An amazingly versatile functional

Martín Escardó

Joint work with Paulo Oliva from Queen Mary, London.

Logic Seminar, Imperial College, 17 Feb 2011

This talk is a selection of work I've published with Paulo

Selection functions everywhere

1. Game theory. Optimal strategies
2. Proof theory. Program extraction from classical proofs with choice.
3. Topology. Tychonoff theorem
4. Logic and higher-type computability. Bar recursion.
5. Category theoory. Strong monad with infinite strength.
6. Fixed points. Bekic's Lemma.

Proof theory

The negative translation of countable choice is not again choice.

But intuitionistic choice and DNS together imply the translation [Spector].

Double-Negation Shift

$$\forall n \in \mathbb{N}(\neg\neg A(n)) \implies \neg\neg\forall n \in \mathbb{N}(A(n)).$$

We have a new translation of classical logic into intuitionistic logic.

Using a monad $JA = ((A \to R) \to A)$ rather than $KA = ((A \to R) \to R)$.

The J-shift is an "infinite strength" of the monad

It realizes the J-translation of countable choice.

Topology and computation

Work in category of (computational) spaces and (computable) continuous maps.

A set $S \subseteq X$ has a selection function $\varepsilon\colon (X \to 2) \to X \iff$ it is compact.

It has a computable selection function iff it is effectively compact.

The infinite strength of the monad implements the Tychonoff Theorem.

Countable products of compact sets are compact.

Game theory

The infinite strength optimally plays finite games of unbounded length.

Selection functions briefly

X set of things.

Goods in a store; possible moves of a game; proofs of a proposition; point of a space.

R set of values.

Prices; outcomes win, lose, draw; how much money you win; true or false; proofs again.

$X \xrightarrow{p} R$ value judgement.

How you value it; how much it costs you; pay-off of a move; propositional function.

$(X \to R) \xrightarrow{\varepsilon} X$ selects something according to some criterion.

The best, the cheapest, any, something odd.

Example 1

X set of goods.

R set of prices.

$X \xrightarrow{p} R$ table of prices.

$(X \to R) \xrightarrow{\varepsilon} X$ selects a cheapest good in a given table.

$(X \to R) \xrightarrow{\phi} R$ determines the lowest price in a given table.

Fundamental equation:

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

The price of a cheapest good is the lowest in the table, of course.

$$\phi = \inf \qquad \varepsilon = \operatorname{arginf},$$

$$p(\operatorname{arginf}(p)) = \inf(p).$$

Example 2

X set of individuals.

R set of booleans $\mathrm{false} = 0 < 1 = \mathrm{true}$.

$X \xrightarrow{p} R$ property.

$(X \to R) \xrightarrow{\varepsilon} X$ selects an individual with the highest truth value.

$(X \to R) \xrightarrow{\phi} R$ determines the highest value of a given property.

Fundamental equation:

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

$$\begin{aligned} \phi &= \sup = \exists \\ \varepsilon &= \mathrm{argsup} = \mathrm{arg}\text{-}\exists = \text{Hilbert's choice operator} \end{aligned}$$

$p(\varepsilon(p)) = \exists(p)$ Hilbert's definition of $\exists$ in his ε-calculus

Maximum-Value Theorem

Let X be a compact non-empty topological space.

Any continuous function $p\colon X \to \mathbb{R}$ attains its maximum value.

This means that there is $a \in X$ such that

$$\sup p = p(a).$$

However, the proof is non-constructive when e.g. $X = [0, 1]$.

A maximizing argument a cannot be algorithmically calculated from p.

Of course, there is a Minimum-Value Theorem too.

Mean-Value Theorem

Any continuous function $p\colon [0,1] \to \mathbb{R}$ attains its mean value.

There is $a \in [0,1]$ such that

$$\int p = p(a).$$

Again this a cannot be found from p using an algorithm.

Universal-Value Theorem

Let X be a non-empty set and $2 = \{0, 1\}$ be the set of booleans.

Any $p\colon X \to 2$ attains its universal value.

There is $a \in X$ such that

$$\forall p = p(a).$$

This is again a classical statement.

This is usually formulated as the Drinker Paradox:

In any inhabited pub there is a person a s.t. if a drinks then everybody drinks.

We've also met the Existential-Value Theorem.

General situation

With ϕ among $\exists, \forall, \sup, \inf, \int, \ldots$,

$$\boxed{\phi(p) = p(a)}$$

for some a depending on p.

In favourable circumstances, the attaining point a can be calculated as

$$a = \varepsilon(p),$$

so that

$$\boxed{\phi(p) = p(\varepsilon(p))}$$

Selection function

Definition.

A selection function for a (logical, arithmetical, . . .) quantifier

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

is a functional

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

such that

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

Monad morphism

Every $\varepsilon\colon (X \to R) \to X$ is the selection function of some $\phi\colon (X \to R) \to R$.

Namely ϕ defined by

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

Also written $\phi = \overline{\varepsilon}$.

This construction defines a monad morphism $\theta\colon J \to K$:

$$\overbrace{(X \to R) \to X}^{JX} \quad \stackrel{\Theta}{\longrightarrow} \quad \overbrace{(X \to R) \to R}^{KX}$$

$$\varepsilon \quad \longmapsto \quad \overline{\varepsilon}$$

This is a morphism from the selection monad to the quantifier monad.

Oh, I mean to the continuation monad.

Units of the monads

$$\begin{array}{ccc} X & \xrightarrow{\eta} & KX \\ x & \longmapsto & \lambda p.p(x). \end{array}$$

(Universally and existentially) quantifies over the singleton $\{x\} \subseteq X$.

$$\eta(x) = \exists_{\{x\}} = \forall_{\{x\}}.$$

$$\begin{array}{ccc} X & \xrightarrow{\eta} & JX \\ x & \longmapsto & \lambda p.x. \end{array}$$

Produces a selection function for the singleton quantifier.

Functors of the monads

Let $f\colon X \to Y$.

$$\begin{array}{ccc} KX & \xrightarrow{Kf} & KY \\ \phi & \longmapsto & \lambda p.\phi(\lambda x.p(f(x))). \end{array}$$

If ϕ quantifies over a set $S \subseteq X$, then $Kf(\phi)$ quantifies over the set $f(S) \subseteq Y$.

$$\begin{array}{ccc} JX & \xrightarrow{Jf} & JY \\ \varepsilon & \longmapsto & \lambda p.f(\varepsilon(\lambda x.p(f(x)))). \end{array}$$

If ε is a selection function for ϕ, then $Jf(\varepsilon)$ is a selection function for $Kf(\phi)$.

Multiplication of the quantifier monad

Involves the perhaps unfamiliar notion of quantification over quantifiers.

$$\begin{array}{rcl} KKX & \xrightarrow{\mu} & KX \\ \Phi & \longmapsto & \lambda p.\Phi(\lambda\phi.\phi(p)). \end{array}$$

Suppose $A \subseteq KX$ is a set such that each $\phi \in A$ existentially quantifies over a set $B_\phi \subseteq X$, i.e.

$$\phi = \exists_{B_\phi}$$

Then the universal quantifier $\forall_A \in KKX$ of the set $A \subseteq KX$ satisfies

$$\mu(\forall_A)(p) = \forall\phi \in A \exists x \in B_\phi(p(x)).$$

Multiplication of the selection monad

Now we have selection functions that select selection functions.

$$
\begin{array}{rcl}
JJX & \xrightarrow{\mu} & JX \\
E & \longmapsto & \lambda p.E(\lambda\varepsilon.p(\varepsilon(p)))(p).
\end{array}
$$

Use the selection function E to find a selection function ε such that $p(\varepsilon(p))$, and apply this resulting selection function to p to find an element of X.

Algebras

$KA \to A$.

$$((A \to R) \to R) \to A.$$

Double-negation elimination.

$JA \to A$.

$$((A \to R) \to A) \to A.$$

Peirce's Law.

Get different proof translation of classical logic into intuitionistic logic.

This gives a better explanation of call/cc.

Strengths

$$\begin{array}{rcl} X \times KY & \xrightarrow{t} & K(X \times Y) \\ (x, \phi) & \longmapsto & \lambda p.\phi(\lambda y.p(x,y)). \end{array}$$

If ϕ quantifies over $S \subseteq Y$, then the resulting quantifier quantifies over $\{x\} \times S \subseteq X \times Y$.

$$\begin{array}{rcl} X \times JY & \xrightarrow{t} & J(X \times Y) \\ (x, \varepsilon) & \longmapsto & \lambda p.(x, \varepsilon(\lambda y.p(x,y))). \end{array}$$

This produces a selection function for the above quantifier.

We have monoidal monad structures

Because we have strong monads on a cartesian closed category.

$$\begin{array}{rcl} TX \times TY & \xrightarrow{\otimes} & T(X \times Y) \\ (u, v) & \longmapsto & (T(\lambda x.t(x, v)))(u). \end{array}$$

Warning. This is one way of getting this.

The other way of getting this gives a different $\otimes$.

| The monads are not commutative. | (And this is good!)

The above choice of $\otimes$ is what we need for our purposes.

("left-to-right" as opposed to "right-to-left".)

Monoidal monad structures

$$\begin{array}{rcl} TX \times TY & \stackrel{\otimes}{\longrightarrow} & T(X \times Y) \\ (u, v) & \longmapsto & (T(\lambda x.t(x, v)))(u). \end{array}$$

1. Sequential games of length two.

2. Binary Tychonoff Theorem.

3. Bekic's Lemma. An element of $JR = KR = ((R \to R) \to R)$ is a fixed point operator iff it is its own selection function. Bekic's formula is given by the product of selection functions.

Examples

$$\begin{array}{rcl} KX \times KY & \stackrel{\otimes}{\longrightarrow} & K(X \times Y) \\ (\exists_A, \exists_B) & \longmapsto & \exists_{A\times B}. \end{array}$$

$$\begin{array}{rcl} KX \times KY & \stackrel{\otimes}{\longrightarrow} & K(X \times Y) \\ (\forall_A, \exists_B) & \longmapsto & \lambda p.\forall x \in A.\exists y \in B.p(x, y). \end{array}$$

What about J?

Well, of course, its product $\otimes$ commutes with that of K.

(Because we have a monad morphism.)

This means:

if

$\varepsilon \in JX$ is a selection function for $\phi \in KX$

$\delta \in JY$ is a selection function for $\gamma \in KY$,

then

$\varepsilon \otimes \delta$ is a selection function for $\phi \otimes \gamma$.

This is good for optimally playing games, as we'll see.

Binary product of quantifiers and of selection functions

In every pub there are a man m and a woman w such that if m buys a drink to w then every man buys a drink to some woman.

Binary product of quantifiers and of selection functions

In every pub there are a man m and a woman w such that if m buys a drink to w then every man buys a drink to some woman.

If $Y =$ set of men and $X =$ set of women, and if we define $\phi = \forall \otimes \exists$ by

$$\phi(p) = (\forall y \in Y \ \exists x \in X \ p(y, x)),$$

then this amounts to saying that

$$\phi(p) = p(a)$$

for a suitable pair $a = (m, w) \in X \times Y$,

This is calculated as $a = (\varepsilon \otimes \delta)(p)$ where $\overline{\varepsilon} = \forall_Y$ and $\overline{\delta} = \exists_X$.

Infinitely iterated left-to-right monoidal monad structure

In certain categories of interest

There is a countable monoidal-monad structure

$$\bigotimes \colon \prod_i J X_i \to J \prod_i X_i$$

uniquely determined by the equation

$$\bigotimes_i \varepsilon_i = \varepsilon_o \otimes \bigotimes_i \varepsilon_{i+1}.$$

(Which turns out to be an instance of a bar recursion scheme.)

Playing games

Products of selection functions compute optimal plays and strategies.

First example

Alternating, two-person game that finishes after exactly n moves.

1. Eloise plays first, against Abelard. One of them wins (no draw).

2. The i-th move is an element of the set X_i.

3. The game is defined by a predicate $p \colon \prod_{i<n} X_i \to \texttt{Bool}$ that tells whether Eloise wins wins a given play $x = (x_0, \ldots, x_{n-1})$.

4. Eloise has a winning strategy for the game p if and only if

$$\exists x_0 \in X \forall x_1 \in Y \exists x_2 \in X_2 \forall x_3 \in X_3 \cdots p(x_0, \ldots, x_{n-1}).$$

First example

4. Eloise has a winning strategy for the game p if and only if

$$\exists x_0 \in X \forall x_1 \in Y \exists x_2 \in X_2 \forall x_3 \in X_3 \cdots p(x_0, \ldots, x_{n-1}).$$

If we define

$$\phi_i = \begin{cases} \exists_{X_i} & \text{if } i \text{ is even,} \\ \forall_{X_i} & \text{if } i \text{ is odd,} \end{cases}$$

then this condition for Eloise having a winning strategy can be equivalently expressed as

$$\left(\bigotimes_{i<n} \phi_i \right)(p).$$

Calculating the optimal outcome of a game

More generally, the value

$$\left(\bigotimes_{i<n} \phi_i\right)(p)$$

gives the optimal outcome of the game.

This takes place when all players play as best as they can.

In the first example, the optimal outcome is `True` if Eloise has a winning strategy, and `False` if Abelard has a winning strategy.

Calculating an optimal play

Suppose each quantifier ϕ_i has a selection function ε_i (thought of as a policy function for the i-th move).

Theorem. The sequence

$$a = (a_0, \ldots, a_{n-1}) = \left(\bigotimes_{i<n} \varepsilon_i \right)(p)$$

is an optimal play.

This means that for every stage $i < n$ of the game, the move a_i is optimal given that the moves $a_0, \ldots, a_{i-1}$ have been played.

Finding an optimal strategy

Theorem. The function $f_k : \prod_{i<k} X_i \to X_k$ defined by

$$f_k(a_1 \ldots a_{k-1}) = \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (\lambda x_k \ldots x_{n-1}.p(a_1 \ldots a_{k-1} x_k \ldots x_{n-1})) \right)_0$$

is an optimal strategy for playing the game.

This means that given that the sequence of moves $a_0, \ldots, a_{k-1}$ have been played, the move $a_k = f_k(a_0, \ldots, a_{k-1})$ is optimal.

Second example

Choose $R = \{-1, 0, 1\}$ instead, with the convention that

$$\begin{cases} -1 = \text{Abelard wins}, \\ \ \ 0 = \text{draw}, \\ \ \ 1 = \text{Eloise wins}. \end{cases}$$

The existential and universal quantifiers get replaced by $\sup$ and $\inf$:

$$\phi_i = \begin{cases} \sup_{X_i} & \text{if } i \text{ is even}, \\ \inf_{X_i} & \text{if } i \text{ is odd}. \end{cases}$$

The optimal outcome is still calculated as $\bigotimes_{i<n} \phi_i$, which amounts to

$$\sup_{x_0 \in X!} \inf_{x_1 \in Y} \sup_{x_2 \in X_2} \inf_{x_3 \in X_3} \cdots p(x_0, \ldots, x_{n-1}).$$

Second example

The optimal outcome is

$$\begin{cases} -1 = \text{Abelard has a winning strategy,} \\ \;\;\,0 = \text{the game is a draw,} \\ \;\;\,1 = \text{Eloise has a winning strategy.} \end{cases}$$

Can compute optimal outcomes, plays and strategies with the same formulas.

Classical choice

The T translation of the axiom of choice, AC, is

$$\mathsf{AC}^T \colon \forall x\,(T\exists y(A(x,y))) \implies T\exists f\forall x(A(x,f(x))).$$

We are interested in $T = J$ or $T = K$.

When T is the identity monad, this is just AC.

Problem: the T-translation of AC is not an instance of AC.

Spector's idea: Implement the T-shift. He did this for $T = K$ and $X = \mathbb{N}$ in the dialectica interpretation, using Spector bar recursion.

Intuitionistic AC together with T-shift gives (classical) AC^T

Proof. 1. Start with (proofs or realizers of)

$$\begin{aligned} \text{AC} &: \quad \forall x\,(\exists y(A(x,y))) \implies \exists f \forall x(A(x,f(x))). \\ T\text{-shift} &: \quad \forall x(T(B(x))) \to T\forall x(B(x)). \end{aligned}$$

2. Now apply the functor T to the proof of AC to get

$$T\text{AC}\colon T\forall x\,(\exists y(A(x,y))) \implies T\exists f \forall x(A(x,f(x))).$$

3. Then take $B = \exists y(A(x,y))$ and compose with T-shift to get

$$\text{AC}^T\colon \forall x\,(T\exists y(A(x,y))) \implies T\exists f \forall x(A(x,f(x))).$$

Corollary

If the T-shift is realizable then so is AC^T.

The J-shift

Theorem. The product functional

$$\bigotimes \colon \prod_{i \in \mathbb{N}} J X_{i \in \mathbb{N}} \to J \prod_{i} X_i$$

realizes the classical J-shift principle

$$\forall i(J(A(i))) \to J \forall i(A(i)).$$

Corollary. $\mathsf{AC}_{\mathbb{N}}^{J}$ is realizable.

This works e.g. in PA^{ω} giving programs from classical proofs in Gödel's System T extended with with a constant $\bigotimes$ and a (bar) recursion equation.

We are implementing examples in Haskell and Agda.

Tychonoff Theorem

$\bigotimes$ implements the fact that countable products of searchable sets are searchable.

The space $2 = \{0, 1\}$ is searchable, and hence so is 2^ω.

In particular, we get a program $\forall\colon (2^\omega \to \mathrm{Bool}) \to \mathrm{Bool}$.

In particular, equality of functions $2^\omega \to \mathbb{N}$ is decidable,

Let's implement and run this in Haskell.

Conclusion

Selection functions everywhere.

- Game theory.
- Proof theory and program extraction.
- Topology.
- Category theory.
- Fixed-point theory.