Промонады, стрелки и нотация Эйнштейна для профункторов

Bartosz Milewski

Перевод:
Геннадий Чернышев
(https://henrychern.wordpress.com/)

В последнее время я работаю с профункторами. Это интересные штучки, как в теории категорий, так и в программировании. В Haskell они составляют основу профункторной оптики, в частности, библиотеки линз.

Кратко о профункторах

Категорное определение профунктора даже еще и не начинает описывать его содержательность. Можно сказать, что это просто функтор от категории произведений $\mathbb{C}^{op} \times \mathbb{D}$ к **Set** (для упрощения будем придерживаться **Set**, хотя имеются обобщения и для других категорий).

Профунктор P (также именуемый распределителем или бимодулем) отображает пару объектов, c из $\mathbb{C}$ и d из $\mathbb{D}$, к множеству $P(c, d)$. Будучи функтором, он также отображает любую пару морфизмов из $\mathbb{C}^{op} \times \mathbb{D}$:

$$f : c' \to c$$
$$g : d \to d'$$

к функции между множествами:

$$P(f, g) \colon P(c, d) \to P(c', d')$$

Обратите внимание, что первый морфизм f действует в направлении, противоположном тому, что обычно ожидается для функторов. Говорят, что этот профуктор является *контравариантным* по своему первому аргументу и *ковариантным* — по второму.

Но что особенного в этой конкретной комбинации категории источника и категории цели?

Hom-профунктор

Ключевым моментом является осознание того, что профунктор обобщает идею hom-функтора. Как и профунктор, hom-функтор отображает пары объектов к множествам. Действительно, для любых двух объектов из $\mathbb{C}$ имеется множество морфизмов между ними, $C(a, b)$.

Также, любая пара морфизмов из $\mathbb{C}$:

$$f : a' \to a$$
$$g : b \to b'$$

может быть поднята к функции, которую обозначим $C(f, g)$, между hom-множествами:

$$C(f, g) \colon C(a, b) \to C(a', b')$$

Действительно, для любого $h \in C(a, b)$ имеем:

$$C(f, g)h = g \circ h \circ f \in C(a', b')$$

Это (плюс функториальные законы) завершает определение функтора от $\mathbb{C}^{op} \times \mathbb{C}$ к **Set**. Таким образом, hom-функтор — это особый случай эндо-профунктора (когда $\mathbb{D}$ совпадает с $\mathbb{C}$). Он контравариантен по первому аргументу и ковариантен по второму.

Для программистов на Haskell приведем определение профунктора из `Data.Profunctor` — библиотеки от Edward Kmett:

```
class Profunctor p where
  dimap :: (a' -> a) -> (b -> b') -> p a b -> p a' b'
```

Функция `dimap` выполняет поднятие пары морфизмов.

Вот свидетельство того, что hom-функтор, который в Haskell представлен стрелкой `->`, является профунктором:

```
instance Profunctor (->) where
  dimap ab cd bc = cd . bc . ab
```

Более того: профунктор вообще может рассматриваться как расширение hom-функтора, образующий мост между двумя категориями. Рассмотрим профунктор P, охватывающий категории $\mathbb{C}$ и $\mathbb{D}$:

$$P \colon \mathbb{C}^{op} \times \mathbb{D} \to \mathbf{Set}$$

Для любых двух объектов из одной категории имеется регулярное hom-множество. Но если взять один объект c из $\mathbb{C}$, а другой объект d — из $\mathbb{D}$, то можно сгенерировать множество $P(c, d)$. Это множество действует как hom-множество. Его элементы называются *гетероморфизмами*, поскольку их можно рассматривать как представляющие морфизм между двумя различными категориями. Что делает их похожими на морфизмы, так это то, что они могут быть скомпонованы с регулярными морфизмами. Предположим, имеется морфизм из $\mathbb{C}$:

$$f \colon c' \to c$$

и гетероморфизм $h \in P(c, d)$. Их композиция - другой гетероморфизм, полученный поднятием пары (f, id_d). На самом деле:

$$P(f, id_d)\colon P(c, d) \to P(c', d)$$

поэтому его действие на h порождает гетероморфизм от c' к d, который можно назвать *композицией* $h \circ f$ гетероморфизма h с морфизмом f. Аналогично, морфизм из $\mathbb{D}$:

$$g\colon d \to d'$$

может быть скомпонован с h подъемом (id_c, g).

На Haskell эта новая композиция будет реализована путем применения `dimap f id` к предкомпозиции `p c d c`

```
f :: c' -> c
```

и `dimap id g` к ее посткомпозиции с

```
g :: d -> d'
```

Вот как можно использовать профунктор, чтобы склеить две категории. Две категории, связанные профунктором, образуют новую категорию, известную как их *коллаж*.

Данный профунктор обеспечивает однонаправленный поток гетероморфизмов от $\mathbb{C}$ к $\mathbb{D}$, поэтому нет возможности скомпоновать два гетероморфизма.

Профункторы как отношения

Необходимость компоновки гетероморфизмов возникает, когда требуется склеить более двух категорий. Ключ к пониманию того, как поступить, исходит из еще одной интерпретации профункторов: как доказательно-релевантных отношений. В классической логике *отношение* между множествами присваивает булевы *истину* или *ложь* каждой паре элементов. Элементы либо связаны, либо нет, другого не дано. В доказательно-релевантной логике нас интересует не только истинность чего-либо, но и предъявление свидетельств доказательства. Таким образом, вместо назначения одного логического значения для каждой пары элементов мы назначаем целое множество. Если множество пусто, элементы не связаны. Если же оно не пустое, то каждый элемент этого множества является отдельным свидетельством связи.

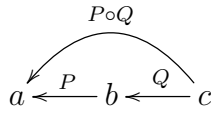
Это определение отношения может быть обобщено на любую категорию. Фактически уже существует естественная связь между объектами в категории — та, которая определяется hom-множествами. Два объекта a и b связаны подобным образом, если hom-множество $C(a, b)$ не пусто, иначе — каждый морфизм в $C(a, b)$ служит свидетельством этого отношения.

С помощью профункторов мы можем определить доказательно-релевантные отношения между объектами, взятыми из разных категорий. Объект c из $\mathbb{C}$ связан с объектом d из $\mathbb{D}$, если $P(c, d)$ — непустое множество. Более того, каждый элемент этого множества служит свидетельством этой связи. Из-за функториальности P это отношение совместимо с категорной структурой, то есть оно хорошо сочетается с отношением, определенным hom-множествами.

В общем случае, композиция двух отношений P и Q, обозначаемая $P \circ Q$, определяется как путь между объектами. Объекты a и c связаны, если существует промежуточный объект b

такой, что $P(a, b)$ и $Q(b, c)$ не являются пустыми. В качестве свидетельства этой связи можно выбрать любую пару элементов, один из $P(a, b)$ и один из $Q(b, c)$.

По соглашению, профунктор $P(a, b)$ изображается стрелкой (часто перечеркнутой) от b к a, $a \not\leftarrow b$.



Композиция профункторов/отношений

Композиция профункторов

Чтобы создать множество всех свидетельств для $P \circ Q$, мы должны собрать их по всем возможным промежуточным объектам и по всем парам свидетельств. Грубо говоря, такая совокупность (по модулю некоторых идентификаций) выражается категорно как ко-конец:

$$(P \circ Q)(a, c) = \int^{b} P(a, b) \times Q(b, c)$$

В качестве напоминания, ко-конец профунктора P — это множество $\int^{a} P(a, a)$, снабженное семейством инъекций

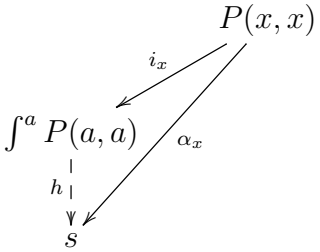
$$i_x \colon P(x, x) \to \int^{a} P(a, a)$$

которое универсально в том смысле, что для любого другого множества s и семейства:

$$\alpha_x \colon P(x, x) \to s$$

существует единственная функция h, которая факторизует их все:

$$\alpha_x = h \circ i_x$$



Универсальное свойство ко-конца

Композиция профункторов может быть переведена на псевдо-Haskell как:

```
type Procompose q p a c = exists b. (p a b, q b c)
```

где ко-конец кодируется как экзистенциальный тип данных. Фактическая реализация (опять же, см. `Data.Profunctor.Composition` от Edward Kmett) выглядит следующим образом:

```
data Procompose q p a c where
  Procompose :: q b c -> p a b -> Procompose q p a c
```

Экзистенциальный квантификатор выражается в терминах GADT (Generalized Algebraic Data Type — обобщенный алгебраический тип данных) со свободным вхождением b внутри конструктора данных.

Соглашение Эйнштейна

К настоящему моменту можно растеряться, наблюдая манипулирование расхождениями объектов, фигурирующих в этих формулах. Например, переменная ко-конца должна появляться под знаком интеграла один раз в ковариантной и один раз в контравариантной позиции, а расхождения справа должны совпадать с расхождениями слева. К счастью, существует прецедент в другой области математики — тензорном исчислении в векторных пространствах с сортом обозначений, учитывающих подобные расхождения. Эйнштейн обобщил и расширил эту нотацию в своей теории относительности. Посмотрим, можно ли адаптировать эту технику к исчислению профункторов.

Прием заключается в том, чтобы записывать контравариантные индексы в виде верхних индексов, а ковариантные — в виде нижних индексов. Итак, с этого момента будем записывать компоненты профунктора p как $p^c{}_d$. Эйнштейн также придумал хитрое соглашение: неявное суммирование по повторному индексу. В случае профункторов суммирование соответствует получению ко-конца. В этих обозначениях ко-конец по профунктору p выглядит как след тензора:

$$p^a{}_a = \int^a p(a, a)$$

Композиция двух профункторов превращается в:

$$(p \circ q)^a{}_c = p^a{}_b \, q^b{}_c = \int^b p(a, b) \times q(b, c)$$

Соглашение по суммированию применяется только к смежным индексам. Когда они разделены явным знаком произведения (или любым другим оператором), ко-конец не предполагается, как в:

$$p^a{}_b \times q^b{}_c$$

(без суммирования).

hom-функтор в категории $\mathbb{C}$ также является профунктором, поэтому его можно записать соответствующим образом:

$$C^a{}_b = C(a, b)$$

Лемма ко-Йонеды приводит к:

$$C^{c}{}_{c'}\, p^{c'}{}_{d} \cong p^{c}{}_{d} \cong p^{c}{}_{d'}\, D^{d'}{}_{d}$$

предполагая, что hom-функторы $C^{c}{}_{c'}$ и $D^{d'}{}_{d}$ ведут себя как дельты Кронекера (в тензорном выражении) или единичные матрицы. Здесь профунктор охватывает две категории:

$$p\colon \mathbb{C}^{op} \times \mathbb{D} \to Set$$

Подъем морфизмов:

$$f : c' \to c$$
$$g : d \to d'$$

может быть записан как:

$$p^{f}{}_{g}\colon p^{c}{}_{d} \to p^{c'}{}_{d'}$$

Имеется еще одна полезная тождественность, которая имеет дело с отображением от ко-конца. Она является следствием того факта, что hom-функтор непрерывен. Это означает, что он отображает (ко-)пределы к пределам. Точнее, поскольку hom-функтор является контравариантным по первой переменной, когда фиксируется целевой объект, он отображает копределы по первой переменной с ограничениями (он также отображает пределы к пределам по второй переменной). Поскольку ко-конец является копределом, а конец — пределом, непрерывность приводит к следующей тождественности:

$$Set(\int^{c} p(c,c), s) \cong \int_{c} Set(p(c,c), s)$$

для любого s. Программисты знают эту идентичность как разбор случаев: функция от типа суммы является произведением функций (одна функция на случай). Если мы интерпретируем ко-конец как экзистенциальный квантификатор, конец эквивалентен универсальному квантификатору.

Применим эту тождественность к отображению от композиции двух профункторов:

$$p^{a}{}_{b}\, q^{b}{}_{c} \to s = Set\Big(\int^{b} p(a,b) \times q(b,c), s\Big)$$

Оно изоморфно

$$\int_{b} Set\Big(p(a,b) \times q(b,c), s\Big)$$

или, после каррирования (используя сопряжение произведения/экспоненциала),

$$\int_{b} Set\Big(p(a,b), q(b,c) \to s\Big)$$

Это дает формулу отображения:

$$p^{a}{}_{b}\, q^{b}{}_{c} \to s \cong p^{a}{}_{b} \to q^{b}{}_{c} \to s$$

с естественной правой частью по b. Опять же, неявное суммирование справа не выполняется, где повторяющиеся индексы разделены стрелкой. Повторяющийся индекс b универсально определяется количественно (до конца), что приводит к естественному преобразованию.

Бикатегория Prof

Поскольку профункторы могут быть скомпонованы, используя формулу ко-конца, естественно спросить, существует ли категория, в которой они работают как морфизмы. Единственная проблема заключается в том, что профункторная композиция удовлетворяет законам ассоциативности и единицы (см. лемму ко-Йонеды) лишь до изоморфизма. Но не беспокойтесь, для этого существуют *бикатегории*. В бикатегории имеются объекты, которые называются 0-клетками; морфизмы, которые называются 1-клетками; и морфизмы между морфизмами, которые называются 2-клетками. Когда говорят, что категорные законы выполняются с точностью до изоморфизма, это означает, что существует обратимая 2-клетка, которая отображает одну сторону закона к другой.

Бикатегория $Prof$ содержит категории в качестве 0-клеток, профункторы как 1-клетки и естественные преобразования как 2-клетки. Естественное преобразование α между профункторами p и q

$$\alpha \colon p \Rightarrow q$$

содержит компоненты, являющиеся функциями:

$$\alpha^{c}{}_{d} \colon p^{c}{}_{d} \to q^{c}{}_{d}$$

удовлетворяющими обычным естественным условиям. Естественные преобразования между профункторами могут быть скомпонованы как функции (это называется вертикальной композицией). На самом деле 2-клетки в любой бикатегории являются составными, и всегда имеются единичные 2-клетки. Отсюда следует, что 1-клетки между любыми двумя 0-клетками образуют категорию, именуемую hom-категорией.

Но существует и другой способ компоновки 2-клеток, который называется горизонтальной композицией. В $Prof$ эта горизонтальная композиция не является обычной горизонтальной композицией естественных преобразований, потому что композиция профункторов не является привычной композицией функторов. Надо построить естественное преобразование между одной композицией профункторов, скажем, $p^{a}{}_{b}\, q^{b}{}_{c}$ и другой, $r^{a}{}_{b}\, s^{b}{}_{c}$, имея в своем распоряжении два естественных преобразования:

$$\alpha : p \Rightarrow r$$
$$\beta : q \Rightarrow s$$

Такая конструкция носит несколько технический характер, поэтому перенесена в приложение. Обозначим эту горизонтальную композицию как:

$$(\alpha \circ \beta)^{a}{}_{c} \colon p^{a}{}_{b}\, q^{b}{}_{c} \to r^{a}{}_{b}\, s^{b}{}_{c}$$

Если одно из естественных преобразований является естественным преобразованием тождественности, скажем, от $p^{a}{}_{b}$ к $p^{a}{}_{b}$, горизонтальная композиция называется *вискерингом* и может быть записана как:

$$(p \circ \beta)^{a}{}_{c} \colon p^{a}{}_{b}\, q^{b}{}_{c} \to p^{a}{}_{b}\, s^{b}{}_{c}$$

Промонады

Тот факт, что монада является моноидом в категории эндофункторов, является счастливой случайностью. Причина в том, что в общем случае монаду можно определить в любой бикатегории, а Cat просто оказывается (строгой) бикатегорией. Она содержит (малые) категории

как 0-клетки, функторы как 1-клетки и естественные преобразования как 2-клетки. Монада определяется как комбинация 0-клетки (категория для определения монады), эндо-1-клетки (эндофунктор в этой категории) и двух 2-клеток, именуемых как умножение и единица, или μ и η, или `join` и `return`.

Поскольку $Prof$ — бикатегория, можно определить в ней монаду и назвать ее промонадой. Промонада состоит из 0-клетки C, которая является категорией; эндо-1-клетки p, которая является профунктором в этой категории; и двух 2-клеток, которые являются естественными преобразованиями:

$$\mu^a{}_b \colon p^a{}_c \, p^c{}_b \to p^a{}_b$$
$$\eta^a{}_b : C^a{}_b \to p^a{}_b$$

Напомним, что $C^a{}_b$ является hom-функтором в категории C, которая, благодаря ко-Йонеде, оказывается единицей профункторной композиции.

Программисты могут узнать здесь элементы `Arrow` из Haskell (см. мое описание о моноидах[1]).

Можно применить тождественность отображения к определению умножения и получить:

$$\mu^a{}_b \colon p^a{}_c \to p^c{}_b \to p^a{}_b$$

Обратите внимание, что это очень похоже на композицию гетероморфизмов. Кроме того, монадическая единица η отображает регулярные морфизмы к гетероморфизмам. Затем, можно построить новую категорию, чьи объекты совпадают с объектами из $\mathbb{C}$, с hom-множествами, задаваемыми профунктором p. То есть, hom-множество от a к b — это множество $p^a{}_b$. Можно определить функтор J, тождественный на объектах, от $\mathbb{C}$ к той категории, действие которой на hom-множествах задается посредством η.

Интересно, что эта конструкция также работает и в противоположном направлении (на что обратил мое внимание Алекс Кэмпбелл). Любой функтор, тождественный на объектах, определяет промонаду. Действительно, заданный функтор J всегда можно превратить в профунктор:

$$p(c, d) = D(J\, c, J\, d)$$

В нотации Эйнштейна это выглядит так:

$$p^c{}_d = D^{J\, c}{}_{J\, d}$$

Поскольку J является тождественностью на объектах, композицию морфизмов из D можно использовать для определения композиции гетероморфизмов. Это, в свою очередь, может использоваться для определения μ, показывая таким образом, что p является промонадой на $\mathbb{C}$.

Заключение

Я понимаю, что затронул некоторые довольно сложные темы в теории категорий, такие как бикатегории и промонады, поэтому немного удивительно, что эти концепции можно проиллюстрировать в Haskell, некоторые из них представлены в популярных библиотеках, таких как `Arrow`, которая имеет приложения в функционально-реактивном программировании.

[1] `https://bartoszmilewski.com/2017/02/09/monoids-on-steroids/`

Я экспериментировал с применением соглашения по суммированию Эйнштейна к профункторам, правда со смешанными результатами. Это, безусловно, работа в развитии, и я приветствую предложения по ее улучшению. Основная проблема заключается в том, что иногда нужно применять сумму (ко-конец), а иногда — произведение (конец) к повторяющимся индексам. Это особенно неудобно при формулировании свойства отображения. Я предлагаю разделить не суммированные индексы знаками произведения или стрелками, но не уверен, насколько хорошо это будет работать.

Приложение: горизонтальная композиция в Prof

В нашем распоряжении имеется два естественных преобразования:

$$\alpha : p \Rightarrow r$$
$$\beta : q \Rightarrow s$$

и следующий ко-конец, который является композицией профункторов p и q:

$$\int^b p(a,b) \times q(b,c)$$

Наша цель — построить элемент целевого ко-конца:

$$\int^b r(a,b) \times s(b,c)$$

Горизонтальная композиция 2-клеток

Чтобы построить элемент ко-конца, нужно предъявить только один элемент из $r(a,b') \times s(b',c)$ для некоторого b'. Будем искать функцию, которая создала бы такой элемент в следующем hom-множестве:

$$Set\Big(\int^b p(a,b) \times q(b,c), r(a,b') \times s(b',c)\Big)$$

Используя нотацию Эйнштейна, можно записать это как:

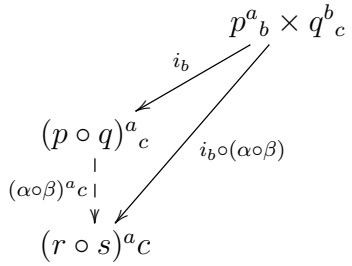
$$p^a{}_b \, q^b{}_c \to r^a{}_{b'} \times s^{b'}{}_c$$

а затем использовать свойство отображения:

$$p^a{}_b \to q^b{}_c \to r^a{}_{b'} \times s^{b'}{}_c$$

Мы можем выбрать b' равным b и реализовать функцию, используя компоненты двух естественных преобразований, $\alpha^a{}_b \times \beta^b{}_c$.

Конечно, так может думать программист. Математик будет использовать универсальное свойство ко-конца $(p \circ q)^a{}_c$, как показано на диаграмме (любезно предоставлено Алексом Кемпбеллом):



Горизонтальная композиция с использованием универсального свойства ко-конца

В Haskell можно определить естественное преобразование между двумя (эндо-) профункторами как полиморфную функцию:

```
newtype PNat p q = PNat (forall a b. p a b -> q a b)
```

Горизонтальная композиция тогда определяется так:

```
horPNat :: PNat p r -> PNat q s -> Procompose p q a c
                                -> Procompose r s a c
horPNat (PNat alpha) (PNat beta) (Procompose pbc qdb) =
        Procompose (alpha pbc) (beta qdb)
```

Acknowledgment

Я благодарен Алексу Кэмпбеллу из Университета Маккуори в Сиднее за обширную помощь в ходе работы над этим текстом.

Дополнительная литература

- Dominique Bourn et Jacques Penon, 2-Catégories Réductibles[2]
- Ross Street, Cauchy characterization of enriched categories[3]

[2] http://tac.mta.ca/tac/reprints/articles/19/tr19.pdf
[3] http://tac.mta.ca/tac/reprints/articles/4/tr4.pdf