POLYNOMIAL LENSES

BARTOSZ MILEWSKI

Lenses seem to pop up in most unexpected places. Recently a new type of lens showed up as a morphism 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.

1. 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 work). In other words, they are fibrations of some sets S and T over N. In programming we call such families *dependent types*. More generally, we can think of these fibrations as functors from a category N to **Set** and the sum as a coend (the summand is in fact profunctorial in n).

A polynomial functor is sometimes written in the exponential form (in **Set** the internal hom is isomorphic to the external hom):

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

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

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

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:

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

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

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

(I used the Fubini theorem to pull the product past the end.)

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

$$\mathbf{Poly}\left(s \times (-)^t, \sum_n a_n \times (-)^{b_n}\right)$$

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

We can now use the currying adjunction to transform the mapping out of a product:

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

We use the Yoneda lemma to eliminate the end. The result is:

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

The set of natural transformation between two arbitrary polynomials is therefore:

$$\mathbf{Poly}(P, Q) \cong \prod_k \mathbf{Set}\left(s_k, \sum_n a_n \times t_k^{b_n}\right)$$

This transformation is called a polynomial lens.

2. 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.

In the case of polynomial lenses, the foci are described by the family a_n. The recomposition replaces this family by a new family b_n. It turns out that the residue can be represented by a doubly-indexed family c_{mn}. If the indexing is done over a category, rather than a set, the components of c will behave like a profunctor over $\mathcal{N}^{op} \times \mathcal{N}$.

The optic is implemented as a coend over all possible residues:

$$\mathbf{Poly}(P,Q) \cong \int^{c_{ki}} \prod_k \mathbf{Set}\left(s_k, \sum_n c_{nk} \times a_n\right) \times \prod_i \mathbf{Set}\left(\sum_m c_{mi} \times b_m, t_i\right)$$

To show that this representation is equivalent to the previous one, let's perform a few transformations. First, we replace the mapping out of a sum with a product of mappings and use the currying adjunction:

$$\mathbf{Poly}(P,Q) \cong \int^{c_{ki}} \prod_k \mathbf{Set}\left(s_k, \sum_n c_{nk} \times a_n\right) \times \prod_i \prod_m \mathbf{Set}\left(c_{mi}, t_i^{b_m}\right)$$

The double product of mappings over i and m can be seen as a natural transformation between two functors from the discrete category $\mathcal{N}^{op} \times \mathcal{N}$.

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

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

$$\prod_k \mathbf{Set}\left(s_k, \sum_n t_k^{b_n} \times a_n\right)$$

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

3. Generalizations

The general optic can be written as a coend over a monoidal category $\mathcal{M}$ whose action is defined in a pair of categories as $L_m a$ and $R_m b$ respectively:

$$O\langle s,t\rangle\langle a,b\rangle = \int^{m\colon \mathcal{M}} \mathcal{C}(s, L_m a) \times \mathcal{D}(R_m b, t)$$

The polynomial lens follows the same pattern. The main observation is that the objects on which this lens acts come from a functor category $[\mathcal{N}, \mathbf{Set}]$. The monoidal category in question is the category of matrices or, if we chose $\mathcal{N}$ to be a category, profunctors $c_{mn}\colon \mathcal{N}^{op} \times \mathcal{N} \to \mathbf{Set}$.

The monoidal structure is given by “matrix multiplication” :

$$\sum_n c_{mn} \times d_{nk}$$

This can be generalized by replacing the sum by a coend, if we treat the c's as profunctors. The monoidal action is then given by a sum (or a coend):

$$(L_c a)_m = \sum_n c_{nm} \times a_n$$

Finally, the two hom-sets in the definition of an optic become sets of natural transformations in the functor category $[\mathcal{N}, \mathbf{Set}]$. Notice that, for $\mathcal{N}$ a discrete category, the natural transformations are products of morphisms.

$$[\mathcal{N}, \mathbf{Set}](s, t) \cong \prod_k \mathbf{Set}\,(s_k, t_k)$$

Taking it all together, we get:

$$\mathbf{Poly}(P, Q) \cong \int^c [\mathcal{N}, \mathbf{Set}]\,(s, L_c a) \times [\mathcal{N}, \mathbf{Set}]\,(R_c b, t)$$

(In our case the action R is the same as L.)

A product of two morphisms going in opposite directions can be expressed as a single morphism in a product category $[\mathcal{N}, \mathbf{Set}]^{op} \times [\mathcal{N}, \mathbf{Set}]$, which gives us an even simpler representation:

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

$$M_c\langle a, b\rangle_{mn} = \langle \sum_i a_i \times c_{im}, \sum_j b_j \times c_{jn}\rangle$$

4. 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 $\mathbf{T}$ of profunctors $P\langle s_m, t_n\rangle$ acting on functor categories $[\mathcal{N}, Set]$, with additional structure given by the following family of transformations, in components:

$$\alpha_{mn}\colon P\langle s_m, t_n\rangle \to P\left\langle \sum_i c_{im} s_i, \sum_j c_{jn} t_j \right\rangle$$

or as a set:

$$\prod_{m,n} \int_{\langle s,t\rangle, c} \mathbf{Set}\left(P\langle s_m, t_n\rangle, P\left\langle \sum_i c_{im} s_i, \sum_j c_{jn} t_j \right\rangle \right)$$

To simplify notation, I'll use $\mathbb{S}$ for $\mathbf{Set}$ and $\mathbb{S}^2$ for $\mathbf{Set}^{op} \times \mathbf{Set}$.

The profunctor representation is based on the double-Yoneda with adjunction trick. We start with the relevant Tambara modules and a forgetful functor U that forgets the Tambara structure. The following end over the whole Tambara category is isomorphic to the action of the profunctor monad Φ on the hom-functor:

$$\int_{P\colon \mathbf{T}} \mathbb{S}\,((UP)\langle a_n, b_m\rangle, (UP)\langle s_i, t_j\rangle) \cong \Phi\big(\mathbb{S}^2(\langle a_n, b_m\rangle, -)\big)\langle s_i, t_j\rangle$$

The monad is given by the formula:

$$(\Phi P)\langle s_i, t_j\rangle = \sum_{m,n} \int^{\langle x,y\rangle} \int^c \mathbb{S}^2 \left(\left\langle \sum_k c_{km} x_k, \sum_l c_{ln} y_l \right\rangle, \langle s_i, t_j\rangle \right) \times P\langle x_m, y_n\rangle$$

As a quick check: The algebras for this monad, that is natural transformations $\Phi P \to P$, are our Tambara modules. Such natural transformations can be written as an end/product:

$$\prod_{i,j} \int_{\langle s,t \rangle} \mathbb{S}\big((\Phi P)\langle s_i, t_j \rangle, P\langle s_i, t_j \rangle\big)$$

We use the currying adjunction and the fact that mapping out of a co-end/sum is isomorphic to the end/product of the mappings:

$$\prod_{i,j,m,n} \int_{\langle s,t \rangle, \langle x,y \rangle, c} \mathbb{S}\left(\mathbb{S}^2 \left(\left\langle \sum_k c_{km} x_k, \sum_l c_{ln} y_l \right\rangle, \langle s_i, t_j \rangle \right), \mathbb{S}(P\langle x_m, y_n \rangle, P\langle s_i, t_j \rangle) \right)$$

The hom functor:

$$\mathbb{S}^2 \left(\left\langle \sum_k c_{km} x_k, \sum_l c_{ln} y_l \right\rangle, \langle s_i, t_j \rangle \right)$$

acts like a Dirac/Kronecker delta function under the end/product. The result is:

$$\prod_{m,n} \int_{\langle x,y \rangle, c} \mathbb{S}\left(P\langle x_m, y_n \rangle, P \left\langle \sum_k c_{km} x_k, \sum_l c_{ln} y_l \right\rangle \right)$$

which is the set of Tambara structures.

The final step is to apply the profunctor monad Φ to the hom-functor:

$$\Phi\big(\mathbb{S}^2(\langle a_n, b_m \rangle, -)\big)\langle s_i, t_j \rangle$$

We get:

$$\sum_{p,q} \int^{\langle x,y \rangle} \int^{c} \mathbb{S}^2 \left(\left\langle \sum_k c_{kp} x_k, \sum_l c_{lq} y_l \right\rangle, \langle s_i, t_j \rangle \right) \times \mathbb{S}^2(\langle a_n, b_m \rangle, \langle x_p, y_q \rangle)$$

We use the co-Yoneda lemma to integrate over $\langle x, y \rangle$ and sum over p, q to get:

$$\prod_{i,j} \int^{c} \mathbb{S}^2 \left(\left\langle \sum_k c_{ki} a_k, \sum_l c_{lj} b_l \right\rangle, \langle s_i, t_j \rangle \right)$$