Computational Content of the Axiom of Choice in Evidenced Frames

Vladimir Ivanov

July 10, 2023

1 The Computational System

The additional codes are $\mathsf{makeaxiom}$, axiom_n for $n \in \mathbb{N}$ and c_f for $f : \mathbb{N} \to C$. The additional reduction relations are

$$\frac{}{\mathsf{makeaxiom} \cdot \overline{k} \downarrow \mathsf{axiom}_k}$$

$$\frac{}{c_f \cdot \overline{n} \downarrow f(n)}$$

The additional termination rules are

$$\frac{}{\mathsf{makeaxiom} \cdot \overline{k} \downarrow}$$

$$\frac{}{c_f \cdot \overline{n} \downarrow}$$

2 The Evidenced Frame

We take the separator $\mathcal{S}_\top$. We interpret non determinism angelically [É]: is there non-deterministic computations?.

For all $n \in \mathbb{N}$, let nat_n be the predicate realized only by $\overline{n}$. We define universal quantification over naturals:

$$\forall n.\phi_n$$

as $\prod\{\mathsf{nat}_n \to \phi_n | n \in \mathbb{N}\}$ [É]: $\prod\{\mathsf{nat}_n \supset \phi_n | n \in \mathbb{N}\}$

We define $\perp$ as the predicate realized by all $\mathsf{axiom}_k, k \in \mathbb{N}$.

We allow arbitrary expressions, and not just pairs of codes, on the left of $\downarrow\downarrow$ [É]: You mean applications of en expression to an expression?.

3 Pragmatics

We write $\lambda x_0 \dots x_k.e[x_0]\dots[x_k]$ for $c_{\lambda^k.e}$. Not that not all lambda terms can be easily encoded, for example, we can't encode $\lambda x.x(\lambda y.xy)$ [É]: why? if you take: $e_2 \triangleq 1 \cdot 0$, $e_1 \triangleq \lambda.e_2$ and $e_0 \triangleq 0 \cdot e_1$, you can encode it with $c_{\lambda.e_0}$, no? and should work around by encoding it as $\lambda x.x((\lambda x', y.x'y)x)$. We leave such conversions implicit. We do not always write the $\cdot$.

We write $\exists x \in X.\phi_x$ for $\coprod\{\phi_x \mid x \in X\}$ and omit the $\in X$ when X is clear from context.

4 Preliminaries

Fact 1. *There exist codes* $\mathsf{intro}_\exists, \mathsf{intro}_\forall, \mathsf{elim}_\exists, \mathsf{elim}_\forall \in \mathcal{S}_\top$ *such that for all family of proposition* $(\phi_n)_{n\in\mathbb{N}}$*, all set* X*, and all family of propositions* $(\psi_x)_{x\in X}$

1. *If for all* n*,* $c \cdot \overline{n} \Downarrow \phi_n$*, then* $c \xrightarrow{\mathsf{intro}_\forall} \forall n.\phi_n$. *[É]: c is not a formula of the language, what do you mean?*
2. *If* $c \models \forall n.\phi_n$ *[É]: what is the definition of* $c \models \phi$*? , then for all* n*,* $\mathsf{elim}_\forall \cdot c \cdot n \models \phi_n$.
3. *For all* x*,* $\psi_x \xrightarrow{\mathsf{intro}_\exists} \exists x.\phi_x$. *[É]: what is the relation between* ψ *and* ϕ*?*
4. *If* $\prod\{\phi_x \mid x \in X\}$ *has a realizer, then there exists* $x \in X$ *such that* $\prod\{\phi_x \mid x \in X\} \xrightarrow{\mathsf{elim}_\exists} \psi_x$

Proof. Take

$$\mathsf{intro}_\forall := \lambda c.\lambda(e_{\mathsf{snd}}; c)$$

$$\mathsf{elim}_\forall := \lambda c, n.c\ n\ n$$

$$\mathsf{intro}_\exists := \lambda c.\lambda(<|c, <|e_{\mathsf{id}}, e_\top|>; e_{\mathsf{eval}}|>; e_{\mathsf{eval}})$$

$$\mathsf{elim}_\exists := <|\lambda(e_{\mathsf{fst}}; e_\top; \lambda(e_{\mathsf{fst};e_{\mathsf{id}}})), |>; e_{\mathsf{eval}}$$

[É]: what does the notation $<|c,|>$ stand for? ☐

Definition 1 (Behaves Like)**.** *We say that code* c'_f *behaves like code* c_f *if* $\forall c_a, c_r.c_f \cdot c_a \downarrow c_r \Rightarrow c'_f \cdot c_a \downarrow c_r$. *If we have an equivalence instead of an implication, we say that they are extensionally equal. [É]: I guess you could define this as an ordering relation* $c' \preccurlyeq c$ *on codes, and indeed the induced equality would be the extensional equality on codes.*

Lemma 1. *If* $c \cdot \overline{n} \Downarrow \bot$ *for all* k *and* $e \Downarrow \bot$*, then* $e[\mathsf{makeaxiom} := c] \Downarrow \bot$.

Proof. Structural induction over the proof of $e \downarrow \mathsf{axiom}_k$. ☐

Lemma 2. *If* c *and* c' *are extensionally equal, so are* $e[x := c]$ *and* $e[x := c']$ *for all expression* e *with a hole* x.

Proof. Structural induction over the proof of $e[x := c] \downarrow c_r$. ☐

5 The Proof

Let

$$KCC' := (\forall n.\nabla\exists i.\neg R(n,i)) \Rightarrow \nabla\exists f.\forall n.\neg R(n,f(n))$$
$$\mathsf{fix}_f := (\lambda x.f(xx))(\lambda x.f(xx))$$
$$\mathsf{fix} := \lambda f.\,\mathsf{fix}_f$$
$$\Phi_{H,P,\phi,L} := P(\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\,L\;n\;(\lambda\_.(\lambda m.Hm(\lambda q.\phi\;((m,q):L)))n))))$$
$$\Phi_{H,P} := \mathsf{fix}_{\lambda\phi L.\Phi_{H,P,\phi,L}}$$
$$H \models \forall n.\nabla\exists i.\neg R(n,i)$$
$$P \models \neg\exists f.\forall n.\neg R(n,f(n))$$

[É]: what is assoc?

It suffices to prove the following lemma, the rest of the proof is stricly identical to the original paper.

Definition 2 (Cache)**.** *A cache is a church encoded list L of church encoded pairs of the form $(\overline{n},q), n\in\mathbb{N}, q\in C$ such that*

- *For all $(\overline{n},q)\in L$ there exists i [É]: what is the type of i? such that $q\models R(n,i)$. [É]: $q\models\neg R(n,i)$?*
- *The first elements of pairs in L are pairwise distinct.*

Lemma 3. *Let L be a chache such that $\Phi_{P,H}\cdot L \not\Downarrow \bot$. Then there exist n and q such that $(n,q):L$ is a cache and $\Phi_{P,H}\cdot((n,q):L)\not\Downarrow\bot$.*

Proof. Define $f_0(n)$ to be i such that $q\models\neg R(n,i)$ if $(n,i)\in L$ [É]: $(n,q)\in L$? for some i ([É]: how do you know that such i exists?, and take $f_0(n)$ to be arbitrary if $n\notin I$.

Claim 1. *We have*

$$P(\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\,L\;n\;(\lambda\_.\,\mathsf{makeaxiom}\,n)))) \Downarrow \bot$$

Let $n\in\mathbb{N}$. Since $\mathsf{axiom}_n\models\bot$, $\lambda\_.\,\mathsf{makeaxiom}\,\overline{n}\models\neg R(n,i)$ for all i. Thus,

$$\mathsf{assoc}\,L\;\overline{n}\;(\lambda\_.\,\mathsf{makeaxiom}\,\overline{n})\models\neg R(n,f(n))$$

, by the previous discussion if n is not in L and by the definition of a cache if n is in L. This holds for all n, so

$$\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\,L\;n\;(\lambda\_.\,\mathsf{makeaxiom}\,n))\models\forall n.\neg R(n,f_0(n))$$

. Therefore,

$$\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\,L\;n\;(\lambda\_.\,\mathsf{makeaxiom}\,n)))\models\exists f.\forall n.\neg R(n,f(n))$$

. This concludes since $P\models\neg\exists f.\forall n.\neg R(n,f(n))$.

Claim 2. $\Phi_{H,P,\Phi_{H,P},L} \not\Downarrow \bot$, *that is (unfolding* Φ*),*

$$P(\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\, L\ n\ (\lambda\_.(\lambda m.H\ m\ (\lambda q.\Phi_{H,P}\ ((m,q):L))n))))) \not\Downarrow \bot$$

Proof. Suppose for the sake of contradiction $\Phi_{H,P,\Phi_{H,P},L} \Downarrow \bot$, that is, $\Phi_{H,P,\Phi_{H,P},L} \downarrow \mathsf{axiom}_k$ for some k. So $(\lambda\phi L'.\Phi_{H,P,\phi L'})\cdot\Phi_{H,P}\cdot L \downarrow \mathsf{axiom}_k$, and since $\Phi_{H,P} = \mathsf{fix}_{\lambda\phi L'.\Phi_{H,P,\phi,L'}}$, we have $\Phi_{H,P}\cdot L \downarrow \mathsf{axiom}_k$, which was supposed to not hold. $\square$

Claim 3. *There exists* $k_{new} \in \mathbb{N}$ *such that*

$$\mathsf{elim}_\forall\, H\ \overline{k_{new}}\ (\lambda q.\Phi_{H,P}((\overline{k_{new}},q):L)) \not\Downarrow \bot$$

Proof. Suppose that $\mathsf{elim}_\forall\, H\ \overline{k}\ (\lambda q.\Phi_{H,P}((\overline{k},q):L)) \downarrow \mathsf{axiom}_l$ for some l for all k. It now suffices to find a contradiction. We have $c\cdot\overline{k} \downarrow \mathsf{axiom}_l$ for some l for all k, where $c := \lambda m.\,\mathsf{elim}_\forall\, H\ m\ (\lambda q.\Phi_{H,P}((m,q):L))$. Thus, c behaves like $\mathsf{makeaxiom}$, so by applying **??** to **??** contradicts **??**. $\square$

Now, let k be as in **??**.

Claim 4. *There exists a* $q \in C$ *such that* $q \models \neg R(k,i)$ *for some* i*.*

Proof. Suppose, for the sake of contradiction, that there is no such q. Since $H \models \forall k.\nabla\exists i.\neg R(n,i)$, it follows from **??** that $\lambda q.\Phi_{H,P}((k,q):L)$ does not realize $\neg\exists i.\neg R(k,i)$. Thus, there exists at least one realizer of $\exists i.\neg R(k,i)$ (since otherwise any code would realize its negation). Now, $\exists i\neg R(k,i) \xrightarrow{\mathsf{elim}_\exists} \neg R(k,i)$ for some i, which concludes that there exists a realizer of $\neg R(k,i)$ for some i. $\square$

Claim 5. k *is not in* L*.*

Proof. For all n, $\mathsf{assoc}\, L\ \overline{n}\ (\lambda m.\,\mathsf{if}\ \ \mathsf{mem}\, m\ L\ \ \mathsf{then}\ \ (\lambda x.x)\ \ \mathsf{else}\ \ \mathsf{makeaxiom}\, m)$ realizes $\neg R(n, f_0(n))$, by the definition of a cache if n is in L and since it the else branch is taken ortherwise. Thus,

$$P(\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\, L\, n\ (\lambda m.\,\mathsf{if}\ \ \mathsf{mem}\, m\ L\ \mathsf{then}\ (\lambda x.x)\ \mathsf{else}\ \mathsf{makeaxiom}\, n)))) \downarrow \mathsf{axiom}$$

since it realizes $\bot$. Then, since for all n, $\mathsf{assoc}\, L\ n\ \ \mathsf{makeaxiom} \models \bot$, we have

$$P(\mathsf{intro}_\exists(\mathsf{intro}_\forall(\lambda n.\,\mathsf{assoc}\, L\ n\ (\lambda m.\,\mathsf{makeaxiom})))) \downarrow \mathsf{axiom}$$

Then, by **??** and **??**,

$$\mathsf{if}\ \ (\mathsf{mem}\,\overline{k}\ L)\ \ \mathsf{then}\ \ (\lambda x.x)\ \ \mathsf{else}\ \ \mathsf{makeaxiom}\,\overline{k} \Downarrow \bot$$

Thus, k is not in L since $\lambda x.x \not\models \bot$. $\square$

We are finished by taking $n = k_{new}$ from **??** and q from **??**. $\square$