Modes of Bar Recursion

Thomas Powell

(based on joint work with Martín Escardó and Paulo Oliva)
Queen Mary, University of London

Birmingham Theory Seminar
3 July 2012

Outline

1. Introduction
   - Bar recursion
   - Overview of talk
2. Modes of bar recursion
   - PS / Finite bar recursion
   - EPS / Spector's bar recursion
   - IPS / Modified bar recursion
   - UR / Berardi-Bezem-Coquand functional
3. Interdefinability results
   - Relationship between EPS and IPS
   - Relationship between IPS and UR

Bar recursion

Primitive recursion: Recursion over the natural numbers. For $n \in \mathbb{N}$:

$$\mathrm{R}(n) := \begin{cases} y & \text{if } n = 0 \\ z_{n-1}(\mathrm{R}(n-1)) & \text{otherwise} \end{cases}$$

Bar recursion

Primitive recursion: Recursion over the natural numbers. For $n \in \mathbb{N}$:

$$R(n) := \begin{cases} y & \text{if } n = 0 \\ z_{n-1}(R(n-1)) & \text{otherwise} \end{cases}$$

Bar recursion: Recursion over well-founded trees. For $s \in T$:

$$B(s^{X^*}) := \begin{cases} Y_s & \text{if } s \text{ is a leaf} \\ Z_s(\lambda x \,.\, B(s * x)) & \text{otherwise} \end{cases}$$

Bar recursion is the wrong way round: $B(s)$ looks at the values of $B(s * x)$ for *extensions* of s!

- **Gödel 1958** Dialectica interpretation of arithmetic

  Arithmetic (induction) $\mapsto$ System T (primitive recursion)

  Spector 1962 Dialectica interpretation of analysis

  Arithmetic + Countable choice $\mapsto$ System T + Bar recursion

- **Gödel 1958** Dialectica interpretation of arithmetic

  Arithmetic (induction) $\mapsto$ System T (primitive recursion)

  Spector 1962 Dialectica interpretation of analysis

  Arithmetic + Countable choice $\mapsto$ System T + Bar recursion

- **Gandy/Hyland 1970s** Γ functional - computable but not S1-S9 definable.

- **Gödel 1958** Dialectica interpretation of arithmetic

  Arithmetic (induction) $\mapsto$ System T (primitive recursion)

  Spector 1962 Dialectica interpretation of analysis

  Arithmetic + Countable choice $\mapsto$ System T + Bar recursion

- **Gandy/Hyland 1970s** Γ functional - computable but not S1-S9 definable.
- **Berardi/Bezem/Coquand 1998** Realizability interpretation of choice. BBC functional (non-sequential form of bar recursion). Game semantics.

- **Gödel 1958** Dialectica interpretation of arithmetic

  $$\text{Arithmetic (induction)} \mapsto \text{System T (primitive recursion)}$$

  Spector 1962 Dialectica interpretation of analysis

  $$\text{Arithmetic} + \text{Countable choice} \mapsto \text{System T} + \text{Bar recursion}$$

- **Gandy/Hyland 1970s** Γ functional - computable but not S1-S9 definable.
- **Berardi/Bezem/Coquand 1998** Realizability interpretation of choice. BBC functional (non-sequential form of bar recursion). Game semantics.
- **Berger/Oliva 2005** Modified realizability interpretation of choice via modified bar recursion. Some interdefinability results.

- **Gödel 1958** Dialectica interpretation of arithmetic

  $$\text{Arithmetic (induction)} \mapsto \text{System T (primitive recursion)}$$

  Spector 1962 Dialectica interpretation of analysis

  $$\text{Arithmetic} + \text{Countable choice} \mapsto \text{System T} + \text{Bar recursion}$$

- **Gandy/Hyland 1970s** Γ functional - computable but not S1-S9 definable.
- **Berardi/Bezem/Coquand 1998** Realizability interpretation of choice. BBC functional (non-sequential form of bar recursion). Game semantics.
- **Berger/Oliva 2005** Modified realizability interpretation of choice via modified bar recursion. Some interdefinability results.
- **Escardó/Oliva 2010-** Product of selection functions. Interdefinability results. Links with game theory made precise.

Theme of talk

Computational aspects of bar recursion

1. Key computational features of different modes of bar recursion.
2. The relative strength of these modes of bar recursion.

To a lesser extent: The semantics of bar recursion (links with language of sequential games).

Why is this important?

Theme of talk

Computational aspects of bar recursion

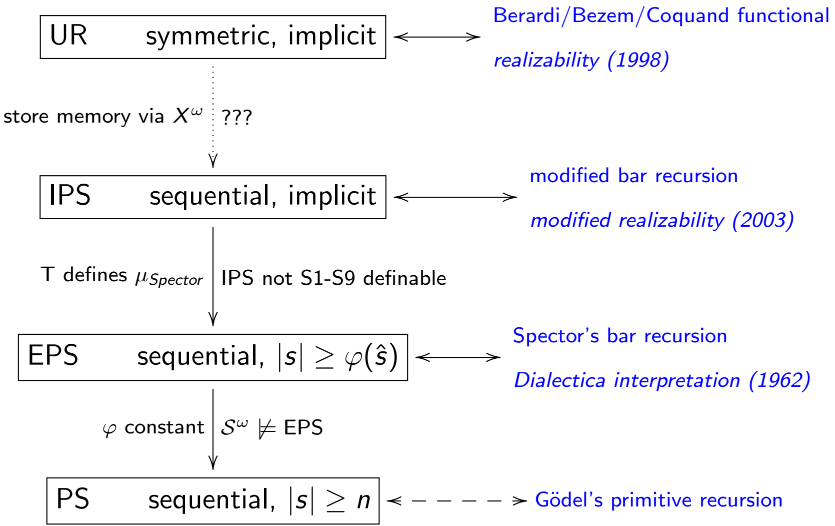
1. Key computational features of different modes of bar recursion.
2. The relative strength of these modes of bar recursion.

To a lesser extent: The semantics of bar recursion (links with language of sequential games).

Why is this important?

- Open questions about an important class of non-primitive recursive functionals.
- Better understand computational content of classical proofs.
- Interesting mathematical problem.

Modes of bar recursion



Outline

1. Introduction
   - Bar recursion
   - Overview of talk
2. Modes of bar recursion
   - PS / Finite bar recursion
   - EPS / Spector's bar recursion
   - IPS / Modified bar recursion
   - UR / Berardi-Bezem-Coquand functional
3. Interdefinability results
   - Relationship between EPS and IPS
   - Relationship between IPS and UR

The finite product of selection functions

Idea: Sequential game with n rounds. Moves of type X, outcome of type R.

- $q\colon X^n \to R$ determines the *outcome* of a play of type X^n (by instead considering $q\colon X^\omega \to R$ type independent of n).
- $\varepsilon_s\colon (X \to R) \to X$ dictates a *strategy* for $|s|$th round given a partial play s^{X^*}.

The finite product of selection functions

Idea: Sequential game with n rounds. Moves of type X, outcome of type R.

- $q\colon X^n \to R$ determines the *outcome* of a play of type X^n (by instead considering $q\colon X^\omega \to R$ type independent of n).
- $\varepsilon_s\colon (X \to R) \to X$ dictates a *strategy* for $|s|$th round given a partial play s^{X^*}.

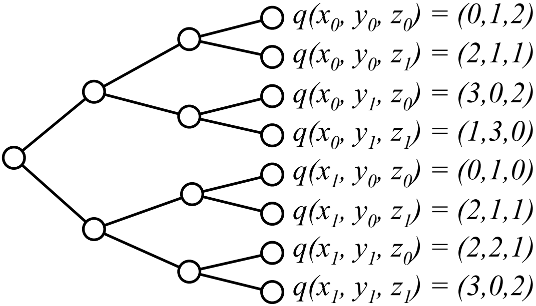
$$\mathrm{PS}^{\varepsilon,q,n}(s^{X^*}) \overset{X^*}{:=} \begin{cases} \langle\rangle & \text{if } |s| \geq n \\ a_s * \mathrm{PS}(s * a_s) & \text{otherwise} \end{cases}$$

where $a_s := \varepsilon_s(\lambda x\ .\ q(s * x * \mathrm{PS}(s * x)))$.

- For $|s| < n$, $\mathrm{PS}(s)$ is the optimal continuation of (of length $n - |s|$) of the play s.
- $\mathrm{PS}(\langle\rangle)$ is an optimal strategy for the game.

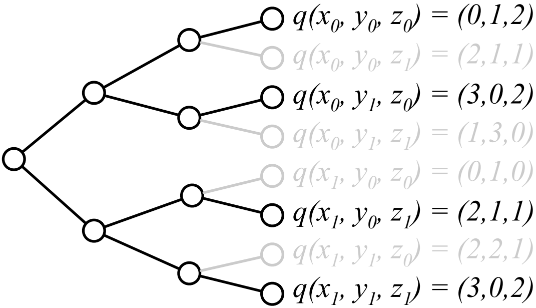
Example 1

$$X = [2] \quad ; \quad R = \mathbb{N} \quad ; \quad n = 3 \quad ; \quad q \colon [2]^3 \to \mathbb{N}^3 \quad ;$$
$$\varepsilon_i(p^{[2]\to\mathbb{N}}) = x \text{ maximising } p(x)_i$$



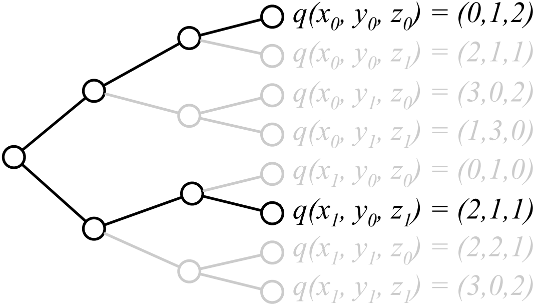
Example 1

$$\mathrm{PS}(x_1, y_0) = \langle \varepsilon_2(z_0 \mapsto 0, z_1 \mapsto 1)\rangle * \mathrm{PS}(x_1, y_0, z_i) = \langle z_1 \rangle * \langle\rangle = \langle z_1 \rangle$$



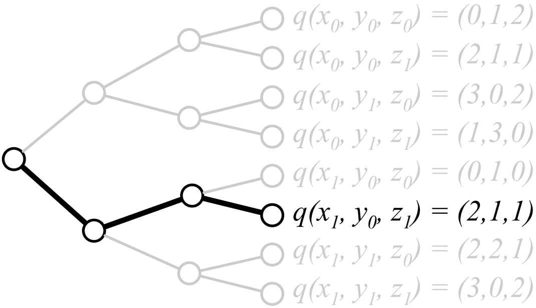
Example 1

$$\mathsf{PS}(x_1) = \langle y_0 \rangle * \mathsf{PS}(x_1, y_0) = \langle y_0, z_1 \rangle$$



Example 1

$$\mathsf{PS}(\langle\rangle) = \langle x_1 \rangle * \mathsf{PS}(x_1) = \langle x_1, y_0, z_1 \rangle$$



Example 2 (Noughts and Crosses)

- $X = \{0, \ldots, 8\}$ and $R = \{1, 0, -1\}$.

Example 2 (Noughts and Crosses)

- $X = \{0, \dots, 8\}$ and $R = \{1, 0, -1\}$.
- $X^{(9)}$ encodes a game (only part of this may be relevant).

$$q(s^{X^{(9)}}) := \begin{cases} 1 & \text{if first player wins} \\ 0 & \text{if players draw} \\ -1 & \text{if second player wins} \end{cases}$$

Example 2 (Noughts and Crosses)

- $X = \{0, \ldots, 8\}$ and $R = \{1, 0, -1\}$.
- $X^{(9)}$ encodes a game (only part of this may be relevant).

$$q(s^{X^{(9)}}) := \begin{cases} 1 & \text{if first player wins} \\ 0 & \text{if players draw} \\ -1 & \text{if second player wins} \end{cases}$$

- $\varepsilon_{2i}(p^{X \to R}) / \varepsilon_{2i+1}(p)$ selects x maximising/minimising $p(x)$.

Example 2 (Noughts and Crosses)

- $X = \{0, \ldots, 8\}$ and $R = \{1, 0, -1\}$.
- $X^{(9)}$ encodes a game (only part of this may be relevant).

$$q(s^{X^{(9)}}) := \begin{cases} 1 & \text{if first player wins} \\ 0 & \text{if players draw} \\ -1 & \text{if second player wins} \end{cases}$$

- $\varepsilon_{2i}(p^{X \to R}) / \varepsilon_{2i+1}(p)$ selects x maximising/minimising $p(x)$.
- $\mathrm{PS}^{\varepsilon,q,n}(\langle\rangle)$ returns an 'optimal' play, resulting in a draw.

Is PS well-defined?

PS well-defined (i.e. defining equations have a unique solution) in any model of system T:

Is PS well-defined?

PS well-defined (i.e. defining equations have a unique solution) in any model of system T:

For $|s| \geq n$ we have $\mathrm{PS}(s) = \langle\rangle$;

Is PS well-defined?

PS well-defined (i.e. defining equations have a unique solution) in any model of system T:

For $|s| \geq n$ we have $\mathrm{PS}(s) = \langle\rangle$;

For $|s| < n$, if $\mathrm{PS}(s * x)$ well-defined for all extensions $s * x$ of s, then so is $\mathrm{PS}(s)$.

Is PS well-defined?

PS well-defined (i.e. defining equations have a unique solution) in any model of system T:

For $|s| \geq n$ we have $\mathrm{PS}(s) = \langle\rangle$;

For $|s| < n$, if $\mathrm{PS}(s * x)$ well-defined for all extensions $s * x$ of s, then so is $\mathrm{PS}(s)$.

By induction on $n \dot{-} |s|$, $\mathrm{PS}(\langle\rangle)$ is well-defined.

Is PS well-defined?

PS well-defined (i.e. defining equations have a unique solution) in any model of system T:

For $|s| \geq n$ we have $\mathrm{PS}(s) = \langle\rangle$;

For $|s| < n$, if $\mathrm{PS}(s * x)$ well-defined for all extensions $s * x$ of s, then so is $\mathrm{PS}(s)$.

By induction on $n \dot{-} |s|$, $\mathrm{PS}(\langle\rangle)$ is well-defined.

Remark. PS is equivalent over a weak base theory to Gödel's primitive recursion in all finite types.

Properties of PS

Properties of PS

Order. Computation carried out sequentially: value of $\mathsf{PS}(\langle\rangle)_1$ depends on the value of $\mathsf{PS}(\langle\rangle)_0$ and so on.

Properties of PS

Order. Computation carried out sequentially: value of $\mathsf{PS}(\langle\rangle)_1$ depends on the value of $\mathsf{PS}(\langle\rangle)_0$ and so on.

Well-foundedness. Underlying tree given *explicitly*:

$$s \text{ a leaf} \Leftrightarrow |s| = n.$$

Properties of PS

Order. Computation carried out sequentially: value of $\text{PS}(\langle\rangle)_1$ depends on the value of $\text{PS}(\langle\rangle)_0$ and so on.

Well-foundedness. Underlying tree given *explicitly*:

$$s \text{ a leaf} \Leftrightarrow |s| = n.$$

Models. Exists in any model of (higher-type) primitive recursion i.e. standard set theoretic model, total continuous functionals...

Properties of PS

Order. Computation carried out sequentially: value of $\mathrm{PS}(\langle\rangle)_1$ depends on the value of $\mathrm{PS}(\langle\rangle)_0$ and so on.

Well-foundedness. Underlying tree given *explicitly*:

$$s \text{ a leaf} \Leftrightarrow |s| = n.$$

Models. Exists in any model of (higher-type) primitive recursion i.e. standard set theoretic model, total continuous functionals...

Semantics. Operation that computes optimal strategies in finite sequential games.

Modes of bar recursion

PS	sequential, $\lvert s \rvert \geq n$

$\lessdot - \gtrdot$ Gödel's primitive recursion

Explicitly iterated product of selection functions (EPS)

Idea: Sequential game with unbounded number of rounds.

- $q \colon X^\omega \to R$ determines outcome of each infinite play X^ω.
- $\varepsilon_s \colon (X \to R) \to X$ dictates a *strategy* for $|s|$th round given any partial play s^{X^*}.
- $\varphi \colon X^\omega \to \mathbb{N}$ gives 'relevant' part of infinite play.

Explicitly iterated product of selection functions (EPS)

Idea: Sequential game with unbounded number of rounds.

- $q \colon X^\omega \to R$ determines outcome of each infinite play X^ω.
- $\varepsilon_s \colon (X \to R) \to X$ dictates a *strategy* for $|s|$th round given any partial play s^{X^*}.
- $\varphi \colon X^\omega \to \mathbb{N}$ gives 'relevant' part of infinite play.

$$\mathrm{EPS}^{\varepsilon,q,\varphi}(s^{X^*}) \overset{X^\omega}{:=} \begin{cases} \mathbf{0}^{X^\omega} & \text{if } |s| \geq \varphi(\hat{s}) \\ a_s * \mathrm{EPS}(s * a_s) & \text{otherwise} \end{cases}$$

where $a_s := \varepsilon_{|s|}(\lambda x \,.\, q(s * x * \mathrm{EPS}(s * x)))$ (and $\hat{s} := s * \mathbf{0}$).

Explicitly iterated product of selection functions (EPS)

Idea: Sequential game with unbounded number of rounds.

- $q \colon X^\omega \to R$ determines outcome of each infinite play X^ω.
- $\varepsilon_s \colon (X \to R) \to X$ dictates a *strategy* for $|s|$th round given any partial play s^{X^*}.
- $\varphi \colon X^\omega \to \mathbb{N}$ gives 'relevant' part of infinite play.

$$\mathrm{EPS}^{\varepsilon,q,\varphi}(s^{X^*}) \overset{X^\omega}{:=} \begin{cases} \mathbf{0}^{X^\omega} & \text{if } |s| \geq \varphi(\hat{s}) \\ a_s * \mathrm{EPS}(s * a_s) & \text{otherwise} \end{cases}$$

where $a_s := \varepsilon_{|s|}(\lambda x \,.\, q(s * x * \mathrm{EPS}(s * x)))$ (and $\hat{s} := s * \mathbf{0}$).

Stopping condition now depends on s!

Is EPS well defined?

For underlying tree to be well-founded, need property that for all infinite sequences α^{X^ω} there must exists n such that $n \geq \varphi(\widehat{[\alpha](n)})$.

Is EPS well defined?

For underlying tree to be well-founded, need property that for all infinite sequences α^{X^ω} there must exists n such that $n \geq \varphi(\widehat{[\alpha](n)})$.

Fails for e.g.

$$\varphi(\alpha) := i + 1 \text{ for least } i(\alpha i = 0), \text{ 0 otherwise.}$$

If $\alpha = \lambda i.1$ then for arbitrary n

Is EPS well defined?

For underlying tree to be well-founded, need property that for all infinite sequences α^{X^ω} there must exists n such that $n \geq \varphi(\widehat{[\alpha](n)})$.

Fails for e.g.

$$\varphi(\alpha) := i + 1 \text{ for least } i(\alpha i = 0), \text{ 0 otherwise.}$$

If $\alpha = \lambda i.1$ then for arbitrary n

$$\varphi(\widehat{[\lambda i.1](n)}) = \varphi(\underbrace{1, \ldots, 1}_{n \text{ times}}, 0, 0, \ldots) = n + 1 > n.$$

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^{\omega}$.

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^\omega$.

$$\mathsf{CONT} : \forall \varphi^{X^\omega \to \mathbb{N}} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \stackrel{X^*}{=} [\beta](N) \to \varphi\alpha = \varphi\beta)$$

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^\omega$.

$$\text{CONT} : \forall \varphi^{X^\omega \to \mathbb{N}} \forall \alpha^{X^\omega} \exists N \forall \beta([\alpha](N) \overset{X^*}{=} [\beta](N) \to \varphi\alpha = \varphi\beta)$$

By CONT, $\varphi(\widehat{[\alpha](n)}) = \varphi\alpha$ for all $n \geq N$, so for $n = \max\{N, \varphi\alpha\}$ we have $n \geq \varphi\alpha = \varphi(\widehat{[\alpha](n)})$.

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^\omega$.

$$\mathrm{CONT}\ :\ \forall\varphi^{X^\omega\to\mathbb{N}}\forall\alpha^{X^\omega}\exists N\forall\beta([\alpha](N)\overset{X^*}{=}[\beta](N)\to\varphi\alpha=\varphi\beta)$$

By CONT, $\varphi(\widehat{[\alpha](n)})=\varphi\alpha$ for all $n\geq N$, so for $n=\max\{N,\varphi\alpha\}$ we have $n\geq\varphi\alpha=\varphi(\widehat{[\alpha](n)})$.

For all α there exists some n such that $\mathrm{EPS}(\widehat{[\alpha](n)})=\mathbf{0}$ and therefore $\mathrm{EPS}([\alpha](n))$ well-defined.

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^\omega$.

$$\mathrm{CONT} : \forall \varphi^{X^\omega \to \mathbb{N}} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \overset{X^*}{=} [\beta](N) \to \varphi\alpha = \varphi\beta)$$

By CONT, $\varphi(\widehat{[\alpha](n)}) = \varphi\alpha$ for all $n \geq N$, so for $n = \max\{N, \varphi\alpha\}$ we have $n \geq \varphi\alpha = \varphi(\widehat{[\alpha](n)})$.

For all α there exists some n such that $\mathrm{EPS}(\widehat{[\alpha](n)}) = \mathbf{0}$ and therefore $\mathrm{EPS}([\alpha](n))$ well-defined.

If $\mathrm{EPS}(s * x)$ well-defined for all extensions $s * x$ of s, then by definition so is $\mathrm{EPS}(s)$.

Theorem. EPS exists in the total continuous functionals $\mathcal{C}^\omega$.

$$\mathrm{CONT}\ :\ \forall\varphi^{X^\omega\to\mathbb{N}}\forall\alpha^{X^\omega}\exists N\forall\beta([\alpha](N)\overset{X^*}{=}[\beta](N)\to\varphi\alpha=\varphi\beta)$$

By CONT, $\varphi(\widehat{[\alpha](n)})=\varphi\alpha$ for all $n\geq N$, so for $n=\max\{N,\varphi\alpha\}$ we have $n\geq\varphi\alpha=\varphi(\widehat{[\alpha](n)})$.

For all α there exists some n such that $\mathrm{EPS}(\widehat{[\alpha](n)})=\mathbf{0}$ and therefore $\mathrm{EPS}([\alpha](n))$ well-defined.

If $\mathrm{EPS}(s*x)$ well-defined for all extensions $s*x$ of s, then by definition so is $\mathrm{EPS}(s)$.

By the principle of *bar induction* $\mathrm{EPS}(\langle\rangle)$ is well-defined.

Properties of EPS

Order. Like PS, computation carried out sequentially: value of $\mathrm{EPS}(\langle\rangle)_1$ depends on the value of $\mathrm{EPS}(\langle\rangle)_1$ and so on.

Properties of EPS

Order. Like PS, computation carried out sequentially: value of $\mathrm{EPS}(\langle\rangle)_1$ depends on the value of $\mathrm{EPS}(\langle\rangle)_1$ and so on.

Well-foundedness. Like PS, underlying tree given *explicitly*.

$$s \text{ a leaf} \Leftrightarrow |s| \geq \varphi(\hat{s}) \wedge \forall t \prec s(|t| < \varphi(\hat{t})).$$

Properties of EPS

Order. Like PS, computation carried out sequentially: value of $\mathrm{EPS}(\langle\rangle)_1$ depends on the value of $\mathrm{EPS}(\langle\rangle)_1$ and so on.

Well-foundedness. Like PS, underlying tree given *explicitly*.

$$s \text{ a leaf} \Leftrightarrow |s| \geq \varphi(\hat{s}) \wedge \forall t \prec s(|t| < \varphi(\hat{t})).$$

Models. Unlike PS, well-foundedness of recursion not provable in T. EPS exists in $\mathcal{C}^\omega$ where CONT holds, but not in the standard model $\mathcal{S}^\omega$.

Properties of EPS

Order. Like PS, computation carried out sequentially: value of $\mathrm{EPS}(\langle\rangle)_1$ depends on the value of $\mathrm{EPS}(\langle\rangle)_1$ and so on.

Well-foundedness. Like PS, underlying tree given *explicitly*.

$$s \text{ a leaf} \Leftrightarrow |s| \geq \varphi(\hat{s}) \wedge \forall t \prec s(|t| < \varphi(\hat{t})).$$

Models. Unlike PS, well-foundedness of recursion not provable in T. EPS exists in $\mathcal{C}^\omega$ where CONT holds, but not in the standard model $\mathcal{S}^\omega$.

Semantics. Operation that computes optimal strategies in unbounded sequential games, relevant part of play α given by $\varphi(\alpha)$.

Modes of bar recursion

EPS sequential, $|s| \geq \varphi(\hat{s})$ $\longleftrightarrow$ Spector's bar recursion

Dialectica interpretation (1962)

φ constant $\downarrow$ $\mathcal{S}^\omega \not\models$ EPS

PS sequential, $|s| \geq n$ $\longleftarrow - - - \rightarrow$ Gödel's primitive recursion

Implicitly iterated product of selection functions

Idea: Sequential game with unbounded number of rounds, but now we forget about the control functional $\varphi \colon X^\omega \to \mathbb{N}$.

Implicitly iterated product of selection functions

Idea: Sequential game with unbounded number of rounds, but now we forget about the control functional $\varphi \colon X^\omega \to \mathbb{N}$.

$$\mathrm{IPS}^{\varepsilon,q}(s^{X^*}) \overset{X^\omega}{:=} a_s * \mathrm{IPS}(s * a_s)$$

where $a_s := \varepsilon_s(\lambda x \, . \, q(s * x * \mathrm{IPS}(s * x)))$.

Implicitly iterated product of selection functions

Idea: Sequential game with unbounded number of rounds, but now we forget about the control functional $\varphi\colon X^\omega \to \mathbb{N}$.

$$\mathrm{IPS}^{\varepsilon,q}(s^{X^*}) \overset{X^\omega}{:=} a_s * \mathrm{IPS}(s * a_s)$$

where $a_s := \varepsilon_s(\lambda x\ .\ q(s * x * \mathrm{IPS}(s * x)))$.

No longer a stopping condition!

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Let $X = \mathbb{N}$, $R = \mathbb{N}^\omega$, $q = \mathrm{id}\colon \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\varepsilon_s(p^{\mathbb{N}\to\mathbb{N}^\omega}) = p(0)_{|s|+1} + 1$.

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Let $X = \mathbb{N}$, $R = \mathbb{N}^\omega$, $q = \mathrm{id}\colon \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\varepsilon_s(p^{\mathbb{N}\to\mathbb{N}^\omega}) = p(0)_{|s|+1} + 1$.

$$\mathrm{IPS}(\langle\rangle)_0 \overset{\mathbb{N}}{=} \varepsilon_{\langle\rangle}(\lambda x \,.\, x * \mathrm{IPS}(x))$$

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Let $X = \mathbb{N}$, $R = \mathbb{N}^\omega$, $q = \mathrm{id} \colon \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\varepsilon_s(p^{\mathbb{N} \to \mathbb{N}^\omega}) = p(0)_{|s|+1} + 1$.

$$\begin{aligned} \mathrm{IPS}(\langle\rangle)_0 &\stackrel{\mathbb{N}}{=} \varepsilon_{\langle\rangle}(\lambda x \, . \, x * \ \mathrm{IPS}(x)) \\ &= \mathrm{IPS}(0)_0 + 1 \end{aligned}$$

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Let $X = \mathbb{N}$, $R = \mathbb{N}^\omega$, $q = \mathrm{id} \colon \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\varepsilon_s(p^{\mathbb{N} \to \mathbb{N}^\omega}) = p(0)_{|s|+1} + 1$.

$$\begin{aligned} \mathrm{IPS}(\langle\rangle)_0 &\stackrel{\mathbb{N}}{=} \varepsilon_{\langle\rangle}(\lambda x \,.\, x * \ \mathrm{IPS}(x)) \\ &= \mathrm{IPS}(0)_0 + 1 \\ &= \varepsilon_{\langle 0\rangle}(\lambda x \,.\, 0 * x * \mathrm{IPS}(0 * x)) + 1 \\ &= \mathrm{IPS}(0,0)_0 + 2 \end{aligned}$$

Why is IPS well defined?

Even in $\mathcal{C}^\omega$ there are obvious instances where IPS is not computable.

Let $X = \mathbb{N}$, $R = \mathbb{N}^\omega$, $q = \mathrm{id} \colon \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\varepsilon_s(p^{\mathbb{N}\to\mathbb{N}^\omega}) = p(0)_{|s|+1} + 1$.

$$\begin{aligned}
\mathrm{IPS}(\langle\rangle)_0 &\stackrel{\mathbb{N}}{=} \varepsilon_{\langle\rangle}(\lambda x \,.\, x * \ \mathrm{IPS}(x)) \\
&= \mathrm{IPS}(0)_0 + 1 \\
&= \varepsilon_{\langle 0\rangle}(\lambda x \,.\, 0 * x * \mathrm{IPS}(0 * x)) + 1 \\
&= \mathrm{IPS}(0,0)_0 + 2 \\
&\ldots \\
&= \mathrm{IPS}(\underbrace{0, \ldots \ 0}_{n \text{ times}})_0 + n \\
&\ldots
\end{aligned}$$

Theorem. IPS exists in $\mathcal{C}^\omega$ whenever the outcome type R in $q\colon X^\omega \to R$ has type level 0 (more generally open, discrete...).

Theorem. IPS exists in $\mathcal{C}^\omega$ whenever the outcome type R in $q\colon X^\omega \to R$ has type level 0 (more generally open, discrete...).

$$\mathrm{CONT} : \forall q^{X^\omega \to R} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \overset{X^*}{=} [\beta](N) \to q\alpha = q\beta)$$

Theorem. IPS exists in $\mathcal{C}^\omega$ whenever the outcome type R in $q \colon X^\omega \to R$ has type level 0 (more generally open, discrete...).

$$\mathsf{CONT} : \forall q^{X^\omega \to R} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \overset{X^*}{=} [\beta](N) \to q\alpha = q\beta)$$

Therefore $\mathsf{IPS}([\alpha](N))_0 = \varepsilon_{[\alpha](N)}(\lambda x \,.\, q(\alpha))$ and by induction

$$\mathsf{IPS}([\alpha](N)) = \lambda k \,.\, \varepsilon_{[\alpha](N) * t_k}(\lambda x \,.\, q\alpha).$$

where $t_k = [\mathsf{IPS}([\alpha](N))](k)$, so $\mathsf{IPS}([\alpha](N))$ well-defined.

Theorem. IPS exists in $\mathcal{C}^\omega$ whenever the outcome type R in $q\colon X^\omega \to R$ has type level 0 (more generally open, discrete...).

$$\mathrm{CONT}\ :\ \forall q^{X^\omega \to R} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \overset{X^*}{=} [\beta](N) \to q\alpha = q\beta)$$

Therefore $\mathrm{IPS}([\alpha](N))_0 = \varepsilon_{[\alpha](N)}(\lambda x \,.\, q(\alpha))$ and by induction

$$\mathrm{IPS}([\alpha](N)) = \lambda k \,.\, \varepsilon_{[\alpha](N) * t_k}(\lambda x \,.\, q\alpha).$$

where $t_k = [\mathrm{IPS}([\alpha](N))](k)$, so $\mathrm{IPS}([\alpha](N))$ well-defined.

If $\mathrm{IPS}(s * x)$ is well-defined for all extension $s * x$ of s, then by definition $\mathrm{IPS}(s)$ is also well-defined.

Theorem. IPS exists in $\mathcal{C}^\omega$ whenever the outcome type R in $q \colon X^\omega \to R$ has type level 0 (more generally open, discrete...).

$$\mathrm{CONT} \ : \ \forall q^{X^\omega \to R} \forall \alpha^{X^\omega} \exists N \forall \beta ([\alpha](N) \overset{X^*}{=} [\beta](N) \to q\alpha = q\beta)$$

Therefore $\mathrm{IPS}([\alpha](N))_0 = \varepsilon_{[\alpha](N)}(\lambda x \, . \, q(\alpha))$ and by induction

$$\mathrm{IPS}([\alpha](N)) = \lambda k \, . \, \varepsilon_{[\alpha](N) * t_k}(\lambda x \, . \, q\alpha).$$

where $t_k = [\mathrm{IPS}([\alpha](N))](k)$, so $\mathrm{IPS}([\alpha](N))$ well-defined.

If $\mathrm{IPS}(s * x)$ is well-defined for all extension $s * x$ of s, then by definition $\mathrm{IPS}(s)$ is also well-defined.

By bar induction $\mathrm{IPS}(\langle\rangle)$ is well-defined.

Properties of IPS

Order. Like EPS, computation carried out sequentially: value of $\mathsf{IPS}(\langle\rangle)_1$ depends on the value of $\mathsf{IPS}(\langle\rangle)_0$ and so on.

Properties of IPS

Order. Like EPS, computation carried out sequentially: value of $\mathrm{IPS}(\langle\rangle)_1$ depends on the value of $\mathrm{IPS}(\langle\rangle)_0$ and so on.

Well-foundedness. Unlike EPS, underlying tree exists *implicitly*, and cannot be written down in system T.

Properties of IPS

Order. Like EPS, computation carried out sequentially: value of $\mathrm{IPS}(\langle\rangle)_1$ depends on the value of $\mathrm{IPS}(\langle\rangle)_0$ and so on.

Well-foundedness. Unlike EPS, underlying tree exists *implicitly*, and cannot be written down in system T.

Models. Like EPS, well-foundedness of recursion not provable in T. IPS exists in $\mathcal{C}^\omega$, where we require additional condition that R has level 0.

Properties of IPS

Order. Like EPS, computation carried out sequentially: value of $\mathsf{IPS}(\langle\rangle)_1$ depends on the value of $\mathsf{IPS}(\langle\rangle)_0$ and so on.

Well-foundedness. Unlike EPS, underlying tree exists *implicitly*, and cannot be written down in system T.

Models. Like EPS, well-foundedness of recursion not provable in T. IPS exists in $\mathcal{C}^\omega$, where we require additional condition that R has level 0.

Semantics. Operation that computes optimal strategies in unbounded sequential games, only finite part of a play considered by continuity of q.

Modes of bar recursion

IPS sequential, implicit $\longleftrightarrow$ modified bar recursion

modified realizability (2003)

EPS sequential, $|s| \geq \varphi(\hat{s})$ $\longleftrightarrow$ Spector's bar recursion

Dialectica interpretation (1962)

φ constant $\downarrow$ $\mathcal{S}^\omega \not\models$ EPS

PS sequential, $|s| \geq n$ $\leftarrow - - - \rightarrow$ Gödel's primitive recursion

Update recursion (UR)

Idea: Compute a sequence, but not sequentially...

$$\mathrm{IPS}'(s) := s * \lambda k \,.\, \varepsilon_{s*t_k}(\lambda x \,.\, q(\mathrm{IPS}'(s * t_k * x)))$$

where $t_k := [\mathrm{IPS}'(s)](I)$.

Update recursion (UR)

Idea: Compute a sequence, but not sequentially...

$$\mathrm{IPS}'(s) := s * \lambda k \,.\, \varepsilon_{s*t_k}(\lambda x \,.\, q(\mathrm{IPS}'(s * t_k * x)))$$

where $t_k := [\mathrm{IPS}'(s)](l)$.

- Suppose that $u\colon X_{\perp}^{\omega}$ is a partial function.
- Let u_k^x denote update of u with x where k not in domain of u.

Update recursion (UR)

Idea: Compute a sequence, but not sequentially...

$$\mathrm{IPS}'(s) := s * \lambda k \, . \, \varepsilon_{s*t_k}(\lambda x \, . \, q(\mathrm{IPS}'(s * t_k * x)))$$

where $t_k := [\mathrm{IPS}'(s)](l)$.

- Suppose that $u \colon X_{\perp}^{\omega}$ is a partial function.
- Let u_k^x denote update of u with x where k not in domain of u.

$$\mathrm{UR}(u) := u \mathbin{@} \lambda k \, . \, \varepsilon_k(\lambda x \, . \, q(\mathrm{UR}(u_k^x))).$$

Update recursion (UR)

Idea: Compute a sequence, but not sequentially...

$$\mathrm{IPS}'(s) := s * \lambda k \, . \, \varepsilon_{s*t_k}(\lambda x \, . \, q(\mathrm{IPS}'(s * t_k * x)))$$

where $t_k := [\mathrm{IPS}'(s)](I)$.

- Suppose that $u\colon X^{\omega}_{\perp}$ is a partial function.
- Let u^x_k denote update of u with x where k not in domain of u.

$$\mathrm{UR}(u) := u \,@\, \lambda k \, . \, \varepsilon_k(\lambda x \, . \, q(\mathrm{UR}(u^x_k))).$$

Recursion is no longer sequential!

Why is UR well defined?

UR(u) requires us to know value of UR(u_k^x) for all *updates* of u (not just extension), so not clear how we can use bar induction to show that UR is total...

Why is UR well defined?

$\mathrm{UR}(u)$ requires us to know value of $\mathrm{UR}(u_k^x)$ for all *updates* of u (not just extension), so not clear how we can use bar induction to show that UR is total...

An *open predicate* on sequences X^ω is one of the form $A(\alpha) = \exists N\, B([\alpha](N))$.

Definition. Update induction is given by the schema

$$\forall u(\forall n \notin \mathrm{dom}(u), x\ A(u_n^x) \to A(u)) \to \forall u\ A(u).$$

Update induction follows from dependent choice.

Theorem. UR exists in $\mathcal{C}^\omega$ for R of type level 0.

Theorem. UR exists in $\mathcal{C}^\omega$ for R of type level 0.

By CONT, the predicate '$q(\mathrm{UR}(u))$ is total' is equivalent to an open predicate on partial sequences $u \colon X_\perp^\omega$, because if $q(\mathrm{UR}(u))$ is total, it must only look at a finite part of u.

Theorem. UR exists in $\mathcal{C}^\omega$ for R of type level 0.

By CONT, the predicate '$q(\mathrm{UR}(u))$ is total' is equivalent to an open predicate on partial sequences $u\colon X_\perp^\omega$, because if $q(\mathrm{UR}(u))$ is total, it must only look at a finite part of u.

If $q(\mathrm{UR}(u_n^x))$ is total for all updates of u, then $\mathrm{UR}(u)$ and hence $q(\mathrm{UR}(u))$ is total.

Theorem. UR exists in $\mathcal{C}^\omega$ for R of type level 0.

By CONT, the predicate '$q(\mathrm{UR}(u))$ is total' is equivalent to an open predicate on partial sequences $u\colon X^\omega_\perp$, because if $q(\mathrm{UR}(u))$ is total, it must only look at a finite part of u.

If $q(\mathrm{UR}(u^x_n))$ is total for all updates of u, then $\mathrm{UR}(u)$ and hence $q(\mathrm{UR}(u))$ is total.

By update induction $q(\mathrm{UR}(u))$ total for all u, and therefore $\mathrm{UR}(u)$ is total for all u.

Properties of UR

Order. Unlike IPS, computation of individual entries carried out independently. Value of $\mathrm{UR}(s)_0$ does not affect value of $\mathrm{UR}(s)_1$ and so on.

Properties of UR

Order. Unlike IPS, computation of individual entries carried out independently. Value of $\mathrm{UR}(s)_0$ does not affect value of $\mathrm{UR}(s)_1$ and so on.

Computation tree. Like IPS, exists *implicitly*, and cannot be written down in system T.

Properties of UR

Order. Unlike IPS, computation of individual entries carried out independently. Value of $\mathrm{UR}(s)_0$ does not affect value of $\mathrm{UR}(s)_1$ and so on.

Computation tree. Like IPS, exists *implicitly*, and cannot be written down in system T.

Well-foundedness. Like IPS, exists in $\mathcal{C}^\omega$ when outcome type R has level 0.

Properties of UR

Order. Unlike IPS, computation of individual entries carried out independently. Value of $\mathrm{UR}(s)_0$ does not affect value of $\mathrm{UR}(s)_1$ and so on.

Computation tree. Like IPS, exists *implicitly*, and cannot be written down in system T.

Well-foundedness. Like IPS, exists in $\mathcal{C}^\omega$ when outcome type R has level 0.

Semantics. Can be viewed as computing an optimal strategy in games where players 'ignore' the others. *Game theoretic semantics not properly formalised for* UR*!*

Modes of bar recursion

UR — symmetric, implicit	⟷	Berardi/Bezem/Coquand functional *realizability (1998)*
IPS — sequential, implicit	⟷	modified bar recursion *modified realizability (2003)*
EPS — sequential, $\lvert s\rvert \geq \varphi(\hat{s})$	⟷	Spector's bar recursion *Dialectica interpretation (1962)*
↓ φ constant, $\mathcal{S}^\omega \not\models$ EPS		
PS — sequential, $\lvert s\rvert \geq n$	⇠ – – – ⇢	Gödel's primitive recursion

Outline

1. Introduction
   - Bar recursion
   - Overview of talk
2. Modes of bar recursion
   - PS / Finite bar recursion
   - EPS / Spector's bar recursion
   - IPS / Modified bar recursion
   - UR / Berardi-Bezem-Coquand functional
3. Interdefinability results
   - Relationship between EPS and IPS
   - Relationship between IPS and UR

T-definablility

Definition A functional Ψ is T-definable from a functional Φ over a theory $\mathcal{S}$ (we write $\mathcal{S} \vdash \Phi \geq_{\mathrm{T}} \Psi$) if there exists a term t in system T such that $t(\Phi)$ satisfies the defining equation of Ψ provably in $\mathcal{S}$.

T-definablility

Definition A functional Ψ is T-definable from a functional Φ over a theory $\mathcal{S}$ (we write $\mathcal{S} \vdash \Phi \geq_{\mathrm{T}} \Psi$) if there exists a term t in system T such that $t(\Phi)$ satisfies the defining equation of Ψ provably in $\mathcal{S}$.

- To show that $\Phi \geq_{\mathrm{T}} \Psi$ generally involves a (clever) construction in T and a (tedious) verification in $\mathcal{S}$.

T-definablility

Definition A functional Ψ is T-definable from a functional Φ over a theory $\mathcal{S}$ (we write $\mathcal{S} \vdash \Phi \geq_{\mathrm{T}} \Psi$) if there exists a term t in system T such that $t(\Phi)$ satisfies the defining equation of Ψ provably in $\mathcal{S}$.

- To show that $\Phi \geq_{\mathrm{T}} \Psi$ generally involves a (clever) construction in T and a (tedious) verification in $\mathcal{S}$.
- To prove that $\Phi \not\geq_{\mathrm{T}} \Psi$ is hard! Tends to involve model theoretic arguments.

T-definablility

Definition A functional Ψ is T-definable from a functional Φ over a theory $\mathcal{S}$ (we write $\mathcal{S} \vdash \Phi \geq_T \Psi$) if there exists a term t in system T such that $t(\Phi)$ satisfies the defining equation of Ψ provably in $\mathcal{S}$.

- To show that $\Phi \geq_T \Psi$ generally involves a (clever) construction in T and a (tedious) verification in $\mathcal{S}$.
- To prove that $\Phi \not\geq_T \Psi$ is hard! Tends to involve model theoretic arguments.

In general our theory $\mathcal{S}$ will be something like $\mathrm{HA}^\omega + \mathrm{CONT} + \mathrm{BI}$.

IPS T-defines EPS (Oliva/Escardo)

IPS T-defines EPS (Oliva/Escardo)

Key observation: The so-called Spector's search functional

$$\mu_{Sp}(\varphi^{X^\omega \to \mathbb{N}})(\alpha^{X^\omega}) := \text{least } n\ (n \geq \varphi(\widehat{[\alpha](n)}))$$

is *definable* in in system T (even if T cannot prove it exists)!

IPS T-defines EPS (Oliva/Escardo)

Key observation: The so-called Spector's search functional

$$\mu_{Sp}(\varphi^{X^\omega \to \mathbb{N}})(\alpha^{X^\omega}) := \text{least } n \; (n \geq \varphi(\widehat{[\alpha](n)}))$$

is *definable* in in system T (even if T cannot prove it exists)!
Define

$$\alpha^\varphi(i) := \begin{cases} 0 & \text{if } \exists k \leq i+1 \; (k \geq \varphi(\widehat{[\alpha](k)})) \\ \alpha(i) & \text{otherwise} \end{cases}$$

IPS T-defines EPS (Oliva/Escardo)

Key observation: The so-called Spector's search functional

$$\mu_{Sp}(\varphi^{X^\omega \to \mathbb{N}})(\alpha^{X^\omega}) := \text{least } n \; (n \geq \varphi(\widehat{[\alpha](n)}))$$

is *definable* in in system T (even if T cannot prove it exists)!
Define

$$\alpha^\varphi(i) := \begin{cases} 0 & \text{if } \exists k \leq i+1 \; (k \geq \varphi(\widehat{[\alpha](k)})) \\ \alpha(i) & \text{otherwise} \end{cases}$$

If $n = \mu_{Sp}(\varphi)(\alpha)$ then $\alpha^\varphi = \widehat{[\alpha](n-1)}$.

Because n is the least we have $\varphi(\alpha^\varphi) > n - 1$ and so $n \leq \varphi(\alpha^\varphi)$.

IPS T-defines EPS (Oliva/Escardo)

Key observation: The so-called Spector's search functional

$$\mu_{Sp}(\varphi^{X^\omega \to \mathbb{N}})(\alpha^{X^\omega}) := \text{least } n \ (n \geq \varphi(\widehat{[\alpha](n)}))$$

is *definable* in in system T (even if T cannot prove it exists)!
Define

$$\alpha^\varphi(i) := \begin{cases} 0 & \text{if } \exists k \leq i+1 \ (k \geq \varphi(\widehat{[\alpha](k)})) \\ \alpha(i) & \text{otherwise} \end{cases}$$

If $n = \mu_{Sp}(\varphi)(\alpha)$ then $\alpha^\varphi = \widehat{[\alpha](n-1)}$.

Because n is the least we have $\varphi(\alpha^\varphi) > n-1$ and so $n \leq \varphi(\alpha^\varphi)$.

Can encode stopping condition $|s| \geq \varphi(\hat{s})$ into $\tilde{\varepsilon}$, $\tilde{q}$ such that $\text{IPS}^{\tilde{\varepsilon},\tilde{q}}$ T-defines EPS.

EPS does not T-define IPS (Oliva/Escardo)

Kleene (1959). Schemes S1-S9 of computations in higher types.

EPS does not T-define IPS (Oliva/Escardo)

Kleene (1959). Schemes S1-S9 of computations in higher types.

Key observations:

- Spector's bar recursion/EPS is S1-S9 computable in $\mathcal{C}^\omega$.
- The fan functional FAN exists in $\mathcal{C}^\omega$ but is not S1-S9 computable in $\mathcal{C}^\omega$.

EPS does not T-define IPS (Oliva/Escardo)

Kleene (1959). Schemes S1-S9 of computations in higher types.

Key observations:

- Spector's bar recursion/EPS is S1-S9 computable in $\mathcal{C}^\omega$.
- The fan functional FAN exists in $\mathcal{C}^\omega$ but is not S1-S9 computable in $\mathcal{C}^\omega$.

FAN is S1-S9 + IPS computable in $\mathcal{C}^\omega$

EPS does not T-define IPS (Oliva/Escardo)

Kleene (1959). Schemes S1-S9 of computations in higher types.

Key observations:

- Spector's bar recursion/EPS is S1-S9 computable in $\mathcal{C}^\omega$.
- The fan functional FAN exists in $\mathcal{C}^\omega$ but is not S1-S9 computable in $\mathcal{C}^\omega$.

FAN is S1-S9 + IPS computable in $\mathcal{C}^\omega$

$\Rightarrow$ IPS is not S1-S9 computable in $\mathcal{C}^\omega$

EPS does not T-define IPS (Oliva/Escardo)

Kleene (1959). Schemes S1-S9 of computations in higher types.

Key observations:

- Spector's bar recursion/EPS is S1-S9 computable in $\mathcal{C}^\omega$.
- The fan functional FAN exists in $\mathcal{C}^\omega$ but is not S1-S9 computable in $\mathcal{C}^\omega$.

FAN is S1-S9 + IPS computable in $\mathcal{C}^\omega$

$\Rightarrow$ IPS is not S1-S9 computable in $\mathcal{C}^\omega$

$\Rightarrow$ IPS is not T-definable from EPS in any theory that has a model in $\mathcal{C}^\omega$.

Modes of bar recursion

UR symmetric, implicit ⟷ Berardi/Bezem/Coquand functional *realizability (1998)*

IPS sequential, implicit ⟷ modified bar recursion *modified realizability (2003)*

T defines $\mu_{Spector}$ | IPS not S1-S9 definable

EPS sequential, $|s| \geq \varphi(\hat{s})$ ⟷ Spector's bar recursion *Dialectica interpretation (1962)*

φ constant | $\mathcal{S}^\omega \not\models$ EPS

PS sequential, $|s| \geq n$ ⇠ ⇢ Gödel's primitive recursion

UR T-defines IPS (unpublished)

How do we simulate a sequential algorithm like IPS with a non-sequential algorithm like UR?

UR T-defines IPS (unpublished)

How do we simulate a sequential algorithm like IPS with a non-sequential algorithm like UR?

Key idea: Use UR to compute a sequence of *sequences*, i.e. moves of type X^ω. Entries may be computed independently, but using sequence types allows us to store recursive calls.

UR T-defines IPS (unpublished)

How do we simulate a sequential algorithm like IPS with a non-sequential algorithm like UR?

Key idea: Use UR to compute a sequence of *sequences*, i.e. moves of type X^ω. Entries may be computed independently, but using sequence types allows us to store recursive calls.

A fairly complex construction with a long and tedious verification. Won't go into any more detail!

Does IPS T-define UR?

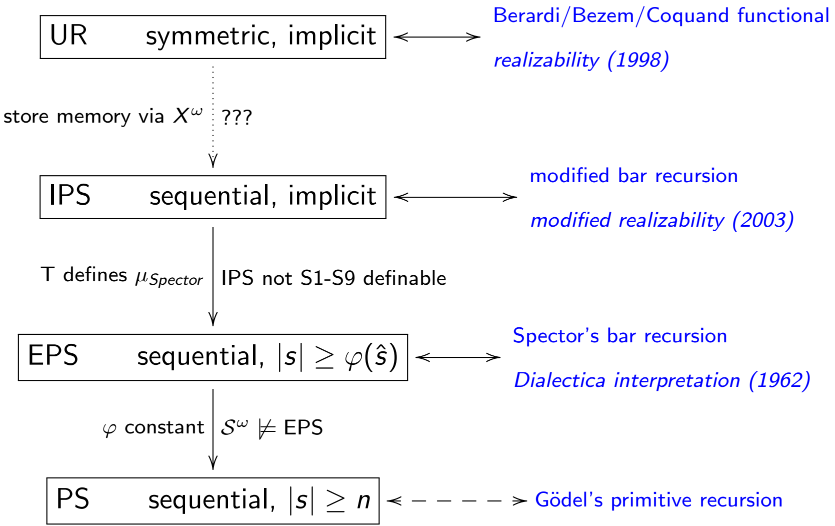
Can we simulate a non-sequential algorithm like UI with a sequential algorithm like IPS?

Does IPS T-define UR?

Can we simulate a non-sequential algorithm like UI with a sequential algorithm like IPS?

Unknown!

Modes of bar recursion



Further questions

The key difference between UR and IPS...

$$\text{IPS} \sim \text{usual order} < \text{on } \mathbb{N}$$
$$\text{UI} \sim \text{discrete order } \mathbb{N}$$

Further questions

The key difference between UR and IPS...

$$\text{IPS} \sim \text{usual order} < \text{on } \mathbb{N}$$
$$\text{UI} \sim \text{discrete order } \mathbb{N}$$

Can we generalise this to associate a form of bar recursion to an arbitrary tree $\prec$?

How is this family of bar recursion functionals related?

New realisers for program extraction?

Direction for future work

Direction for future work

- Complete interdefinability question for main known modes of bar recursion.

Direction for future work

- Complete interdefinability question for main known modes of bar recursion.
- Formulate a uniform framework in which they can be compared, to better understand their behaviour and semantics.

Direction for future work

- Complete interdefinability question for main known modes of bar recursion.
- Formulate a uniform framework in which they can be compared, to better understand their behaviour and semantics.
- Look at new modes of bar recursion. New realizers for proof interpretations? How do they fit into current picture?

Direction for future work

- Complete interdefinability question for main known modes of bar recursion.
- Formulate a uniform framework in which they can be compared, to better understand their behaviour and semantics.
- Look at new modes of bar recursion. New realizers for proof interpretations? How do they fit into current picture?
- Develop some new results and machinery in theory of higher-type computability.

References

Spector's bar recursion
C. Spector. Provably recursive functionals of analysis: A consistency proof of analysis by an extension of principles in current intuitionistic mathematics. *in Recursive Function Theory: Proc. Symposia in Pure Mathematics, 5:1-27*. 1962

Modified bar recursion
U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science, 16(2):163-183*. 2006

UI/BBC **functional**
S. Berardi, M. Bezem and T. Coquand. On the computational content of the axiom of choice. *Journal of Symbolic Logic, 63(2):600-622.* 1998
U. Berger. A computational interpretation of open induction. *in Proc. Nineteenth Annual IEEE Symposium of Logic in Computer Science, 326-334.* 2004

Products of selection functions
M. Escardó and P. Oliva. Selection functions, bar recursion and backward induction. *Mathematical Structures in Computer Science, 20(2):127-168*. 2010
M. Escardó and P. Oliva. Bar recursion and products of selection functions. *Submitted for publication, preprint on authors' webpage*.