Bar Recursion and the Product of Selection Functions

Paulo Oliva

(joint work with Martín Escardó)

Queen Mary, University of London, UK

CiE'2010

Special Session on Proof Theory and Computation

Azores, 4 July 2010

Outline

1. Bar Recursion
2. Selection Functions (and Generalised Quantifiers)
3. Iterated Products and Bar Recursion
4. Three Remarks

Outline

1. Bar Recursion
2. Selection Functions (and Generalised Quantifiers)
3. Iterated Products and Bar Recursion
4. Three Remarks

Background

1958 Gödel's dialectica interpretation of arithmetic

Arithmetic $\mapsto$ System T (primitive recursive functionals)

Background

1958 Gödel's dialectica interpretation of arithmetic

Arithmetic $\mapsto$ System T (primitive recursive functionals)

1959 Kreisel (mod) realizability interpretation of arithmetic

Background

1958 Gödel's dialectica interpretation of arithmetic

Arithmetic $\mapsto$ System T (primitive recursive functionals)

1959 Kreisel (mod) realizability interpretation of arithmetic

1962 Spector extends dialectica interpretation to analysis

Analysis $\mapsto$ System T + **bar recursion**

Background

1958 Gödel's dialectica interpretation of arithmetic

Arithmetic $\mapsto$ System T (primitive recursive functionals)

1959 Kreisel (mod) realizability interpretation of arithmetic

1962 Spector extends dialectica interpretation to analysis

Analysis $\mapsto$ System T + **bar recursion**

1998 Berardi et al. extend Kreisel interpretation to analysis
A new (modifed) form of bar recursion is used

Primitive Recursion and Bar Recursion

Primitive recursion

Define $f(n)$ based on $f(i)$, for $i < n$

Good definition since natural numbers are well-founded

Primitive Recursion and Bar Recursion

Primitive recursion

Define $f(n)$ based on $f(i)$, for $i < n$

Good definition since natural numbers are well-founded

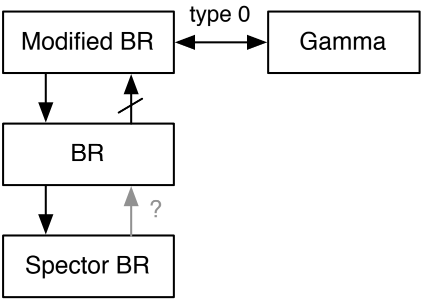
Bar recursion

Define $f(s)$ based on $f(s * x)$, for all extensions $s * x$

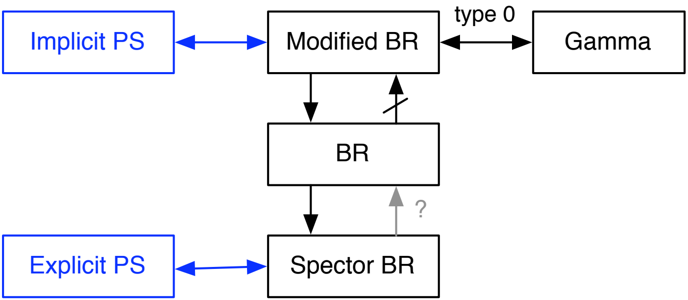
Good definition if tree is well-founded (no infinite branches)

$$f(s) = \begin{cases} g(s) & \text{if } s \text{ is a leaf} \\ h(s, \lambda x. f(s * x)) & \text{otherwise} \end{cases}$$

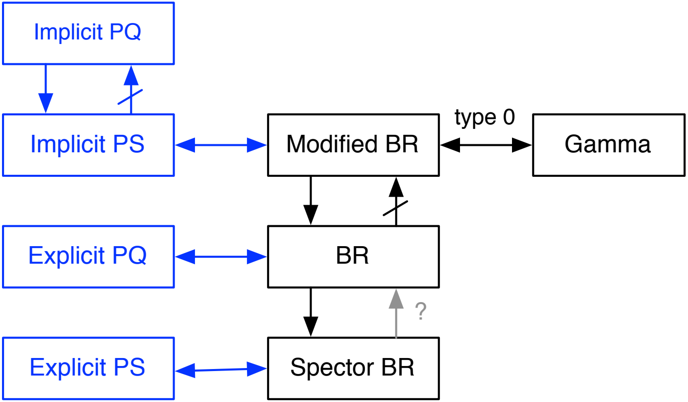
Executive Summary



Executive Summary



Executive Summary



Outline

1. Bar Recursion
2. Selection Functions (and Generalised Quantifiers)
3. Iterated Products and Bar Recursion
4. Three Remarks

Generalised quantifiers

$$\phi : (X \to R) \to R$$

Generalised quantifiers

$$\phi : (X \to R) \to R$$

For instance

Operation	ϕ	:	$(X \to R) \to R$
Quantifiers	$\forall_X, \exists_X$	:	$(X \to \mathbb{B}) \to \mathbb{B}$
Integration	$\int_0^1$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Supremum	$\sup_{[0,1]}$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Limit	$\lim$	:	$(\mathbb{N} \to R) \to R$
Fixed point operator	fix_X	:	$(X \to X) \to X$

Generalised quantifiers

$$\phi : (X \to R) \to R \qquad (\equiv K_R X)$$

For instance

Operation	ϕ	:	$(X \to R) \to R$
Quantifiers	$\forall_X, \exists_X$	:	$(X \to \mathbb{B}) \to \mathbb{B}$
Integration	$\int_0^1$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Supremum	$\sup_{[0,1]}$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Limit	$\lim$	:	$(\mathbb{N} \to R) \to R$
Fixed point operator	fix_X	:	$(X \to X) \to X$

Nested quantifiers $\equiv$ single quantifier on **product space**

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x, y)$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{=} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{\equiv} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

$$\sup_x \int_0^1 p(x,y)dy \quad \stackrel{\mathbb{R}}{\equiv} \quad (\sup \otimes \int)(p^{[0,1]^2 \to \mathbb{R}})$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{\equiv} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

$$\sup_x \int_0^1 p(x,y)dy \quad \stackrel{\mathbb{R}}{\equiv} \quad (\sup \otimes \textstyle\int)(p^{[0,1]^2 \to \mathbb{R}})$$

Definition (Product of Generalised Quantifiers)

Given $\phi \colon KX$ and $\psi \colon KY$ define $\phi \otimes \psi \; : \; K(X \times Y)$

$$(\phi \otimes \psi)(p) :\stackrel{R}{\equiv} \phi(\lambda x^X . \psi(\lambda y^Y . p(x,y)))$$

where $p \colon X \times Y \to R$.

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

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

Definition (Selection Functions)

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

$$\phi(p) = p(\varepsilon p)$$

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

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

Definition (Selection Functions)

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

$$\phi(p) = p(\varepsilon p)$$

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

Definition (Attainable Quantifiers)

A generalised quantifier $\phi \colon KX$ is called **attainable** if it has a selection function $\varepsilon \colon JX$.

For Instance

- $\sup\colon K_{\mathbb{R}}[0,1]$ is an attainable quantifier since

$$\sup(p) = p(\mathrm{argsup}(p))$$

For Instance

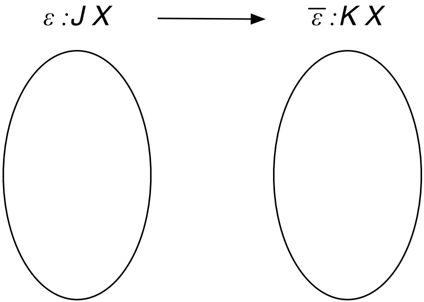
- $\sup\colon K_{\mathbb{R}}[0,1]$ is an attainable quantifier since

$$\sup(p) = p(\mathrm{argsup}(p))$$

- $\mathrm{fix}\colon K_X X$ is an attainable quantifier since

$$\mathrm{fix}(p) = p(\mathrm{fix}(p))$$

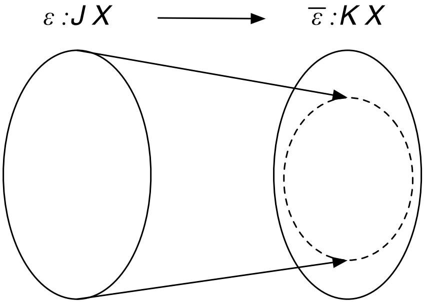
Selection Functions and Generalised Quantifiers



Every selection function $\varepsilon \colon JX$ defines a quantifier $\overline{\varepsilon} \colon KX$

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

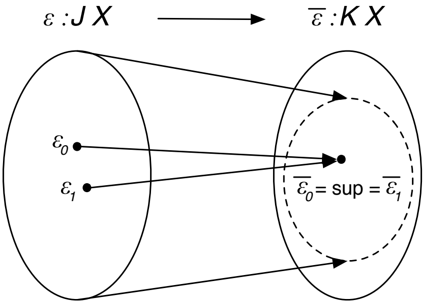
Selection Functions and Generalised Quantifiers



Not all quantifiers are attainable, e.g. $R = \{0, 1\}$

$$\phi(p) \;=\; 0$$

Selection Functions and Generalised Quantifiers



Different ε might define same ϕ, e.g. $X = [0,1]$ and $R = \mathbb{R}$

$$\begin{aligned} \varepsilon_0(p) &= \mu x.\sup p = p(x) \\ \varepsilon_1(p) &= \nu x.\sup p = p(x) \end{aligned}$$

Quantifier Elimination

Suppose $\exists x\, p(x) = p(\varepsilon p)$ and $\forall y\, p(y) = p(\delta p)$.

Quantifier Elimination

Suppose $\exists x\, p(x) = p(\varepsilon p)$ and $\forall y\, p(y) = p(\delta p)$. Then

$$\exists x \forall y\; p(x, y) \quad = \quad \exists x\; p(x, b(x))$$

where

$$b(x) \quad = \quad \delta(\lambda y . p(x, y))$$

Quantifier Elimination

Suppose $\exists x\, p(x) = p(\varepsilon p)$ and $\forall y\, p(y) = p(\delta p)$. Then

$$\begin{aligned} \exists x \forall y\, p(x,y) &= \exists x\, p(x, b(x)) \\ &= p(a, b(a)) \end{aligned}$$

where

$$\begin{aligned} b(x) &= \delta(\lambda y . p(x,y)) \\ a &= \varepsilon(\lambda x . p(x, b(x))). \end{aligned}$$

Definition (Product of Selection Functions)

Given $\varepsilon\colon JX$ and $\delta\colon JY$ define $\varepsilon \otimes \delta\colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \overset{X \times Y}{:=} (a, b(a))$$

where

$$\begin{array}{lcl} a & := & \varepsilon(\lambda x.p(x, b(x))) \\ b(x) & := & \delta(\lambda y.p(x, y)). \end{array}$$

Definition (Product of Selection Functions)

Given $\varepsilon \colon JX$ and $\delta \colon JY$ define $\varepsilon \otimes \delta \colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \overset{X \times Y}{:=} (a, b(a))$$

where

$$\begin{array}{lcl} a & := & \varepsilon(\lambda x.p(x, b(x))) \\ b(x) & := & \delta(\lambda y.p(x, y)). \end{array}$$

Lemma

$\overline{\varepsilon \otimes \delta} = \overline{\varepsilon} \otimes \overline{\delta}$

Why Should We Care?

The product of selection functions...

- computes optimal plays in sequential games
- can be used for backtracking with pruning
- finds strategies in Nash equilibria (backward induction)
- computational content of Tychonoff's theorem
- construction that prod of searchable sets is searchable
- is behind construction in proof of Bekič's lemma
- **solves Spector's equations**
- **realizes classical axiom of choice**

Outline

1. Bar Recursion
2. Selection Functions (and Generalised Quantifiers)
3. Iterated Products and Bar Recursion
4. Three Remarks

Iterated Product: Two Possibilities

Binary product goes from $JX \times JY$ to $J(X \times Y)$.

Can we go from $\Pi_{i \in \mathbb{N}} JX_i$ to $J(\Pi_{i \in \mathbb{N}} X_i)$?

Iterated Product: Two Possibilities

Binary product goes from $JX \times JY$ to $J(X \times Y)$.

Can we go from $\Pi_{i \in \mathbb{N}} JX_i$ to $J(\Pi_{i \in \mathbb{N}} X_i)$?

Yes, in two ways.

Iterated Product: Two Possibilities

Binary product goes from $JX \times JY$ to $J(X \times Y)$.

Can we go from $\Pi_{i \in \mathbb{N}} JX_i$ to $J(\Pi_{i \in \mathbb{N}} X_i)$?

Yes, in two ways.

1. Assume R is discrete (and $\Pi_{i \in \mathbb{N}} X_i \to R$ continuous)

$$\mathsf{IPS}_n(\varepsilon) \overset{J\Pi_{i=n}^{\infty} X_i}{=} \varepsilon_n \otimes \mathsf{IPS}_{n+1}(\varepsilon)$$

Iterated Product: Two Possibilities

Binary product goes from $JX \times JY$ to $J(X \times Y)$.

Can we go from $\Pi_{i\in\mathbb{N}} JX_i$ to $J(\Pi_{i\in\mathbb{N}} X_i)$?

Yes, in two ways.

1. Assume R is discrete (and $\Pi_{i\in\mathbb{N}} X_i \to R$ continuous)

$$\mathsf{IPS}_n(\varepsilon) \stackrel{J\Pi_{i=n}^{\infty} X_i}{=} \varepsilon_n \otimes \mathsf{IPS}_{n+1}(\varepsilon)$$

2. Assume $l(\cdot) \colon R \to \mathbb{N}$ (and $l \circ q$ continuous/majorizable)

$$\mathsf{EPS}^l_n(\varepsilon) \stackrel{J\Pi_{i=n}^{\infty} X_i}{=} \lambda q. \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ (\varepsilon_n \otimes \mathsf{EPS}_{n+1}(\varepsilon))(q) & \text{otherwise.} \end{cases}$$

What about Quantifiers?

1. Schema

$$\mathsf{IPQ}_n(\phi) \stackrel{K\Pi_{i=n}^{\infty} X_i}{=} \phi_n \otimes \mathsf{IPQ}_{n+1}(\phi)$$

not well-defined even when R discrete and q continuous.

What about Quantifiers?

1. Schema

$$\mathsf{IPQ}_n(\phi) \overset{K\Pi_{i=n}^{\infty}X_i}{=} \phi_n \otimes \mathsf{IPQ}_{n+1}(\phi)$$

not well-defined even when R discrete and q continuous.

2. On the other hand (under assumptions above)

$$\mathsf{EPQ}^l_n(\phi) \overset{K\Pi_{i=n}^{\infty}X_i}{=} \lambda q. \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ (\phi_n \otimes \mathsf{EPQ}_{n+1}(\phi))(q) & \text{otherwise} \end{cases}$$

uniquely defines a functional.

Results 1/4

Definition

We denote by $\otimes_d$ a dependent version of $\otimes$ having type

$$JX \times (X \to JY) \to J(X \times Y)$$

Results 1/4

Definition

We denote by $\otimes_d$ a dependent version of $\otimes$ having type

$$JX \times (X \to JY) \to J(X \times Y)$$

Theorem

Iteration of simple product is (prim. rec.) equivalent to iteration of dependent product (same for EPS*)*

$$\mathsf{IPS}_s(\varepsilon) = \varepsilon_s \otimes_d \lambda x^{X_{|s|}}.\mathsf{IPS}_{s*x}(\varepsilon).$$

Proof idea.

Use mapping $(X \to JY) \to J(X \to Y)$. □

Results 2/4

Theorem

$$\mathsf{EPS}^l_n(\varepsilon)(q) = \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ (\varepsilon_n \otimes \mathsf{EPS}^l_{n+1}(\varepsilon))(q) & \text{otherwise} \end{cases}$$

is primitive recursively equivalent to Spector's bar rec., i.e.

$$\mathsf{SBR}^\omega_s(\varepsilon)(q) = \begin{cases} \hat{s} & \text{if } \omega(\hat{s}) < |s| \\ \mathsf{SBR}^\omega_{s*c}(\varepsilon)(q) & \text{otherwise,} \end{cases}$$

where $c = \varepsilon_s(\lambda x^{X_{|s|}}.\mathsf{SBR}^\omega_{s*x}(\varepsilon)(q))$.

Results 3/4

Theorem

IPS *is primitive recursively equivalent to*

$$\mathsf{MBR}_s(\varepsilon)(q) = \varepsilon_s(\lambda x^{X_{|s|}}.q_x(\mathsf{MBR}_{s*x}(\varepsilon)(q_x))),$$

where $\varepsilon_s \colon (X_n \to R) \to \Pi_{i \geq n} X_i$.

Proof idea.

(1) Think of

$$(X_n \to R) \to \Pi_{i \geq n} X_i$$

as **skewed selection functions**.

(2) Define product of such selection functions.

(3) Show binary products are **uniformly** inter-definable. □

Results 4/4

Theorem

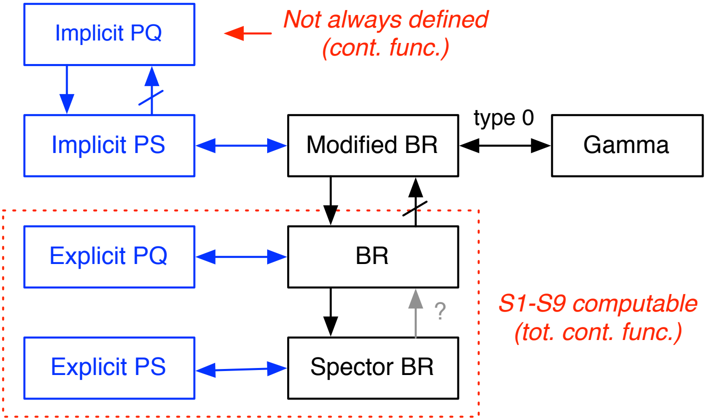
$$\mathsf{EPQ}^l_s(\phi)(q) = \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ (\phi_s \otimes_d \lambda x.\mathsf{EPQ}^l_{s*x}(\phi))(q) & \text{otherwise} \end{cases}$$

is primitive recursively equivalent to bar recursion, i.e.

$$\mathsf{BR}^\omega_s(\phi)(q) = \begin{cases} \hat{s} & \text{if } \omega(\hat{s}) < |s| \\ \phi_s(\lambda x.\mathsf{BR}^\omega_{s*x}(\phi)(q)) & \text{otherwise.} \end{cases}$$

Question. Is simple (non-dependent) EPQ sufficient?

Summary


Implicit PQ
Not always defined
(cont. func.)
Implicit PS
Modified BR
type 0
Gamma
Explicit PQ
BR
?
S1-S9 computable
(tot. cont. func.)
Explicit PS
Spector BR

Outline

1 Bar Recursion

2 Selection Functions (and Generalised Quantifiers)

3 Iterated Products and Bar Recursion

4 Three Remarks

Remark 1: On Strong Monads

K and J are strong monads, i.e. for $T \in \{J, K\}$

- $A \to TA$
- $T^2 A \to TA$
- $(A \wedge TB) \to T(A \wedge B)$

$\overline{(\cdot)} \colon J \to K$ is a monad morphism

J (but not K) also satisfies (used for Main Result 1)

$$(A \to JB) \to J(A \to B).$$

Remark 2: On Negative Translations

J gives rise to a new form of “negative” translation (presented by Martín Escardó on Tuesday)

$$\begin{aligned} KA &\equiv \neg\neg A \\ JA &\equiv (\neg A \to A) \end{aligned}$$

If $\bot \to A$ they are the same, but in ML J is stronger

Modified bar recursion witnesses J-shift

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

and hence double negation (K) shift when $\bot \to A(n)$

Remark 3: On Games and Optimal Plays

General notion of game based on generalised quantifiers

If quantifiers attainable, product s.f. computes optimal play

Arithmetic	$\mapsto$	Finite games of **fixed** length
Analysis	$\mapsto$	Finite games of **unbounded** length

References

M. Escardó and P. Oliva
Selection functions, bar recursion and backward induction
MSCS, 20(2):127-168, 2010

M. Escardó and P. Oliva
The Peirce translation and the double negation shift
LNCS, CiE'2010

M. Escardó and P. Oliva
Computational interpretations of analysis via products of selection functions
LNCS, CiE'2010