Instances of Bar Recursion as Products of Selection Functions

Paulo Oliva

Queen Mary, University of London, UK

(joint work with Martín Escardó)

North American Annual Meeting

Washington, 17 March 2010

- Gödel (1956)

$$\text{Arithmetic} \quad \overset{\text{(dialectica)}}{\Rightarrow} \quad \mathsf{T}$$

- Gödel (1956)

$$\text{Arithmetic} \quad \stackrel{\text{(dialectica)}}{\Rightarrow} \quad \mathrm{T}$$

- Spector (1962)

$$\text{Analysis} \quad \stackrel{\text{(dialectica)}}{\Rightarrow} \quad \mathrm{T} + \text{bar recursion}$$

- Gödel (1956)

$$\text{Arithmetic} \quad \stackrel{\text{(dialectica)}}{\Rightarrow} \quad \text{T}$$

- Spector (1962)

$$\text{Analysis} \quad \stackrel{\text{(dialectica)}}{\Rightarrow} \quad \text{T} + \text{bar recursion}$$

- Bar recursion = recursion on well-founded trees

- Gödel (1956)

$$\text{Arithmetic} \overset{\text{(dialectica)}}{\Rightarrow} \text{T}$$

- Spector (1962)

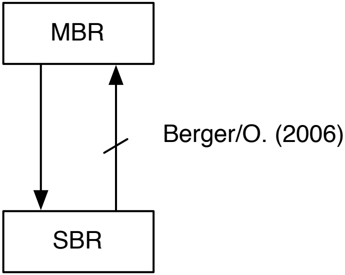
$$\text{Analysis} \overset{\text{(dialectica)}}{\Rightarrow} \text{T} + \text{bar recursion}$$

- Bar recursion = recursion on well-founded trees

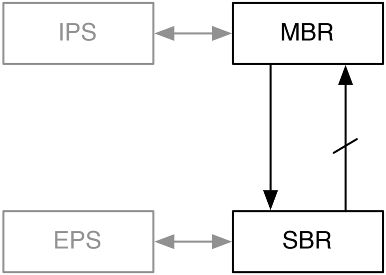
- Berardi et al. (1999) and Berger/O. (2005)

$$\text{Analysis} \overset{\text{(realizability)}}{\Rightarrow} \text{T} + \text{modified BR}$$

Bar Recursion – Overview



Bar Recursion – Overview



Outline

Outline

$$KX \equiv (X \to R) \to R$$

$$JX \equiv (X \to R) \to X$$

$$
\begin{aligned}
KX &\equiv (X \to R) \to R \\
JX &\equiv (X \to R) \to X
\end{aligned}
$$

Call elements $\phi \colon KX$ **generalised quantifiers**

$$KX \equiv (X \to R) \to R$$
$$JX \equiv (X \to R) \to X$$

Call elements $\phi \colon KX$ **generalised quantifiers**

Call elements $\varepsilon \colon JX$ **selection functions**

$$KX \equiv (X \to R) \to R$$
$$JX \equiv (X \to R) \to X$$

Call elements $\phi\colon KX$ **generalised quantifiers**

Call elements $\varepsilon\colon JX$ **selection functions**

$\varepsilon\colon JX$ is a **selection function for** $\phi\colon KX$ if

$$\phi(p) \stackrel{R}{=} p(\varepsilon p)$$

holds for all $p\colon X \to R$

$$KX \equiv (X \to R) \to R$$
$$JX \equiv (X \to R) \to X$$

Call elements $\phi\colon KX$ **generalised quantifiers**

Call elements $\varepsilon\colon JX$ **selection functions**

$\varepsilon\colon JX$ is a **selection function for** $\phi\colon KX$ if

$$\phi(p) \stackrel{R}{=} p(\varepsilon p)$$

holds for all $p\colon X \to R$

Remark: Not all quantifiers have a selection function

Products

Consider two products

$$\begin{aligned} \otimes_{\mathsf{q}} \quad &: \quad KX \times KY \;\rightarrow\; K(X \times Y) \\ \otimes_{\mathsf{s}} \quad &: \quad JX \times JY \;\rightarrow\; J(X \times Y) \end{aligned}$$

Products

Consider two products

$$\begin{array}{rcccl} \otimes_{\mathsf{q}} & : & KX \times KY & \to & K(X \times Y) \\ \otimes_{\mathsf{s}} & : & JX \times JY & \to & J(X \times Y) \end{array}$$

defined as

$$(\phi \otimes_{\mathsf{q}} \psi)(p^{X \times Y \to R}) \overset{R}{:=} \phi(\lambda x^X . \psi(\lambda y^Y . p(x, y)))$$

Products

Consider two products

$$\begin{array}{rcrcl} \otimes_{\mathsf{q}} & : & KX \times KY & \to & K(X \times Y) \\ \otimes_{\mathsf{s}} & : & JX \times JY & \to & J(X \times Y) \end{array}$$

defined as

$$(\phi \otimes_{\mathsf{q}} \psi)(p^{X \times Y \to R}) \stackrel{R}{:=} \phi(\lambda x^X . \psi(\lambda y^Y . p(x, y)))$$

and

$$(\varepsilon \otimes_{\mathsf{s}} \delta)(p^{X \times Y \to R}) \stackrel{X \times Y}{:=} (a, b(a))$$

where $a := \varepsilon(\lambda x^X . p(x, b(x)))$ and $b(x) := \delta(\lambda y^Y . p(x, y))$

$J \mapsto K$

Given sel. fct. $\varepsilon \colon JX$ we define a quantifier $\overline{\varepsilon} \colon KX$ as

$$\overline{\varepsilon} p := p(\varepsilon p)$$

$J \mapsto K$

Given sel. fct. $\varepsilon \colon JX$ we define a quantifier $\overline{\varepsilon} \colon KX$ as

$$\overline{\varepsilon}p := p(\varepsilon p)$$

Theorem

$\overline{\varepsilon \otimes_{\mathsf{s}} \delta} = \overline{\varepsilon} \otimes_{\mathsf{q}} \overline{\delta}$

Product in Practice

Quantifier	**Sel. fct.**	
fix	fix	Bekič's lemma
sup	argsup	Backward induction
$\exists$	ε term	Epsilon method
$\exists$	search	Backtracking
$\overline{\varepsilon}$	ε	Bar recursion

Product in Practice

Quantifier	Sel. fct.	
fix	fix	Bekič's lemma
sup	argsup	Backward induction
$\exists$	ε term	Epsilon method
$\exists$	search	Backtracking
$\overline{\varepsilon}$	ε	Bar recursion

In general, product computes optimal strategies and outcome

Outline

1. Selection Functions
2. Modified Bar Recursion
3. Spector's Bar Recursion

Interpreting Classical Analysis

$$\mathsf{PA}^2 + \mathsf{CA}$$

Interpreting Classical Analysis

$$\mathsf{PA}^2 + \mathsf{CA}$$

$$\Downarrow$$

$$\mathsf{PA}^\omega + \mathsf{AC}_0$$

Interpreting Classical Analysis

$$\mathsf{PA}^2 + \mathsf{CA}$$

$$\Downarrow$$

$$\mathsf{PA}^\omega + \mathsf{AC}_0$$

$$\Downarrow \quad \text{(neg trans)}$$

$$\mathsf{HA}^\omega + \mathsf{AC}_0^N$$

Interpreting Classical Analysis

$$\mathsf{PA}^2 + \mathsf{CA}$$

$$\Downarrow$$

$$\mathsf{PA}^\omega + \mathsf{AC}_0$$

$$\Downarrow \qquad \text{(neg trans)}$$

$$\mathsf{HA}^\omega + \mathsf{AC}_0^N$$

$$\Downarrow$$

$$\underbrace{\mathsf{HA}^\omega + \mathsf{AC}_0}_{\text{semi-int.}} + \mathsf{DNS}$$

Double Negation Shift

The **double negation shift** DNS

$$\forall n \neg\neg A(n) \rightarrow \neg\neg \forall n A(n)$$

Double Negation Shift

The **double negation shift** DNS

$$\forall n \neg\neg A(n) \to \neg\neg \forall n A(n)$$

corresponds to the type

$$\Pi_n K A_n \to K \Pi_n A_n$$

Double Negation Shift

The **double negation shift** DNS

$$\forall n \neg\neg A(n) \rightarrow \neg\neg\forall n A(n)$$

corresponds to the type

$$\Pi_n K A_n \rightarrow K \Pi_n A_n$$

If $\perp \rightarrow A_n$, this is equivalent to

$$\Pi_n J A_n \rightarrow J \Pi_n A_n$$

Double Negation Shift

The **double negation shift** DNS

$$\forall n \neg\neg A(n) \to \neg\neg \forall n A(n)$$

corresponds to the type

$$\Pi_n K A_n \to K \Pi_n A_n$$

If $\bot \to A_n$, this is equivalent to

$$\Pi_n J A_n \to J \Pi_n A_n$$

The type of the **countable product** of selection functions!

Implicitly Controlled Product

Given a family of selection functions $\{\varepsilon_i\}_{i \in \mathbb{N}}$, let

$$\mathsf{IPS}_i(\varepsilon) = \varepsilon_i \otimes_{\mathsf{s}} (\mathsf{IPS}_{i+1}(\varepsilon))$$

Implicitly Controlled Product

Given a family of selection functions $\{\varepsilon_i\}_{i \in \mathbb{N}}$, let

$$\mathsf{IPS}_i(\varepsilon) = \varepsilon_i \otimes_{\mathsf{s}} (\mathsf{IPS}_{i+1}(\varepsilon))$$

Theorem

$$\mathsf{IPS} \quad \mathsf{mr} \quad \forall n^{\mathbb{N}} JA(n) \to J \forall n^{\mathbb{N}} A(n)$$

Implicitly Controlled Product

Given a family of selection functions $\{\varepsilon_i\}_{i \in \mathbb{N}}$, let

$$\mathsf{IPS}_i(\varepsilon) = \varepsilon_i \otimes_{\mathsf{s}} (\mathsf{IPS}_{i+1}(\varepsilon))$$

Theorem

$$\mathsf{IPS} \quad \mathsf{mr} \quad \forall n^{\mathbb{N}} J A(n) \to J \forall n^{\mathbb{N}} A(n)$$

Theorem

IPS *is primitive recursively equivalent to* MBR

Outline

Interpreting Analysis via Dialectica Interpretation

Spector reduced interpretation of DNS to the following:

Spector's Equation

Given $\omega, \varepsilon_{(\cdot)}$ and q find $n, \alpha, p_{(\cdot)}$ satisfying

$$\begin{array}{lcl} n & \stackrel{\mathbb{N}}{=} & \omega(\alpha) \\ \alpha(n) & \stackrel{A_n}{=} & \varepsilon_n(p_n) \\ p_n(\alpha(n)) & \stackrel{R}{=} & q(\alpha) \end{array}$$

Explicitly Controlled Product

A solution to these equations can be computed with

$$\mathsf{EPS}_s(\varepsilon) = \begin{cases} \lambda q.\mathbf{0} & \text{if } \omega_s(\mathbf{0}) < |s| \\ \varepsilon_i \otimes_{\mathsf{s}} \lambda x.(\mathsf{EPS}_{s*x}(\varepsilon)) & \text{otherwise} \end{cases}$$

Product of sel. fcts. with *explicty control* ω on termination

Explicitly Controlled Product

A solution to these equations can be computed with

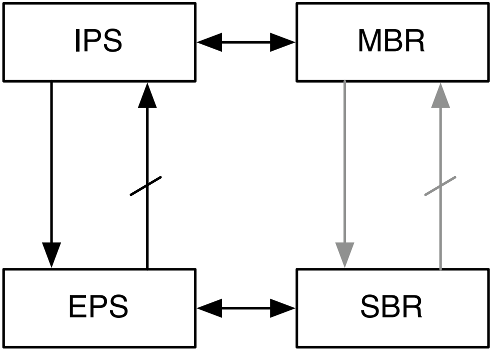
$$\mathsf{EPS}_s(\varepsilon) = \begin{cases} \lambda q.\mathbf{0} & \text{if } \omega_s(\mathbf{0}) < |s| \\ \varepsilon_i \otimes_{\mathsf{s}} \lambda x.(\mathsf{EPS}_{s*x}(\varepsilon)) & \text{otherwise} \end{cases}$$

Product of sel. fcts. with *explicty control* ω on termination

Theorem

EPS *is primitive recursively equivalent to* SBR

Summary



For Details See:

M. Escardó and P. Oliva
Selection functions, bar recursion and backward induction
Mathematical Structures in Computer Science, to appear

M. Escardó and P. Oliva
Instances of bar recursion as iterated products of selection functions and quantifiers
Computability in Europe CiE'2010