC uses the instruction γ (which ensures $\daleth 2 = 2^2$).

It is a parallel instruction : in order to run $\gamma \star \xi \centerdot \eta \centerdot \zeta \centerdot \pi$

we must launch the three processes $\xi \star \pi, \eta \star \pi, \zeta \star \pi$.

References

[J]. **T. Jech** *The axiom of choice.* (1973)

[Ka]. **A. Karagila** *Realizing realizability results with classical constructions.* arXiv:1905.08202 (2019).

[Kr1]. **J-L. Krivine** *Realizability algebras : a program to well order R.* Logical Methods in Comp. Sc., vol. 7, 3:02 (2011) p. 1-47.

[Kr2]. **J-L. Krivine** *Realizability algebras III : some examples.* Math. Struct. in Comp. Sc. vol. 28, 1 (2018) p. 45-76.