Bar recursion over finite partial functions

Thomas Powell
(joint work with Paulo Oliva)

University of Innsbruck

CCC 2015
Kochel am See, Germany
15 September 2015

Outline

Spector's bar recursion $\mathsf{BR}^{g,h,\varphi}\colon \rho^* \to \sigma$ is defined by

$$\mathsf{BR}^{g,h,\varphi}(s) =_\sigma \begin{cases} g(s) & \text{if } \varphi(\hat{s}) < \mathrm{len}(s) \\ h_s(\lambda x \, . \, \mathsf{BR}^{g,h,\varphi}(s * x)) & \text{otherwise} \end{cases}$$

Spector's bar recursion $\mathsf{BR}^{g,h,\varphi} \colon \rho^* \to \sigma$ is defined by

$$\mathsf{BR}^{g,h,\varphi}(s) =_\sigma \begin{cases} g(s) & \text{if } \varphi(\hat{s}) < \mathrm{len}(s) \\ h_s(\lambda x \, . \, \mathsf{BR}^{g,h,\varphi}(s * x)) & \text{otherwise} \end{cases}$$

Recursion

- input $s \colon X^*$ a finite sequence;
- recursive calls made over all one-element extensions $s * x$ of s;
- g assigns base values to the recursion.

Spector's bar recursion $\mathsf{BR}^{g,h,\varphi}\colon \rho^* \to \sigma$ is defined by

$$\mathsf{BR}^{g,h,\varphi}(s) =_\sigma \begin{cases} g(s) & \text{if } \varphi(\hat{s}) < \text{len}(s) \\ h_s(\lambda x \,.\, \mathsf{BR}^{g,h,\varphi}(s * x)) & \text{otherwise} \end{cases}$$

Recursion

- input $s\colon X^*$ a finite sequence;
- recursive calls made over all one-element extensions $s * x$ of s;
- g assigns base values to the recursion.

Termination

- $\hat{s} := \lambda k. \begin{cases} s(k) & \text{if } k < |s| \\ 0 & \text{otherwise} \end{cases}$ i.e. a canonical embedding of s into $\rho^{\mathbb{N}}$;
- $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$ controls the recursion, terminating it when *Spector's point* $\varphi(\hat{s})$ is less than the length of the input i.e. $\varphi(\hat{s}) < \text{len}(s)$ holds.

Bar recursion exists in continuous models by a standard argument:

Bar recursion exists in continuous models by a standard argument:

1. EXTEND. If $\mathsf{BR}(s_0) = \bot$ there exists an infinite sequence $s_0 \prec s_1 \prec s_2 \prec \ldots$ satisfying

$$\varphi(\hat{s}_i) \geq \mathrm{len}(s_i) \quad \text{and} \quad s_{i+1} = s_i * x_i \quad \text{and} \quad \mathsf{BR}(s_{i+1}) = \bot$$

Bar recursion exists in continuous models by a standard argument:

1. EXTEND. If $\mathsf{BR}(s_0) = \bot$ there exists an infinite sequence $s_0 \prec s_1 \prec s_2 \prec \dots$ satisfying

$$\varphi(\hat{s}_i) \geq \mathrm{len}(s_i) \quad \text{and} \quad s_{i+1} = s_i * x_i \quad \text{and} \quad \mathsf{BR}(s_{i+1}) = \bot$$

2. LIMIT. Let $\alpha \colon \rho^{\mathbb{N}}$ be the domain-theoretic limit of the s_i i.e.

$$\alpha := \bigsqcup_{i \in \mathbb{N}} s_i = \lambda k \, . \, s_{k+1}(k)$$

Bar recursion exists in continuous models by a standard argument:

1. Extend. If $\mathsf{BR}(s_0) = \bot$ there exists an infinite sequence $s_0 \prec s_1 \prec s_2 \prec \dots$ satisfying

$$\varphi(\hat{s}_i) \geq \mathrm{len}(s_i) \quad \text{and} \quad s_{i+1} = s_i * x_i \quad \text{and} \quad \mathsf{BR}(s_{i+1}) = \bot$$

2. Limit. Let $\alpha\colon \rho^{\mathbb{N}}$ be the domain-theoretic limit of the s_i i.e.

$$\alpha := \bigsqcup_{i \in \mathbb{N}} s_i = \lambda k \, . \, s_{k+1}(k)$$

3. Continuity. The value of $\varphi(\alpha)$ depends only on some finite initial segment $[\alpha(0), \dots, \alpha(N-1)]$ of its argument.

Take any $M \geq N, \varphi(\alpha) + 1$. Then

$$\varphi(\hat{s}_M) \underbrace{=}_{\text{continuity: } N \leq M} \varphi(\alpha) < \varphi(\alpha) + 1 \leq M \leq \mathrm{len}(s_M)$$

What is so useful about bar recursion? One answer: *self-reference*.

Suppose $Q(s,i)$ is some predicate on $\rho^* \times \mathbb{N}$, and that whenever

$$\forall k < \mathrm{len}(s)\ Q(s,k)$$

we can compute an extension $a_s\colon \rho$ such that

$$\forall k < \mathrm{len}(s * a_s)\ Q(s * a_s, k).$$

Then bar recursion allows us to compute a chain $[] \prec s_1 \prec s_2 \prec \ldots \prec s_M$ with

$$\forall k < \mathrm{len}(s_i)\ Q(s_i, k)$$

for each i, and moreover s_M is a leaf with $\varphi(\hat{s}_M) < \mathrm{len}(s_M)$ therefore we have

$$\boxed{Q(s_M, \varphi(\hat{s}_M))}$$

We will see why this in important in Part 2!

Symmetric bar recursion $\mathsf{sBR}^{g,h,\varphi}\colon \rho^\dagger \to \sigma$ is defined by

$$\mathsf{sBR}^{g,h,\varphi}(u) =_\sigma \begin{cases} g(u) & \boxed{\text{if } \varphi(\hat{u}) \in \mathrm{dom}(u)} \\ h_s(\lambda x \, . \, \mathsf{sBR}^{\phi,b,\varphi}(\boxed{u \oplus (\varphi(\hat{u}), x)})) & \text{otherwise} \end{cases}$$

Symmetric bar recursion $\mathsf{sBR}^{g,h,\varphi} \colon \rho^\dagger \to \sigma$ is defined by

$$\mathsf{sBR}^{g,h,\varphi}(u) =_\sigma \begin{cases} g(u) & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ h_s(\lambda x \, . \, \mathsf{sBR}^{\phi,b,\varphi}(u \oplus (\varphi(\hat{u}), x))) & \text{otherwise} \end{cases}$$

Recursion

- input $u \colon \rho^\dagger$ a finite partial function;
- recursive calls made over one-element domain-theoretic extensions $u \oplus (\varphi(\hat{u}), x)$ of u;
- g assigns base values to the recursion.

Symmetric bar recursion $\mathsf{sBR}^{g,h,\varphi}\colon \rho^\dagger \to \sigma$ is defined by

$$\mathsf{sBR}^{g,h,\varphi}(u) =_\sigma \begin{cases} g(u) & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ h_s(\lambda x \,.\, \mathsf{sBR}^{\phi,b,\varphi}(u \oplus (\varphi(\hat{u}), x))) & \text{otherwise} \end{cases}$$

Recursion

- input $u\colon \rho^\dagger$ a finite partial function;
- recursive calls made over one-element domain-theoretic extensions $u \oplus (\varphi(\hat{u}), x)$ of u;
- g assigns base values to the recursion.

Termination

- $\hat{u} := \lambda k. \begin{cases} u(k) & \text{if } k \in \mathrm{dom}(u) \\ 0 & \text{otherwise} \end{cases}$, a canonical embedding of u into $\rho^{\mathbb{N}}$;
- $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$ controls recursion, terminating when Spector's point is in the *domain* of u i.e. $\varphi(\hat{u}) \in \mathrm{dom}(u)$ holds.

Need to adapt standard argument:

Need to adapt standard argument:

1. EXTEND. If $\mathsf{sBR}(u_0) = \bot$ there exists an infinite sequence $u_0 \sqsubset u_1 \sqsubset u_2 \sqsubset \ldots$ satisfying

$$n_i := \varphi(\hat{u}_i) \notin \mathrm{dom}(u_i) \quad \text{and} \quad u_{i+1} = u_i \oplus (n_i, x_i) \quad \text{and} \quad \mathsf{sBR}(u_{i+1}) = \bot$$

Need to adapt standard argument:

1. EXTEND. If $\mathsf{sBR}(u_0) = \bot$ there exists an infinite sequence $u_0 \sqsubset u_1 \sqsubset u_2 \sqsubset \ldots$ satisfying

$$n_i := \varphi(\hat{u}_i) \notin \mathrm{dom}(u_i) \quad \text{and} \quad u_{i+1} = u_i \oplus (n_i, x_i) \quad \text{and} \quad \mathsf{sBR}(u_{i+1}) = \bot$$

2. LIMIT. Let $\alpha \colon \mathbb{N} \to \rho_\bot$ be the domain-theoretic limit of the u_i i.e.

$$\alpha := \bigsqcup_{i \in \mathbb{N}} u_i = \lambda k \,.\, \begin{cases} u_{i(k)}(k) & \text{where } i(k) \text{ least s.t. } k \in \mathrm{dom}(u_{i(k)}) \\ \text{undefined} & \text{if no such index exists.} \end{cases}$$

Let $\hat{\alpha} \colon \rho^{\mathbb{N}}$ denote the canonical extenion:

$$\hat{\alpha} = \lambda k \,.\, \begin{cases} u_{i(k)}(k) & \text{where } i(k) \text{ least s.t. } k \in \mathrm{dom}(u_{i(k)}) \\ 0_\rho & \text{if no such index exists.} \end{cases}$$

3. Continuity. The value of $\varphi(\hat{\alpha})$ depends only on some finite initial segment $[\hat{\alpha}(0), \ldots, \hat{\alpha}(N-1)]$ of its argument.

Take any $M \geq N, \varphi(\hat{\alpha}) + 1$. Since $\alpha = \bigsqcup u_i$ there exists some I such that

$$\forall i < M(u_I(i) = \alpha(i)), \text{ or equivalently, } \forall i < M(\hat{u}_I(i) = \hat{\alpha}(i))$$

which implies that

$$n_I := \varphi(\hat{u}_I) \underbrace{=}_{\text{continuity: } N \leq M} \varphi(\hat{\alpha}) < \varphi(\hat{\alpha}) + 1 \leq M.$$

Since $n_I \notin \mathrm{dom}(u_I)$ and $n_I < M$ we have $n_I \notin \mathrm{dom}(\alpha)$. But

$$u_{I+1} = u_I \oplus (n_I, x_I),$$

and since $u_{I+1} \sqsubset \alpha$ we have $n_I \in \mathrm{dom}(\alpha)$, a contradiction.

Therefore $n_I = \varphi(\hat{u}_I) \in \mathrm{dom}(u_I)$.

Summary: Two ways of achieving self-reference

Spector's bar recursion

$$\mathsf{BR}^{g,h,\varphi}(s) =_\sigma \begin{cases} g(s) & \text{if } \varphi(\hat{s}) < \text{len}(s) \\ h_s(\lambda x \,.\, \mathsf{BR}^{g,h,\varphi}(s * x)) & \text{otherwise} \end{cases}$$

makes recursive calls over the tree $s_0 \prec s_1 \prec s_2 \prec \dots$ until it reaches a leaf s_M such that $\varphi(\hat{s_M}) < \text{len}(s_M)$. This tree is well-founded in continuous models.

Symmetric bar recursion generalises this idea:

$$\mathsf{BR}^{g,h,\varphi}(u) =_\sigma \begin{cases} g(u) & \text{if } \varphi(\hat{u}) \in \text{dom}(u) \\ h_s(\lambda x \,.\, \mathsf{BR}^{\phi,b,\varphi}(u \oplus (\varphi(\hat{u}), x))) & \text{otherwise} \end{cases}$$

making recursive calls over the tree $u_0 \sqsubset u_1 \sqsubset u_2 \sqsubset \dots$ until it reaches a leaf u_M such that $\varphi(\hat{u}_M) \in \text{dom}(u_M)$. This tree is well-founded in continuous models.

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

THEOREM 1. BR is primitive recursively definable from sBR, provably in $\mathsf{E\text{-}HA}^\omega$ (extensional Heyting arithmetic in all finite types).

Proof. Straightforward.

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

THEOREM 1. BR is primitive recursively definable from sBR, provably in $\mathsf{E\text{-}HA}^\omega$ (extensional Heyting arithmetic in all finite types).

Proof. Straightforward.

THEOREM 2. sBR is primitive recursively definable from BR, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{DC}$.

Proof. Fairly complex. Need to move up a type level to define sBR.

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

THEOREM 1. BR is primitive recursively definable from sBR, provably in $\mathsf{E\text{-}HA}^\omega$ (extensional Heyting arithmetic in all finite types).

Proof. Straightforward.

THEOREM 2. sBR is primitive recursively definable from BR, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{DC}$.

Proof. Fairly complex. Need to move up a type level to define sBR.

COROLLARY 1. Both the Kleene-Kreisel continuous functionals $\mathcal{C}^\omega$ and the strongly majorizable functionals $\mathcal{M}^\omega$ are a model of sBR.

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

THEOREM 1. BR is primitive recursively definable from sBR, provably in $\mathsf{E\text{-}HA}^\omega$ (extensional Heyting arithmetic in all finite types).

Proof. Straightforward.

THEOREM 2. sBR is primitive recursively definable from BR, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{DC}$.

Proof. Fairly complex. Need to move up a type level to define sBR.

COROLLARY 1. Both the Kleene-Kreisel continuous functionals $\mathcal{C}^\omega$ and the strongly majorizable functionals $\mathcal{M}^\omega$ are a model of sBR.

COROLLARY 2. The tree $u_0 \sqsubset u_1 \sqsubset u_2 \ldots$ with leaves $u_i \in \mathrm{dom}(\hat{u}_i)$ is well-founded in any model of $\mathsf{E\text{-}HA}^\omega + \mathsf{sBR}$, including $\mathcal{C}^\omega$ and $\mathcal{M}^\omega$.

Some facts about symmetric bar recursion, taken from (Oliva/P. 2015).

THEOREM 1. BR is primitive recursively definable from sBR, provably in $\mathsf{E\text{-}HA}^\omega$ (extensional Heyting arithmetic in all finite types).

Proof. Straightforward.

THEOREM 2. sBR is primitive recursively definable from BR, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{DC}$.

Proof. Fairly complex. Need to move up a type level to define sBR.

COROLLARY 1. Both the Kleene-Kreisel continuous functionals $\mathcal{C}^\omega$ and the strongly majorizable functionals $\mathcal{M}^\omega$ are a model of sBR.

COROLLARY 2. The tree $u_0 \sqsubset u_1 \sqsubset u_2 \ldots$ with leaves $u_i \in \mathrm{dom}(\hat{u}_i)$ is well-founded in any model of $\mathsf{E\text{-}HA}^\omega + \mathsf{sBR}$, including $\mathcal{C}^\omega$ and $\mathcal{M}^\omega$.

COROLLARY 3. sBR is S1-S9 computable in $\mathcal{C}^\omega$, and thus strictly weaker than modified bar recursion/Gandy-Hyland Γ functional.

Outline

1. Backward recursion in the continuous functionals
2. The computational interpretation of countable choice
3. Backward recursion as a learning realizer

What is the computational meaning of a Π_3-theorem?

$$P :\equiv \forall a^\rho \exists x^\sigma \forall y^\tau A(a, x, y)$$

What is the computational meaning of a Π_3-theorem?

$$P :\equiv \forall a^\rho \exists x^\sigma \forall y^\tau A(a,x,y)$$

In general we cannot hope to produce a direct computable witness for $\exists x$. But suppose we double negate and Skolemize:

$$\begin{aligned} \neg P &\leftrightarrow \exists a \forall x \exists y \neg A(a,x,y) \\ &\leftrightarrow \exists a, p^{\sigma \to \tau} \forall x \neg A(a,x,p(x)) \\ \neg\neg P &\leftrightarrow \forall a, p \exists x \neg\neg A(a,x,p(x)) \\ &\leftrightarrow \forall a, p \exists x\ A(a,x,p(x)) \end{aligned}$$

What is the computational meaning of a Π_3-theorem?

$$P :\equiv \forall a^\rho \exists x^\sigma \forall y^\tau A(a,x,y)$$

In general we cannot hope to produce a direct computable witness for $\exists x$. But suppose we double negate and Skolemize:

$$\begin{aligned}\neg P &\leftrightarrow \exists a \forall x \exists y \neg A(a,x,y)\\ &\leftrightarrow \exists a, p^{\sigma\to\tau} \forall x \neg A(a,x,p(x))\\ \neg\neg P &\leftrightarrow \forall a, p \exists x \neg\neg A(a,x,p(x))\\ &\leftrightarrow \forall a, p \exists x\ A(a,x,p(x))\end{aligned}$$

We can typically extract some *indirect* computable witness

$$X \colon \rho \to (\sigma \to \tau) \to \sigma$$

for $\exists x$ in $\neg\neg P$, i.e.

$$\forall a, p\ A(a, X_{a,p}, p(X_{a,p})).$$

In the statement

$$\forall a \exists x \forall y A(a, x, y),$$

x is an ideal object which works for *all* y. On the other hand, in the statement

$$\forall a, p \exists x A(a, x, p(x))$$

x is a finitary approximation to ideal object, which works for just $p(x)$. The function p can be seen as determining the size, or ‘quality’, of this approximation.

$\boxed{I}$ There exists an ideal object x which works for all y.

$\boxed{I'}$ For arbitrary p, there is an approximation x to an ideal object which works for $p(x)$.

Over classical logic $\boxed{I} \leftrightarrow \boxed{I'}$, but $\boxed{I'}$ is intuitionistically weak enough to admit a computational interpretation.

Example

By the least element principle we can prove

$$P :\equiv \forall f \colon \mathbb{N} \to \mathbb{N}\ \exists x \in \mathbb{N}\ \forall y \in \mathbb{N}\ .\ f(x) \leq f(y).$$

However, there is no computable witness $F \colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ for $\exists x$.

Example

By the least element principle we can prove

$$P :\equiv \forall f\colon \mathbb{N} \to \mathbb{N}\ \exists x \in \mathbb{N}\ \forall y \in \mathbb{N}\ .\ f(x) \leq f(y).$$

However, there is no computable witness $F\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ for $\exists x$. But over classical logic P is equivalent to

$$\forall f, p\colon \mathbb{N} \to \mathbb{N}\ \exists x\ f(x) \leq f(p(x)).$$

and $\exists x$ must be witnessed for some $x \leq p^{(f(0))}(0)$, else we'd have

$$\underbrace{f(0) > f(p(0)) > f(p^{(2)}(0)) > \ldots > f(p^{(f(0))}(0)) > f(p^{(f(0)+1)}(0))}_{f(0)+1 \text{ times}} \geq 0$$

Example

By the least element principle we can prove

$$P :\equiv \forall f\colon \mathbb{N} \to \mathbb{N}\ \exists x \in \mathbb{N}\ \forall y \in \mathbb{N}\ .\ f(x) \leq f(y).$$

However, there is no computable witness $F\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ for $\exists x$. But over classical logic P is equivalent to

$$\forall f, p\colon \mathbb{N} \to \mathbb{N}\ \exists x\ f(x) \leq f(p(x)).$$

and $\exists x$ must be witnessed for some $x \leq p^{(f(0))}(0)$, else we'd have

$$\underbrace{f(0) > f(p(0)) > f(p^{(2)}(0)) > \ldots > f(p^{(f(0))}(0)) > f(p^{(f(0)+1)}(0))}_{f(0)+1 \text{ times}} \geq 0$$

Various choices of p yield e.g.

$\exists x\ \forall y \in [0, 1000000]\ .\ f(x) \leq f(y)$

$\exists x\ \forall y \in [2^x, 2^{2^{2^x}}]\ .\ f(x) \leq f(y)$

A well-known technique of extracting computable witnesses for negated theorems in this way is:

$$\mathcal{T}_{\text{class}} \vdash P \quad \underbrace{\Rightarrow}_{\text{negative translation}} \quad \mathcal{T}_{\text{int}} \vdash P^N \quad \underbrace{\Rightarrow}_{\text{Dialectica interpretation}} \quad \mathsf{T}_\lambda \vdash \forall y |P^N|^t_y$$

Theorem (Gödel, 1930s). If $\mathsf{PA} \vdash P$ then $\mathsf{T}_\lambda \vdash \forall y |P^N|^t_y$, where T_λ is the system of primitive recursive functionals in all finite types.

Corollary. If

$$\mathsf{PA} \vdash \forall a^\rho \exists x^\sigma \forall y^\tau A(a, x, y),$$

then there is a primitive recursive functional $X \colon \rho \to (\sigma \to \tau) \to \sigma$ satisfying

$$\forall a, p^{\sigma \to \tau} A(a, X_{a,p}, p(X_{a,p}))$$

and an algorithm for formally extracting such an X from the proof.

What is the computational content of the axiom of countable choice?

$$\mathsf{AC} : \forall n^{\mathbb{N}} \exists x^{\rho} \forall y^{\sigma} A(n, x, y) \to \exists f^{\mathbb{N} \to \rho} \forall n, y A(n, f(n), y).$$

What is the computational content of the axiom of countable choice?

$$\mathsf{AC}: \forall n^{\mathbb{N}} \exists x^{\rho} \forall y^{\sigma} A(n, x, y) \to \exists f^{\mathbb{N} \to \rho} \forall n, y A(n, f(n), y).$$

First, let's interpret the premise and conclusion seperately:

$$\forall n, p^{\rho \to \sigma} \exists x \; A(n, x, p(x)) \to \forall \varphi^{\rho^{\mathbb{N}} \to \mathbb{N}}, q^{\rho^{\mathbb{N}} \to \sigma} \exists f \; A(\varphi(f), f(\varphi(f)), q(f))$$

Premise: For each n there exists a finitary (pointwise) approximation x to the ideal object which works for $p(x)$.

Conclusion: There exists a finitary (global) approximation f to the ideal choice sequence which works for $q(f)$ at point $\varphi(f)$.

What is the computational content of the axiom of countable choice?

$$\mathsf{AC}\,:\;\forall n^{\mathbb{N}}\exists x^{\rho}\forall y^{\sigma}A(n,x,y)\rightarrow\exists f^{\mathbb{N}\rightarrow\rho}\forall n,yA(n,f(n),y).$$

First, let's interpret the premise and conclusion seperately:

$$\forall n,p^{\rho\rightarrow\sigma}\exists x\;A(n,x,p(x))\rightarrow\forall\varphi^{\rho^{\mathbb{N}}\rightarrow\mathbb{N}},q^{\rho^{\mathbb{N}}\rightarrow\sigma}\exists f\;A(\varphi(f),f(\varphi(f)),q(f))$$

Premise: For each n there exists a finitary (pointwise) approximation x to the ideal object which works for $p(x)$.

Conclusion: There exists a finitary (global) approximation f to the ideal choice sequence which works for $q(f)$ at point $\varphi(f)$.

$$\underbrace{\forall X^{\mathbb{N}\rightarrow(\rho\rightarrow\sigma)\rightarrow\rho}\forall\varphi,q\exists f}\;[\forall n,p\;A(n,X_{n,p},p(X_{n,p}))\rightarrow A(\varphi(f),f(\varphi(f)),q(f))].$$

Comp. Interpretation: For any pointwise realizer X of the premise of AC, and parameters φ, q, there is a global approximation f to a choice sequence in φ and q.

For an arbitrary sequence $s\colon \rho^*$ define an extension $s \preceq \mathsf{E}_s$ using bar recursion:

$$\mathsf{E}_s = \begin{cases} s & \text{if } \varphi(\hat{s}) < \mathrm{len}(s) \\ \mathsf{E}_{s * a_s} & \text{otherwise.} \end{cases}$$

where $a_s := X_{\mathrm{len}(s), \lambda x. q(\hat{\mathsf{E}}_{s*x})}$.

Suppose that $\hat{s}$ is an approximation to a choice sequence which works for $q(\hat{\mathsf{E}}_s)$ at all points $i < \mathrm{len}(s)$:

$$\mathrm{App}(s) \ : \ \forall i < \mathrm{len}(s)\ A(i, s(i), q(\hat{\mathsf{E}}_s))$$

but $\varphi(\hat{s}) \geq \mathrm{len}(s)$. Then since $A(\mathrm{len}(s), \overbrace{X_{\mathrm{len}(s), \lambda x. q(\hat{\mathsf{E}}_{s*x})}}^{a_s}, \overbrace{q(\hat{\mathsf{E}}_{s*a_s})}^{p(a_s)})$ holds we have $\mathsf{E}_s = \mathsf{E}_{s*a_s}$ and

$$\mathrm{App}(s) \Rightarrow \mathrm{App}(s * a_s)$$

i.e. we can build a better approximation $\widehat{s * a_s}$, which works for $q(\hat{\mathsf{E}}_{s*a_s})$ at all points $i < \mathrm{len}(s) + 1$.

If $\mathrm{App}(s_0)$ there exists a sequence $s_0 \prec s_1 \prec \ldots$ of progressively better approximations:

$$\varphi(\hat{s}_i) \geq \mathrm{len}(s_i) \quad \text{and} \quad s_{i+1} = s_i * a_{s_i} \quad \text{and} \quad \mathrm{App}(s_{i+1}) \quad \text{and} \quad \mathsf{E}_{s_i} = \mathsf{E}_{s_{i+1}}.$$

But at some point we reach a leaf $\varphi(\hat{s}_M) < \mathrm{len}(s_M)$, and then $\mathsf{E}_{s_M} = s_M$ and

$$\begin{aligned} \mathrm{App}(s_M) &\equiv \forall i < \mathrm{len}(s_M)\ A(i, s_M(i), q(\hat{\mathsf{E}}_{s_M})) \\ &\Rightarrow A(\varphi(\hat{s}_M), \hat{s}_M(\varphi(\hat{s}_M)), q(\hat{s}_M)). \end{aligned}$$

Thus $F_{X,\varphi,q} = \mathsf{E}_{s_0} = \ldots = \mathsf{E}_{s_M} = \hat{s}_M$ is a sufficiently good approximation.

If $\mathrm{App}(s_0)$ there exists a sequence $s_0 \prec s_1 \prec \ldots$ of progressively better approximations:

$$\varphi(\hat{s}_i) \geq \mathrm{len}(s_i) \quad \text{and} \quad s_{i+1} = s_i * a_{s_i} \quad \text{and} \quad \mathrm{App}(s_{i+1}) \quad \text{and} \quad \mathsf{E}_{s_i} = \mathsf{E}_{s_{i+1}}.$$

But at some point we reach a leaf $\varphi(\hat{s}_M) < \mathrm{len}(s_M)$, and then $\mathsf{E}_{s_M} = s_M$ and

$$\begin{aligned} \mathrm{App}(s_M) &\equiv \forall i < \mathrm{len}(s_M)\ A(i, s_M(i), q(\hat{\mathsf{E}}_{s_M})) \\ &\Rightarrow A(\varphi(\hat{s}_M), \hat{s}_M(\varphi(\hat{s}_M)), q(\hat{s}_M)). \end{aligned}$$

Thus $F_{X,\varphi,q} = \mathsf{E}_{s_0} = \ldots = \mathsf{E}_{s_M} = \hat{s}_M$ is a sufficiently good approximation.

Theorem. $\hat{\mathsf{E}}_{[]}$ is a sufficiently good approximation to a choice sequence.

Corollary (Spector 1962). If $\mathsf{PA} + \mathsf{AC} \vdash P$ then $\mathsf{T}_{\lambda_{\mathrm{BR}}} \vdash \forall y |P^N|^t_y$, where $\mathsf{T}_{\lambda_{\mathrm{BR}}}$ is the system of primitive recursive functionals in all finite types together with Spector's bar recursion.

For an arbitrary partial function $u\colon \rho^\dagger$ define an extension $u \sqsubset \mathsf{U}_u$ as:

$$\mathsf{U}_u = \begin{cases} u & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ \mathsf{U}_{s*(n_u,a_u)} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$ and $a_u := X_{n_u, \lambda x. q(\hat{U}_{u \oplus (n_u,x)})}$.

Suppose that $\hat{u}$ is an approximation to a choice sequence which works for $q(\hat{\mathsf{U}}_u)$ at all points $i \in \mathrm{dom}(u)$:

$$\mathrm{App}(u) \;:\; \forall i \in \mathrm{dom}(u)\; A(i, u(i), q(\hat{\mathsf{U}}_u))$$

but $\varphi(\hat{u}) \notin \mathrm{dom}(u)$. Then since $A(n_u, \overbrace{X_{n_u, \lambda x. q(\hat{\mathsf{U}}_{u\oplus(n_u,x)})}}^{a_u}, \overbrace{q(\hat{U}_{u\oplus(n_u,a_u)})}^{p(a_u)})$ holds we have $\mathsf{U}_u = \mathsf{U}_{u\oplus(n_u,a_u)}$ and

$$\mathrm{App}(u) \Rightarrow \mathrm{App}(u \oplus (n_u, a_u))$$

i.e. we can build a better approximation $u \oplus \widehat{(n_u, a_u)}$, which works for $q(\hat{U}_{u\oplus(n_u,a_u)})$ at all points $i \in \mathrm{dom}(u) \cup \{n_u\}$.

If App(u_0) there exists a sequence $u_0 \sqsubset u_1 \sqsubset \ldots$ of progressively better approximations:

$$n_u := \varphi(\hat{u}_i) \notin \mathrm{dom}(u_i) \quad \text{and} \quad u_{i+1} = u_i \oplus (n_{u_i}, a_{u_i}) \quad \text{and} \quad \mathrm{App}(u_{i+1}).$$

But at some point we reach a leaf $\varphi(\hat{u}_M) \in \mathrm{dom}(u_M)$, and then $\mathsf{U}_{u_M} = u_M$ and

$$\begin{aligned} \mathrm{App}(u_M) &\equiv \forall i \in \mathrm{dom}(u_M)\ A(i, u_M(i), q(\hat{\mathsf{U}}_{u_M})) \\ &\Rightarrow A(\varphi(\hat{u}_M), \hat{u}_M(\varphi(\hat{u}_M)), q(\hat{u}_M)). \end{aligned}$$

Thus $F_{X,\varphi,q} = \mathsf{U}_{u_0} = \ldots = \mathsf{U}_{u_M} = \hat{u}_M$ is a sufficiently good approximation.

If $\mathrm{App}(u_0)$ there exists a sequence $u_0 \sqsubset u_1 \sqsubset \ldots$ of progressively better approximations:

$$n_u := \varphi(\hat{u}_i) \notin \mathrm{dom}(u_i) \quad \text{and} \quad u_{i+1} = u_i \oplus (n_{u_i}, a_{u_i}) \quad \text{and} \quad \mathrm{App}(u_{i+1}).$$

But at some point we reach a leaf $\varphi(\hat{u}_M) \in \mathrm{dom}(u_M)$, and then $\mathsf{U}_{u_M} = u_M$ and

$$\begin{aligned} \mathrm{App}(u_M) &\equiv \forall i \in \mathrm{dom}(u_M)\ A(i, u_M(i), q(\hat{\mathsf{U}}_{u_M})) \\ &\Rightarrow A(\varphi(\hat{u}_M), \hat{u}_M(\varphi(\hat{u}_M)), q(\hat{u}_M)). \end{aligned}$$

Thus $F_{X,\varphi,q} = \mathsf{U}_{u_0} = \ldots = \mathsf{U}_{u_M} = \hat{u}_M$ is a sufficiently good approximation.

THEOREM. $\mathsf{U}_\emptyset$ is a sufficiently good approximation to a choice sequence.

COROLLARY (Oliva/P. 2015). If $\mathsf{PA} + \mathsf{AC} \vdash P$ then $\mathsf{T}_{\lambda_{\mathrm{sBR}}} \vdash \forall y |P^N|^t_y$, where $\mathsf{T}_{\lambda_{\mathrm{sBR}}}$ is the system of primitive recursive functionals in all finite types together with symmetric bar recursion.

Summary

In order to give a general computational interpretation to countable choice, need:

$$\text{Gödel's } \mathsf{T} + \text{backward recursion.}$$

Spector's original bar recursion is one possibility.

Summary

In order to give a general computational interpretation to countable choice, need:

$$\text{Gödel's } \mathsf{T} + \text{backward recursion.}$$

Spector's original bar recursion is one possibility.

What advantage does symmetric bar recursion have?

$$\text{control parameter } \varphi \;\approx\; \text{proof-theoretic environment}$$

Spector only cares whether or not $\varphi(\hat{s}_i) < \text{len}(s_i)$, and insists on building approximations sequentially. But if we care about point $n = 1,000,000$ do we really need to compute $n = 0, 1, \ldots, 999, 999$ first?

Symmetric bar recursion uses φ to drive the construction of the approximation.

We would expect symmetric bar recursion to produce algorithms that are (a) more efficient and (b) more intuitive.

Outline

Let us consider a countable sequence of instances of Σ^0_1-LEM:

$$\forall n^{\mathbb{N}}(\exists x^{\mathbb{N}} P_n(x) \vee \forall y \neg P_n(y)).$$

where $P_n(x)$ is quantifier-free. The finitary intepretation is

$$\forall n, p^{\mathbb{N}\to\mathbb{N}} \exists x (P_n(x) \vee \neg P_n(p(x))).$$

Let us consider a countable sequence of instances of Σ^0_1-LEM:

$$\forall n^{\mathbb{N}}(\exists x^{\mathbb{N}} P_n(x) \vee \forall y \neg P_n(y)).$$

where $P_n(x)$ is quantifier-free. The finitary intepretation is

$$\forall n, p^{\mathbb{N}\to\mathbb{N}} \exists x (P_n(x) \vee \neg P_n(p(x))).$$

This is realized by

$$X_{n,p} := \begin{cases} 0 & \text{if } \neg P_n(p(0)) \\ p(0) & \text{otherwise} \end{cases}$$

in other words, the realizer decides which branch of the standard Herbrand disjunction holds:

$$[P_n(0) \vee \neg P_n(p(0))] \vee [P_n(p(0)) \vee P_n(p(p(0)))].$$

By axiom of choice there exists a comprehension $f\colon \mathbb{N} \to \rho$ such that

$$\forall n(P_n(f(n)) \vee \forall y \neg P_n(y)).$$

The finitary interpretation is

$$\forall \varphi, q \exists f(P_{\varphi f}(f(\varphi f)) \vee \neg P_{\varphi f}(qf))$$

i.e. there exists an approximation f to a comprehesion function which works for qf at point φf.

By axiom of choice there exists a comprehension $f\colon \mathbb{N} \to \rho$ such that

$$\forall n(P_n(f(n)) \vee \forall y \neg P_n(y)).$$

The finitary interpretation is

$$\forall \varphi, q \exists f(P_{\varphi f}(f(\varphi f)) \vee \neg P_{\varphi f}(qf))$$

i.e. there exists an approximation f to a comprehesion function which works for qf at point φf.

This is realized by $F_{\varphi,q} := \hat{\mathsf{U}}^{X,\varphi,q}_{\emptyset}$ or $\hat{\mathsf{E}}^{X,\varphi,q}_{[]}$ where X is realizer to Σ^0_1-LEM on previous slide.

The standard realizer $\hat{\mathsf{E}}_s$ of comprehension, using Spector's bar recursion, is well-known and widely studied. So let's look at the symmetric realizer:

$$\hat{\mathsf{U}}_u = \begin{cases} \hat{u} & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ \hat{\mathsf{U}}_{u \oplus (n_u, a_u)} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$ and

$$a_u := X_{n_u, \lambda x. q(\hat{U}_{u \oplus (n_u, x)})} = \begin{cases} 0 & \text{if } \neg P_{n_u}(q(\hat{\mathsf{U}}_{u \oplus (n_u, 0)})) \\ q(\hat{\mathsf{U}}_{u \oplus (n_u, 0)}) & \text{otherwise} \end{cases}$$

The standard realizer $\hat{\mathsf{E}}_s$ of comprehension, using Spector's bar recursion, is well-known and widely studied. So let's look at the symmetric realizer:

$$\hat{\mathsf{U}}_u = \begin{cases} \hat{u} & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ \hat{\mathsf{U}}_{u \oplus (n_u, a_u)} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$ and

$$a_u := X_{n_u, \lambda x . q(\hat{U}_{u \oplus (n_u, x)})} = \begin{cases} 0 & \text{if } \neg P_{n_u}(q(\hat{\mathsf{U}}_{u \oplus (n_u, 0)})) \\ q(\hat{\mathsf{U}}_{u \oplus (n_u, 0)}) & \text{otherwise} \end{cases}$$

Note $\varphi(u \widehat{\oplus (n_u}, 0)) = \varphi(\hat{u}) = n_u \in \mathrm{dom}(u \oplus (n_u, 0))$, therefore

$$\hat{\mathsf{U}}_{u \oplus (n_u, 0)} = u \widehat{\oplus (n_u}, 0) = \hat{u}.$$

The standard realizer $\hat{\mathsf{E}}_s$ of comprehension, using Spector's bar recursion, is well-known and widely studied. So let's look at the symmetric realizer:

$$\hat{\mathsf{U}}_u = \begin{cases} \hat{u} & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ \hat{\mathsf{U}}_{u \oplus (n_u, a_u)} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$ and

$$a_u = \begin{cases} 0 & \text{if } \neg P_{n_u}(q(\hat{u})) \\ q(\hat{u}) & \text{otherwise} \end{cases}$$

Note $\varphi(u \widehat{\oplus (n_u}, 0)) = \varphi(\hat{u}) = n_u \in \mathrm{dom}(u \oplus (n_u, 0))$, therefore

$$\hat{\mathsf{U}}_{u \oplus (n_u, 0)} = u \widehat{\oplus (n_u}, 0) = \hat{u}.$$

The standard realizer $\hat{\mathsf{E}}_s$ of comprehension, using Spector's bar recursion, is well-known and widely studied. So let's look at the symmetric realizer:

$$\hat{\mathsf{U}}_u = \begin{cases} \hat{u} & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \\ \hat{\mathsf{U}}_{u \oplus (n_u, 0)} & \text{if } \neg P_{n_u}(q(\hat{u})) \\ \hat{\mathsf{U}}_{u \oplus (n_u, q(\hat{u}))} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$.

Note $\varphi(u \widehat{\oplus (n_u}, 0)) = \varphi(\hat{u}) = n_u \in \mathrm{dom}(u \oplus (n_u, 0))$, therefore

$$\hat{\mathsf{U}}_{u \oplus (n_u, 0)} = u \widehat{\oplus (n_u}, 0) = \hat{u}.$$

The standard realizer $\hat{\mathsf{E}}_s$ of comprehension, using Spector's bar recursion, is well-known and widely studied. So let's look at the symmetric realizer:

$$\hat{\mathsf{U}}_u = \begin{cases} \hat{u} & \text{if } \varphi(\hat{u}) \in \mathrm{dom}(u) \vee \neg P_{n_u}(q(\hat{u})) \\ \hat{\mathsf{U}}_{u \oplus (n_u, q(\hat{u}))} & \text{otherwise} \end{cases}$$

where $n_u := \varphi(\hat{u})$.

Start with $u_0 := \emptyset$ and let $n_0 := \varphi(\hat{u}_0)$:

$$\hat{u}_0 = 0, 0, 0, \ldots$$

If $n_0 \in \emptyset$ or $\neg P_{n_0}(q(\hat{u}_0))$ then we're done. Otherwise update as $u_1 := (n_0, q(\hat{u}_0))$:

$$\hat{u}_1 = 0, 0, \ldots, 0, \underbrace{q(\hat{u}_0)}_{n_0}, 0, \ldots$$

If $n_1 := \varphi(\hat{u}_1) \in \{n_0\}$ or $\neg P_{n_1}(q(\hat{u}_1))$ we're done. Otherwise update as $u_2 := (n_0, a_0) \oplus (n_1, q(\hat{u}_1))$:

$$\hat{u}_2 = 0, 0, \ldots, 0, \underbrace{q(\hat{u}_0)}_{n_0}, 0, \ldots, 0, \underbrace{q(\hat{u}_1)}_{n_1}, 0, \ldots$$

If $n_2 := \varphi(\hat{u}_2) \in \{n_0, n_1\}$ or $\neg P_{n_2}(q(\hat{u}_2))$ we're done. Otherwise update again...

$$\hat{u}_3 = 0, 0, \ldots, 0, \underbrace{q(\hat{u}_2)}_{n_2}, 0, \ldots, 0, \underbrace{q(\hat{u}_0)}_{n_0}, 0, \ldots, 0, \underbrace{q(\hat{u}_1)}_{n_1}, 0, \ldots$$

$$\vdots$$

We have an increasing sequence of approximations $u_0 \sqsubset u_1 \sqsubset u_2 \sqsubset \ldots$ satisfying

$$\forall k \in \mathrm{dom}(u_i)\ P_k(u_i(k))$$

Eventually must hit a point M such that $n_M \notin \mathrm{dom}(u_M)$ and

$$\neg P_{n_M}(q(\hat{u}_M)),$$

or $n_M \in \mathrm{dom}(u_M)$ and thus

$$P_{n_M}(u_M(n_M)),$$

i.e. (recall $n_M = \varphi(\hat{u}_M)$):

$$P_{\varphi(\hat{u}_M)}(\hat{u}_M(\varphi(\hat{u}_M))) \vee \neg P_{\varphi(\hat{u}_M)}(q(\hat{u}_M))$$

and so $\hat{u}_M$ is a sufficiently good approximation to a comprehension function.

Symmetric bar recursion $\approx$ Learning procedure

By $\mathcal{L}_{\varphi,q,P}$ we mean the following algorithm:

`TEST`(u): `Does` $\varphi(\hat{u}) \in \mathrm{dom}(u) \vee \neg P_{\varphi(\hat{u})}(q(\hat{u}))$ `hold?`

`YES` $\rightsquigarrow$ `Terminate.`

`NO` $\rightsquigarrow$ `Update with new information:` $u \to u \oplus (\varphi(\hat{u}), q(\hat{u}))$

Symmetric bar recursion ≈ Learning procedure

By $\mathcal{L}_{\varphi,q,P}$ we mean the following algorithm:

`TEST`(u): `Does` $\varphi(\hat{u}) \in \mathrm{dom}(u) \vee \neg P_{\varphi(\hat{u})}(q(\hat{u}))$ `hold?`

`YES` $\rightsquigarrow$ `Terminate.`

`NO` $\rightsquigarrow$ `Update with new information:` $u \to u \oplus (\varphi(\hat{u}), q(\hat{u}))$

Proposition. Suppose that in PA we can derive

$$\forall x[\mathsf{CA}(P_x) \to \exists y A_0(x,y)].$$

Then there is some learning procedure $\mathcal{L}_{\varphi,q,P_x}$ and a primitive recursive function g such that

$$\forall x A_0(x, g(\mathcal{L}_{\varphi,q,P_x}, x))$$

Example. In PA we can derive

$$\forall H^{(\mathbb{N}\to\mathbb{N})\to\mathbb{N}}[\mathsf{CA}(P_F) \to \exists \alpha^{\mathbb{N}\to\mathbb{N}}, \beta^{\mathbb{N}\to\mathbb{N}}, i^{\mathbb{N}}(\alpha(i) \neq \beta(i) \wedge H\alpha = H\beta)].$$

Example. In PA we can derive

$$\forall H^{(\mathbb{N}\to\mathbb{N})\to\mathbb{N}}[\mathsf{CA}(P_F) \to \exists \alpha^{\mathbb{N}\to\mathbb{N}}, \beta^{\mathbb{N}\to\mathbb{N}}, i^{\mathbb{N}}(\alpha(i) \neq \beta(i) \wedge H\alpha = H\beta)].$$

An algorithm for finding α, β and i can be formally extracted, which uses the following learning procedure:

Define the sequence of functions $\gamma_i \colon \mathbb{N} \to \mathbb{N}$ by

$$\gamma_i := \lambda k \,.\, \begin{cases} 1 & \text{if } k \in D_i \\ 0 & \text{otherwise,} \end{cases}$$

where

$$D_0 := \emptyset \qquad D_{i+1} := D_i \cup \{H(\gamma_i)\}.$$

We have $\gamma_i(k) = 1$ iff $H(\gamma_j) = k$ for some $j < i$. Stop at the first point M such that $H(\gamma_M) \in D_M$. This means that for some $j < M$ have $H(\gamma_j) = H(\gamma_M)$.

Set $\alpha, \beta := \gamma_M, \gamma_j$. These differ at point $i = H(\gamma_M)$.

Start with $s_0 := \langle\rangle$:

$$\hat{s}_0 = 0, 0, 0, \ldots$$

Search for the least $n_0 \leq \varphi(s_0)$ such that $\neg P_{n_0}(q(\hat{s}_0))$ otherwise we're done. Else, update as $s_1 := \langle 0, 0, \ldots, q(\hat{s}_0)\rangle$:

$$\hat{s}_1 = 0, 0, \ldots, 0, \underbrace{q(\hat{s}_0)}_{n_0}, 0, \ldots$$

Search for the least $n_1 \leq \max(n_0, \varphi(\hat{s}_1))$ with $n_1 \leq n_0$ satisfying $\neg P_{n_1}(q(\hat{s}_1))$. If $n_1 > n_0$ set $s_2 := \langle 0, 0, \ldots, 0, q(\hat{s}_0), 0, \ldots, 0, q(\hat{s}_1)\rangle$:

$$\hat{s}_2 = 0, 0, \ldots, 0, \underbrace{q(\hat{s}_0)}_{n_0}, 0, \ldots, 0, \underbrace{q(\hat{s}_1)}_{n_1}, 0, \ldots$$

else if $n_1 < n_0$ set $s_2 := \langle 0, 0, \ldots, q(\hat{s}_1)\rangle$:

$$\hat{s}_2 = 0, 0, \ldots, 0, \underbrace{q(\hat{s}_1)}_{n_1}, 0, \ldots$$

The witness $q(\hat{s}_0)$ for $\exists x P_{n_0}(x)$ is erased!

Tests indicate that, on the whole, the highly intuitive algorithm given by symmetric bar recursion performs much better than the traditional one based on Spector.

$H_n(\gamma) =$ least $i \leq n$ such that $\gamma i < \gamma(i+1)$, else n if none exist :

	Spector	Symmetric
$n = 3$	4 / 316	4 / 52
$n = 4$	5 / 688	5 / 64
$n = 5$	6 / 1444	6 / 76

$H_n(\gamma) = \Pi_{i=0}^{n-1}(1+i)^{1+\gamma i}$:

	Spector	Symmetric
$n = 3$	577 / 2350	1 / 12
$n = 4$	577 / 365700	1 / 12

Directions for future research

1. A more detailed investigation into the behaviour of programs extracted using symmetric bar recursion. Can we give concise, intuitive computational interpretations of well-known proofs which use countable choice?
2. Have already suggested that the Dialectica interpretation of analysis is linked to learning. How are extracted programs related to those obtained using e.g. ϵ-calculus, or Aschieri-Berardi interactive learning realizability?
3. Can we take advantage of symmetric bar recursion's flexibility to extend Dialectica to more general choice principles over arbitrary discrete domains:

$$\mathsf{AC}_{D,X} \;:\; \forall d^D \exists x^X A(d,x) \rightarrow \exists f^{D\rightarrow X} \forall d A(d, fd).$$