Bar Recursion Tutorial

Paulo Oliva

Queen Mary University of London
p.oliva@qmul.ac.uk

Continuity, Computability, Constructivity

Kyoto, 29 May 2026

Historical perspetive

- 1956 - Gödel's Dialectica interpretation of arithmetic in system T
- 1962 - Spector extends T with bar recursion (BR) to interpret analysis
- 1971 - Scarpellini model of BR via continuous functionals
- 1985 - Bezem model $\mathscr{M}$ of BR via majorizable functionals
- 1990 - Berger BR-like algorithm for calculating FAN functional
- 1990s - BR and model $\mathscr{M}$ crucial for Kohlenbach's proof mining
- 1998 - Berardi, Bezem and Coquand novel BR for realizability
- 2005 - Berger/O. modified bar recursion (MBR)
- 2010 - Escardo/O. bar recursion and higher-order games strategies
- 2017 - Escardo/O. monadic BR and the Herbrand interpretation

Quantifier elimination

Suppose we had $\varepsilon_{A(x)}$-terms for every unary quantifier-free formula $A(x)$, i.e.

$$\exists x\, A(x) \quad \Leftrightarrow \quad A(\varepsilon_{A(x)})$$

We can then define $\delta_{A(y)} := \varepsilon_{\neg A(y)}$ so

$$\forall y\, A(y) \quad \Leftrightarrow \quad A(\delta_{A(y)})$$

Indeed:

$$\begin{aligned} \forall y\, A(y) &\Leftrightarrow \neg \exists y \neg A(y) \\ &\Leftrightarrow \neg\neg A(\varepsilon_{\neg A(y)}) \\ &\Leftrightarrow \neg\neg A(\delta_{A(y)}) \\ &\Leftrightarrow A(\delta_{A(y)}) \end{aligned}$$

We can iterate these to reduce every formula to a quantifier-free formula.

Consider the formula $\exists x \forall y\, A(x, y)$

Define $b(x) := \delta_{A(x,y)}$ so

$$\exists x \forall y A(x, y) \quad \Leftrightarrow \quad \exists x A(x, b(x))$$

Define $a := \varepsilon_{A(x,b(x))}$ so

$$\exists x A(x, b(x)) \quad \Leftrightarrow \quad A(a, b(a))$$

Hence

$$\exists x \forall y A(x, y) \quad \Leftrightarrow \quad A(a, b(a))$$

Quantifier elimination

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

(1) define $b(x) := \delta_{A(x,y)}$

(2) define $a := \varepsilon_{A(x,b(x))}$

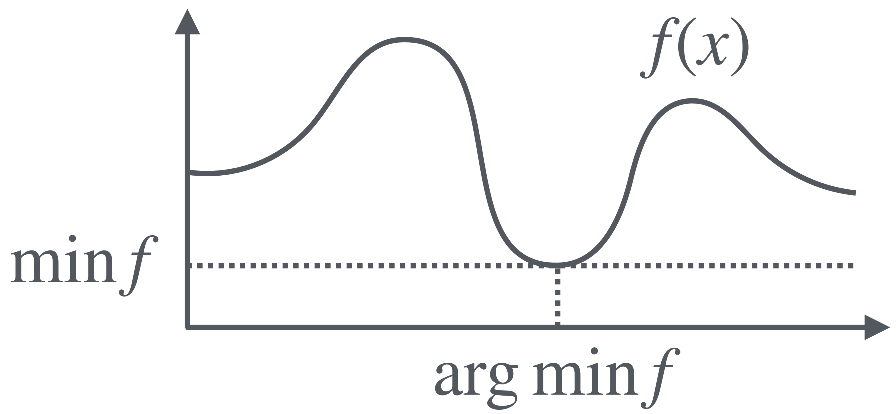
(3) consider $(a, b(a))$

$$A(a, b(a))$$

min-max elimination

Suppose we can calculate arg min of any function $f \colon X \to \mathbb{R}$, i.e.

$$\min_{x \colon X} f(x) = f(\arg\min_{x \colon X}(f(x)))$$



Suppose we can also calculate the arg max of any $g \colon Y \to \mathbb{R}$, i.e.

$$\max_{y \colon Y} g(y) = g(\arg\max_{y \colon Y}(g(y)))$$

We can iterate these to calculate the min-max of any $F \colon X \times Y \to \mathbb{R}$

Define $b(x) := \arg\max_{y \colon Y} F(x, y)$ so

$$\min_{x \colon X} \max_{y \colon Y} F(x, y) = \min_{x \colon X} F(x, b(x))$$

Define $a := \arg\min_{x \colon X} F(x, b(x))$ so

$$\min_{x \colon X} F(x, b(x)) = F(a, b(a))$$

Hence

$$\min_{x \colon X} \max_{y \colon Y} F(x, y) = F(a, b(a))$$

min-max elimination

$$\min_{x \colon X} \max_{y \colon Y} F(x, y)$$

$$\updownarrow$$

(1) define $b(x) := \arg\max_{y \colon Y} F(x, y)$

(2) define $a := \arg\min_{x \colon X} F(x, b(x))$

(3) consider $(a, b(a))$

$$F(a, b(a))$$

Bekić lemma

Suppose we have a fixed point operator fix^X for X, i.e. for any $f \colon X \to X$

$$\mathrm{fix}^X(f) \;=\; f(\mathrm{fix}^X(f))$$

Suppose we also have a fixed point operator fix^Y for Y, i.e. given $g \colon Y \to Y$

$$\mathrm{fix}^Y(g) \;=\; g(\mathrm{fix}^Y(g))$$

We can combine these to obtain fixed points for $F \colon X \times Y \to X \times Y$

Assume projections $\pi_X \colon X \times Y \to X$ and $\pi_Y \colon X \times Y \to Y$

Define $b(x) := \mathrm{fix}^Y(\lambda y \,.\, \pi_Y(F(x, y)))$ so

$$b(x) \;=\; \pi_Y(F(x, b(x)))$$

Define $a := \mathrm{fix}^X(\lambda x \,.\, \pi_X(F(x, b(x)))$ so

$$a \;=\; \pi_X(F(a, b(a)))$$

And, by property of $b(x)$,

$$b(a) \;=\; \pi_Y(F(a, b(a)))$$

Hence $(a, b(a)) = F(a, b(a))$.

Bekić lemma

$$\mathrm{fix}^{X \times Y}(F)$$

(1) define $b(x) := \mathrm{fix}^{Y}(\lambda y \,.\, \pi_Y(F(x, y)))$

(2) define $a := \mathrm{fix}^{X}(\lambda x \,.\, \pi_X(F(x, b(x)))$

(3) consider $(a, b(a))$

$$F(a, b(a)) = (a, b(a))$$

Agenda

- Part I: The binary product of selection functions
  (Selection functions and quantifiers)
- Part II: The finite product of selection functions
  (Higher-order sequential games)
- Part III: The infinite product of selection functions
  (Spector's and modified bar recursion)
- Part IV: The selection monad transformer
  (Turbo-charged bar recursion)

Agenda

- Part I: The binary product of selection functions
  (Selection functions and quantifiers)
- Part II: The finite product of selection functions
  (Higher-order sequential games)
- Part III: The infinite product of selection functions
  (Spector's and modified bar recursion)
- Part IV: The selection monad transformer
  (Turbo-charged bar recursion)

Mathematical Structures in Computer Science

Article contents

Selection functions, bar recursion and backward induction

Published online by Cambridge University Press: **25 March 2010**

MARTÍN ESCARDÓ and PAULO OLIVA

Show author details

Article Metrics

Save PDF | Share | Cite | Rights & Permissions

Abstract

Core share and HTML view are not available for this content. However, as you have access to this content, a full PDF is available via the 'Save PDF' action button.

Bar recursion arises in constructive mathematics, logic, proof theory and higher-type computability theory. We explain bar recursion in terms of sequential games, and show how it can be naturally understood as a generalisation of the principle of backward induction that arises in game theory. In summary, bar recursion calculates optimal plays and optimal strategies, which, for particular games of interest, amount to equilibria. We consider finite games and continuous countably infinite games, and relate the two. The above development is followed by a conceptual explanation of how the finite version of the main form of bar recursion considered here arises from a strong monad of selections functions that can be defined in any cartesian closed category. Finite bar recursion turns out to be a well-known morphism available in any strong monad, specialised to the selection monad.

Quantifiers and selection functions

Definition (Escardó/O.'2010).

We call functionals of the form

$$\varphi \;:\; (X \to R) \to R$$

R-valued quantifiers, or simply **quantifiers**

These generalise

$$\exists, \forall \;:\; (X \to \mathbb{B}) \to \mathbb{B} \qquad \text{and} \qquad \min, \max \;:\; (X \to \mathbb{R}) \to \mathbb{R}$$

We call functionals of the form

$$\varepsilon \;:\; (X \to R) \to X$$

R-valued selection functions, or simply **selection functions**

These generalise

$$\varepsilon, \delta \;:\; (X \to \mathbb{B}) \to X \qquad \text{and} \qquad \arg\min, \arg\max \;:\; (X \to \mathbb{R}) \to X$$

Product of quantifiers

Definition (Escardó/O.'2010).

Given **quantifiers** $\varphi\colon (X \to R) \to R$ and $\psi\colon (Y \to R) \to R$, define

$$\varphi \otimes \psi \ : \ (X \times Y \to R) \to R$$

as follows

$$(\varphi \otimes \psi)(q) := \varphi(\lambda x \, . \, \psi(\lambda y \, . \, q(x, y)))$$

We call this construction the **binary product of quantifiers**

Examples.

- $R = \mathbb{B}$
- $\varphi = \exists^{(X \to \mathbb{B}) \to \mathbb{B}}$ and $\psi = \forall^{(Y \to \mathbb{B}) \to \mathbb{B}}$
- Given a formula $q(x, y)$
- $(\varphi \otimes \psi)(q) \ = \ \exists x \forall y \, q(x, y)$

- $R = \mathbb{R}$
- $\varphi = \min^{(X \to \mathbb{R}) \to \mathbb{R}}$ and $\psi = \max^{(X \to \mathbb{R}) \to \mathbb{R}}$
- Given a function $q(x, y)$
- $(\varphi \otimes \psi)(q) \ = \ \min_x \max_y q(x, y)$

Product of selection functions

Definition (Escardó/O.'2010).

Given **selection functions** $\varepsilon \colon (X \to R) \to X$ and $\delta \colon (Y \to R) \to Y$, define

$$\varepsilon \otimes \delta \ : \ (X \times Y \to R) \to X \times Y$$

as follows

$$(\varepsilon \otimes \delta)(q) := (a, b(a))$$

where

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

We call this construction the **binary product of selection functions**

This is the essential combinatorial construction behind the three examples at the start of the talk (Hilbert's ε-terms, arg min and arg min, and fixed point operators)

Attainability

Definition (Escardó/O.'2010).

We say that a **quantifier** $\varphi \colon (X \to R) \to R$ is **attainable** if for some **selection function** $\varepsilon \colon (X \to R) \to X$ we have

$$\varphi(p) \;=\; p(\varepsilon(p))$$

for all $p \colon X \to R$

Examples.

- $\varepsilon \colon (X \to \mathbb{B}) \to X$ attains $\exists \colon (X \to \mathbb{B}) \to \mathbb{B}$ — $\exists x\, p(x) = p(\varepsilon(p))$
- $\arg\min \colon (X \to \mathbb{R}) \to X$ attains $\min \colon (X \to \mathbb{R}) \to \mathbb{R}$ — $\min(p) = p(\arg\min(p))$
- $\mathrm{fix}^X \colon (X \to X) \to X$ attains itself — $\mathrm{fix}(p) = p(\mathrm{fix}(p))$

Main lemma

Lemma (Escardó/O.'2010).

Suppose

(1) $\varepsilon \colon (X \to R) \to X$ attains $\varphi \colon (X \to R) \to R$

(2) $\delta \colon (Y \to R) \to Y$ attains $\psi \colon (Y \to R) \to R$

Then $\varepsilon \otimes \delta \colon (X \times Y \to R) \to X \times Y$ attains $\varphi \otimes \psi \colon (X \times Y \to R) \to R$

Proof.

$$\begin{aligned}
(\varphi \otimes \psi)(q) &\equiv \varphi(\lambda x \,.\, \psi(\lambda y \,.\, q(x,y))) && \text{(def)} \\
&= \varphi(\lambda x \,.\, q(x, b(x))) && (2) \\
&= q(a, b(a)) && (1) \\
&\equiv q((\varepsilon \otimes \delta)(q)) && \text{(def)}
\end{aligned}$$

where $b(x) = \delta(\lambda y \,.\, q(x,y))$ and $a = \varepsilon(\lambda x \,.\, q(x, b(x)))$

From selection functions to quantifiers

Definition (Escardó/O.'2010).

Every **selection function** $\varepsilon\colon (X \to R) \to X$ gives rise to a quantifier, which we shall denote by $\overline{\varepsilon}\colon (X \to R) \to R$ as

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

So, one can say that selection functions carry more information than quantifiers

Lemma (Escardó/O.'2010).

This mapping from selection functions to quantifiers commutes with the corresponding products

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

for $\varepsilon\colon (X \to R) \to X$ and $\delta\colon (Y \to R) \to Y$

A tale of two monads...

continuation monad

$$K_R X = (X \to R) \to R$$

$$\eta^K_X \colon X \to K_R X$$
$$\beta^K_{X,Y} \colon K_R X \to (X \to K_R Y) \to K_R Y$$
$$\otimes^K_{X,Y} \colon K_R X \times K_R Y \to K_R(X \times Y)$$

selection monad

$$J_R X = (X \to R) \to X$$

$$\eta^J_X \colon X \to J_R X$$
$$\beta^J_{X,Y} \colon J_R X \to (X \to J_R Y) \to J_R Y$$
$$\otimes^J_{X,Y} \colon J_R X \times J_R Y \to J_R(X \times Y)$$

Lemma (Escardó/O.'2010).

Mapping $\overline{(\,\cdot\,)} \colon J_R X \to K_R X$ is a monad morphism

In Haskell...

```
{- Continuation monad -}

newtype K r x = K {quant :: (x -> r) -> r}

instance Functor (K r) where
    fmap = liftM

instance Applicative (K r) where
    pure x = K (\k -> k x)
    (<*>) = ap

instance Monad (K r) where
    return = pure
    phi >>= f = K (\p -> quant phi (\gamma -> quant (f gamma) p))
```

Continuation monad

```
{- Selection monad -}

newtype J r x = J {selection :: (x -> r) -> x}

instance Functor (J r) where
   fmap = liftM

instance Applicative (J r) where
    pure x = J (\p -> x)
    (<*>) = ap

instance Monad (J r) where
    return = pure
    e >>= f = J (\p -> b p (a p))
        where
            a p = selection e $ (\x -> p (b p x))
            b p x = selection (f x) p
```

Selection monad

Realizability, Dialectica and selection functions

Dialectica interpretation of $\neg\neg A$

The Dialectica interpretation of

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

is:

$$\exists \varepsilon \forall p\, A(\varepsilon(p), p(\varepsilon(p)))$$

where

$$\varepsilon : (X \to Y) \to X$$

A selection function!

(classical) realizability of $\neg\neg A$

Assume $R(y)$ are realizers for $\bot$, then, the classical realizability of

$$\neg\neg\exists x^X A^N(x)$$

is:

$$\exists \varphi \forall p (\forall x (A^N(x) \to R(p(x))) \to R(\varphi(p)))$$

where

$$\varphi : (X \to Y) \to Y$$

since we have $\mathrm{efq} : \bot \to \exists x^X A^N(x)$

$$\mathrm{efq} \circ \varphi : (X \to Y) \to X$$

Agenda

- Part I: The binary product of selection functions
  (Selection functions and quantifiers)
- Part II: The finite product of selection functions
  (Higher-order sequential games)
- Part III: The infinite product of selection functions
  (Spector's and modified bar recursion)
- Part IV: The selection monad transformer
  (Turbo-charged bar recursion)

The finite product of selection functions

Definition (Escardó/O.'2010).

We can iterate the binary product of quantifiers and selection function to obtain finite products of these:

Given **quantifiers** $\varphi_i \colon (X_i \to R) \to R$, for $0 \leq i < n$, we can define

$$\left(\bigotimes_{i=0}^{n-1} \varphi_i\right) := \varphi_0 \otimes \varphi_1 \otimes \ldots \otimes \varphi_{n-1} \quad : \quad \left(\prod_{i<n} X_i \to R\right) \to R$$

Given **selection functions** $\varepsilon_i \colon (X_i \to R) \to X_i$, for $0 \leq i < n$, we can define

$$\left(\bigotimes_{i=0}^{n-1} \varepsilon_i\right) := \varepsilon_0 \otimes \varepsilon_1 \otimes \ldots \otimes \varepsilon_{n-1} \quad : \quad \left(\prod_{i<n} X_i \to R\right) \to \prod_{i<n} X_i$$

Theorem (Escardó/Powell/O.'2011).

These finite products are equivalent to Gödel primitive recursion

Finite bar recursion

Definition (Escardó/O.'2010).

The **product of quantifiers construction**

$$\left(\bigotimes_{i<n} \varphi_i\right) : \left(\prod_{i<n} X_i \to R\right) \to R$$

is equivalent to the following **finite bar recursion** construction

$$\mathrm{fBR}(\varphi, q, n, s) =_R \begin{cases} q(\hat{s}) & \text{if } n < |s| \\ \varphi_{|s|}(\lambda x \,.\, \mathrm{fBR}(\varphi, q, n, s * x)) & \text{if } n \geq |s| \end{cases}$$

where $q \colon \Pi_i X_i \to R$ and $\hat{s} \colon \Pi_i X_i$ is the infinite extension of the finite sequence $s \colon \Pi_{i<n} X_i$ with some fixed constant value (e.g. 0)

Finite bar recursion

Definition (Escardó/O.'2010).

The <u>product of selection functions construction</u>

$$\left(\bigotimes_{i<n} \varepsilon_i\right) : \left(\prod_{i<n} X_i \to R\right) \to \prod_{i<n} X_i$$

is equivalent to the following <u>(special) finite bar recursion</u> construction

$$\mathrm{sBR}(\varepsilon, q, n, s) =_{\Pi_i X_i} \begin{cases} \hat{s} & \text{if } n < |s| \\ \mathrm{sBR}(\varphi, q, n, s * a) & \text{if } n \geq |s| \end{cases}$$

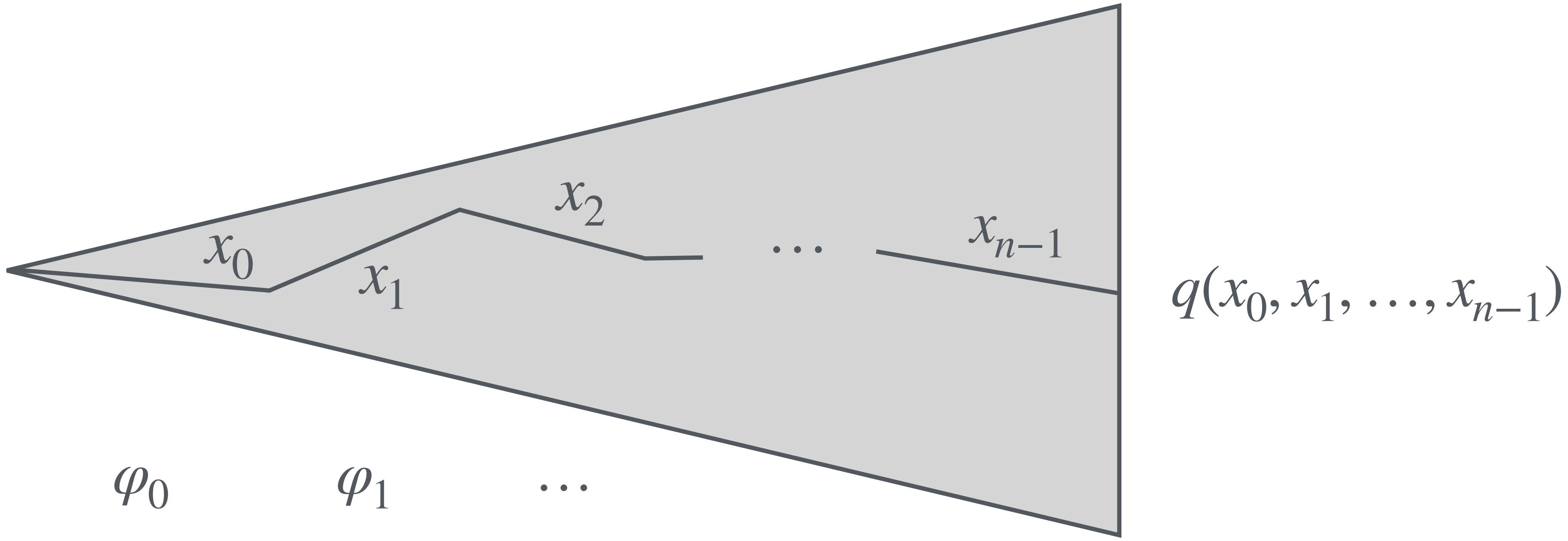
where $a =_{X_{|s|}} \varepsilon_{|s|}(q \circ b)$ and $b(x) =_{\Pi_i X_i} \mathrm{sBR}(\varphi, q, n, s * x)$

Higher-order games

Definition (Escardó/O.'2010).

A sequential **higher-order game** of n-rounds is a tuple $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$ where:

- R is the type of outcomes
- X_i is the type of moves at round i
- $q\colon \Pi_{i<n} X_i \to R$ is the outcome function
- $\varphi_i\colon (X_i \to R) \to R$ are quantifiers describing the "goal" at round i



Higher-order games

Definition (Escardó/O.'2010).

A sequential **higher-order game** of n-rounds is a tuple $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$ where:

- R is the type of outcomes
- X_i is the type of moves at round i
- $q \colon \Pi_{i<n} X_i \to R$ is the outcome function
- $\varphi_i \colon (X_i \to R) \to R$ are quantifiers describing the "goal" at round i

Example 1 (maximising n-players).

- $R = \mathbb{R}^n$
- X_i some choice of actions for player i
- $q(\vec{x}) \colon \mathbb{R}^n$ the game's payoff function (i-th player receives $\pi_i(q(\vec{x}))$)
- $\varphi_i = \max_i$, maximising players (of type $(X_i \to \mathbb{R}^n) \to \mathbb{R}^n$)

Higher-order games

Definition (Escardó/O.'2010).

A sequential **higher-order game** of n-rounds is a tuple $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$ where:

- R is the type of outcomes
- X_i is the type of moves at round i
- $q\colon \Pi_{i<n} X_i \to R$ is the outcome function
- $\varphi_i\colon (X_i \to R) \to R$ are quantifiers describing the "goal" at round i

Example 2 (SAT).

- $R = \mathbb{B}$
- $X_i = \mathbb{B}$
- $q(\vec{x})\colon \mathbb{B}$ some Boolean formula over n atomic propositions
- $\varphi_i = \exists$, existential quantifiers (of type $(\mathbb{B} \to \mathbb{B}) \to \mathbb{B}$)

Higher-order games

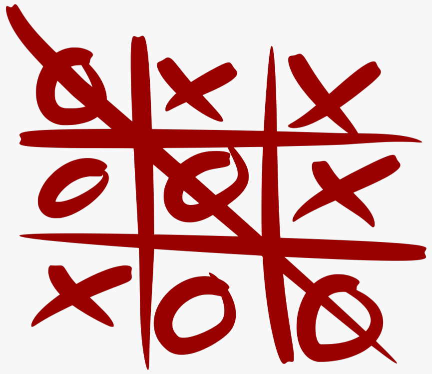
Definition (Escardó/O.'2010).

A sequential **higher-order game** of n-rounds is a tuple $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$ where:

- R is the type of outcomes
- X_i is the type of moves at round i
- $q \colon \Pi_{i<n} X_i \to R$ is the outcome function
- $\varphi_i \colon (X_i \to R) \to R$ are quantifiers describing the "goal" at round i

Example 3 (tic-tac-toe).

- $R = \{-1, 0, 1\}$
- $X_i = \{1, \ldots, 9\}$ — grid position
- $q(\vec{x})$ decides outcome of game
- $\varphi_{2i} = \max$ and $\varphi_{2i+1} = \min$



Higher-order games

Definition (Escardó/O.'2010).

A sequential **higher-order game** of n-rounds is a tuple $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$ where:

- R is the type of outcomes
- X_i is the type of moves at round i
- $q \colon \Pi_{i<n} X_i \to R$ is the outcome function
- $\varphi_i \colon (X_i \to R) \to R$ are quantifiers describing the "goal" at round i

Definition (Escardó/O.'2010).

Given a higher-order game of n-rounds $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$

- $(\bigotimes_{i<n} \varphi_i)(q) \colon R$ is the **optimal outcome** of the game
- A strategy $\sigma_i \colon \Pi_{j<i} X_j \to X_i$ is **optimal** if for all $x_1, \ldots, x_{i-1}$

$$q(x_1, \ldots, x_{i-1}, \sigma) = \varphi_i(\lambda x_i' \,.\, q(x_1, \ldots, x_{i-1}, x_i', \sigma))$$

Main result

Theorem (Escardó/O.'2010).

Given a higher-order game of n-rounds $(R, (X_i)_{i<n}, q, (\varphi_i)_{i<n})$, if each game quantifier

$$\varphi_i \colon (X_i \to R) \to R$$

is attainable by a selection function

$$\varepsilon_i \colon (X_i \to R) \to X_i$$

then **product of these selection functions calculates an optimal play** in the game, i.e.

$$\left(\bigotimes \varepsilon_i\right)(q) = \text{optimal play}$$

and hence, can be used to define **optimal (sub-game perfect) optimal strategies**

The product of selection functions generalises the "**backward induction**" algorithm from Game Theory (the particular case when $R = \mathbb{R}^n$ and $\varphi_i = \max_i$)

Higher-order games

R — outcomes

X_i — moves

$q\colon \Pi_i X_i \to R$ — outcome function

$\varphi_i\colon (X_i \to R) \to R$ — goal functions

$\varepsilon_i\colon (X_i \to R) \to X_i$ — selection functions

$\left(\bigotimes \varphi_i\right)(q)$ — optimal outcome

$\left(\bigotimes \varepsilon_i\right)(q)$ — optimal strategy

A SAT solver

```
ghci> :type sequence
sequence :: (Monad m) => [m a] -> m [a]
```

```
sequence :: [J r a] -> J r [a]
```

```
{- Selection monad -}
newtype J r x = J {selection :: (x -> r) -> x}
   ...

-- Selection function for predicates over one variable:
one_choice :: J Bool Bool
one_choice = J (\p -> p True)

-- Selection function for predicates over n variables:
n_choices :: Int -> J Bool [Bool]
n_choices n = sequence (replicate n one_choice)

-- Find a satisfying assignment for predicates with n variables:
sat :: Int -> ([Bool] -> Bool) -> [Bool]
sat n = (take n) . (selection (n_choices n))
```


Coffee
BREAK

Agenda

- Part I: The binary product of selection functions
  (Selection functions and quantifiers)
- Part II: The finite product of selection functions
  (Higher-order sequential games)
- Part III: The infinite product of selection functions
  (Spector's and modified bar recursion)
- Part IV: The selection monad transformer
  (Turbo-charged bar recursion)

Quantifier elimination

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

$$\updownarrow$$

(1) define $b(x) := \delta_{A(x,y)}$

(2) define $a := \varepsilon_{A(x,b(x))}$

(3) consider $(a, b(a))$

$$A(a, b(a))$$

Product of selection functions

Definition (Escardó/O.'2010).

Given **selection functions** $\varepsilon\colon (X \to R) \to X$ and $\delta\colon (Y \to R) \to Y$, define

$$\varepsilon \otimes \delta \ : \ (X \times Y \to R) \to X \times Y$$

as follows

$$(\varepsilon \otimes \delta)(q) := (a, b(a))$$

where

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

We call this construction the **binary product of selection functions**

This is the essential combinatorial construction behind the three examples at the start of the talk (Hilbert's ε-terms, arg min and arg min, and fixed point operators)

Infinite pigeonhole principle

Infinite pigeonhole principle (IPP).

Any colouring of $\mathbb{N}$ with n colours will use one of the colours infinitely often

$$\forall n^{\mathbb{N}} \forall f^{\mathbb{N} \to [n]} \exists c < n \forall i^{\mathbb{N}} \exists j > i \, (c = f(j)))$$

ND-interpretation of IPP.

Given $n^{\mathbb{N}}$, colouring $f^{\mathbb{N} \to [n]}$ and selection functions $\varepsilon_i^{(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}}$, for $i < n$, find colour $c < n$ and $p^{\mathbb{N} \to \mathbb{N}}$ such that

$$p(\varepsilon_i p) > \varepsilon_i p \quad \text{and} \quad c = f(p(\varepsilon_i p))$$

Theorem (O.'2006, Escardó/O'2010).

The finite product of selection functions (finite BR) solves IPP$^{\text{ND}}$, namely

$$c = f(\max((\otimes_{i<n} \varepsilon_i)(\max))$$

Binary, finite, infinite...

Given two selection functions $\varepsilon : J_R X$ and $\delta : J_R Y$ we have defined **binary product**:

$$\varepsilon \otimes \delta : J_R(X \times Y)$$

For every $n : \mathbb{N}$, given finite sequence $(\varepsilon_i)_{i<n} : \Pi_{i<n} J_R X_i$ we have defined **finite product**:

$$\bigotimes_{i<n} \varepsilon_i : J_R \Pi_{i<n} X_i$$

Given infinite sequence $(\varepsilon_i)_{i:\mathbb{N}} : \Pi_{i:\mathbb{N}} J_R X_i$ we now consider **infinite product**:

$$\bigotimes_{i:\mathbb{N}} \varepsilon_i : J_R \Pi_{i:\mathbb{N}} X_i$$

```haskell
-- Selection function for one variable:
one_choice :: J Bool Bool
one_choice = J (\p -> p True)

-- Selection function for Cantor space:
cantor_choice :: J Bool [Bool]
cantor_choice = sequence (repeat one_choice)
```

selection function for Cantor space

```haskell
-- Existential quantifier for Cantor space
forsome :: ([Bool] -> Bool) -> Bool
forsome q = q (selection cantor_choice q)

-- Universal quantification for Cantor space
forevery :: ([Bool] -> Bool) -> Bool
forevery q = not (forsome (not . q))
```

exists and forall quantifiers for Cantor space

```haskell
-- Calculate FAN functional
-- Ulrich Berger (1990)
fan :: ([Bool] -> Bool) -> Int
fan q = go 0
 where
  go n | constant n q = n
       | otherwise = go (n + 1)

  constant :: Int -> ([Bool] -> Bool) -> Bool
  constant n q = forevery (\xs ->
                 forevery (\ys ->
                   take n xs /= take n ys ||
                   q xs == q ys))
```

modulus of uniform continuity for Cantor space

Spector 1962

The Principles.

$$\begin{aligned} \mathbf{LEM} &: \forall n^{\mathbb{N}}(A_n \vee \neg A_n) \\ \mathbf{CA} &: \exists f^{\mathbb{N}\to\mathbb{B}} \forall n^{\mathbb{N}}(f(n) \leftrightarrow A_n) \\ \mathbf{AC}_0 &: \forall n^{\mathbb{N}} \exists x^{\sigma} A_n(x) \to \exists \alpha^{\sigma^{\mathbb{N}}} \forall n\, A_n(\alpha_n) \\ \mathbf{DNS} &: \forall n^{\mathbb{N}} \neg\neg A_n \to \neg\neg \forall n^{\mathbb{N}} A_n \end{aligned}$$

Theorem (Spector'62).

- If $\mathbf{LEM} \vdash A$ then $\vdash A^N$ (negative translation)
- $\mathbf{LEM} + \mathbf{AC}_0 \vdash \mathbf{CA}$
- $\mathbf{DNS} + \neg\neg\mathbf{AC}_0 \vdash (\mathbf{AC}_0)^N$
- Bar recursion solves Dialectica interpretation of **DNS**

Challenge.

Functional (Dialectica) interpretation of

$$\forall n^{\mathbb{N}} \neg\neg \exists x^X \forall y^R A_n(x, y) \to \neg\neg \exists \alpha^{X^{\mathbb{N}}} \forall n, y\, A_n(\alpha_n, y)$$

Problem A (solves challenge).

Given $\varepsilon_n^{(X \to R) \to X}$, $\omega^{X^{\mathbb{N}} \to \mathbb{N}}$ and $q^{X^{\mathbb{N}} \to R}$ find $\alpha^{X^{\mathbb{N}}}$, $n^{\mathbb{N}}$ and $p^{X \to R}$ such that

$$A_n(\varepsilon_n(p), p(\varepsilon_n(p))) \;\to\; A_m(\alpha_m, q(\alpha))$$

where $m = \omega(\alpha)$.

Problem B (solution for B gives solution for A).

global selection

Given $\varepsilon_n^{(X \to R) \to X}$, $\omega^{X^{\mathbb{N}} \to \mathbb{N}}$ and $q^{X^{\mathbb{N}} \to R}$ find $\alpha^{X^{\mathbb{N}}}$, $n^{\mathbb{N}}$ and $p^{X \to R}$ such that

local selections

$$\alpha_n = \varepsilon_n(p_n)$$
$$p_n(\alpha_n) = q(\alpha)$$

for $n \leq \omega(\alpha)$.

Interpreting $\mathbf{AC}_0$ (classically)

Problem B (solution for B gives solution for A).

Given $\varepsilon_n^{(X \to R) \to X}$, $\omega^{X^{\mathbb{N}} \to \mathbb{N}}$ and $q^{X^{\mathbb{N}} \to R}$ find $\alpha^{X^{\mathbb{N}}}$, $n^{\mathbb{N}}$ and $p^{X \to R}$ such that

$$\alpha_n = \varepsilon_n(p_n)$$
$$p_n(\alpha_n) = q(\alpha)$$

for $n \leq \omega(\alpha)$.

iterated product of selection function

Solution for B (Spector bar recursion).

Define $\mathrm{SBR}(s) : X^{\mathbb{N}}$ as follows ($s : X^*$ and $\hat{s} = s * 0 \ldots$)

$$\mathrm{SBR}(s) = \begin{cases} \hat{s} & \text{if } \omega(\hat{s}) < |s| \\ \mathrm{SBR}(s * a) & \text{if } \omega(\hat{s}) \geq |s| \end{cases}$$

where $a =_X \varepsilon_{|s|}(q \circ b_s)$ and $b_s(x) =_{\Pi_i X_i} \mathrm{SBR}(s * x)$.

Take $\alpha = \mathrm{SBR}(\langle\ \rangle)$ and $p_n = q \circ b_{\overline{\alpha}(n)}$, where $\overline{\alpha}(n) = \langle \alpha_0, \alpha_1, \ldots, \alpha_{n-1} \rangle$.

Interpreting $\mathbf{AC}_0$ (classically)

Solution for B (Spector bar recursion).

Define $\mathrm{SBR}(s) : X^{\mathbb{N}}$ as follows ($s : X^*$ and $\hat{s} = s * 0 \ldots$)

$$\mathrm{SBR}(s) = \begin{cases} \hat{s} & \text{if } \omega(\hat{s}) < |s| \\ \mathrm{SBR}(s * a) & \text{if } \omega(\hat{s}) \geq |s| \end{cases}$$

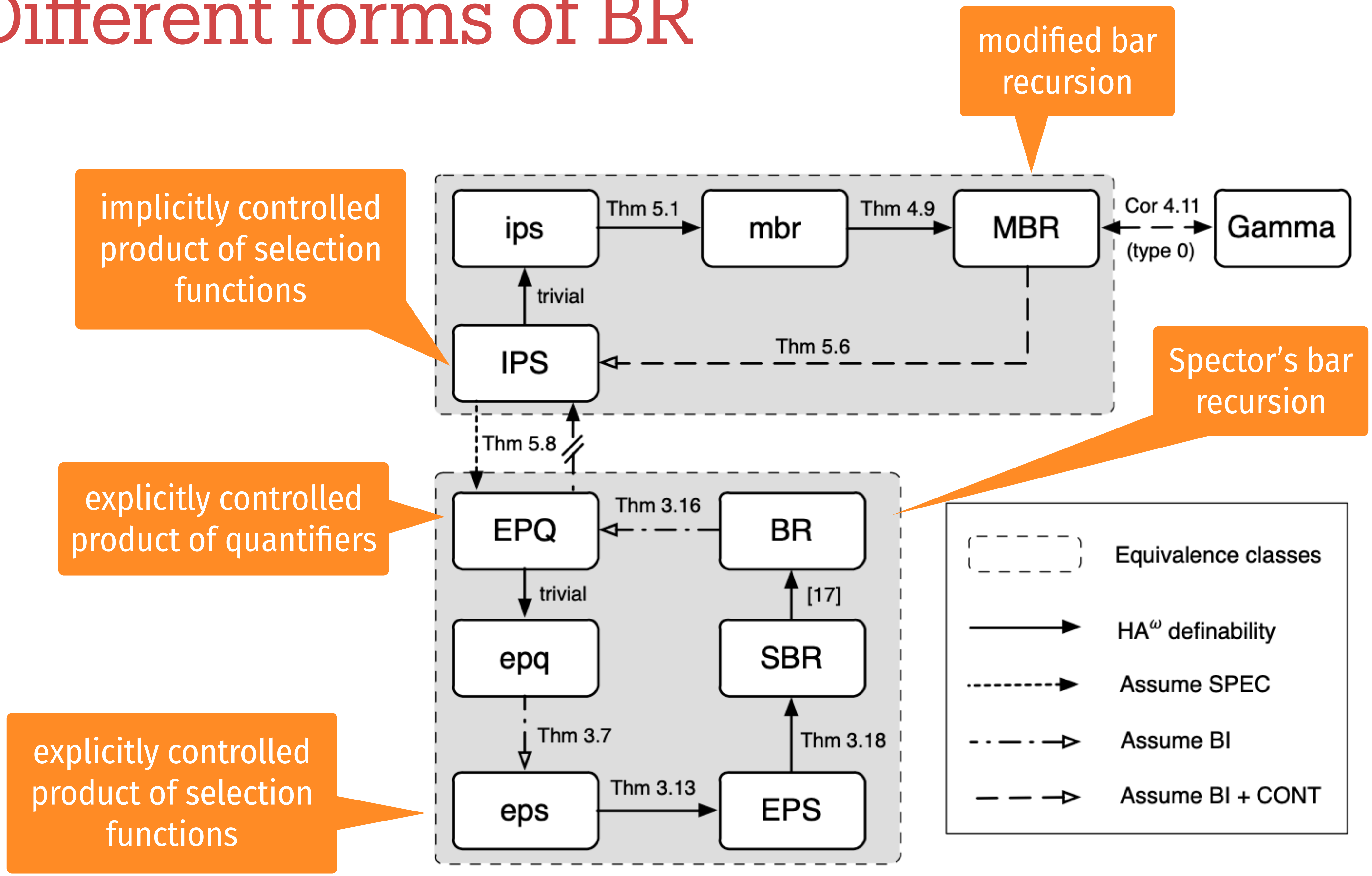
where $a =_X \varepsilon_{|s|}(q \circ b_s)$ and $b_s(x) =_{\Pi_i X_i} \mathrm{SBR}(s * x)$.

Take $\alpha = \mathrm{SBR}(\langle\ \rangle)$ and $p_n = q \circ b_{\overline{\alpha}(n)}$, where $\overline{\alpha}(n) = \langle \alpha_0, \alpha_1, \ldots, \alpha_{n-1} \rangle$.

Main Lemmas (Spector'62).

- If $\alpha = \mathrm{SBR}(\langle\ \rangle)$ then $\alpha = \mathrm{SBR}(\overline{\alpha}(n))$, for all n (hence $p_n(\alpha_n) = q(\alpha)$)
- If $\alpha = \mathrm{SBR}(\langle\ \rangle)$ and $n = \omega(\alpha)$ then $\omega(\overline{\alpha, n}) \geq n$
- If $\alpha = \mathrm{SBR}(\langle\ \rangle)$ and $n = \omega(\alpha)$ then $\alpha_n = \varepsilon_n(q \circ b_{\overline{\alpha}(n)})$ (hence $\alpha_n = \varepsilon_n(p_n)$)

Different forms of BR



M. Escardó and P. Oliva, Bar recursion and the product of selection functions
The Journal of Symbolic Logic, 80(1):1-28, 2015

Theorem (Escardó/O.'2015).

There are two main forms of (unbounded) bar recursion.

Given selection functions

$$\varepsilon_s \colon (X_{|s|} \to R) \to X_{|s|}$$
$$q \colon \Pi_i X_i \to R$$
$$\omega \colon \Pi_i X_i \to \mathbb{N}$$

Spector's "explicitly controlled" bar recursion is defined as

$$\mathrm{SBR}(s) = \begin{cases} \mathbf{0} & \text{if } \omega(\hat{s}) < |s| \\ a * \mathrm{SBR}(s * a) & \text{if } \omega(\hat{s}) \geq |s| \end{cases}$$

where $a =_X \varepsilon_{|s|}(q \circ b_s)$ and $b_s(x) =_{\Pi_i X_i} s * x * \mathrm{SBR}(s * x)$

And the **modified "implicitly controlled" bar recursion**

$$\mathrm{MBR}(s) = a * \mathrm{MBR}(s * a)$$

type R needs to be discrete (e.g. $\mathbb{N}$)

where $a =_X \varepsilon_{|s|}(q \circ b_s)$ and $b_s(x) =_{\Pi_i X_i} s * x * \mathrm{MBR}(s * x)$

The model $\mathscr{M}$

Theorem (Bezem'1985).

- The full set-theoretic type structure $\mathscr{S}$ is NOT a model of T + BR
- The model $\mathscr{M}$ of **strongly majorizable functionals** is a model of T + BR

Model $\mathscr{M}$ construction.

Define the **strong majorizability relations** for each type X inductively as:

$$n \leq^*_{\mathbb{N}} m \;:\equiv\; n \leq m$$

$$f \leq^*_{X \to Y} g \;:\equiv\; \forall a^X \forall x \leq^*_X a (fx \leq^*_Y ga \wedge gx \leq^*_Y ga)$$

Then define:

$$\mathscr{M}_{\mathbb{N}} \;:\equiv\; \mathbb{N}$$

$$\mathscr{M}_{X \to Y} \;:\equiv\; \{f \colon \mathscr{M}_X \to \mathscr{M}_Y \;:\; \exists g^{\mathscr{M}_X \to \mathscr{M}_Y} (f \leq^*_{X \to Y} g)\}$$

Agenda

- Part I: The binary product of selection functions
  (Selection functions and quantifiers)
- Part II: The finite product of selection functions
  (Higher-order sequential games)
- Part III: The infinite product of selection functions
  (Spector's and modified bar recursion)
- Part IV: The selection monad transformer
  (Turbo-charged bar recursion)

Continuation and selection monads

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

quantifier

$$\varepsilon \colon (X \to R) \to X$$

selection function

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

attainability
(ε attains ϕ)

Selection monad transformer

$$T : \text{Type} \to \text{Type}$$

strong monad

$$\alpha : TR \to R$$

T-algebra

$$\varepsilon : (X \to R) \to TX$$

monadic selection

M. Escardó and P. Oliva
The Herbrand functional interpretation of the double negation shift
The Journal of Symbolic Logic, 82(2):590-607, 2017

Generalised product of selection functions

Theorem (Escardó/O.'2017).

For any strong monad M let $J_R^M X \equiv (X \to MR) \to MX$ (free algebra case)

$J_R^M \colon \mathrm{Type} \to \mathrm{Type}$ is also a **strong monad**. In particular, we also have

$$\otimes \colon J_R^M X \times J_R^M Y \to J_R^M (X \times Y).$$

Given $\varepsilon \colon J_R^M X$ and $\delta \colon J_R^M Y$ and $q \colon X \times Y \to MR$ let

$$b(x^X) :=_{MY} \delta(\lambda y^Y . q(x, y))$$

and

$$a :=_{MX} \varepsilon(\lambda x . q^\dagger(x, \kappa(x))$$

where $q^\dagger \colon X \times MY \to MR$. Then

$$(\varepsilon \otimes \delta)(q) =_{M(X \times Y)} a \otimes_M b^\dagger(a)$$

where $b^\dagger \colon MX \to MY$

Generalised product of selection functions

Theorem (Escardó/O.'2017).

For any strong monad M let $J_R^M X \equiv (X \to MR) \to MX$

J_R^M: Type $\to$ Type is also a **strong monad**. In particular, we also have

$$\otimes \colon J_R^M X \times J_R^M Y \to J_R^M (X \times Y).$$

Game-theoretic interpretation.

Consider the finite powerset monad ($MX = X^*$)

Selection functions then are "multi-valued"

Before: $\varepsilon_n^{(X \to R) \to X}$ finds a **good move** given a **game continuation** (selection function)

Now: $\varepsilon_n^{(X \to R^*) \to X^*}$ finds a **set of moves** containing a good move or all good moves

Herbrand interpretation of $\mathbf{AC}_0$

Problem D.

Let formula $A_n(x^X, y^{R*})$ be given, anti-monotone in y^{R*}.

Suppose $\varepsilon_n^{(X \to R*) \to X*}$ are such that, for all $n^{\mathbb{N}}$ and $p^{X \to R*}$,

$$\exists x \in \varepsilon_n(p)\, A_n(x, p(x)).$$

Then, given $\omega^{X^{\mathbb{N}} \to \mathbb{N}}$ and $q^{X^{\mathbb{N}} \to R*}$ find finite set $A \subset X^{\mathbb{N}}$ such that

$$\exists \alpha \in A\, \forall n \leq \omega(\alpha)\, A_n(\alpha_n, q(\alpha)).$$

Game-theoretic interpretation.

Premise: $\varepsilon_n^{(X \to R*) \to X*}$ finds a finite set of moves containing a **good move** for round n

Conclusion: A is a finite set which contains **good play**

Herbrand interpretation of $\mathbf{AC}_0$

Problem D.

Let formula $A_n(x^X, y^{R*})$ be given, anti-monotone in y^{R*}.

Suppose $\varepsilon_n^{(X \to R*) \to X*}$ are such that, for all $n^{\mathbb{N}}$ and $p^{X \to R*}$,

$$\exists x \in \varepsilon_n(p)\, A_n(x, p(x)).$$

Then, given $\omega^{X^{\mathbb{N}} \to \mathbb{N}}$ and $q^{X^{\mathbb{N}} \to R*}$ find finite set $A \subset X^{\mathbb{N}}$ such that

$$\exists \alpha \in A\, \forall n \leq \omega(\alpha)\, A_n(\alpha_n, q(\alpha)).$$

Solution for D (bar recursion with finite sets).

Define $B(s) \subset X^{\mathbb{N}}$ as follows ($s : X^*$ and $\hat{s} = s * 0 \ldots$)

$$\mathrm{HBR}(s) = \begin{cases} \{\hat{s}\} & \text{if } \omega(\hat{s}) < |s| \\ \cup_{a \in A} \mathrm{HBR}(s * a) & \text{if } \omega(\hat{s}) \geq |s| \end{cases}$$

where $A =_{X*} \varepsilon_{|s|}(\lambda x. \cup_{\alpha \in b_s(x)} q(\alpha))$ and $b_s(x) = \mathrm{HBR}(s * x)$. Take $A = \mathrm{HBR}(\langle\, \rangle)$.

Herbrand interpretation of $\mathbf{AC}_0$

Solution for D (bar recursion with finite sets).

Define $B(s) \subset X^{\mathbb{N}}$ as follows ($s : X^*$ and $\hat{s} = s * 0...$)

$$\mathrm{HBR}(s) = \begin{cases} \{\hat{s}\} & \text{if } \omega(\hat{s}) < |s| \\ \cup_{a \in A} \mathrm{HBR}(s * a) & \text{if } \omega(\hat{s}) \geq |s| \end{cases}$$

where $A =_{X^*} \varepsilon_{|s|}(\lambda x. \cup_{\alpha \in b_s(x)} q(\alpha))$ and $b_s(x) = \mathrm{HBR}(s * x)$. Take $A = \mathrm{HBR}(\langle\,\rangle)$.

Theorem (Escardó/O.'2017).

The above construction solves the Herbrand functional interpretation of $\mathbf{DNS}$ and hence $\mathbf{AC}_0$ (classically).

Current work and open questions

- Selection functions with **<u>powerset monad</u>** to calculate all optimal plays
- Selection functions with **<u>Giry (distribution) monad</u>** to model stochastic players
- Selection functions with **<u>"state" monad</u>** to implement alpha-beta pruning of min-max search
- **<u>Question</u>**: Very few Proof Mining case studies has needed bar recursion. Why? (most mathematics seems very "tame")
- **<u>Question</u>**: Girard/Krivine have interpreted analysis via polymorphism. Exact relation between BR and polymorphism?

K. Gödel
Über eine bisher noch nicht benützte Erweiterung des finiten Standpunk
Dialectica, 12:280–287, 1958

C. Spector
Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics
Recursive Function Theory: Proc. Symposia in Pure Mathematics, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962

B. Scarpellini
A model for barrecursion of higher types
Compositio Matematica 23(3):123-153, 1971

M. Bezem
Strongly Majorizable Functionals of Finite Type: A Model for Barrecursion Containing Discontinuous Functionals
The Journal of Symbolic Logic, 50(3):652–660, 1985

S. Berardi, M. Bezem, T. Coquand
On the Computational Content of the Axiom of Choice
The Journal of Symbolic Logic, 63(2):600-622, 1998

U. Berger, P. Oliva
Modified Bar Recursion
Mathematical Structures in Computer Science, 16(2):163-183, 2006

P. Oliva
Understanding and using Spector's bar recursive interpretation of classical analysis
Proceedings of CiE 2006 LNCS 3988:423-434, Springer, 2006

U. Kohlenbach
Applied Proof Theory: Proof Interpretations and their Use in Mathematics
Springer Monographs in Mathematics, 2008

M. Escardó and P. Oliva
Selection functions, bar recursion and backward induction
Mathematical Structures in Computer Science, 20(2):127-168, 2010

M. Escardó, P. Oliva and T. Powell
System T and the product of selection functions
Computer Science Logic, 2011

M. Escardó and P. Oliva
Sequential games and optimal strategies
Proceedings of the Royal Society A, 467:1519-1545, 2011

M. Escardó and P. Oliva
Bar recursion and products of selection functions
The Journal of Symbolic Logic, 80(1):1-28, 2015

M. Escardó and P. Oliva
The Herbrand functional interpretation of the double negation shift
The Journal of Symbolic Logic, 82(2):590-607, 2017