Bar recursion in classical realizability : Dependent choice and Continuum hypothesis

Jean-Louis Krivine

I.R.I.F. - University Paris-Diderot

krivine@pps.univ-paris-diderot.fr

Marseille, August 30, 2016

Brief history

The bar recursion operator was introduced by C. Spector in 1962
in order to prove (?) the consistency of Analysis, i.e. :
2nd order Arithmetic + DC (*axiom of dependent choice*)
or CC (*axiom of countable choice)* which is slightly weaker.
In 1998, S. Berardi, M. Bezem and T. Coquand used a similar operator,
to obtain *programs* from *classical proofs in Analysis*.
In 2001 this fundamental work was refined and
translated in *denotational semantics (domains)* by U. Berger and P. Oliva.
In 2013, T. Streicher managed to use this operator in *classical realizability*
which permits to get programs from proofs in *set theory with dependent choice*.
Moreover, as we shall see today, we can also use the following axioms :
Well ordering of $\mathbb{R}$ (and therefore *Ultrafilters on* $\mathscr{P}(\mathbb{N})$) and *Continuum hypothesis*.

What is the bar recursion operator ?

Simply a λ-term, somewhat like the fixpoint operator
just a little more complicated. But its execution is difficult to understand.
Define first the λ-term $\chi = \lambda k \lambda f \lambda z \lambda n(\texttt{if}\ n < k\ \texttt{then}\ fn\ \texttt{else}\ z)$.
If f is a function of domain $\mathbb{N}$, then χkfz is the same function
in $[0 \cdots k-1]$ and the constant z in $[k \cdots \infty]$.
Let $k^+ = \lambda f \lambda x(f)(k)fx$ be the successor of the integer k.
Define now a λ-term $\Phi = \Psi GU$ which depends on two arbitrary terms G, U :

$$\Phi kf = (U)(\chi kf)(G)\lambda z(\Phi k^+)(\chi)kfz$$

The recursive definition of Ψ is therefore :

$$\Psi = \lambda g \lambda u \lambda k \lambda f(u)(\chi kf)(g)\lambda z(\Psi guk^+)(\chi)kfz.$$

We are interested mainly by the value $\Phi 0$; indeed, the bar recursion operator is :

$$\text{BR} = \lambda g \lambda u \Psi gu0.$$

The programming language : the BBC-algebra

In classical realizability, we use a *realizability algebra* which is a complicated form of *combinatory algebra*. Let us describe the particular one we need here which is simply the λ-calculus with some additions.

We call it the BBC-algebra (for Berardi-Bezem-Coquand).

Λ is the set of closed λ-terms with the following supplementary instructions :

cc (call/cc), A (abort), p (stop) and a (very big) set of *oracles* :

there is an oracle $\wedge_i t_i$ for *every infinite sequence* $t_i (i \in \mathbb{N})$ of terms.

They are needed for the theory but (fortunately) do not appear in real programs.

Π is the set of *stacks* (or environments) which are *finite* sequences of terms.

We write such a stack $\pi = t_0 \bullet t_1 \bullet \ldots \bullet t_{n-1} \bullet \pi_0$ with $t_i \in \Lambda$; π_0 is the *empty stack*.

Define the *continuation* $\mathsf{k}_\pi = \lambda x (\mathsf{A})(x) t_0 \ldots t_{n-1}$.

Because of the oracles, the cardinality of Λ and Π is $2^{\aleph_0}$.

Execution of processes

We execute not a term but a *process* i.e. a pair $t \star \pi$ ($t \in \Lambda, \pi \in \Pi$). The rules are :

$\mathsf{p} \star \pi \succ$ (stop)

$tu \star \pi \succ t \star u . \pi$ (push)

$\lambda x\, t \star u . \pi \succ t[u/x] \star \pi$ (pop)

$\mathsf{A} \star t . \pi \succ t \star \pi_0$ (delete the stack) or (abort)

$\mathsf{cc} \star t . \pi \succ t \star \mathsf{k}_\pi . \pi$ (save the stack)

$\wedge_i t_i \star \underline{n} . \pi \succ t_n \star \pi$ (oracle) ; $\underline{n}$ is a Church integer.

This last rule is never used in real computations.

From the rule for A, it follows easily that :

$\mathsf{k}_\pi \star t . \pi' \succ t \star \pi$ (restore the stack).

Finally, we define $\perp\!\!\!\perp$: the set of all processes which reduce to $\mathsf{p} \star \pi$ with $\pi \in \Pi_\perp$ some fixed set of stacks.

Formulas and realizability

Usual set theory ZF with the relation symbols $\in, \subset$ and function symbols ;

ZF$_\varepsilon$ is a conservative extension of ZF with a new relation symbol :

ε (strong, non extensional membership relation).

We use only $\top, \bot, \to, \forall$ as logical symbols.

For each formula F, we define two values, by induction :

the *truth value* $|F| \subset \Lambda$; the *falsity value* $\|F\| \subset \Pi$.

They are connected by the relation $t \in |F| \Leftrightarrow (\forall \pi \in \|F\|)(t \star \pi \in \perp\!\!\!\perp)$.

If $t \in |F|$, we say that *t realizes F* and we write $t \Vdash F$.

Definition by recurrence :

$\|\bot\| = \Pi$; $\|\top\| = \emptyset$; $\|a \not\varepsilon b\| = \{\pi; (a, \pi) \in b\}$; $a \neq b \equiv \top$ or $\bot$.

$\|F \to G\| = \{t \centerdot \pi; t \Vdash F, \pi \in \|G\|\}$; $\|\forall x\, F[x]\| = \bigcup_x \|F[x]\|$.

Proofs and programs

Proofs are done by means of *classical natural deduction*. Therefore :
Each proof gives a program written in λ_{cc}-calculus (λ-calculus with cc).
We call such a program a *proof-like term*.
The essential property is the *adequation lemma* : If $\vdash t : F$ then $t \Vdash F$
i.e. : any term you get from a proof of a formula F is a realizer of F.

If we want to get (useful) programs from proofs in ZF_ε
we must realize each axiom of ZF_ε by a proof-like term.
This is done, once and for all, by the theory of classical realizability
for every realizability algebra.

Proofs and programs

Now, if we want to get programs from proofs in $\mathrm{ZF}_\varepsilon + \mathrm{CC}$ (countable choice) we must realize CC by a proof-like term.

The same for the *Well ordering of* $\mathbb{R}$ or the *Continuum hypothesis*.

We now show that *in the particular realizability algebras* we have just defined the bar recursion BR realizes CC

and there is a (not so simple) proof-like term which realizes the Continuum hypothesis and even the stronger axiom :

Every real is constructible.

Realizing countable choice

The axiom of countable choice is : $\forall x \exists y F[x,y] \to \exists f \forall n^{\text{int}} F[n, f(n)]$.

The quantifier $\forall n^{\text{int}}$ is restricted to $\mathbb{N}$. It is realized as follows (Kleene realizability) :

$t \Vdash \forall n^{\text{int}} F[n]$ iff $t\underline{n} \Vdash F[n]$ for all integers n.

Now, we have to show $\mathrm{BR}GU \Vdash \perp$ with the hypotheses :

$G \Vdash \neg\forall y \neg F[x,y]$ for all x and $U \Vdash \neg\forall n^{\text{int}} F[n, f(n)]$ for all f.

With the above recursive definition $\Phi k\phi = (U)(\chi k\phi)(G)\lambda z(\Phi k^{+})(\chi)k\phi z$.

We have to show $\Phi\underline{0} \Vdash \perp$ and, in fact, we show $\Phi\underline{0}\phi_0 \Vdash \perp$ *for every* ϕ_0.

Now suppose $\Phi\underline{0}\phi_0 \not\Vdash \perp$. We define recursively $\phi_k \in \Lambda$ such that :

$\Phi\underline{k}\phi_k \not\Vdash \perp$ and $\phi_k\underline{n} \Vdash F[n, f_k(n)]$ for $n < k$ (for some function f_k) ;

there is no condition on $\phi_k\underline{n}$ for $n \geq k$. But we want $f_k \subset f_{k+1}$.

Realizing countable choice (cont.)

We have $\Phi\,\underline{k}\phi_k = (U)(\chi\,\underline{k}\phi_k)(G)\tau_k \not\Vdash \bot$ (recurrence hypothesis)

with $\tau_k = \lambda z(\Phi\underline{k}^+)(\chi)\underline{k}\phi_k z$. It follows that $(\chi\,\underline{k}\phi_k)(G)\tau_k \not\Vdash \forall n^{\text{int}} F[n, f_k(n)]$.

Therefore $(G)\tau_k \not\Vdash \bot$ so that $\tau_k \not\Vdash \forall y \neg F[k, y]$.

It follows that there exist $\zeta \in \Lambda$ and a_k such that :

$\zeta \Vdash F[k, a_k]$ and $\tau_k \zeta \not\Vdash \bot$. But we have $\tau_k \zeta = \Phi\underline{k}^+ \phi_{k+1}$ with $\phi_{k+1} = (\chi)\underline{k}\phi_k \zeta$.

This gives the recurrence step.

Observe that, until now, our reasoning is valid *for every realizability algebra*.

The sequences ϕ_k, f_k are increasing : the $(k+1)$th function is an extension of the kth.

Let f, ϕ be their extensions to the whole of $\mathbb{N}$; ϕ *is given by an oracle*.

By construction of ϕ, we have $\phi \Vdash \forall n^{\text{int}} F[n, f(n)]$ and therefore $U\phi \Vdash \bot$.

Realizing countable choice (cont.)

Now, the realizability algebra has the following property, known as :

Continuity. If ϕ is an oracle and $U\phi \Vdash \perp$, then there exists an integer k such that $U\psi \Vdash \perp$ for every ψ such that $\psi\underline{n} = \phi\underline{n}$ for $n < k$.

Proof. The execution of $U\phi$ is finite, thus uses only finitely many $\phi\underline{n}$. ∎

But then, we can take $\psi = (\chi\underline{k}\phi_k)\eta$ for any $\eta \in \Lambda$ and, in particular : $\psi = (\chi\underline{k}\phi_k)(G)\tau_k$.

Now, we have $U\psi = \Phi\underline{k}\phi_k \nVdash \perp$ by construction of ϕ_k. Contradiction !

Realizing more axioms

It would be nice to find a λ_{cc}-term which realizes the *full axiom of choice*.

In addition to CC, this is only done for the following particular case :

There exists a well-ordering on $\mathbb{R}$.

This implies the existence of *ultrafilters on* $\mathscr{P}(\mathbb{N})$ which is useful for proving combinatorial properties in arithmetic (Ramsey theory).

Moreover, this well-ordering is isomorphic to $\aleph_1$, which is

The continuum hypothesis (CH).

The programs for these axioms contain BR but are more complicated.

For the moment, their behaviour is not well understood.

Sketch of proof

In fact, we have shown more than CC, namely :

$$\mathrm{BR} \Vdash \forall x \exists y\, F[x,y] \to \exists f \forall n^{\mathrm{int}} F[n, \mathrm{app}(f,n)]$$

where app is, in set theory, a new functional symbol for *application*.

Suppose that $F[x,y]$ defines a real, i.e. an application $\mathbb{N} \to \{0,1\}$.

We can replace $F[x,y]$ with $F[x,y] \wedge (y=0 \vee y=1)$ so that we have

$$\mathrm{BR}' \Vdash \exists f \forall n^{\mathrm{int}} \big(F[n, \mathrm{app}(f,n)] \wedge (\mathrm{app}(f,n)=0 \vee \mathrm{app}(f,n)=1)\big).$$

Now, the general theory of classical realizability gives a realizer for :

$$\forall f \left(\forall n^{\mathrm{int}} (\mathrm{app}(f,n)=0 \vee \mathrm{app}(f,n)=1) \to f \text{ is a constructible real}\right)$$

Thus, the following is realized when $F[x,y]$ defines a real :

$$\exists f \left((f \text{ is a constructible real}) \wedge \forall n^{\mathrm{int}} F[n, \mathrm{app}(f,n)]\right)$$

i.e. *Every real is constructible.*

Everything we want follows from this !

Using these axioms

Any proof of a formula F by means of the axioms of ZF + DC + CH gives a program (proof-like term) $\theta \Vdash F$. This is often very useful.

The simplest well kown example is $F \equiv \forall m^{\mathrm{int}} \exists n^{\mathrm{int}} (f(m,n) = 0)$.

Then, we have $\theta \underline{m} \Vdash \neg \forall n^{\mathrm{int}} (f(m,n) \neq 0)$.

Now, choose $\perp\!\!\!\perp = \{\mathsf{p} \star \underline{n} \cdot \pi \,;\, \pi \in \Pi, f(m,n) = 0\}$; then $\mathsf{p} \Vdash \forall n^{\mathrm{int}} (f(m,n) \neq 0)$.

Thus $\theta \underline{m} \star \mathsf{p} \cdot \pi_0 \in \perp\!\!\!\perp$ and therefore $\theta \underline{m} \star \mathsf{p} \cdot \pi_0 \succ \mathsf{p} \star \underline{n} \cdot \pi$ with $f(m,n) = 0$.

The program θ computes a solution of $f(m,n) = 0$ for every m.

Observe that, since it comes from a proof, it contains no oracle.