Computational Content of the Classical Axiom of Countable Choice

Vladimir Ivanov

July 17, 2023

1 The Computational System

Definition 1 (Lambda Terms)**.** *Let* Var *be a countably infinite set of variable names. The notions of free variable and closed term are defined as usual. We define the set* Λ_{open} *of not necessarily closed lambda terms, and take* Λ *to be the set of closed terms of* Λ_{open}*. When we say lambda term, we mean closed lambda term.* Λ_{open} *is defined as follows*

$$
\begin{aligned}
\Lambda_{open} := & \mid var \in \mathsf{Var} & \\
& \mid \Lambda_{open}\ \Lambda_{open} & \\
& \mid \lambda x.\Lambda_{open} & \textit{where } x \in \mathsf{Var} \\
& \mid 0 \mid \mathsf{succ} \mid \mathsf{rec}_{\mathbb{N}} & \textit{constructors and the recursor for } \mathbb{N} \\
& \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{rec}_{\mathsf{bool}} & \textit{constructors and the recursor for } \mathsf{bool} \\
& \mid \Phi & \textit{the bar recursion operator} \\
& \mid \mathsf{cc} \mid \mathsf{k}_\pi & \textit{where } \pi \textit{ is a stack}
\end{aligned}
$$

Definition 2 (Stacks)**.** *A stack is a finite list of closed lambda terms. We let* Π *be the set of all stacks. We write* $t \cdot \pi$ *for prepending a lambda term to a stack and* $\pi \cdot \pi'$ *for concatenating two stacks. We write* π_{empty} *for the empty stack. We sometimes omit the* $\cdot$*.*

Definition 3 (Process)**.** *A process is a pair* $\langle t \mid \pi \rangle$ *of a lambda term and a stack. We write* $\Lambda \times \Pi$ *for the set of all processes.*

Definition 4 (Reduction Relation)**.** *The big step reduction relation* $\succ$ *is the* *smallost transitive and reflexive relation which staisfies:*

$$
\begin{aligned}
\langle tu \mid \pi \rangle &\succ \langle t \mid u\pi \rangle \\
\langle \lambda x.t \mid u\pi \rangle &\succ \langle t[x := u] \mid \pi \rangle \\
\langle \mathsf{rec}_{\mathbb{N}} \mid t_0 \cdot t_{\mathsf{succ}} \cdot 0 \cdot \pi \rangle &\succ \langle t_0 \mid \pi \rangle \\
\langle \mathsf{rec}_{\mathbb{N}} \mid t_0 \cdot t_{\mathsf{succ}} \cdot \mathsf{succ}\ n \cdot \pi \rangle &\succ \langle \mathsf{rec}_{\mathbb{N}}\ (t_{\mathsf{succ}}\ t_0)\ t_{\mathsf{succ}}\ n \mid \pi \rangle \\
\langle \mathsf{rec}_{\mathsf{bool}} \mid t_{\mathsf{true}} \cdot t_{\mathsf{false}} \cdot \mathsf{true} \cdot \pi \rangle &\succ \langle t_{\mathsf{true}} \mid \pi \rangle \\
\langle \mathsf{rec}_{\mathsf{bool}} \mid t_{\mathsf{true}} \cdot t_{\mathsf{false}} \cdot \mathsf{false} \cdot \pi \rangle &\succ \langle t_{\mathsf{false}} \mid \pi \rangle \\
\langle \Phi \mid H \cdot P \cdot C \cdot \overline{k} \rangle &\succ P\ (C; \geq \overline{k} \mapsto H\ (\lambda z.\ \Phi\ H\ P\ \overline{k+1}\ (C; \geq \overline{k} \mapsto z))) \\
\langle \mathsf{cc} \mid t \cdot \pi \rangle &\succ \langle t \mid \mathsf{k}_\pi \cdot \pi \rangle \\
\langle \mathsf{k}_\pi \mid t \cdot \pi' \rangle &\succ \langle t \mid \pi \rangle
\end{aligned}
$$

All the syntactic sugar used in the rule for Φ will be defined in the following subsection.

For $t, u \in \Lambda$, we write $t \succ u$ for $\forall \pi \in \Pi. \langle t \mid \pi \rangle \succ \langle u \mid \pi \rangle$.

1.1 Syntactic Sugar and Special Terms

naturals For $n \in \mathbb{N}$, we write $\overline{k}$ for $\mathsf{succ}^n\ 0$.

if then else We write $\mathsf{if}\ t\ \mathsf{then}\ u\ \mathsf{else}\ v$ for $\mathsf{rec}_{\mathsf{bool}}\ u\ v\ t$.

comparison For $t, u \in \Lambda$, we use $\mathsf{rec}_{\mathbb{N}}$ to define a term $t \leq u$ such that for all $n, m \in \Lambda$

$$
\begin{aligned}
\overline{n} \leq \overline{m} &\succ \mathsf{true} && \text{if } n \leq m \\
\overline{n} \leq \overline{m} &\succ \mathsf{false} && \text{otherwise}
\end{aligned}
$$

function cons Let $f, t, n \in \Lambda$. Define

$$(f; \geq n \mapsto t) \in \Lambda := \lambda k.\ \mathsf{if}\ n\ \leq\ k\ \mathsf{then}\ t\ \mathsf{else}\ f\ k$$

This term satisfies, for all $f, t \in \Lambda, n, m \in \mathbb{N}$,

$$
\begin{aligned}
(f; \geq \overline{n} \mapsto t)\ \overline{m} &\succ f\ \overline{m} && \text{if } m < n \\
(f; \geq \overline{n} \mapsto t)\ \overline{m} &\succ t && \text{if } m \geq n
\end{aligned}
$$

We omit parentheses and write $(f; \geq k_1 \mapsto t_1; \geq k_2 \mapsto t_2; \ldots; \geq k_n \mapsto t_n)$ for $((\ldots((f; \geq k_1 \mapsto t_1); \geq k_2 \mapsto t_2))\ldots)); \geq k_n \mapsto t_n)$

2 Realizability

2.1 Logic

We take some model of $\mathbb{N}$.

Atoms are defined as equalities of expressions built from variables and some fixed (possibly zero-ary) functions $f_1 : \mathbb{N}^{k_1} \to \mathbb{N}, \ldots, f_n : \mathbb{N}^{k_n} \to \mathbb{N}$. We define formulas as

$$
\begin{aligned}
F, G, \cdots :=\ & \mid A && \text{where } A \text{ is an atom} \\
& \mid x && \text{where } x \text{ is a variable name taken from countably infinite set} \\
& \mid \forall x \in X.F && \text{where } X \text{ is a set and } x \text{ is a variable name free in } F \\
& \mid \forall^{\mathbb{N}} n.F && \text{where } x \text{ is a variable name free in } F \\
& \mid F \to G
\end{aligned}
$$

Later, when we say formula, we mean formula with no free variables.

Note that, since cc realizes Peirce's law, we do not need existential quantification since we can encode it as $\neg\forall\neg$ by De Morgan's law.

2.2 The Realizability Relation

Throughout the rest of the section, we fix a pole $⊥\!\!⊥ \subseteq \Lambda \times \Pi$ be a set of processes which is closed by anti-reduction, that is, for all processos p and q, if $p \succ q$ and $q \in ⊥\!\!⊥$ then $p \in ⊥\!\!⊥$.

For a formula F, define $\|F\| \subseteq \Pi$ and $|F| \subseteq \Lambda$ by structural induction over the syntax of F as follows:

$$
\begin{aligned}
\|\bot\| &:= \Pi \\
\|a\| &:= \Pi \quad \text{if } a \text{ is an atom which is true in the model of } \mathbb{N} \\
\|a\| &:= \emptyset \quad \text{if } a \text{ is an atom which is false in the model of } \mathbb{N} \\
\|\forall x \in X.G\| &:= \bigcup_{x_0 \in X} \|G[x := x_0]\| \\
\|G \to H\| &:= \{t\pi \mid t \in |G|, \pi \in \|H\|\} \\
\|\forall^{\mathbb{N}} x.G\| &:= \{\overline{n}\pi \mid n \in \mathbb{N}, \pi \in \|G[x := n]\|\} \\
|F| &:= \{t \in \Lambda \mid \forall \pi \in \|F\|. \langle t \mid \pi \rangle \in ⊥\!\!⊥\}
\end{aligned}
$$

2.3 Properties

Propetry 1. *If $t \succ u$ and $u \Vdash t$, then $t \Vdash u$.*

Propetry 2. *$t \Vdash A \to B$ if and only if $\forall u \Vdash A. tu \Vdash B$.*

Propetry 3. *If $t \Vdash \bot$ then $t \Vdash F$ for all formula F.*

Propetry 4 (Consistency)**.** *There is no realizer of $\bot$.*

Propetry 5 (Continuity)**.** *Let t be a term with one free variable x. Let F be a formula. Let $u_0, u_1, \dots \in \Lambda$. Suppose that $t[x := \lambda\!\!\lambda\, n.u_n] \Vdash F$. Then there exists $N \in \mathbb{N}$ such that for all $f \in \Lambda$ such that $\forall n < N.\ f\ \overline{n} \succ u_n$ we have $t[x := f] \Vdash F$.*

3 Realizability of Countable Choice

Throughout this section, let I be a set and $R : \mathbb{N} \times I \to \mathcal{P}(\Pi)$. We define

$$AC_{\mathbb{N}} := (\forall n.\neg\forall i.\neg R(n,i)) \to \neg\forall f.\neg\forall n.R(n,f(n))$$

Theorem 1. *We have*

$$\lambda H.\ \lambda P.\ \Phi\ H\ P\ \overline{0}\ \overline{0} \Vdash AC_{\mathbb{N}}$$

Proof. Let

$$\begin{aligned} H &\Vdash \forall n.\neg\forall i.\neg R(n,i) \\ P &\Vdash \forall f.\neg\forall^{\mathbb{N}} n.R(n,f(n)) \end{aligned}$$

By propetry 2, it is necessarily and sufficient to show that

$$\Phi\ H\ P\ \overline{0}\ \overline{0} \Vdash \bot$$

Definition 5 ($<k$ -cache)**.** *Let $k \in \mathbb{N}$. A $<k$ -cache is a term $C \in \Lambda$ such that $\forall n < k.\exists i.C\ \overline{n} \Vdash R(n,i)$.*

Lemma 1. *Let $k \in \mathbb{N}$. Let C be a $<k$ -cache. Suppose that $\Phi\ H\ P\ C\ \overline{k} \nVdash \bot$. Then, there exists i_k and $r_k \Vdash R(k,i_k)$ such that $\Phi\ H\ P\ (C; \geq \overline{k} \mapsto r_k)\ \overline{k+1} \nVdash \bot$.*

Note that $(C; \geq \overline{k} \mapsto r_k)$ is then a $<k+1$ -cache.

Proof. There exist $r_k \in \Lambda, i_k \in I$ such that $r_k \Vdash R(k,i_k)$. Indeed, if none did exists, then for all i, any term would realize $\neg R(k,i)$, thus, any term would realize $\forall i.\neg R(k,i)$. Thus, H applied to any term would realize $\bot$, which contradicts consistency.

Now, suppose that for all r_k, i_k such that $r_k \Vdash R(k,i_k)$,

$$\Phi\ H\ P\ (C; \geq \overline{k} \mapsto r_k)\ \overline{k+1} \Vdash \bot$$

it suffices to find a contradiction. Then, for all i,

$$\lambda z.\ \Phi\ H\ P\ (C; \geq \overline{k} \mapsto z)\ \overline{k+1} \Vdash \neg R(k,i)$$

and so by the hypothesis on H and then by ???,

$$\begin{aligned} H\ (\lambda z.\ \Phi\ H\ P\ (C; \geq \overline{k} \mapsto z)\ \overline{k+1} \Vdash \neg R(k,i)) &\Vdash \bot \\ &\Vdash R(n,i) \quad \text{for all } n \text{ and } i \end{aligned}$$

Now, by definition of a cache, for all $n < k$, $C\ \overline{n} \Vdash R(n,i)$ for some i. Let $f(n)$ be such an i for $n < k$ and be arbitrary for $n \geq k$. Then,

$$(C; \geq \overline{k} \mapsto H\ (\lambda z.\ \Phi\ H\ P\ (C; \geq \overline{k} \mapsto z)\ \overline{k+1} \Vdash \neg R(k,i))) \Vdash \forall^{\mathbb{N}} n.R(n, f(n))$$

since this term applied to $n < k$ realizes $R(n, f(n))$ by the definition of a cache and this term applied to $n \geq k$ realizes $R(n, f(n))$ by the previous discussion. Thus,

$$P\ (C; \geq \overline{k} \mapsto H\ (\lambda z.\ \Phi\ H\ P\ (C; \geq \overline{k} \mapsto z)\ \overline{k+1} \Vdash \neg R(k,i))) \Vdash \forall^{\mathbb{N}} n.R(n, f(n)) \Vdash \bot$$

But

$$\Phi\ H\ P\ C\ \overline{k} \succ P\ (C; \geq \overline{k} \mapsto H\ (\lambda z.\ \Phi\ H\ P\ (C; \geq \overline{k} \mapsto z)\ \overline{k+1}))$$

So

$$\Phi\ H\ P\ C\ \overline{k} \Vdash \bot$$

which was supposed to not hold. $\square$

Now, suppose

$$\Phi\ H\ P\ \overline{0}\ \overline{0} \nVdash \bot$$

it is necessary and sufficient to find a contradiction.

By applying lemma 1 repeatedly, we get a sequence $r_0, r_1, \ldots$ such that each r_n realizes $R(n,i)$ for some i. For each n, take $f_0(n)$ to be such an i.

Then, for all n, $(\lambda\!\!\lambda n.r_n)\overline{n} \Vdash R(n, f_0(n))$. Thus, $\lambda\!\!\lambda n.r_n \Vdash \forall^{\mathbb{N}} n.R(n, f_0(n))$ and so

$$P\ (\lambda\!\!\lambda n.r_n) \Vdash \bot$$

But then, by continuity, there exists N such that for all C, if $\forall n < N.\ C\ \overline{n} \succ r_n$ then $P\ C \Vdash \bot$. Thus, taking $C = (\overline{0}; \geq \overline{0} \mapsto r_0; \geq \overline{1} \mapsto r_1; \ldots; \geq \overline{N-1} \mapsto r_{N-1}; \geq \overline{N} \mapsto H\ (\lambda z.\ \Phi\ H\ P\ \overline{N+1}\ (C; \geq \overline{k} \mapsto z)))$, we have $P\ C \Vdash \bot$ since $\forall n < N.C\ \overline{n} \succ r_n$. But $\Phi\ H\ P\ \overline{0}\ \overline{0} \succ P\ C$, so $\Phi\ H\ P\ overline0\ \overline{0} \Vdash \bot$, which was supposed to not hold.

$\square$