The ubiquitous selection monad

Martín Escardó
University of Birmingham, UK

Joint work with Paulo Oliva, Queen Mary, London, UK.

Wessex Seminar, Swansea, 7th April 2011

Seemingly disparate constructions

1. Game theory.

   Optimal plays of sequential games of unbounded length.

2. Proof theory. Double Negation Shift:

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

3. Topology. Tychonoff Theorem:

   X_i compact $\implies \prod_i X_i$ also compact.

4. Higher-type computation. Computational Tychonoff Theorem:

   X_n exhaustibly searchable $\implies \prod_n X_n$ also exhaustibly searchable.

   The point is that we get infinite exhaustively searchable sets.

What do they have in common?

Implemented/realized by a certain infinite product of selection functions.

Explained by a certain selection monad.

Countable-product functional

In a simply typed formalism:

$$(\mathbb{N} \to ((X \to R) \to X)) \to ((\mathbb{N} \to X) \to R) \to (\mathbb{N} \to X).$$

In a dependently typed formalism:

$$\left(\prod_{n\in\mathbb{N}} (X_n \to R) \to X_n\right) \to \left(\left(\prod_{n\in\mathbb{N}} X_n\right) \to R\right) \to \left(\prod_{n\in\mathbb{N}} X_n\right).$$

There is structure in the above types

Write $JX = ((X \to R) \to X)$ where R is fixed in advance.

In a simply typed formalism:

$$(\mathbb{N} \to JX) \to J(\mathbb{N} \to X).$$

In a dependently typed formalism:

$$\prod_{n \in \mathbb{N}} JX_n \to J \prod_{n \in \mathbb{N}} X_n.$$

Selection functions $(X \to R) \to X$

X set of things.

Goods in a store; possible moves of a game; proofs of a proposition; points 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:

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

This says that the price of a cheapest good is the lowest in the table.

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

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

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 if the set X is infinite.

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 $\exists, \forall, \sup, \inf, \int, \ldots$, we have that

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

for some a depending on p.

In favourable circumstances, 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 $\phi = \overline{\varepsilon}$ defined by

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

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

$$\overbrace{(X \to R) \to X}^{JX} \xrightarrow{\Theta} \overbrace{(X \to R) \to R}^{KX}$$

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

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

Oh, I mean to the continuation monad.

Units of the monads

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

Quantifies over the singleton $\{x\} \subseteq X$.

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

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

Produces a selection function for the singleton quantifier.

Functors of the monads

Let $f \colon X \to Y$.

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

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

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

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

Multiplications

They can be explained in intuitive terms, but this takes some time.

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

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

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.

Monad algebras

$KA \to A$.

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

Double-negation elimination.

Explains the Gödel–Gentzen translation of classical into intuitionistic logic.

$JA \to A$.

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

Peirce's Law.

Get different proof translation of classical into intuitionistic logic.

Aside: we get a more conceptual explanation of call/cc

The type of call/cc can be written as $JKX \to KX$.
(An instance of Peirce's Law, as discovered by Tim Griffin.)

Its λ-term can be reconstructed as follows:

1. KX is a K-algebra, with structure map $\mu\colon KKX \to KX$.

2. Because we have a morphism $J \xrightarrow{\theta} K$, every K-algebra is a J-algebra:

$$JA \xrightarrow{\theta_A} KA \xrightarrow{\alpha} A.$$

3. Call/cc is what results for $A = KX$ and $\alpha = \mu$:

$$JKX \xrightarrow{\theta_{KX}} KKX \xrightarrow{\mu} KX.$$

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 $t(x, \phi)$ 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 $T = J$ and $T = K$ on a ccc.

$$
\begin{array}{rcll}
TX \times TY & \xrightarrow{\otimes} & T(X \times Y) & \\
(u, v) & \longmapsto & (T(\lambda x. t_{X,Y}(x, v)))(u) & \longleftarrow \text{we want this one,} \\
(u, v) & \longmapsto & (T(\lambda y. t_{Y,X}(u, x)))(v) & \longleftarrow \text{not this one.}
\end{array}
$$

The monads are not commutative.

And this is good!

Examples

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

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

The other choice of $\otimes$ concatenates the quantifiers in reverse order.

Because we have a strong monad morphism:

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

In other words:

Theorem.

If

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

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

then

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

Binary product of quantifiers and selection functions

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

Binary product of quantifiers and selection functions

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

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

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

then our claim amounts to

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

for a suitable pair $a = (b, c) \in X \times Y$,

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

The infinite strength of the selection monad

In certain categories of interest

There is a countable monoidal-monad structure

$$\bigotimes : \prod_n JX_n \to J \prod_n X_n$$

uniquely determined by the equation

$$\bigotimes_n \varepsilon_n = \varepsilon_o \otimes \bigotimes_n \varepsilon_{n+1}.$$

Turns out to be an instance of the bar recursion scheme.

The continuation monad lacks infinite strength

However, if a sequence of quantifiers ϕ_n have selection functions ε_n,

then their product can be defined as

$$\bigotimes_n \phi_n = \overline{\bigotimes_n \varepsilon_n}$$

and satisfies

$$\bigotimes_n \phi_n = \phi_o \otimes \bigotimes_n \phi_{n+1}.$$

This is useful for various applications.

Playing games

Products of selection functions compute optimal plays and strategies.

First example

Alternating, two-person game.

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 $p\colon \prod_i X_i \to \texttt{Bool}$
   that tells whether Eloise 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

$$\exists x_0 \in X \forall x_1 \in Y \exists x_2 \in X_2 \forall x_3 \in X_3 \cdots p(x_0, x_1, x_2, x_3, \ldots).$$

First example

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

$$\exists x_0 \in X \forall x_1 \in Y \exists x_2 \in X_2 \forall x_3 \in X_3 \cdots p(x_0, x_1, x_2, x_3, \ldots).$$

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 amounts to

$$\left(\bigotimes_i \phi_i \right) (p).$$

Second example

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

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

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 calculated as $\bigotimes_i \phi_i$, which amounts to

$$\sup_{x_0 \in X_0} \inf_{x_1 \in Y} \sup_{x_2 \in X_2} \inf_{x_3 \in X_3} \cdots p(x_0, x_1, x_2, x_3, \ldots).$$

General non-history dependent case

A sequential game is given by

1. Sets of moves $X_0, X_1, X_2, \ldots$.
2. A set R of possible outcomes.
3. An outcome function $p\colon \prod_i X_i \to R$,
4. A quantifier $\phi_i \in KX_i$ for each stage of the game.
5. Optionally a selection function ε_i for the quantifier ϕ_i.

These are games in normal form.

For games in extensive form, the outcome function is presented as a tree.

Calculating the optimal outcome of a game

The value

$$\left(\bigotimes_i \phi_i\right)(p)$$

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.

Theorem. The sequence

$$a = (a_0, a_1, \ldots, a_i, \ldots) = \left(\bigotimes_i \varepsilon_i \right)(p)$$

is an optimal play.

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

Calculating an optimal strategy

For a partial play $a \in \prod_{i<k} X_i$, we have a subgame $p_a \colon \prod_{i \geq k} X_i \to R$,

$$p_a(x) = p(a \cdot x).$$

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

$$f_k(a) = \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (p_a) \right)_0$$

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.

Program extraction from classical proofs with choice

Start with intuitionistic choice

$$\forall i \in I \, (\exists x \in X_i \, (A(i,x))) \implies \exists \vec{x} \in \prod_i X_i \, (\forall i \in I \, (A(i,x_i)))\,.$$

Apply the T-translation, say for $T = K$ or $T = J$:

$$\forall i \in I \, (T\exists x \in X_i \, (A(i,x))) \implies T\exists \vec{x} \in \prod_i X_i \, (\forall i \in I \, (A(i,x_i)))\,.$$

Is that realizable?

The J-shift

Think of $JA = ((A \to R) \to A)$ as a logical modality.

Theorem

The product functional $\bigotimes$: $\prod_n JX_n \to J\left(\prod_n X_n\right)$ realizes the J-shift

$$\forall n(J(A(n)) \to J\left(\forall n(A(n))\right).$$

To guess the theorem, apply Curry–Howard.

To prove it, use bar induction.

Countable choice

1. Start again with intuitionistic choice, but countable this time:

$$\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x))) \implies \exists \vec{x} \in \prod_n X_n\,(\forall n \in \mathbb{N}\,(A(n,x_n)))\,.$$

2. Apply the functor J:

$$J(\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x)))) \implies J\exists \vec{x} \in \prod_n X_n\,(\forall n \in \mathbb{N}\,(A(n,x_n)))\,.$$

3. Finally pre-compose with the instance of the J-shift

$$\forall n \in \mathbb{N}\,(J\exists x \in X_n\,(A(n,x))) \implies J\,(\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x))))\,.$$

Theorem. The J-translation of countable choice is realizable.

Countable choice

1. Start again with intuitionistic choice, but countable this time:

$$\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x))) \implies \exists \vec{x} \in \prod_n X_n\,(\forall n \in \mathbb{N}\,(A(n,x_n)))\,.$$

2. Apply the functor J:

$$\underline{J(\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x))))} \implies J\exists \vec{x} \in \prod_n X_n\,(\forall n \in \mathbb{N}\,(A(n,x_n)))\,.$$

3. Finally pre-compose with the instance of the J-shift

$$\forall n \in \mathbb{N}\,(J\exists x \in X_n\,(A(n,x))) \implies \underline{J\,(\forall n \in \mathbb{N}\,(\exists x \in X_n\,(A(n,x))))}.$$

Theorem. The J-translation of countable choice is realizable.

Topology and computation

I need a large class of topological spaces to formulate a computational theorem.

Kleene–Kreisel spaces are good for total higher-type computation.

But perhaps a bit limited.

Enlarge by closing under retracts.

Denote by $2 = \{0, 1\}$ the space of booleans.

Effective compactness

1. Theorem (topological).

   A space X is compact $\iff$ has a continuous quantifier $(X \to 2) \to 2$.

2. Definition (computational).

   A space X is effectively compact if it has a computable quantifier $(X \to 2) \to 2$.

3. Theorem (computational).

   A space X is effectively compact $\iff$
   it has a computable selection function $(X \to 2) \to X$.

This says that two different, common forms of exhaustive search are equivalent.

Computational Tychonoff Theorem

Theorem

Effectively compact spaces are closed under the formation of countable products.

This is implemented again by the infinite product of selection functions.

We have a Haskell implementation that runs fast in counter-intuitive examples.

Conclusion

Selection functions everywhere.

Appendix

The monads defined as Kleisli triples

Define the (internalized) extension operators:

$$\begin{array}{rcl} (X \to KY) & \to & (KX \to KY) \\ f & \mapsto & f^{\sharp} \\ f & \mapsto & \lambda\phi.\lambda p.(\phi(\lambda x.p(fx))). \end{array}$$

Example: Take $X = Y = \mathbb{N}$ and $f(k)(p) = \exists n < k(p(k))$. Then

$$f^{\sharp}(\forall_N)(p) = \forall k \in \mathbb{N}(\exists n < k(p(k))).$$

Kleisli extension for J

$$\begin{array}{rcl}(X \to JY) & \to & (JX \to JY) \\ g & \mapsto & g^\sharp.\end{array}$$

For a suitable $x \in X$ constructed from ε, we will define: $g^\sharp(\varepsilon)(p) = gxp$.

Before such an $x \in X$ is available, we abstract it: $\lambda x.gxp$.

But now we can apply ε to this, to find $x_0 \stackrel{\text{def}}{=} \varepsilon(\lambda x.gxp) \in X$, and define

$$g^\sharp(\varepsilon)(p) = gx_0p.$$

Expanding the definition, we get $g^\sharp(\varepsilon)(p) = g(\varepsilon(\lambda x.gxp))p$.

Extension operators related by the monad morphism

$$\overline{g^\sharp(\varepsilon)} = \left(\lambda x.\overline{g(x)}\right)^\sharp(\overline{\varepsilon}).$$

In other words:

Theorem. Let $f\colon X \to KY$ and $g\colon X \to JY$.

If $g(x) \in JY$ is a selection function of the quantifier $f(x) \in KY$ for all $x \in X$,

and $\varepsilon \in KY$ is a selection function for the quantifier $\phi \in KX$,

then $g^\sharp(\varepsilon) \in JX$ is a selection function for the quantifier $f^\sharp(\phi) \in KX$.