POLYNOMIAL LENSES

BARTOSZ MILEWSKI

1. Motivation

Lenses seem to pop up in most unexpected places. Recently a new type of lens showed up as a set of morphisms between polynomial functors. This lens seemed to not fit the usual classification of optics, so it was not immediately clear that it had an existential representation using coends and, consequently a profunctor representation using ends. A profunctor representation of optics is of special interest since it lets us compose optics using standard function composition. In this post I will show how the polynomial lens fits into the framework of general optics.

2. Polynomial Functors

A polynomial functor in **Set** can be written as a sum (coproduct) of representables:

$$P(y) = \sum_{n \in N} s_n \times \mathbf{Set}(t_n, y)$$

The two families of sets, s_n and t_n are indexed by elements of the set N (in particular, you may think of it as a set of natural numbers, but any set will do). In other words, they are fibrations of some sets S and T over N. In programming we call such families *dependent types*. We can also think of these fibrations as functors from a category N to **Set**.

Since in **Set** the internal hom is isomorphic to the external hom, a polynomial functor is sometimes written in the exponential form, which makes it look more like an actual polynomial or a power series:

$$P(y) = \sum_{n \in N} s_n \times y^{t_n}$$

or, by representing all sets s_n as sums of singlentons:

$$P(y) = \sum_{n \in N} y^{t_n}$$

I will also use the notation $[t_n, y]$ for the internal hom:

$$P(y) = \sum_{n \in N} s_n \times [t_n, y]$$

Polynomial functors form a category **Poly** in which morphisms are natural transformations. Consider two polynomial functors P and Q. A natural transformation between them can be written as an end. Let's first expand the source functor:

$$\mathbf{Poly}\left(\sum_n s_n \times [t_n, -], Q\right) = \int_{y \colon \mathbf{Set}} \mathbf{Set}\left(\sum_n s_n \times [t_n, y], Q(y)\right)$$

The mapping out of a sum is isomorphic to a product of mappings:

$$\cong \prod_n \int_y \mathbf{Set}\left(s_n \times [t_n, y], Q(y)\right)$$

We can see that a natural transformation between polynomials can be reduced to a product of natural transformations out of monomials. Let's therefore consider a mapping out of a monomial:

$$\int_y \mathbf{Set}\left(s \times [t, y], \sum_n a_n \times [b_n, y]\right)$$

We can now use the currying adjunction:

$$\int_y \mathbf{Set}\left([t, y], \mathbf{Set}\left(s, \sum_n a_n \times [b_n, y]\right)\right)$$

We can now use the Yoneda lemma to eliminate the end. This will simply replace y with t in the target of the natural transformation:

$$\mathbf{Set}\left(s, \sum_n a_n \times [b_n, t]\right)$$

The set of natural transformation between two arbitrary polynomials is called the polynomial lens:

$$\mathbf{PolyLens}\langle s, t\rangle\langle a, b\rangle = \prod_k \mathbf{Set}\left(s_k, \sum_n a_n \times [b_n, t_k]\right)$$

Using dependent-type language, we can describe the polynomial lens as acting on a whole family of types at once. For a given value of type s_k it determines the index n. The interesting part is that this index and, consequently, the type of the focus a_n and the type on the new focus b_n depends not only on the type but also on the *value* of the argument s_k.

Here's a simple example: consider a family of node-counted trees. In this case s_k is a type of a tree with k nodes. For a given node count we can still have trees with a different number of leaves. We could define a poly-lens for such trees that focuses on the leaves. For a given tree it would produce a counted vector a_n of leaves and a function that takes a counted vector b_n (same size, but different type of leaf) and return a new tree t_k.

3. Poly Lens as an Optic

A lens is just a special case of optics. Optics have a very general representation as existential types or, categorically speaking, as coends.

The general idea is that optics describe various modes of decomposing a type into the focus (or multiple foci) and the residue. This residue is an existential type. Its only property is that it can be combined with a new focus (or foci) to produce a new composite.

The question is, what's the residue in the case of a polynomial lens? The intuition from the counted-tree example tells us that such residue should be parameterized by both, the number of nodes, and the number of leaves. It encodes the shape of the tree, with placeholders replacing the leaves.

In general, the residue will be a doubly-indexed family c_{mn} and the existential form of poly-lens will be implemented as a coend over all possible residues:

$$\mathbf{Pl}\langle s,t\rangle\langle a,b\rangle \cong \int^{c_{ki}} \prod_k \mathbf{Set}\left(s_k, \sum_n a_n \times c_{nk}\right) \times \prod_i \mathbf{Set}\left(\sum_m b_m \times c_{mi}, t_i\right)$$

To see that this representation is equivalent to the previous one let's first rewrite a mapping out of a sum as a product of mappings:

$$\prod_i \mathbf{Set}\left(\sum_m b_m \times c_{mi}, t_i\right) \cong \prod_i \prod_m \mathbf{Set}\left(b_m \times c_{mi}, t_i\right)$$

and use the currying adjunction to get:

$$\prod_i \prod_m \mathbf{Set}\left(c_{mi}, [b_m, t_i]\right)$$

The main observation is that, if we treat the set N as a discrete category $\mathcal{N}$, a product of mappings can be considered a natural transformation between functors from $\mathcal{N}$. Functors from a discrete category are just mappings of objects, and naturality conditions are trivial.

A double product could be considered a natural transformation from a product category. And since a discrete category is its own opposite, we can rewrite our mappings as natural transformations:

$$\prod_i \prod_m \mathbf{Set}\left(c_{mi}, [b_m, t_i]\right) \cong [\mathcal{N}^{op} \times \mathcal{N}, \mathbf{Set}]\left(c_{=-}, [b_=, t_-]\right)$$

The indexes were replaced by placeholders. This notation underscores the interpretation of b and t as functors (co-presheaves) from $\mathcal{N}$ to **Set**, and c as a profunctor on $\mathcal{N}^{op} \times \mathcal{N}$.

We can therefore use the co-Yoneda lemma to eliminate the coend over c_{ki}. The result is:

$$
\begin{aligned}
&\mathbf{Pl}\langle s, t\rangle\langle a, b\rangle \\
&\cong \int^{c_{ki}} \prod_k \mathbf{Set}\left(s_k, \sum_n a_n \times c_{nk}\right) \times [\mathcal{N}^{op} \times \mathcal{N}, \mathbf{Set}]\,(c_{=-}, [b_=, t_-]) \\
&\cong \prod_k \mathbf{Set}\left(s_k, \sum_n a_n \times [b_n, t_k]\right)
\end{aligned}
$$

which is exactly the original polynomial-to-polynomial transformation.

I haven't shown yet that the existential form of the polynomial lens follows the pattern of a monoidal action. This will become clear in a more general setting.

4. Generalizations

We've seen that the polynomial lens can be described as an optic in terms on functors and profunctors on a discrete category $\mathcal{N}$. With just a few modifications, we can make $\mathcal{N}$ non-discrete. The trick is to replace sums with coends and products with ends; and, when appropriate, interpret ends as natural transformations.

We'll write the residues c_{mn} as profunctors:

$$c\langle m, n\rangle \colon \mathcal{N}^{op} \times \mathcal{N} \to \mathbf{Set}$$

They are objects in the monoidal category in which the tensor product is given by profunctor composition:

$$(c \diamond c')\langle m, n\rangle = \int^{k \colon \mathcal{N}} c\langle m, k\rangle \times c'\langle k, n\rangle$$

and the unit is the hom-functor $\mathcal{N}(m, n)$.

In the case of $\mathcal{N}$ a discrete category, these definitions decay to standard matrix multiplication and the Kronecker delta.

$$\sum_k c_{mk} \times c'_{kn}$$

We define the monoidal action of the profunctor c acting on a co-presheaf a as:

$$(c \bullet a)(m) = \int^{n \colon \mathcal{N}} a(n) \times c\langle n, m\rangle$$

Finally, the two hom-sets in the definition of an optic become sets of natural transformations in the functor category $[\mathcal{N}, \mathbf{Set}]$.

$$\mathbf{Po}\langle s, t\rangle\langle a, b\rangle \cong \int^{c \colon [\mathcal{N}^{op} \times \mathcal{N}, Set]} [\mathcal{N}, \mathbf{Set}]\,(s, c \bullet a) \times [\mathcal{N}, \mathbf{Set}]\,(c \bullet b, t)$$

Or, using the end notation for natural transformations:

$$\int^{c} \left(\int_{m} \mathbf{Set}\left(s(m), (c \bullet a)(m)\right) \times \int_{n} \mathbf{Set}\left((c \bullet b)(n), t(n)\right) \right)$$

As before, we can eliminate the coend if we can isolate c in the second hom-set using a set of transformations:

$$\begin{aligned} &\int_{n} \mathbf{Set}\left(\int^{k} b(k) \times c\langle k, n\rangle, t(n) \right) \\ \cong &\int_{n} \int_{k} \mathbf{Set}\left(b(k) \times c\langle k, n\rangle, t(n)\right) \\ \cong &\int_{n,k} \mathbf{Set}\left(c\langle k, n\rangle, [b(k), t(n)]\right) \end{aligned}$$

I used the fact that a mapping out of a coend is an end. The final result is:

$$\mathbf{Po}\langle s, t\rangle\langle a, b\rangle \cong \int_{m} \mathbf{Set}\left(s(m), \int^{j} a(j) \times [b(j), t(m)] \right) \cong [\mathcal{N}, \mathbf{Set}](s, [b, t]\bullet a)$$

When $\mathcal{N}$ is discrete, this formula decays to the one for polynomial lens.

5. Profunctor Representation

Since the poly-lens is a special case of general optic, it automatically has a profunctor representation. The trick is to define a generalized Tambara module, that is a category $\mathcal{T}$ of profunctors of the type:

$$P \colon [\mathcal{N}, \mathbf{Set}]^{op} \times [\mathcal{N}, \mathbf{Set}] \to \mathbf{Set}$$

with additional structure given by the following family of transformations, in components:

$$\alpha_{c,s,t} \colon P\langle s, t\rangle \to P\left\langle c \bullet s, c \bullet t\right\rangle$$

The profunctor representation of the polynomial lens is then given by an end over all profunctors in this Tambara category:

$$\mathbf{Po}\langle s, t\rangle\langle a, b\rangle \cong \int_{P \colon \mathcal{T}} \mathbf{Set}\left((UP)\langle a, b\rangle, (UP)\langle s, t\rangle\right)$$