A Constructive Proof of Dependent Choice, Compatible with Classical Logic

Hugo Herbelin

LICS 2012

27 June 2012

Computing with the axiom of dependent choice

Axiom of dependent choice (a key axiom of real analysis):

$$DC : \forall x^A \exists y^A \, P(x,y) \;\Rightarrow\; \forall x_0 \, \exists f^{A \Rightarrow A} \, (f(0) = x_0 \wedge \forall n \, P(f(n), f(n+1)))$$

- Directly realisable in intuitionistic logic as an instance of full (intensional) axiom of choice
- Provable in intuitionistic logic thanks to Martin-Löf's strong existential elimination
- "Dialectica" interpretation in classical logic using bar recursion by Spector [1962]
- Modified realisability interpretation in the negative translation of classical logic by Berardi-Bezem-Coquand [1998] then Berger-Oliva [2005]
- Classical realisability interpretation by Krivine [2002]
- Our approach: the *proof-as-program* correspondence
  - use Martin-Löf's strong existential elimination but constrain it so as to be compatible with classical logic
  - turn countable universal quantification into an infinite conjunction and evaluate its proofs lazily

Proving full (intensional) choice in intuitionistic logic

Using Martin-Löf's strong elimination of existential (i.e. strong sums, or Σ-types)

$$\frac{\Gamma \vdash p : \exists x^T\, A(x)}{\Gamma \vdash \mathtt{prf}\, p : A(\mathtt{wit}\, p)}$$

the (intensional) axiom of choice gets provable:

$$\begin{array}{rcl} AC_{A,B} & \triangleq & \lambda H.(\lambda x.\mathtt{wit}\,(H\,x), \lambda x.\mathtt{prf}\,(H\,x)) \\ & : & \forall x^A \exists y^B\, P(x,y) \Rightarrow \exists f^{A \Rightarrow B}\, \forall x^A\, P(x, f(x)) \end{array}$$

Unrestricted strong elimination of existential is computationally incompatible with classical logic

Consider computational classical logic:

$$\frac{\Gamma, \alpha : A^{\perp\!\!\!\perp} \vdash p : A}{\Gamma \vdash \texttt{catch}_\alpha\, p : A} \qquad \frac{\Gamma \vdash p : A \qquad (\alpha : A^{\perp\!\!\!\perp}) \in \Gamma}{\Gamma \vdash \texttt{throw}_\alpha\, p : C}$$

Example, Drinker Paradox:

$$\begin{array}{rcl} \mathsf{DP} & \triangleq & \texttt{catch}_\alpha.(x_0, \lambda y.\lambda H_x.\texttt{catch}_\beta \texttt{throw}_\alpha(y, \lambda y'.\lambda H_y.\texttt{throw}_\beta H_y)) \\ & : & \exists x\, \forall y\, (P(x) \Rightarrow P(y)) \end{array}$$

Such a proof backtracks on the choice of a witness... how to interpret $\texttt{wit}\,\mathsf{DP}$ in such a way that $\texttt{prf}\,\mathsf{DP} : \forall y\, (P(\texttt{wit}\,\mathsf{DP}) \Rightarrow P(y))$?

Unrestricted strong elimination of existential is computationally incompatible with classical logic

In particular, in the proof of choice,

$$\begin{array}{rcl} AC_{A,B} & \triangleq & \lambda H.(\lambda x.\texttt{wit}\,(H\,x), \lambda x.\texttt{prf}\,(H\,x)) \\ & : & \forall x^A \exists y^B\, P(x,y) \Rightarrow \exists f^{A\Rightarrow B}\, \forall x^A\, P(x, f(x)) \end{array}$$

if $Hx : \exists y\, P(x,y)$ is classically proved then what $\texttt{wit}\,(H\,x)$ should be is unclear, and how to keep it "synchronised" with $\texttt{prf}\,(H\,x)$ is even more unclear.

A trick to recover *countable* choice

Turn $\forall x \exists y\, P(x,y)$ into a infinite conjunction $\exists y\, P(0,y) \wedge \exists y\, P(1,y) \wedge\ \ldots$ and prove instead

$$\begin{array}{rcl} AC'_{\mathbb{N},B} & \triangleq & \lambda H.(\lambda n.\mathtt{wit}\,(\mathtt{nth}\,n\,H), \lambda n.\mathtt{prf}\,(\mathtt{nth}\,n\,H)) \\ & : & (\exists y\, P(0,y) \wedge \exists y\, P(1,y) \wedge\ \ldots) \Rightarrow \exists f^{A \Rightarrow B}\, \forall x^A\, P(x, f(x)) \end{array}$$

Now, the infinite conjunction is a "positive" object and we just have to evaluate it in (lazy) call-by-value order to ensure that at the time `wit` and `prf` are called, the underlying stream is evaluated at this position.

A dependent classical arithmetic in finite types: dPA^ω

We are now going to incrementally define a dependent classical arithmetic in finite types by extending (a type-theoretic presentation) of HA^ω with

- native coinductive formulae

- classical logic

- a restriction of strong elimination of existential compatible with classical logic

The underlying intuitionistic arithmetic in finite types: HA^ω

(the language of expressions: system T)

$$
\begin{array}{lcl}
T, U & ::= & \mathbb{N} \mid T \Rightarrow U \\
t, u & ::= & x \mid 0 \mid S(t) \mid \texttt{rec}\ t\ \texttt{of}\ [t \mid (x,y).t] \mid \lambda x.t \mid t\,t
\end{array}
$$

$$
\frac{(x:T) \in \Gamma}{\Gamma \vdash x : T} \qquad \frac{\Gamma, x : U \vdash t : T}{\Gamma \vdash \lambda x.t : U \Rightarrow T} \qquad \frac{\Gamma \vdash t : U \Rightarrow T \qquad \Gamma \vdash u : U}{\Gamma \vdash t\,u : T}
$$

$$
\frac{}{\Gamma \vdash 0 : \mathbb{N}} \qquad \frac{\Gamma \vdash t : \mathbb{N}}{\Gamma \vdash S(t) : \mathbb{N}} \qquad \frac{\Gamma \vdash t : \mathbb{N} \qquad \Gamma \vdash t_0 : U \qquad \Gamma, x : \mathbb{N}, y : U \vdash t_S : U}{\Gamma \vdash \texttt{rec}\ t\ \texttt{of}\ [t_0 \mid (x,y).t_S] : U}
$$

$$
\begin{array}{lcl}
(\lambda x.t)\,u & \equiv & t[x \leftarrow u] \\
\texttt{rec}\ 0\ \texttt{of}\ [t_0 \mid (x,y).t_S] & \equiv & t_0 \\
\texttt{rec}\ S(t)\ \texttt{of}\ [t_0 \mid (x,y).t_S] & \equiv & t_S[x \leftarrow t][y \leftarrow \texttt{rec}\ t\ \texttt{of}\ [t_0 \mid (x,y).t_S]]
\end{array}
$$

The underlying intuitionistic arithmetic in finite types: HA^ω

(formulae and equational theory)

$$A, B \ ::= \ t = u \mid \top \mid \bot \mid A \Rightarrow B \mid A \wedge B \mid A \vee B \mid \forall x^T A \mid \exists x^T A$$

$$\begin{array}{ll} 0 = 0 & \equiv \ \top \\ 0 = S(u) & \equiv \ \bot \\ S(t) = 0 & \equiv \ \bot \\ S(t) = S(u) & \equiv \ t = u \end{array}$$

The underlying intuitionistic arithmetic in finite types: HA^ω

(inference rules)

$$\frac{(a:A)\in\Gamma}{\Gamma\vdash a:A}\qquad\frac{\Gamma\vdash p:A\qquad\Gamma,a:A\vdash q:B}{\Gamma\vdash \texttt{let}\ a=p\ \texttt{in}\ q:B}\qquad\frac{\Gamma\vdash p:A\qquad A\equiv B}{\Gamma\vdash p:B}\qquad\frac{}{\Gamma\vdash ():\top}\qquad\frac{\Gamma\vdash p:\bot}{\Gamma\vdash \texttt{exfalso}\ p:C}$$

$$\frac{\Gamma\vdash p_1:A_1\qquad\Gamma\vdash p_2:A_2}{\Gamma\vdash (p_1,p_2):A_1\wedge A_2}\qquad\frac{\Gamma\vdash p:A_1\wedge A_2\qquad\Gamma,a_1:A_1,a_2:A_2\vdash q:B}{\Gamma\vdash \texttt{split}\ p\ \texttt{as}\ (a_1,a_2)\ \texttt{in}\ q:B}$$

$$\frac{\Gamma\vdash p:A_i}{\Gamma\vdash \iota_i(p):A_1\vee A_2}\qquad\frac{\Gamma\vdash p:A_1\vee A_2\qquad\Gamma,a_1:A_1\vdash p_1:B\qquad\Gamma,a_2:A_2\vdash p_2:B}{\Gamma\vdash \texttt{case}\ p\ \texttt{of}\ [a_1.p_1\,|\,a_2.p_2]:B}$$

$$\frac{\Gamma,a:A\vdash p:B}{\Gamma\vdash\lambda a.p:A\Rightarrow B}\qquad\frac{\Gamma\vdash p:A\Rightarrow B\quad\Gamma\vdash q:A}{\Gamma\vdash p\,q:B}\qquad\frac{\Gamma,x:T\vdash p:A(x)}{\Gamma\vdash\lambda x.p:\forall x^T\,A(x)}\qquad\frac{\Gamma\vdash p:\forall x^T\,A(x)\qquad\Gamma\vdash t:T}{\Gamma\vdash pt:A(t)}$$

$$\frac{\Gamma\vdash p:A(t)\qquad\Gamma\vdash t:T}{\Gamma\vdash (t,p):\exists x^T\,A(x)}\qquad\frac{\Gamma\vdash p:\exists x^T\,A(x)\qquad\Gamma,x:T,a:A(x)\vdash q:B}{\Gamma\vdash \texttt{dest}\ p\ \texttt{as}\ (x,a)\ \texttt{in}\ q:B}$$

$$\frac{t\equiv u}{\Gamma\vdash \texttt{refl}:t=u}\qquad\frac{\Gamma\vdash p:t=u\qquad\Gamma\vdash q:P(t)}{\Gamma\vdash \texttt{subst}\,p\,q:P(u)}\qquad\frac{\Gamma\vdash t:\mathbb{N}\qquad\Gamma\vdash p:P(0)\qquad\Gamma,x:T,a:P(x)\vdash q:P(S(x))}{\Gamma\vdash \texttt{ind}\ t\ \texttt{of}\ [p\,|\,(x,a).q]:P(t)}$$

The underlying intuitionistic arithmetic in finite types: HA^ω

(call-by-value evaluation semantics, minimal part)

$$
\begin{array}{lcl}
(\lambda a.q)\,p & \to & \mathtt{let}\ a = p\ \mathtt{in}\ q \\
(\lambda x.p)\,t & \to & p[x \leftarrow t] \\
\mathtt{case}\ \iota_i(p)\ \mathtt{of}\ [a_1.p_1 \,|\, a_2.p_2] & \to & \mathtt{let}\ a_i = p\ \mathtt{in}\ p_i \\
\mathtt{dest}\ (t,p)\ \mathtt{as}\ (x,a)\ \mathtt{in}\ q & \to & \mathtt{let}\ a = p\ \mathtt{in}\ q[x \leftarrow t] \\
\mathtt{split}\ (p_1,p_2)\ \mathtt{as}\ (a_1,a_2)\ \mathtt{in}\ q & \to & \mathtt{let}\ a_1 = p_1\ \mathtt{in}\ \mathtt{let}\ a_2 = p_2\ \mathtt{in}\ q \\
\mathtt{let}\ a = b\ \mathtt{in}\ q & \to & q[a \leftarrow b] \\
\mathtt{let}\ a = \lambda b.q\ \mathtt{in}\ q & \to & q[a \leftarrow \lambda b.q] \\
\mathtt{let}\ a = \lambda x.p\ \mathtt{in}\ q & \to & q[a \leftarrow \lambda x.t] \\
\mathtt{let}\ a = ()\ \mathtt{in}\ q & \to & q[a \leftarrow ()] \\
\mathtt{let}\ a = \iota_i(p)\ \mathtt{in}\ q & \to & \mathtt{let}\ b = p\ \mathtt{in}\ q[a \leftarrow \iota_i(b)] \\
\mathtt{let}\ a = (t,p)\ \mathtt{in}\ q & \to & \mathtt{let}\ b = p\ \mathtt{in}\ q[a \leftarrow (t,b)] \\
\mathtt{let}\ a = (p_1,p_2)\ \mathtt{in}\ q & \to & \mathtt{let}\ a_1 = p_1\ \mathtt{in}\ \mathtt{let}\ a_2 = p_2\ \mathtt{in}\ q[a \leftarrow (a_1,a_2)] \\
\mathtt{subst}\ \mathtt{refl}\ p & \to & p \\
\mathtt{ind}\ 0\ \mathtt{of}\ [p \,|\, (x,a).q] & \to & p \\
\mathtt{ind}\ S(t)\ \mathtt{of}\ [p \,|\, (x,a).q] & \to & q[x \leftarrow t][a \leftarrow \mathtt{ind}\ t\ \mathtt{of}\ [p \,|\, (x,a).q]]
\end{array}
$$

The underlying intuitionistic arithmetic in finite types: HA^ω

(call-by-value evaluation semantics, intuitionistic part)

$$\begin{array}{lcl} F[\texttt{exfalso}\ p] & \to & \texttt{exfalso}\ p \\ \texttt{exfalso}\ \texttt{exfalso}\ p & \to & \texttt{exfalso}\ p \end{array}$$

where elementary evaluation contexts are defined by

$$\begin{array}{lll} F[\,] & ::= & \iota_i([\,]) \mid ([\,], p) \mid (V, [\,]) \mid (t, [\,]) \\ & \mid & \texttt{case}\ [\,]\ \texttt{of}\ [a_1.p_1 \mid a_2.p_2] \mid \texttt{split}\ [\,]\ \texttt{as}\ (a_1, a_2)\ \texttt{in}\ q \mid \texttt{subst}\,[\,]\,p \\ & \mid & \texttt{dest}\ [\,]\ \texttt{as}\ (x, a)\ \texttt{in}\ p \mid [\,]\,q \mid [\,]\,t \mid \texttt{let}\ a = [\,]\ \texttt{in}\ q \end{array}$$

HA^ω has coinductive formulae

For instance, the infinite conjunction $P(0) \wedge P(1) \wedge \ldots$ can be represented by

$$\exists f^{\mathbb{N} \Rightarrow \mathbb{N}} \, (f(0) = 1 \wedge \forall n \, (f(n) = 1 \Rightarrow (P(n) \wedge f(S(n)) = 1))$$

(standard second order encoding, using quantification over functions rather than on predicates)

For convenience, add primitive cofixpoints to HA^ω

$$\frac{\Gamma, f : T \Rightarrow \mathbb{N}, x : T, b : f(x) = 1 \vdash p : A \qquad f(\_) = 1 \text{ possibly occurs in positive } A}{\Gamma \vdash \texttt{cofix}^t_{bx} p : \nu^t_{fx} A}$$

with equation

$$\nu^t_{fx} A \;\equiv\; A[x \leftarrow t][f(y) = 1 \leftarrow \nu^y_{fx} A]$$

For instance, $\nu^3_{fx}(P(x) \wedge f(S(x)) = 1)$ represents $P(3) \wedge P(4) \wedge \ \ldots$

Extend evaluation semantics of HA^ω to cofixpoints

(unfolding of cofixpoints is by need)

$$
\begin{array}{lcl}
\texttt{case cofix}^t_{bx}p\ \texttt{of}\ [a_1.p_1 \,|\, a_2.p_2] & \to & \texttt{let}\ c = \texttt{cofix}^t_{bx}p\ \texttt{in case}\ c\ \texttt{of}\ [a_1.p_1 \,|\, a_2.p_2] \\
\texttt{dest cofix}^t_{bx}p\ \texttt{as}\ (x,a)\ \texttt{in}\ q & \to & \texttt{let}\ c = \texttt{cofix}^t_{bx}p\ \texttt{in dest}\ c\ \texttt{as}\ (x,a)\ \texttt{in}\ q \\
\texttt{split cofix}^t_{bx}p\ \texttt{as}\ (a_1,a_2)\ \texttt{in}\ q & \to & \texttt{let}\ c = \texttt{cofix}^t_{bx}p\ \texttt{in split}\ c\ \texttt{as}\ (a_1,a_2)\ \texttt{in}\ q \\
\texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in exfalso}\ q & \to & \texttt{exfalso let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ q \\
F[\texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ q] & \to & \texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ F[q] \\
\texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ D[\texttt{case}\ a\ \texttt{of}\ [a_1.p_1 \,|\, a_2.p_2]] & \to & \\
\multicolumn{3}{r}{\texttt{let}\ a = p[b \leftarrow \lambda y.\texttt{cofix}^y_{bx}p][x \leftarrow t]\ \texttt{in}\ D[\texttt{case}\ a\ \texttt{of}\ [a_1.p_1 \,|\, a_2.p_2]]} \\
\texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ D[\texttt{split}\ a\ \texttt{as}\ (a_1,a_2)\ \texttt{in}\ q] & \to & \\
\multicolumn{3}{r}{\texttt{let}\ a = p[b \leftarrow \lambda y.\texttt{cofix}^y_{bx}p][x \leftarrow t]\ \texttt{in}\ D[\texttt{split}\ a\ \texttt{as}\ (a_1,a_2)\ \texttt{in}\ q]} \\
\texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ D[\texttt{dest}\ a\ \texttt{as}\ (x,a')\ \texttt{in}\ q] & \to & \\
\multicolumn{3}{r}{\texttt{let}\ a = p[b \leftarrow \lambda y.\texttt{cofix}^y_{bx}p][x \leftarrow t]\ \texttt{in}\ D[\texttt{dest}\ a\ \texttt{as}\ (x,a')\ \texttt{in}\ q]}
\end{array}
$$

where

$$D[\,]\ ::=\ [\,] \mid D[F[\,]] \mid \texttt{let}\ a = \texttt{cofix}^t_{bx}p\ \texttt{in}\ D[\,]$$

Extension to a classical arithmetic in finite types: PA^ω

$$\frac{\Gamma, \alpha : A^{\perp\!\!\!\perp} \vdash p : A}{\Gamma \vdash \mathtt{catch}_\alpha\, p : A} \qquad \frac{\Gamma \vdash p : A \qquad (\alpha : A^{\perp\!\!\!\perp}) \in \Gamma}{\Gamma \vdash \mathtt{throw}_\alpha\, p : C}$$

Classical arithmetic in finite types: PA^ω

(call-by-value evaluation semantics, classical part)

$$
\begin{array}{lcl}
F[\mathtt{throw}_\alpha\, p] & \to & \mathtt{throw}_\alpha\, p \\
F[\mathtt{catch}_\alpha\, p] & \to & \mathtt{catch}_\alpha F[p[\alpha \leftarrow F]] \\
\mathtt{exfalso}\ \mathtt{throw}_\beta\, p & \to & \mathtt{throw}_\beta\, p \\
\mathtt{exfalso}\ \mathtt{catch}_\beta\, p & \to & \mathtt{exfalso}\ p[\alpha \leftarrow \mathtt{exfalso}\ [\,]] \\
\mathtt{throw}_\beta\, \mathtt{exfalso}\ p & \to & \mathtt{exfalso}\ p \\
\mathtt{throw}_\beta\, \mathtt{throw}_\alpha\, p & \to & \mathtt{throw}_\alpha\, p \\
\mathtt{throw}_\beta\, \mathtt{catch}_\alpha\, p & \to & \mathtt{throw}_\beta\, p[\alpha \leftarrow \beta] \\
\mathtt{catch}_\alpha\, \mathtt{throw}_\alpha\, p & \to & \mathtt{catch}_\alpha\, p \\
\mathtt{catch}_\beta\, \mathtt{catch}_\alpha\, p & \to & \mathtt{catch}_\beta\, p[\alpha \leftarrow \beta] \\
\mathtt{let}\ a = \mathtt{cofix}^t_{bx} p\ \mathtt{in}\ \mathtt{throw}_\alpha\, q & \to & \mathtt{throw}_\alpha\, \mathtt{let}\ a = \mathtt{cofix}^t_{bx} p\ \mathtt{in}\ q \\
\mathtt{let}\ a = \mathtt{cofix}^t_{bx} p\ \mathtt{in}\ \mathtt{catch}_\alpha\, q & \to & \mathtt{catch}_\alpha\, \mathtt{let}\ a = \mathtt{cofix}^t_{bx} p\ \mathtt{in}\ q
\end{array}
$$

dPA^ω: Adding (restricted) strong elimination of existential to PA^ω

Replace weak elimination of existential by

$$\frac{\Gamma \vdash p : \exists x^T\, A(x) \qquad p \text{ is N-elimination-free}}{\Gamma \vdash \mathtt{prf}\, p : A(\mathtt{wit}\, p)}$$

where

- a value is N-elimination-free
- if p, q, p_1 and p_2 is N-elimination-free then $\mathtt{prf}\, p$, $\mathtt{ind}\ t\ \mathtt{of}\ [p_1 \,|\, (x,a).p_2]$, $\mathtt{case}\ a\ \mathtt{of}\ [a_1.p_1 \,|\, a_2.p_2]$, $\mathtt{dest}\ q\ \mathtt{as}\ (x,a)\ \mathtt{in}\ p$ and $\mathtt{split}\ q\ \mathtt{as}\ (a_1,a_2)\ \mathtt{in}\ p$ are N-elimination-free.

Dependent choice is now provable!

$$
\begin{array}{lll}
DC & \triangleq & \lambda a.\lambda x_0.\mathtt{let}\ b = \mathtt{s}\,a\,x_0\ \mathtt{in} \\
 & & (\lambda n.\mathtt{wit}\,(\mathtt{nth}_D\,n\,(x_0,b)), \\
 & & \ (\mathtt{refl}, \lambda n.\pi_1(\mathtt{prf}\,(\mathtt{prf}\,(\mathtt{nth}_D\,n\,(x_0,b)))))) \\
 & : & \forall x \exists y\, P(x,y) \Rightarrow \\
 & & \forall x_0\, \exists f\, (f(0) = x_0 \wedge \forall n\, P(f(n), f(S(n))))
\end{array}
$$

where

$$
\begin{array}{lll}
\mathtt{nth}_D\,n & : & \exists x\, R_D(x) \Rightarrow \exists x\, R_D(x) \\
\mathtt{nth}_D\,n & \triangleq & \lambda b.\mathtt{ind}\ n\ \mathtt{of}\ [b \,|\, (m,c).\mathtt{dest}\ c\ \mathtt{as}\ (x,d)\ \mathtt{in} \\
 & & \qquad\qquad (\mathtt{wit}\,(\mathtt{prf}\,d), \pi_2(\mathtt{prf}\,(\mathtt{prf}\,d)))] \\
\mathtt{s}\,a\,x & : & R_D(x) \\
\mathtt{s}\,a\,x & \triangleq & \mathtt{cofix}^{x}_{bn}(\mathtt{dest}\ a\,n\ \mathtt{as}\ (y,c)\ \mathtt{in}\ (y,(c,by)))
\end{array}
$$

(s is a stream of type $R_D(x_0) \triangleq \exists x_1\, (P(x_0,x_1) \wedge \exists x_2\, (P(x_1,x_2) \wedge\ \ldots))$ obtained by recursively applying the hypothesis)

Conjecture: dPA^{ω} exactly captures the strength of dependent choice.

A proof of countable choice

$$
\begin{array}{lcl}
AC_{\mathbb{N}} & \triangleq & \lambda a.\texttt{let}\ b = \texttt{cofix}^{0}_{bn}(a\,n, b(Sn))\ \texttt{in} \\
 & & (\lambda n.\texttt{wit}\,(\texttt{nth}_C\,n\,b), \lambda n.\texttt{prf}\,(\texttt{nth}_C\,n\,b)) \\
 & : & \forall n \exists y\, P(n,y) \Rightarrow \exists f\, \forall n\, P(n, f(n))
\end{array}
$$

where

$$
\begin{array}{lcl}
\texttt{nth}_C\,n & : & R_C(0) \Rightarrow R_C(n) \\
\texttt{nth}_C\,n & \triangleq & \lambda b.\pi_1(\texttt{ind}\ n\ \texttt{of}\ [b \,|\, (m,c).\pi_2(c)])
\end{array}
$$

(s is the stream of type $R_C(0) \triangleq \exists y\, P(0,y) \wedge \exists y\, P(1,y) \wedge\ \ldots$ extracted from the hypothesis)

Conjecture: one exactly captures the strength of countable choice if we remove the `prf` case from the definition of N-elimination-free.

Properties of dPA^ω

Subject reduction: if $\Gamma \vdash p : A$ and $p \rightarrow q$ then $\Gamma \vdash q : A$

(Claimed) Normalisation: if $\Gamma \vdash p : A$ then p normalises

Progress: if $\vdash p : A$ and p not a value then p reduces

Evaluation: $\vdash p : A$ then $\vdash V : A$ for some V s.t. $P \xrightarrow{*} V$

Conservativity over HA^ω for closed $\forall$-$\Rightarrow$-ν-`wit`-free and Σ_1^0-formulae: if $\vdash T$ and T is $\forall$-$\Rightarrow$-ν-`wit`-free or Σ_1^0 then $\vdash_{HA^\omega} T$

Consistency: $\nvdash \bot$

Comparison with realisability-based approaches

Krivine's realiser only supports choice over predicates (i.e. A is of the form $B \Rightarrow \mathsf{Prop}$). It works by "quoting" the predicates so as to be able to well-order these and to select the minimal one along this order. Existence of a minimal element crucially needs classical logic.

As rephrased by Berger [2004], Coquand-Berardi-Bezem's realiser of countable choice [1998] builds a choice function by *update* induction. Initially, the choice function returns a dummy value everywhere. Each time a proof of $P(n, f(n))$ is requested, the proof of $\exists y\, P(n, y)$ together with a continuation that updates the choice function. If, later on, the proof of some $P(n, f(n))$ has already been asked, the former value is retrieved.

As rephrased by Escardó and Oliva [2010], Spector's realiser can be seen as the computation of a controlled product of selection functions, with default value assigned beyond the point the construction stops being under control.

In our case, no choice function is actually constructed. Only approximations are computed and there is no need to give default values.

Summary

By adding an appropriate intuitionistically-restricted rule for strong elimination of existential to PA^ω, we computationally capture the strength of either countable choice or dependent choice.

This can be turned into a Martin-Löf-style type theory by allowing dependent products with the restriction that they are instantiated only by N-elimination-free expressions (as done in the paper).

Provides with an intuitionistic proof of (a weak form of) bar induction compatible with classical logic:

$$\forall f\, \exists n\, B(f_{|n}) \Rightarrow \forall g \begin{pmatrix} \forall l\, (B(l) \Rightarrow g(l) = 0) \wedge \\ \forall l\, (\forall x\, g(l \star x) = 0 \Rightarrow g(l) = 0) \end{pmatrix} \Rightarrow g(\langle\rangle) = 0$$

Our proofs use a weak form of effect (lazy evaluation) and this suggests that a proof-theoretic investigation of classical call-by-need λ-calculus is worth being conducted...