Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 308 (2014) 25–48

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Coalgebraic Update Lenses

Danel Ahman[1]

Laboratory for Foundations of Computer Science, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9LE, United Kingdom

Tarmo Uustalu[2]

Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia

Abstract

Lenses are mathematical structures used in the context of bidirectional transformations.
In this paper, we introduce update lenses as a refinement of ordinary (asymmetric) lenses in which we distinguish between views and updates. In addition to the set of views, there is a monoid of updates and an action of the monoid on the set of views. Decoupling updates from views allows for other ways of changing the source than just merging a view into the source. We also consider a yet finer dependently typed version of update lenses.
We give a number of characterizations of update lenses in terms of bialgebras and coalgebras, including analogs to O'Connor's coalgebraic and Johnson, Rosebrugh and Wood's algebraic characterizations of ordinary lenses. We consider conversion of views and updates, a tensor product of update lenses and composition of update lenses.

Keywords: lenses, comonads, monads, coalgebras, algebras, bialgebras, distributive laws, liftings, monoid acts, directed containers

1 Introduction

Lenses are studied in the context of bidirectional transformations. A lens is a structure on two sets mediating actions on them. Think of making sure that two copies of a database, one on a server and one on a client computer, remain in agreement no matter how the database is changed on the client's side; this must also work when the client only sees a part of the database, a "view".

Many different variations of lenses have been considered in the literature. In the seminal work of Foster et al. [8], a *lens* between sets X and S was defined as a pair of maps $\mathsf{lkp} : X \to S$ and $\mathsf{upd} : X \times S \to X$, subject to appropriate

[1] d.ahman@ed.ac.uk

[2] tarmo@cs.ioc.ee

http://dx.doi.org/10.1016/j.entcs.2014.10.003

round-trip laws. Here, X is to be thought of as a set of sources and S as a set of views. O'Connor [18] and also Power and Shkaravska [19] showed that such lenses are nothing but coalgebras of the array comonad for S. Johnson, Rosebrugh and Wood [14] and Gibbons and Johnson [10] characterized lenses as algebras for a certain monad on $\mathbf{Set}/S$.

Other variants of lenses include, for example, quotient lenses [9], symmetric lenses [12], (symmetric) edit lenses [11], and (asymmetric) delta lenses [6].

In this paper, we define a variation of ordinary (asymmetric) lenses, which we call *update lenses* (this name is justified by their practical motivation and agrees with the terminology of our prior related work [3] on update monads). In short, update lenses differ from ordinary state-based lenses in that they distinguish between views and updates. In addition to the set of views S, there is also a monoid $(P, \mathsf{o}, \oplus)$ of updates and an action $\downarrow$ of $(P, \mathsf{o}, \oplus)$ on S. The update map thus becomes $\mathsf{upd} : X \times P \to X$, allowing for other ways of changing the source than just merging a view into the source. There is also a dependently typed version where every view $s : S$ comes with its own set of updates $P\,s$.

We give a number of characterizations of update lenses in terms of bialgebras, algebras and coalgebras, which we derive from standard facts about distributive laws between comonads/functors, compatible composition of comonads, liftings of comonads/functors to categories of coalgebras, distributive laws of monads/functors over comonads/functors, cofree comonads, adjoint monads and comonads etc. [7,5,4,20,21]. We study conversion of views and updates, a tensor of update lenses, composition of update lenses. In the appendix, we also discuss a specialization, initializable update lenses.

Of the related work, the closest to ours is that of Hofmann, Pierce and Wagner [11] on (symmetric) edit lenses. They analyze symmetric lenses, and recognize that edits (in our terminology updates) often carry a natural monoidal structure, and are thus best modeled using monoids. As a result of the symmetric setting, edit lenses consist of two monoids, with the upd operations given by a suitable variation of monoid homomorphisms. A different line of related work is that of Diskin, Xiong, and Czarnecki [6] and Johnson and Rosebrugh [15], on (asymmetric) delta lenses. The latter group of authors also introduced a subclass of delta lenses, called c-lenses, with a more succinct definition. Both groups of authors investigate adding more structure to the sets of sources and views, taking them to be reflexive graphs (or, equivalently, categories). The nodes model sources or views, and arrows model deltas (in our terminology updates) between them. The lkp and upd operations rely on functoriality to guarantee the correct and composable translation of deltas.

For readability, we develop all our mathematics for **Set** (and categories built on **Set**), but for nearly everything we do it could be replaced by any category with finite products or any Cartesian closed category.

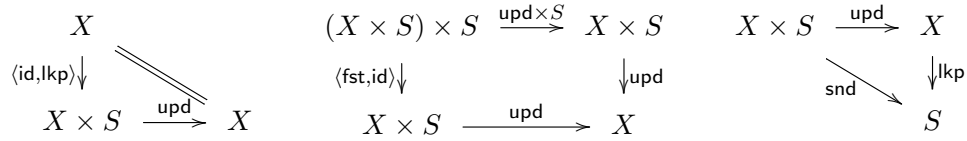
2 Ordinary lenses

Ordinary lenses: the definition

Given a set S (of views), an *ordinary* (asymmetric) *lens* for S is a set X (of sources) and maps $\mathsf{lkp} : X \to S$ (viewing a source) and $\mathsf{upd} : X \times S \to X$ (updating a source) [3] satisfying the conditions

$$x = \mathsf{upd}\,(x, \mathsf{lkp}\,x) \qquad \mathsf{upd}\,(\mathsf{upd}\,(x,s),s') = \mathsf{upd}\,(x,s') \qquad \mathsf{lkp}\,(\mathsf{upd}\,(x,s)) = s$$

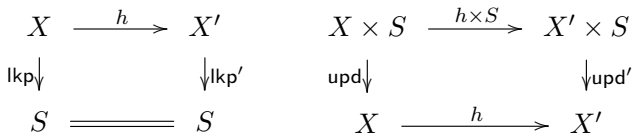
i.e., making the following three diagrams commute



A *map* between two ordinary lenses $(X, \mathsf{lkp}, \mathsf{upd})$, $(X', \mathsf{lkp}', \mathsf{upd}')$ is a map $h : X \to X'$ (conversion of a source) satisfying

$$\mathsf{lkp}\,x = \mathsf{lkp}'\,(h\,x) \qquad h\,(\mathsf{upd}\,(x,s)) = \mathsf{upd}'\,(h\,x, s)$$

i.e.,



Ordinary lenses for S and maps between them form a category $\mathbf{Lens}\,S$.

Running example: editing a bookshop database

Consider modelling a database of a small bookshop. We will work with a very simple database, consisting of only one table, containing information about the identity (e.g., book title and author names, or ISBN), price and quantity of every book in the shop. We let *book*, *price*, *quantity* stand for the sets of allowed values of these data.

We let the set of sources X (the set of database states) be

$$X = \Sigma B \subseteq \mathit{book}.\, B \to \mathit{price} \times \mathit{quantity}$$

Every such database state consists of a set of books B currently in the database, and an assignment of *price* and *quantity* to each book in B.

A simple example of a set of views S for the above set of sources X can be given by

$$S = \Sigma B \subseteq \mathit{book}.\, B \to \mathit{price}$$

[3] We use the letter $\underline{X}$ for the source set and $\underline{S}$ for the view set, and the names lkp for get and upd for put. This may at first feel confusing, but agrees with what is mnemonic for algebras of update monads [3].

This set of views can be used to define an ordinary lens for editing book prices as follows.

We define $\mathsf{lkp} : X \to S$ by simply discarding the *quantity* field of each book, that is

$$\mathsf{lkp}\,(B, v) = (B, \mathsf{fst} \circ v)$$

We define $\mathsf{upd} : X \times S \to X$ by removing all books that are in the database but not in the given view, editing the prices and preserving the quantities of all books that are both in the database and the view, and adding all books that are only in the view with quantity 0. In detail, upd is defined by

$$\mathsf{upd}\,((B, v), (B', v')) = (B', \lambda b.\, \mathsf{if}\ b \in B\ \mathsf{then}\ (v'\, b, \mathsf{snd}\,(v\, b))\ \mathsf{else}\ (v'\, b, 0))$$

Observe that, for this bookshop database to satisfy the 3rd ordinary lens law, the view argument passed to upd has to contain all the books we want to leave in the database after the update. Also, as we cannot restrict which books can be edited, we have to allow the lens to be able to add new books to the source, with some default quantity, here 0.

We claim that such forced choices are a real shortcoming of ordinary lenses, and address it in the rest of this paper by developing the theory of update lenses.

Ordinary lenses: some alternative descriptions

Johnson et al. [14] pointed out that the category of lenses for S is the same as the category of algebras of the following monad (T_1, η_1, μ_1) on $\mathbf{Set}/S$:

$$\begin{gathered} T_1\,(X, \mathsf{lkp}) = (X \times S, \mathsf{snd}) \\ \eta_1\,\{X, \mathsf{lkp}\}\,x = (x, \mathsf{lkp}\,x) \\ \mu_1\,((x, s), s') = (x, s') \end{gathered}$$

Indeed, an algebra of (T_1, η_1, μ_1) is a map $\mathsf{upd} : (X \times S, \mathsf{snd}) \to (X, \mathsf{lkp})$ in $\mathbf{Set}/S$, i.e., a map $\mathsf{upd} : X \times S \to X$ satisfying $\mathsf{lkp}\,(\mathsf{upd}\,(x, s)) = s$, which must moreover satisfy the conditions of an algebra structure of (T_1, η_1, μ_1). The latter amounts exactly to $x = \mathsf{upd}\,(x, \mathsf{lkp}\,x)$ and $\mathsf{upd}\,(\mathsf{upd}\,(x, s), s') = \mathsf{upd}\,(x, s')$.

O'Connor [18] and also Power and Shkaravska [19] observed that the category of lenses is isomorphic to the category of coalgebras of the following comonad (D, ε, δ) on $\mathbf{Set}$ (Power and Shkaravska call it the *array comonad* for S):

$$\begin{gathered} D\,X = S \times (S \to X) \\ \varepsilon\,(s, v) = v\, s \\ \delta\,(s, v) = (s, \lambda s'.\, (s', v)) \end{gathered}$$

Explictly, a coalgebra of (D, ε, δ) is a set X and a map $\mathsf{act} : X \to S \times (S \to X)$ satisfying

$$\begin{gathered} x = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ v\, s \\ \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ (s, \lambda s'.\, \mathsf{act}\,(v\, s')) = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ (s, \lambda s'.\, (s', v)) \end{gathered}$$

The isomorphism assigns to a lens structure $(\mathsf{lkp}, \mathsf{upd})$ on a set X the (D, ε, δ)-coalgebra structure $\mathsf{act} = \langle \mathsf{lkp}, \mathsf{cur}\,\mathsf{upd} \rangle$ on X.

A further characterization, due to Power and Shkaravska [4], is only an equivalence of categories, not an isomorphism. The category of lenses for S is equivalent to **Set**. With a set R, one associates the lens $(X, \mathsf{lkp}, \mathsf{upd})$ defined by $X = S \times R$, $\mathsf{lkp}\,(s, r) = s$, $\mathsf{upd}\,((s, r), s') = (s', r)$. In the converse direction, a lens $(X, \mathsf{lkp}, \mathsf{upd})$ is sent to the set $R = \{x : X \mid \mathsf{lkp}\,x = s_0\}$ where s_0 is some chosen element of S. The choice of s_0 only makes the difference of an isomorphism: for any $s, s' : S$, the function $\lambda x.\, \mathsf{upd}\,(x, s') : \{x : X \mid \mathsf{lkp}\,x = s\} \to \{x : X \mid \mathsf{lkp}\,x = s'\}$ is an isomorphism by the 1st and 3rd lens conditions. (In the special case $S = 0$, we take $R = 0$.)

Intuitively, X is decomposed into two parts: S, which can be both viewed and updated, and R, which can be neither viewed nor updated.

Composition of ordinary lenses

Rather than thinking of S (the set of views) as a fixed set, we can let it vary. We can then define that an (ordinary) lens between two sets X and Y is a pair of maps $\mathsf{lkp} : X \to Y$ and $\mathsf{upd} : X \times Y \to X$ satisfying the three ordinary lens laws.

For any set X, there is the identity lens $(\mathsf{id}\,\{X\}, \mathsf{fst})$. Given two lenses $(\mathsf{lkp}, \mathsf{upd}) : X \to Y$ and $(\mathsf{lkp}', \mathsf{upd}') : Y \to Z$, we define their composition to be $(\mathsf{lkp}'', \mathsf{upd}'') : X \to Z$ where $\mathsf{lkp}''\,x = \mathsf{lkp}'\,(\mathsf{lkp}\,x)$ and $\mathsf{upd}''\,(x, z) = \mathsf{upd}\,(x, \mathsf{upd}'\,(\mathsf{lkp}\,x, z))$. Sets and lenses between them form a category.

3 Update lenses

In this paper, we are interested in a more fine-grained variation of ordinary lenses that we call update lenses.

Update lenses: the definition

We define an *act* to be given by a set S (of views), a monoid $(P, \mathsf{o}, \oplus)$ (of updates) and an action $\downarrow$ of the monoid on the set (describing the outcome of applying any given update on any given view). [5]

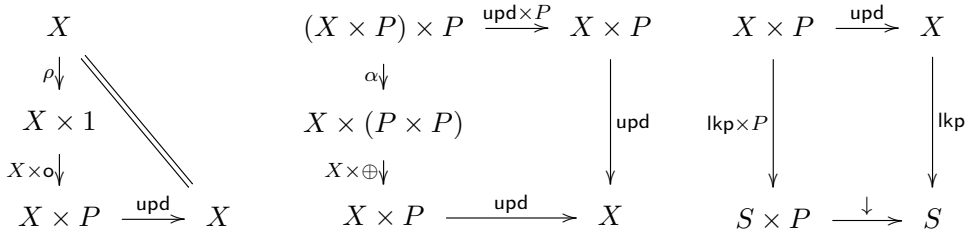
An *update lens* for an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ is a set X (of sources) and maps $\mathsf{lkp} : X \to S$ (viewing a source) and $\mathsf{upd} : X \times P \to X$ (updating a source) satisfying the conditions

$$x = \mathsf{upd}\,(x, \mathsf{o}) \qquad \mathsf{upd}\,(\mathsf{upd}\,(x, p), p') = \mathsf{upd}\,(x, p \oplus p') \qquad \mathsf{lkp}\,(\mathsf{upd}\,(x, p)) = \mathsf{lkp}\,x \downarrow p$$

[4] J. Power, O. Shkaravska, Arrays with garbage, Slides from a talk at the Estonian Computer Science Theory Days at Viinistu, Oct. 2005, http://cs.ioc.ee/~tarmo/tday-viinistu/shkaravska-slides.pdf.

[5] In the literature, the pair $(S, \downarrow)$ is often called a $(P, \mathsf{o}, \oplus)$-*set*. We will occasionally use this terminology too.

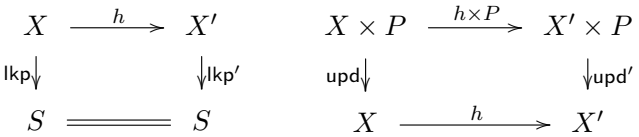
i.e.,



A *map* between two update lenses $(X, \mathsf{lkp}, \mathsf{upd})$, $(X', \mathsf{lkp}', \mathsf{upd}')$ is a map $h : X \to X'$ (conversion of a source) satisfying

$$\mathsf{lkp}\, x = \mathsf{lkp}'\, (h\, x) \qquad h\, (\mathsf{upd}\, (x, p)) = \mathsf{upd}'\, (h\, x, p)$$

i.e.,



Update lenses for an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ and maps between them form a category **ULens** $(S, (P, \mathsf{o}, \oplus), \downarrow)$.

Running example: editing a bookshop database

We now revisit the bookshop example from Section 2.

We first recall that it was somewhat inconvenient to use the ordinary bookshop lens to edit the price of a particular book. In particular, we could not simply list the books whose prices we wanted to edit, but had to also list all the other books. The main cause for this inconvenience was the single set of views S that got used in both lkp and upd. Here we illustrate how decoupling the updates from the views helps us reuse the same set of views for different update strategies.

We keep the set of sources X, the set of views S and the viewing map lkp as before, but let $(P, \mathsf{o}, \oplus)$, $\downarrow$, upd vary depending on what we want to do with the database.

For the first example, we revisit editing book prices by taking P to be $P = (\mathit{book} \times \mathit{price})^*$. Any update $ps : P$ is to be read as a sequence of price modifications of single books. The monoid structure $(\mathsf{o}, \oplus)$ is that of free monoids (with $\mathsf{o} = []$ and $\oplus = +\!\!+$), while $\downarrow$ and upd are defined below.

$$(B, v) \downarrow [] = (B, v)$$
$$(B, v) \downarrow ((b, p) :: ps) = (B, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ p\ \mathsf{else}\ v\, b') \downarrow ps$$

$$\mathsf{upd}\, ((B, v), []) = (B, v)$$
$$\mathsf{upd}\, ((B, v), (b, p) :: ps) = \mathsf{upd}\, ((B, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ (p, \mathsf{snd}\, (v\, b'))\ \mathsf{else}\ v\, b'), ps)$$

(With this definition, if some book we want to modify is not in the database, it will not be added to the database!)

Thanks to the additional freedom offered by update lenses, we can equally well modify the book prices relative to the prices in the database. For this purpose, we can define P to be $(\mathit{book} \times \mathit{change})^*$ (let us agree that a price can only be a non-negative number, but a change can also be negative). The monoid structure $(P, \mathsf{o}, \oplus)$ is as above, but the definitions of $\downarrow$ and upd are adjusted accordingly.

$$(B, v) \downarrow [] = (B, v)$$
$$(B, v) \downarrow ((b, c) :: ps) = (B, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ \max(0, v\, b' + c))\ \mathsf{else}\ v\, b') \downarrow ps$$

$$\mathsf{upd}\,((B, v), []) = (B, v)$$
$$\mathsf{upd}\,((B, v), (b, c) :: ps) =$$
$$\mathsf{upd}\,((B, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ (\max(0, \mathsf{fst}\,(v\, b') + c), \mathsf{snd}\,(v\, b'))\ \mathsf{else}\ v\, b'), ps)$$

As an additional treat, we can also use the same set of views S to perform both deletions and additions of books. We choose P to be $((\mathit{book} \times \mathit{price} \times \mathit{quantity}) + \mathit{book})^*$ where the first summand stands for addition of a book and the second summand for deletion of a book. The monoid structure $(P, \mathsf{o}, \oplus)$ is again that of free monoids.

$$(B, v) \downarrow [] = (B, v)$$
$$(B, v) \downarrow (\mathsf{inl}\,(b, p, q) :: ps) = (B \cup \{b\}, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ p\ \mathsf{else}\ v\, b') \downarrow ps$$
$$(B, v) \downarrow (\mathsf{inr}\, b :: ps) = (B \backslash \{b\}, v|_{B \backslash \{b\}}) \downarrow ps$$

$$\mathsf{upd}\,((B, v), []) = (B, v)$$
$$\mathsf{upd}\,((B, v), \mathsf{inl}\,(b, p, q) :: ps) = \mathsf{upd}\,((B \cup \{b\}, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ (p, q)\ \mathsf{else}\ v\, b'), ps)$$
$$\mathsf{upd}\,((B, v), \mathsf{inr}\, b :: ps) = \mathsf{upd}\,((B \backslash \{b\}, v|_{B \backslash \{b\}}), ps)$$

Of course, we can also combine the addition and deletion of entries with modification of entries, if needed.

Notice that when changing a book's price, or when adding a new book, we have to introduce additional default behavior when the absolute value of a negative change is greater than the book's price resp. if the book is already in the database. This is needed because every update must always be applicable. We will address this deficiency in Section 7 by introducing a dependently typed version of update lenses.

Update lenses as bialgebras of a functor and monad

The category of lenses for $(S, (P, \mathsf{o}, \oplus), \downarrow)$ admits several alternative characterizations. We will now consider these in turn. The ultimate insight will be that update lenses are coalgebras of an appropriate comonad. We will arrive there in several small steps through intermediate characterizations that are also of independent value. We only use standard facts about monads, comonads, distributive laws, liftings, cofree comonads, adjoint monads and comonads. These appear scattered in the literature, some references include [7,5,4,20,21].

We begin by noting that a map lkp from a set X to S is nothing but a coalgebra structure on X of the functor F_0 defined by $F_0\,X = S$. An action upd of $(P, \mathsf{o}, \oplus)$ on X is exactly an algebra structure on X of the monad (T_1, η_1, μ_1) defined by

$$\begin{gathered} T_1\,X = X \times P \\ \eta_1\,x = (x, \mathsf{o}) \\ \mu_1\,((x,p),p') = (x, p \oplus p') \end{gathered}$$

Finally, an action $\downarrow : S \times P \to S$ of $(P, \mathsf{o}, \oplus)$ on S is exactly a distributive law of the monad (T_1, η_1, μ_1) over the functor F_0.

Therefore, an update lens $(X, \mathsf{lkp}, \mathsf{upd})$ for $(S, (P, \mathsf{o}, \oplus), \downarrow)$ is nothing but a $\downarrow$-bialgebra of the functor F_0 and monad (T_1, η_1, μ_1), i.e., a triple of a set X, map $\mathsf{lkp} : X \to S$ and algebra structure $\mathsf{upd} : X \times P \to X$ of (T_1, η_1, μ_1) cohering in the sense of the condition

$$\begin{array}{ccccc} X \times P & \xrightarrow{\mathsf{upd}} & X & \xrightarrow{\mathsf{lkp}} & S \\ {\scriptstyle \mathsf{lkp} \times P}\downarrow & & & & \| \\ S \times P & & \xrightarrow{\quad\downarrow\quad} & & S \end{array}$$

In the same way, update lens maps are bialgebra maps. So the category of update lenses is the same as

(0) the category of $\downarrow$-bialgebras of the functor F_0 and monad (T_1, η_1, μ_1).

It follows that this category can also be described as:

(0') the category of coalgebras of the $\downarrow$-lifting $F_0^{\downarrow}$ of F_0 to the category of algebras of (T_1, η_1, μ_1);

(0") the category of algebras of the $\downarrow$-lifting $(T_1^{\downarrow}, \eta_1^{\downarrow}, \mu_1^{\downarrow})$ of (T_1, η_1, μ_1) to the category of coalgebras of F_0.

Let us spell out (0') and (0").

(0'): The category of algebras of (T_1, η_1, μ_1) is $(P, \mathsf{o}, \oplus)$**-Set**. The $\downarrow$-lifting $F_0^{\downarrow}$ of F_0 to $(P, \mathsf{o}, \oplus)$**-Set** is defined by $F_0^{\downarrow}(X, \mathsf{upd}) = (S, \downarrow)$. A coalgebra of $F_0^{\downarrow}$ is therefore given by a $(P, \mathsf{o}, \oplus)$-set (X, upd) with a map lkp to $(S, \downarrow)$. So the category of coalgebras of $F_0^{\downarrow}$ is $(P, \mathsf{o}, \oplus)$**-Set**$/(S, \downarrow)$.

(0"): The category of coalgebras of F_0 is **Set**$/S$. The $\downarrow$-lifting $(T_1^{\downarrow}, \eta_1^{\downarrow}, \mu_1^{\downarrow})$ of (T_1, η_1, μ_1) is defined by

$$T_1^{\downarrow}\,(X, \mathsf{lkp}) = (X \times P, \lambda(x,p).\,\mathsf{lkp}\,x \downarrow p)$$

An algebra of $(T_1^{\downarrow}, \eta_1^{\downarrow}, \mu_1^{\downarrow})$ is an object (X, lkp) of **Set**$/S$ with a map upd from $(X \times P, \lambda(x,p).\,\mathsf{lkp}\,x \downarrow p)$ satisfying the conditions of a monad algebra.

Update lenses as bialgebras of a comonad and monad

Let $(D_0, \varepsilon_0, \delta_0)$ be the cofree comonad on the functor F_0, explicitly given by

$$\begin{gathered} D_0\, X = S \times X \\ \varepsilon_0\, (s, x) = x \\ \delta_0\, (s, x) = (s, (s, x)) \end{gathered}$$

The category of coalgebras of F_0 (i.e., the category $\mathbf{Set}/S$) is isomorphic to the category of coalgebras of $(D_0, \varepsilon_0, \delta_0)$, whose objects are given by a set X and map $\mathsf{dlkp} : X \to S \times X$ satisfying

$$\begin{array}{ccc} X & \xrightarrow{\mathsf{dlkp}} & S \times X \\ & \searrow & \downarrow{\scriptstyle \mathsf{snd}} \\ & & X \end{array} \qquad \begin{array}{ccc} X & \xrightarrow{\mathsf{dlkp}} & S \times X \\ {\scriptstyle \mathsf{dlkp}}\downarrow & & \downarrow{\scriptstyle \langle \mathsf{fst}, \mathsf{id} \rangle} \\ S \times X & \xrightarrow[S \times \mathsf{dlkp}]{} & S \times (S \times X) \end{array}$$

The isomorphism assigns to any F_0-coalgebra structure $\mathsf{lkp} : X \to S$ on a set X the $(D_0, \varepsilon_0, \delta_0)$-coalgebra structure $\mathsf{dlkp} = \langle \mathsf{lkp}, \mathsf{id} \rangle : X \to S \times X$.

Furthermore, the distributive law $\downarrow$ of (T_1, η_1, μ_1) over F_0 induces a distributive law λ of (T_1, η_1, μ_1) over $(D_0, \varepsilon_0, \delta_0)$, explicitly given by

$$\begin{gathered} \lambda : (S \times X) \times P \to S \times (X \times P) \\ \lambda\, ((s, x), p) = (s \downarrow p, (x, p)) \end{gathered}$$

As a result, the category of update lenses, i.e, the category of $\downarrow$-bialgebras of F_0 and (T_1, η_1, μ_1), is isomorphic to

(1) the category of λ-bialgebras of $(D_0, \varepsilon_0, \delta_0)$ and (T_1, η_1, μ_1).

Explictly, the objects of this category are triples of a set X, $(D_0, \varepsilon_0, \delta_0)$-coalgebra structure $\mathsf{dlkp} : X \to S \times X$ and (T_1, η_1, μ_1)-algebra structure $\mathsf{upd} : X \times P \to X$ satisfying

$$\begin{array}{ccccc} X \times P & \xrightarrow{\mathsf{upd}} & X & \xrightarrow[= \langle \mathsf{lkp}, \mathsf{id} \rangle]{\mathsf{dlkp}} & S \times X \\ {\scriptstyle \mathsf{dlkp} \times P}\downarrow & & & & \uparrow{\scriptstyle S \times \mathsf{upd}} \\ (S \times X) \times P & & \xrightarrow{\lambda} & & S \times (X \times P) \end{array}$$

This category can also be described as:

(1') the category of coalgebras of the λ-lifting $(D_0^\lambda, \varepsilon_0^\lambda, \delta_0^\lambda)$ of $(D_0, \varepsilon_0, \delta_0)$ to the category of algebras of (T_1, η_1, μ_1);

(1") the category of algebras of the λ-lifting $(T_1^\lambda, \eta_1^\lambda, \mu_1^\lambda)$ of (T_1, η_1, μ_1) to the category of coalgebras of $(D_0, \varepsilon_0, \delta_0)$.

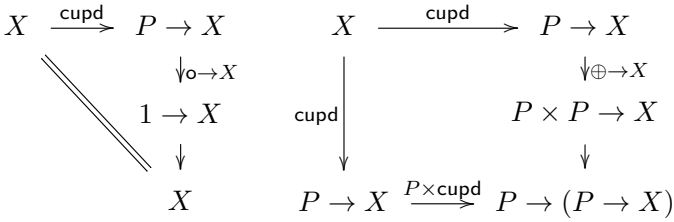
Update lenses as pairs of coalgebras of a functor and comonad

Instead of replacing the functor F_0 with a comonad, we could replace the monad (T_1, η_1, μ_1) with a different comonad. This is what we will embark on now.

Consider the following comonad $(D_1, \varepsilon_1, \delta_1)$:

$$D_1\, X = P \to X$$
$$\varepsilon_1\, v = v\, \mathsf{o}$$
$$\delta_1\, v = \lambda p.\, \lambda p'.\, v\, (p \oplus p')$$

The functor D_1 is the right adjoint of the functor T_1. What is more, the comonad $(D_1, \varepsilon_1, \delta_1)$ is the corresponding "right adjoint" of the monad (T_1, η_1, μ_1) in the sense of [7]. As a consequence, the category of algebras of (T_1, η_1, μ_1) is isomorphic to the category of coalgebras of $(D_1, \varepsilon_1, \delta_1)$. Explicitly, a coalgebra of $(D_1, \varepsilon_1, \delta_1)$ is an object X with a map $\mathsf{cupd} : X \to (P \to X)$ satisfying

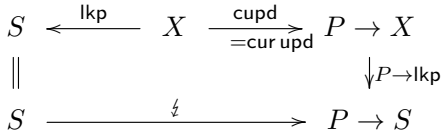


The isomorphism assigns to a (T_1, η_1, μ_1)-algebra structure $\mathsf{upd} : X \times P \to X$ on a set X the $(D_1, \varepsilon_1, \delta_1)$-coalgebra structure $\mathsf{cupd} = \mathsf{cur}\, \mathsf{upd} : X \to (P \to X)$.

Further the distributive law $\downarrow$ of (T_1, η_1, μ_1) over F_0 induces a distributive law $\lightning$ of F_0 over $(D_1, \varepsilon_1, \delta_1)$, namely $\lightning = \mathsf{cur} \downarrow$. It follows that the category of $\downarrow$-bialgebras of F_0 and (T_1, η_1, μ_1) is isomorphic to

(2) the category of $\lightning$-matching pairs of coalgebras of F_0 and $(D_1, \varepsilon_1, \delta_1)$.

Explicitly, the objects of this category are given by triples of a set X, map $\mathsf{lkp} : X \to S$ and coalgebra structure $\mathsf{cupd} : X \to (P \to X)$ of $(D_1, \varepsilon_1, \delta_1)$ satisfying



This category can also be described as:

(2') the category of coalgebras of the lifting $F_0^{\lightning}$ of the functor F_0 to the category of coalgebras of $(D_1, \varepsilon_1, \delta_1)$.

Update lenses as pairs of coalgebras of two comonads

We now combine what we did in the last two paragraphs. We replace both F_0 with $(D_0, \varepsilon_0, \delta_0)$ and (T_1, η_1, μ_1) with $(D_1, \varepsilon_1, \delta_1)$. Both the distributive law λ of (T_1, η_1, μ_1) over $(D_0, \varepsilon_0, \delta_0)$ and the distributive law $\lightning$ of F_0 over $(D_1, \varepsilon_1, \delta_1)$ give us the following distributive law θ of $(D_0, \varepsilon_0, \delta_0)$ over $(D_1, \varepsilon_1, \delta_1)$:

$$\theta : S \times (P \to X) \to (P \to S \times X)$$
$$\theta\, (s, v) = \lambda p.\, (s \downarrow p, v\, p)$$

The category of update lenses, i.e., the category of $\downarrow$-bialgebras of the functor F_0 and monad (T_1, η_1, μ_1), is therefore isomorphic to

(3) the category of θ-matching pairs of coalgebras of $(D_0, \varepsilon_0, \delta_o)$ and $(D_1, \varepsilon_1, \delta_1)$.

Explicitly, an object of this category is given by a set X, $(D_0, \varepsilon_0, \delta_0)$-coalgebra structure $\mathsf{dlkp} : X \to S \times X$ and $(D_1, \varepsilon_1, \delta_1)$-coalgebra structure $\mathsf{cupd} : X \to (P \to X)$ satisfying

$$\begin{array}{ccccc} S \times X & \xleftarrow[=\langle \mathsf{lkp}, \mathsf{id} \rangle]{\mathsf{dlkp}} & X & \xrightarrow[=\mathsf{cur\ upd}]{\mathsf{cupd}} & P \to X \\ {\scriptstyle S \times \mathsf{cupd}}\downarrow & & & & \downarrow{\scriptstyle P \to \mathsf{dlkp}} \\ S \times (P \to X) & & \xrightarrow{\quad\theta\quad} & & P \to S \times X \end{array}$$

The same category can also be described as

(3') the category of coalgebras of the θ-lifting $(D_0^\theta, \varepsilon_0^\theta, \delta_0^\theta)$ of the comonad $(D_0, \varepsilon_0, \delta_0)$ to the category of coalgebras of $(D_1, \varepsilon_1, \delta_1)$.

Update lenses as coalgebras of a comonad

We have two comonads $(D_0, \varepsilon_0, \delta_0)$ and $(D_1, \varepsilon_1, \delta_1)$ with a distributive law θ of the former over the latter. This gives a compatible composite comonad (D, ε, δ):

$$\begin{gathered} D\,X = (D_0 \cdot D_1)\,X = S \times (P \to X) \\ \varepsilon\,(s, v) = v\,\mathsf{o} \\ \delta\,(s, v) = (s, \lambda p.\,(s \downarrow p, \lambda p'.\,v\,(p \oplus p'))) \end{gathered}$$

The category of update lenses, i.e., of θ-matching pairs of coalgebras of $(D_0, \varepsilon_0, \delta_0)$ and $(D_1, \varepsilon_1, \delta_1)$, is isomorphic to

(4) the category of (D, ε, δ)-coalgebras.

Explicitly, a (D, ε, δ)-coalgebra is a set X with a map $\mathsf{act} : X \to S \times (P \to X)$ satisfying

$$x = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ v\,\mathsf{o}$$

$$\mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ (s, \lambda p.\,\mathsf{act}\,(v\,p)) = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\,x\ \mathsf{in}\ (s, \lambda p.\,(s \downarrow p, \lambda p'.\,v\ (p \oplus p')))$$

i.e.,

$$\begin{array}{ccc} X & \xrightarrow{\mathsf{act}} & S \times (P \to X) \\ & \searrow & \downarrow{\scriptstyle \varepsilon} \\ & & X \end{array} \qquad \begin{array}{ccc} X & \xrightarrow{\qquad\mathsf{act}\qquad} & S \times (P \to X) \\ {\scriptstyle \mathsf{act}}\downarrow & & \downarrow{\scriptstyle \delta} \\ S \times (P \to X) & \xrightarrow{S \times (P \to \mathsf{act})} & S \times (P \to S \times (P \to X)) \end{array}$$

The isomorphism associates to an update lens structure $(\mathsf{lkp}, \mathsf{upd})$ on a set X a (D, ε, δ)-coalgebra structure $\mathsf{act} = \langle \mathsf{lkp}, \mathsf{cur\ upd} \rangle$.

Following two routes (0)-(1)-(3)-(4) and (0)-(2)-(3)-(4), we have derived that update lenses for $(S, (P, \mathsf{o}, \oplus), \downarrow)$ are essentially the same as coalgebras of the comonad

(D, ε, δ). The algebraic characterization (0") is analogous to that of Johnson et al. [14] for ordinary lenses for a set S and the coalgebraic characterization (4) is analogous to O'Connor's [18], while perhaps the most basic one is (0'), which says that an update lens is a $(P, \mathsf{o}, \oplus)$-set with a map to $(S, \downarrow)$. Many of the other alternative characterizations of update lenses are without counterparts for ordinary lenses.

We have no opportunity to discuss this here in any detail, but update lenses for $(S, (P, \mathsf{o}, \oplus), \downarrow)$, i.e., coalgebras of the comonad (D, ε, δ), comodel the same (generally large) Lawvere theory that is modelled by the algebras of what we have elsewhere [3] called the update monad for $(S, (P, \mathsf{o}, \oplus), \downarrow)$. This monad (T, η, μ) is defined by

$$\begin{gathered} T\,X = S \to P \times X \\ \eta\,x = \lambda s.\,(\mathsf{o}, x) \\ \mu\,f = \lambda s.\,\mathsf{let}\ (p, g) \leftarrow f\,s\ \mathsf{in}\ \mathsf{let}\ (p', x) \leftarrow g\,(s \downarrow p)\ \mathsf{in}\ (p \oplus p', x) \end{gathered}$$

Update lenses: an equivalent characterization

Here is a different characterization, which is only an equivalence of categories, not an isomorphism.

We first observe that any act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ defines a category $\langle\!\langle S, (P, \mathsf{o}, \oplus), \downarrow \rangle\!\rangle$ where an object is an element s of S and a map between $s, s' : S$ is an element p of P such that $s \downarrow p = s'$. The identity on s is o and the composition of p and p' is $p \oplus p'$.

The category $\mathbf{ULens}\,(S, (P, \mathsf{o}, \oplus), \downarrow)$ is equivalent to the functor category $[\langle\!\langle S, (P, \mathsf{o}, \oplus), \downarrow \rangle\!\rangle, \mathbf{Set}]$. A functor $R : \langle\!\langle S, (P, \mathsf{o}, \oplus), \downarrow \rangle\!\rangle \to \mathbf{Set}$ is mapped to the update lens $(X, \mathsf{lkp}, \mathsf{upd})$ defined by $X = \Sigma s : S.\,R\,s$, $\mathsf{lkp}\,(s, r) = s$ and $\mathsf{upd}\,((s, r), p) = (s \downarrow p, R\,p\,r)$. In the converse direction, an update lens $(X, \mathsf{lkp}, \mathsf{upd})$ is mapped to the functor R defined by $R\,s = \{x : X \mid \mathsf{lkp}\,x = s\}$, $R\,p\,x = \mathsf{upd}\,(x, p)$.

Turning ordinary lenses into update lenses

Given an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$, any ordinary lens $(X, \mathsf{lkp} : X \to S, \mathsf{upd} : X \times S \to X)$ for S defines an update lens $(X, \mathsf{lkp}, \mathsf{upd}' : X \times P \to X)$ for $(S, (P, \mathsf{o}, \oplus), \downarrow)$ via $\mathsf{upd}'\,(x, p) = \mathsf{upd}\,(x, \mathsf{lkp}\,x \downarrow p)$. And any map h between two ordinary lenses for S is also a map between the corresponding update lenses for $(S, (P, \mathsf{o}, \oplus), \downarrow)$. This gives us a functor from $\mathbf{Lens}\,S$ to $\mathbf{ULens}\,(S, (P, \mathsf{o}, \oplus), \downarrow)$.

This functor is "caused" by a morphism τ between the comonads (D, ε, δ), $(D', \varepsilon', \delta')$ where $D\,X = S \times (S \to X)$ and $D'\,X = S \times (P \to X)$. It is defined by $\tau\,(s, v) = (s, \lambda p.\,v\,(s \downarrow p))$. Any morphism between two comonads gives a functor between the corresponding categories of coalgebras.

4 Conversions of views and updates

So far we have seen that update lens maps $h : (X, \mathsf{lkp}, \mathsf{upd}) \to (X', \mathsf{lkp}', \mathsf{upd}')$ model the conversion from a source set X to a source set X', for a fixed act $(S, (P, \mathsf{o}, \oplus), \downarrow)$.

In this section, we discuss how maps between acts give rise to conversions between view and update sets, for a fixed source set X.

A *map* (t, q) between acts $(S, (P, \mathsf{o}, \oplus), \downarrow)$ and $(S', (P', \mathsf{o}', \oplus'), \downarrow')$ consists of a function (conversion of views) $t : S \to S'$ and a monoid homomorphism (conversion of updates) $q : (P', \mathsf{o}', \oplus') \to (P, \mathsf{o}, \oplus)$, such that

$$t\,(s \downarrow q\,p) = t\,s \downarrow' p$$

(Notice the reversed direction of the monoid homomorphism! In the literature, e.g., [17], one typically requires that $q : (P, \mathsf{o}, \oplus) \to (P', \mathsf{o}', \oplus')$ and $t\,(s \downarrow p) = t\,s \downarrow' q\,p$, but this is not what is needed here.) Acts form a category **Act**.

We say that a *conversion of views and updates* induced by a morphism (t, q) between acts $(S, (P, \mathsf{o}, \oplus), \downarrow)$, $(S', (P', \mathsf{o}', \oplus'), \downarrow')$ is a functor

$$\mathbf{ULens}\,(t, q) : \mathbf{ULens}\,(S, (P, \mathsf{o}, \oplus), \downarrow) \to \mathbf{ULens}\,(S', (P', \mathsf{o}', \oplus'), \downarrow')$$

defined as

$$\begin{gathered}\mathbf{ULens}\,(t, q)\,(X, \mathsf{lkp}, \mathsf{upd}) = (X, t \circ \mathsf{lkp}, \mathsf{upd} \circ (X \times q)) \\ \mathbf{ULens}\,(t, q)\,h = h\end{gathered}$$

We note that **ULens** is a functor from **Act** to **CAT**.

From the functor **ULens** we can build the total category **ULensTot** of all update lenses by using the Grothendieck construction (see, e.g., [13, §1.10]), i.e., $\mathbf{ULensTot} = \int \mathbf{ULens}$.

In detail, an object of **ULensTot** is an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ together with an update lens $(X, \mathsf{lkp}, \mathsf{upd})$ for that act, i.e., an object of $\mathbf{ULens}\,(S, (P, \mathsf{o}, \oplus), \downarrow)$. A map between two objects $((S, (P, \mathsf{o}, \oplus), \downarrow), (X, \mathsf{lkp}, \mathsf{upd}))$ and $((S', (P', \mathsf{o}', \oplus'), \downarrow'), (X', \mathsf{lkp}', \mathsf{upd}'))$ is an act map $(t, q) : (S, (P, \mathsf{o}, \oplus), \downarrow) \to (S', (P', \mathsf{o}', \oplus'), \downarrow')$ paired with a map $h : \mathbf{ULens}\,(t, q)\,(X, \mathsf{lkp}, \mathsf{upd}) \to (X', \mathsf{lkp}', \mathsf{upd}')$ of update lenses for $(S', (P', \mathsf{o}', \oplus'), \downarrow')$.

Conversions of views and updates as mappings of comonad coalgebras to comonad coalgebras

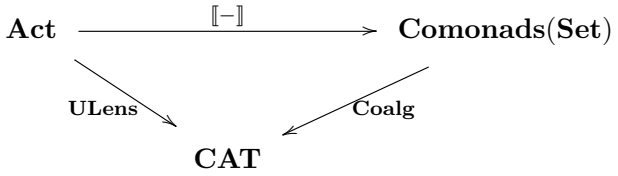
Just as every act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ defines a comonad $[\![S, (P, \mathsf{o}, \oplus), \downarrow]\!] = (D, \varepsilon, \delta)$, every morphism (t, q) between two acts $(S, (P, \mathsf{o}, \oplus), \downarrow)$ and $(S', (P', \mathsf{o}', \oplus'), \downarrow')$ defines a morphism $[\![t, q]\!] = \tau$ between the comonads $[\![S, (P, \mathsf{o}, \oplus), \downarrow]\!]$ and $[\![S', (P', \mathsf{o}', \oplus'), \downarrow']\!]$. Explictly, this natural transformation is defined by

$$\begin{gathered}\tau : \forall \{X\}.\, (S \times (P \to X)) \to S' \times (P' \to X) \\ \tau\,(s, v) = (t\,s, v \circ q)\end{gathered}$$

$[\![-]\!]$ forms a functor from **Act** to $\mathbf{Comonads}(\mathbf{Set})$.

As already said earlier, a morphism τ between two comonads (D, ε, δ) and $(D', \varepsilon', \delta')$ determines a functor between the corresponding categories of coalgebras, sending a (D, ε, δ)-coalgebra (X, act) to the $(D', \varepsilon', \delta')$-coalgebra $(X, \tau\,\{X\} \circ \mathsf{act})$.

While the category $\mathbf{ULens}\,(S,(P,\mathsf{o},\oplus),\downarrow)$ is isomorphic to the category of coalgebras of the comonad $[\![S,(P,\mathsf{o},\oplus),\downarrow]\!]$, the functor $\mathbf{ULens}\,(t,q) : \mathbf{ULens}\,(S,(P,\mathsf{o},\oplus),\downarrow) \to \mathbf{ULens}\,(S',(P',\mathsf{o}',\oplus'),\downarrow')$ is isomorphic to the functor between the categories of coalgebras of $[\![S,(P,\mathsf{o},\oplus),\downarrow]\!]$ and $[\![S',(P',\mathsf{o}',\oplus'),\downarrow']\!]$, induced by the comonad morphism $[\![t,q]\!]$. In summary, we have that the following diagram commutes up to isomorphism:



5 Tensoring update lenses

Sometimes we might want to collect multiple update lenses into a single update lens structure. We show how this can be done with a tensor product on the category of all update lenses.

We define the unit act I to be the act $(1,(1,!,!),!)$.

For any two acts $(S_0,(P_0,\mathsf{o}_0,\oplus_0),\downarrow_0)$ and $(S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1)$, we define their tensor product by

$$(S_0,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1) \otimes (S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1) = (S_0 \times S_1,(P_0 \times P_1,\mathsf{o},\oplus),\downarrow)$$

where

$$\mathsf{o} = (\mathsf{o}_0,\mathsf{o}_1) \qquad (p_0,p_1) \oplus (p_0',p_1') = (p_0 \oplus_0 p_0', p_1 \oplus_1 p_1')$$
$$(s_0,s_1) \downarrow (p_0,p_1) = (s_0 \downarrow_0 p_0, s_1 \downarrow_1 p_1)$$

I and $\otimes$ make **Act** into a monoidal category.

Now for the act I we have a unit lens $J = (1,!,!)$.

And for two update lenses $(X_0,\mathsf{lkp}_0,\mathsf{upd}_0)$ and $(X_1,\mathsf{lkp}_1,\mathsf{upd}_1)$ for acts $(S_0,(P_0,\mathsf{o}_0,\oplus_0),\downarrow_0)$ resp. $(S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1)$, there is the tensor update lens

$$(X_0,\mathsf{lkp}_0,\mathsf{upd}_0) \boxtimes (X_1,\mathsf{lkp}_1,\mathsf{upd}_1) = (X_0 \times X_1,\mathsf{lkp},\mathsf{upd})$$

for the act $(S_0,(P_0,\mathsf{o}_0,\oplus_0),\downarrow_0) \otimes (S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1)$ with

$$\mathsf{lkp}\,(x_0,x_1) = (\mathsf{lkp}_0\,x_0, \mathsf{lkp}_1\,x_1) \qquad \mathsf{upd}\,(x_0,x_1)\,(p_0,p_1) = (\mathsf{upd}_0\,(x_0,p_0), \mathsf{upd}_1\,(x_1,p_1))$$

With this we have shown that **ULens** is a lax monoidal functor from **Act** to **CAT** (wrt. the $(I,\otimes)$ monoidal structure on **Act** and the product monoidal structure on **CAT**), where the monoidality witnesses are

$$J : 1 \to \mathbf{ULens}\,I$$
$$\boxtimes : \mathbf{ULens}\,(S_0,(P_0,\mathsf{o}_0,\oplus_0),\downarrow_0) \times \mathbf{ULens}\,(S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1) \to \mathbf{ULens}((S_0,(P_0,\mathsf{o}_0,\oplus_0),\downarrow_0) \otimes (S_1,(P_1,\mathsf{o}_1,\oplus_1),\downarrow_1))$$

It is also evident that $(I, \otimes)$ and $(J, \boxtimes)$ endow the total category **ULensTot** with a monoidal structure. The unit is (I, J) and the tensor of $((S_0, (P_0, \mathsf{o}_0, \oplus_0), \downarrow_0), (X_0, \mathsf{lkp}_0, \mathsf{upd}_0))$ and $((S_1, (P_1, \mathsf{o}_1, \oplus_1), \downarrow_1), (X_1, \mathsf{lkp}_1, \mathsf{upd}_1))$ is $((S_0, (P_0, \mathsf{o}_0, \oplus_0), \downarrow_0) \otimes (S_1, (P_1, \mathsf{o}_1, \oplus_1), \downarrow_1)), (X_0, \mathsf{lkp}_0, \mathsf{upd}_0) \boxtimes (X_1, \mathsf{lkp}_1, \mathsf{upd}_1))$.

The tensor product on **ULensTot** is kind of a "parallel composition" of update lenses.

The category of acts also carries a product monoidal structure. However, due to its involved definition and lack of space, we refrain from discussing it here. For the dependently typed version of acts discussed in Section 7, we refer the reeader to [2] for details.

6 Composition of update lenses

We have already seen how to convert sources (for a fixed act) and views/updates (for a fixed set of sources). We now discuss a notion of composition of two update lenses where the view set of one update lens is the source set of another.

We develop the definition in two steps, starting from the alternative characterization (0') of update lenses from Section 3.

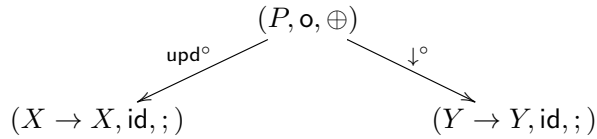
We recall from (0') that an update lens $(X, \mathsf{lkp} : X \to S, \mathsf{upd} : X \times P \to X)$ for an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ is essentially the same as an object $((X, \mathsf{upd}), \mathsf{lkp} : (X, \mathsf{upd}) \to (S, \downarrow))$ of $(P, \mathsf{o}, \oplus)$-**Set**$/(S, \downarrow)$. If we keep $(P, \mathsf{o}, \oplus)$ fixed, but let $(S, \downarrow)$ vary, we can say that an update lens for $(P, \mathsf{o}, \oplus)$ is a $(P, \mathsf{o}, \oplus)$-set map $\mathsf{lkp} : (X, \mathsf{upd}) \to (Y, \mathsf{upd}')$. The identity update lens on (X, upd) is then $\mathsf{id}\,\{X\}$ and the composition of two update lenses $\mathsf{lkp} : (X, \mathsf{upd}) \to (Y, \mathsf{upd}')$ and $\mathsf{lkp}' : (Y, \mathsf{upd}') \to (Z, \mathsf{upd}'')$ is $\mathsf{lkp}' \circ \mathsf{lkp}$.

To be able to compose update lenses for possibly different monoids, we need to let $(P, \mathsf{o}, \oplus)$ vary too. We achieve this by switching to a more involved category where again lenses are maps.

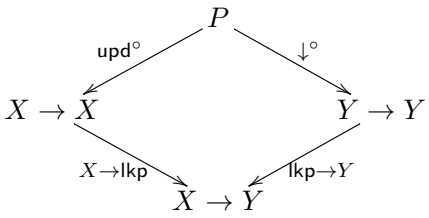
An object of this category is just a set. But a map between two sets X, Y is given by a monoid $(P, \mathsf{o}, \oplus)$ with actions upd and $\downarrow$ on X resp. Y (up to isomorphism in the choice of these data) and a map $\mathsf{lkp} : X \to Y$ satisfying $\mathsf{lkp}\,(\mathsf{upd}\,(x, p)) = \mathsf{lkp}\,x \downarrow p$.

The identity map on X is $(P, \mathsf{o}, \oplus, \mathsf{upd}, \downarrow, \mathsf{lkp})$ where $P = X \to X$, $\mathsf{o} = \mathsf{id}$, $f \oplus g = g \circ f$, $\mathsf{upd}\,(x, f) = x \downarrow f = f\,x$, $\mathsf{lkp} = \mathsf{id}$. The composition of two maps $(P_0, \mathsf{o}_0, \oplus_0, \mathsf{upd}_0, \downarrow_0, \mathsf{lkp}_0) : X \to Y$ and $(P_1, \mathsf{o}_1, \oplus_1, \mathsf{upd}_1, \downarrow_1, \mathsf{lkp}_1) : Y \to Z$ is $(P, \mathsf{o}, \oplus, \mathsf{upd}, \downarrow, \mathsf{lkp})$ where $P = \{(p_0, p_1) : P_0 \times P_1 \mid \forall y : Y.\, y \downarrow_0 p_0 = \mathsf{upd}_1\,(y, p_1)\}$, $\mathsf{o} = (\mathsf{o}_0, \mathsf{o}_1)$, $(p_0, p_1) \oplus (p_0', p_1') = (p_0 \oplus_0 p_0', p_1 \oplus_1 p_1')$, $\mathsf{upd}\,(x, (p_0, p_1)) = \mathsf{upd}_0\,(x, p_0)$, $z \downarrow (p_0, p_1) = z \downarrow_1 p_1$, $\mathsf{lkp} = \mathsf{lkp}_1 \circ \mathsf{lkp}_0$.

This treatment of an update lens between X and Y as a map between X and Y arises from analyzing the data $(P, \mathsf{o}, \oplus, \mathsf{upd}, \downarrow)$ as a span of monoid morphisms to the endomap monoids $(X \to X, \mathsf{id}, ;)$ and $(Y \to Y, \mathsf{id}, ;)$

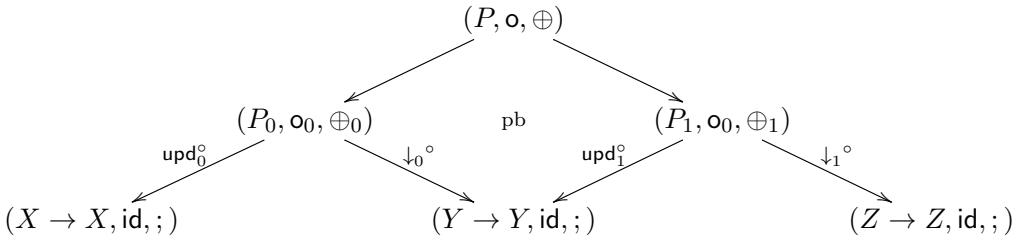


and lkp as a map between X and Y satisfying



where $\mathsf{upd}^\circ : P \to (X \to X)$ and $\downarrow^\circ : P \to (Y \to Y)$ are obtained from upd and $\downarrow$ by swapping the arguments and currying.

Under this view, it is only natural to take the identity update lens on X to be given the span of identity monoid morphisms with the vertex $(X \to X, \mathsf{id}, ;)$ and the identity map on X. The composition of lenses $(P_0, \mathsf{o}_0, \oplus_0, \mathsf{upd}_0, \downarrow_0, \mathsf{lkp}_0) : X \to Y$ and $(P_1, \mathsf{o}_1, \oplus_1, \mathsf{upd}_1, \downarrow_1, \mathsf{lkp}_1) : Y \to Z$ can be taken to be given by the following span of monoid morphisms constructed with a pullback



and by the composition of lkp_0 and lkp_1 as maps.

7 A dependently typed generalization

It often happens that every view comes with a specific set of safe or allowed updates. For example, if we recall our bookshop database, then one expects to be able to edit and delete only the books appearing in the view and to add only the books that are not in the view. As we saw in Section 3, update lenses do not offer this flexibility; one has to provide $\downarrow$ and upd with suitable additional default behavior. The reason for this is that, with a single monoid of updates, all updates must act on every possible view.

We propose to remove the need for such additional default behavior by introducing a dependently typed version of update lenses. This development is based on our previous work with Chapman [2] on directed containers, a specialization of Abbott, Altenkirch and Ghani's containers [1]. Due to lack of space, we must keep the presentation very brief and refer the interested reader to our earlier work [2,3].

Containers, directed containers

Recall that a *container* is given by a set S and a S-indexed family P of sets. A *map* between two containers (S, P) and (S', P') is given by functions $t : S \to S'$ and

$q : \Pi\{s : S\}.\, P'\,(t\,s) \to P\,s$. Containers and maps between them form a category **Cont**. Any container (S, P) gives rise to a set functor $[\![S, P]\!]^{\mathrm{c}}$ defined by

$$[\![S, P]\!]^{\mathrm{c}} X = \Sigma s : S.P\,s \to X$$
$$[\![S, P]\!]^{\mathrm{c}} h = \lambda(s, v).(s, h \circ v)$$

Any container map (t, q) defines a natural transformation $[\![t, q]\!]^{\mathrm{c}} : [\![S, P]\!]^{\mathrm{c}} \to [\![S', P']\!]^{\mathrm{c}}$ by $[\![t, q]\!]^{\mathrm{c}} = \lambda(s, v).\,(t\,s, v \circ q)$. $[\![-]\!]^{\mathrm{c}}$ is a fully faithful functor from **Cont** to $[\mathbf{Set}, \mathbf{Set}]$.

A *directed container* $(S, P, \downarrow, \mathsf{o}, \oplus)$ is given by a container (S, P) and operations

$$\downarrow : \Pi s : S.\, P\,s \to S$$
$$\mathsf{o} : \Pi\{s : S\}.\, P\,s$$
$$\oplus : \Pi\{s : S\}.\, \Pi p : P\,s.\, P\,(s \downarrow p) \to P\,s$$

satisfying five conditions

$$s \downarrow \mathsf{o} = s \qquad s \downarrow (p \oplus p') = (s \downarrow p) \downarrow p'$$
$$p \oplus \mathsf{o} = p \qquad \mathsf{o} \oplus p = p \qquad (p \oplus p') \oplus p'' = p \oplus (p' \oplus p'')$$

Directed containers are a dependently typed generalization of acts. Here we do not have a single monoid and not a family of monoids either, but rather a single monoid-like structure spread out over multiple carriers $P\,s$. If one ignores the dependent typing of the above five equations, they are exactly the equations of an act.

A *map* between directed containers $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ is a map (t, q) between the underlying containers (S, P) and (S', P') that satisfies

$$t\,(s \downarrow q\,p) = t\,s \downarrow' p \qquad \mathsf{o} = q\,\mathsf{o}' \qquad q\,p \oplus q\,p' = q\,(p \oplus' p')$$

Of course, a map between directed containers is a dependently typed generalization of a map between acts. Directed containers form a category **DCont**.

A directed container $(S, P, \downarrow, \mathsf{o}, \oplus)$ determines a comonad $[\![S, P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}} = (D, \varepsilon, \delta)$ given by

$$D\,X = [\![S, P]\!]^{\mathrm{c}}\,X = \Sigma s : S.\, P\,s \to X$$
$$\varepsilon\,(s, v) = v\,\mathsf{o}$$
$$\delta\,(s, v) = (s, \lambda p.\,(s \downarrow p, \lambda p'.\, v\,(p \oplus p')))$$

For a map (t, q) between two directed containers $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$, the natural transformation $[\![t, q]\!]^{\mathrm{c}}$ is a comonad map between $[\![S, P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}}$ and $[\![S', P', \downarrow', \mathsf{o}', \oplus']\!]^{\mathrm{dc}}$.

The fully faithful functor $[\![-]\!]^{\mathrm{c}} : \mathbf{Cont} \to [\mathbf{Set}, \mathbf{Set}]$ lifts to a fully faithful functor $[\![-]\!]^{\mathrm{dc}} : \mathbf{DCont} \to \mathbf{Comonads}(\mathbf{Set})$. In fact, $[\![-]\!]^{\mathrm{dc}}$ is the pullback in **CAT** of $[\![-]\!]^{\mathrm{c}}$ along $U : \mathbf{Comonads}(\mathbf{Set}) \to [\mathbf{Set}, \mathbf{Set}]$.

Dependently typed update lenses

A *dependently typed update lens* (X, act) for a directed container $(S, P, \downarrow, \mathsf{o}, \oplus)$ is a set X (of sources) and a map $\mathsf{act} : X \to \Sigma s : S.\, P\, s \to X$ satisfying

$$x = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\, x\ \mathsf{in}\ v\, \mathsf{o}$$
$$\mathsf{let}\ (s, v) \leftarrow \mathsf{act}\, x\ \mathsf{in}\ (s, \lambda p.\, \mathsf{act}\, (v\, p)) = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\, x\ \mathsf{in}\ (s, \lambda p.\, (s \downarrow p, \lambda p'.\, v\ (p \oplus p')))$$

A *map* between dependently typed update lenses (X, act) and (X', act') is a map $h : X \to X'$ (conversion of a source) satisfying

$$\mathsf{act}'\, (h\, x) = \mathsf{let}\ (s, v) \leftarrow \mathsf{act}\, x\ \mathsf{in}\ (s, h \circ v)$$

The category of dependently typed update lenses is precisely the category of coalgebras of the comonad (D, ε, δ) defined above.

While simply-typed update lenses fail to subsume ordinary lenses (since an arbitrary unstructured set S does not carry monoid structure), an ordinary lens $(X, \mathsf{lkp}, \mathsf{upd})$ for a set S is a dependently typed update lens for $(S, P, \downarrow, \mathsf{o}, \oplus)$ given by $P\, s = S$, $s \downarrow s' = s'$, $\mathsf{o}\, \{s\} = s$, $s' \oplus \{s\}\, s'' = s''$.

Running example: editing a bookshop database

We briefly revisit our bookshop database example in the context of dependently typed update lenses. We define a dependently typed update lens structure for editing book prices in a type-safe way.

We keep the set of sources X and the set of views S as before. We define the S-indexed family P of updates inductively as heterogeneous lists by the two rules

$$\frac{}{[] : P\, (B, v)} \qquad \frac{v\, b + c \geq 0 \quad ps : P\, (B, \lambda b' \in B.\, \mathsf{if}\ b = b'\ \mathsf{then}\ v'\, b + c\ \mathsf{else}\ v'\, b)}{(b, c) :: ps : P\, (B, v)}$$

Here, $P\, (B, v)$ encodes sequences of single book price changes whose application is guaranteed to lead to no negative price in the database, if all prices in the given view (B, v) of the database are nonnegative.

The monoid structure $(\mathsf{o}, \oplus)$ is again that of nil and append with the dependently typed $\downarrow$ and act defined as below.

$$(B, v) \downarrow [] = (B, v)$$
$$(B, v) \downarrow ((b, c) :: ps) = (B, \lambda b' \in B.\, \mathsf{if}\ b = b'\ \mathsf{then}\ v\, b' + c\ \mathsf{else}\ v\, b') \downarrow ps$$

$$\mathsf{act}\, (B, v) = ((B, \mathsf{fst} \circ v), \lambda ps.\, \mathsf{act}'\, (B, v)\, ps)$$

where

$$\mathsf{act}' : \Pi (B, v) : X.\, P\, (B, \mathsf{fst} \circ v) \to X$$
$$\mathsf{act}'\, (B, v)\, [] = (B, v)$$
$$\mathsf{act}'\, (B, v)\, ((b, c) :: ps) =$$
$$\qquad \mathsf{act}'\, (B, \lambda b'.\, \mathsf{if}\ b' = b\ \mathsf{then}\ (\mathsf{fst}\, (v\, b') + c, \mathsf{snd}\, (v\, b'))\ \mathsf{else}\ v\, b')\, ps$$

8 Conclusions and future work

Combining insights from our work on update monads [3] with existing knowledge about ordinary (asymmetric) lenses [18,19,14,10], we were led to a concept of update lenses.

Update lenses are a refinement of ordinary lenses that we find both practically meaningful and theoretically elegant. Most interestingly perhaps, update lenses admit decompositions that ordinary lenses do not enjoy: they can be seen as pairs of matching coalgebras, bialgebras etc. These characterizations arise naturally from various kinds of distributive laws.

We wish to continue this work by turning to symmetric variations of update lenses. We expect to be able to build on both the original work on symmetric lenses [12,11] and the newest categorical ideas of Johnson and Rosebrugh [16]. We also plan to find out the precise connections to delta and c-lenses [15].

Acknowledgement

Tarmo Uustalu thanks Andreas Abel for reminding him of O'Connor's work. This ongoing work is being supported by the University of Edinburgh Principal's Career Development PhD Scholarship, the ERDF funded Estonian CoE project EXCS and ICT National Programme project "Coinduction", the Estonian Ministry of Education and Research target-financed research theme 0140007s12 and the Estonian Science Foundation grant no. 9475.

References

[1] Abbott, M., T. Altenkirch, N. Ghani. *Containers: constructing strictly positive types,* Theor. Comput. Sci. **342**(1), 2005, pp. 3–27.

[2] Ahman, D., J. Chapman, and T. Uustalu, *When is a container a comonad?* Log. Methods in Comput. Sci. **10**(3), 2014, article 14.

[3] Ahman, D. and T. Uustalu, *Update monads: cointerpreting directed containers,* in: R. Matthes and A. Schubert, eds., "Proc. of 19th Int. Conf. on Types for Proofs and Programs, TYPES '13 (Toulouse, Apr. 2013)," Leibniz Int. Proc. in Informatics **26**, Dagstuhl Publishing, 2014, pp. 1–23.

[4] Barr, M. and C. Wells, *Toposes, Triples and Theories,* Grundlehren der mathematischen Wissenschaften **278**, Springer, 1984.

[5] Beck, J., *Distributive laws,* in: B. Eckmann, ed., "Seminar on Triples and Categorical Homology, ETH 1966/67," Lect. Notes in Math. **80**, Springer, 1969, pp. 119–140.

[6] Diskin, Z., Y. Xiong, and K. Czarnecki, *From state- to delta-based bidirectional model transformations: the asymmetric case,* J. of Object Technology **10**, 2011, article 6.

[7] Eilenberg, S. and J. Moore, *Adjoint functors and triples,* Illinois J. of Math. **9**(3), 1965, pp 381–398.

[8] Foster, J. N., M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt, *Combinators for bidirectional tree transformations: a linguistic approach to the view-update problem,* ACM Trans. on Program. Lang. and Syst. **29**(3), 2007, article 17.

[9] Foster, J. N., A. Pilkiewicz, and B. C. Pierce, *Quotient lenses,* in: "Proc. of 13th ACM SIGPLAN Int. Conf. on Functional Programming, ICFP '08 (Victoria, BC, Sept. 2008)," ACM, 2008, pp. 383–396.

[10] Gibbons, J. and M. Johnson, *Relating algebraic and coalgebraic descriptions of lenses,* in: F. Hermann and J. Voigtländer, eds., "Proc. of 1st Int. Wksh. on Bidirectional Transformations, BX 2012 (Tallinn, March 2012)," Electron. Commun. of EASST **49**, 2012, 16 pp.

[11] Hofmann, M., B. C. Pierce, and D. Wagner, *Edit lenses,* in: "Proc. of 39th Ann. ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '12 (Philadelphia, PA, January 2012)," ACM, 2012, pp. 495–508.

[12] Hofmann, M., B. C. Pierce, and D. Wagner, *Symmetric lenses,* in: "Proc. of 38th Ann. ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '11 (Austin, TX, Jan. 2011)," ACM, 2011, pp. 371–384.

[13] Jacobs, B., *Categorical Logic and Type Theory,* Studies in Logic and the Foundations of Mathematics **141**, North Holland, 1999.

[14] Johnson, M., R. Rosebrugh, and R. J. Wood, *Algebras and update strategies,* J. of Univ. Comput. Sci **16**(5), 2010, pp. 729–748.

[15] Johnson, M. and R. Rosebrugh, *Delta lenses and opfibrations,* in: P. Stevens and J. F. Terwilliger, "Proc. of 2nd Int. Wksh. on Bidirectional Transformations, BX 2013 (Rome, March 2013)," Electron. Commun. of EASST **57**, 2013, 18 pp.

[16] Johnson, M. and R. Rosebrugh, *Spans of lenses,* in: K. Selcuk-Candan et al., eds., "Proc. of Wkshs. of EDBT/ICDT 2014 Joint Conf. (Athens, March 2014)," *CEUR Wksh. Proc.* **1133**, RWTH Aachen, 2014, pp. 112–118.

[17] Kilp, M., U. Knauer, A. V. Mikhalev, *Monoids, Acts and Categories: With Applications to Wreath Products and Graphs,* De Gruyter Expositions in Mathematics **29**, De Gruyter, 2000.

[18] O'Connor, R., *Functor is to lens as applicative is to biplate: introducing multiplate,* arXiv preprint 1103.2841, 2011. (Paper presented at 2011 ACM SIGPLAN Wksh. on Generic Programming, WGP '11, Tokyo, Sept. 2011.)

[19] Power, J. and O. Shkaravska, *From comodels to coalgebras: state and arrays,* in: J. Adámek and S. Milius, eds., "Proc. of 7th Int. Wksh. on Coalgebraic Methods in Computer Science, CMCS '04 (Barcelona, March 2004)," Electr. Notes in Theor. Comput. Sci. **106**, Elsevier, 2004, pp. 297–314.

[20] Power, J. and H. Watanabe, *Combining a monad and a comonad,* Theor. Comput. Sci. **280**(1-2), 2002, pp. 137–162.

[21] Tanaka, M., *Pseudo-distributive laws and unified framework for variable binding,* PhD thesis, LFCS, Univ. of Edinburgh, 2005.

A Initializable ordinary and update lenses

In this section we discuss a further specialization that is applicable to both ordinary and update lenses, initializability.

A.1 Initializable ordinary lenses

In the lenses literature [9,12], one sometimes equips an ordinary lens $(X, \mathsf{lkp}, \mathsf{upd})$ for a set S with an additional map $\mathsf{create} : S \to X$ (creation of a source from a view) satisfying

$$\mathsf{lkp}\,(\mathsf{create}\,s) = s \qquad \mathsf{upd}\,(\mathsf{create}\,s, s') = \mathsf{create}\,s'$$

i.e.,

$$\begin{array}{ccc} S & \xrightarrow{\mathsf{create}} & X \\ & \searrow & \downarrow{\scriptstyle \mathsf{lkp}} \\ & & S \end{array} \qquad\qquad \begin{array}{ccc} S \times S & \xrightarrow{\mathsf{create} \times S} & X \times S \\ {\scriptstyle \mathsf{snd}}\downarrow & & \downarrow{\scriptstyle \mathsf{upd}} \\ S & \xrightarrow{\mathsf{create}} & X \end{array}$$

We call such a structure $(X, \mathsf{lkp}, \mathsf{upd}, \mathsf{create})$ an *initializable lens*.

A *map* between two initializable lenses $(X, \mathsf{lkp}, \mathsf{upd}, \mathsf{create})$ and $(X', \mathsf{lkp}', \mathsf{upd}', \mathsf{create}')$ is a map h between $(X, \mathsf{lkp}, \mathsf{upd})$ and $(X', \mathsf{lkp}', \mathsf{upd}')$ that also satisfies

$$h\,(\mathsf{create}\,s) = \mathsf{create}'\,s$$

i.e.,

$$\begin{array}{ccc} S & = & S \\ {\scriptstyle \mathsf{create}}\downarrow & & \downarrow{\scriptstyle \mathsf{create}'} \\ X & \xrightarrow{h} & X' \end{array}$$

Of course initializable lenses for S and maps between them form a category.

A.2 Initializable update lenses

Update lenses admit this specialization too.

Initializable update lenses: the definition

We define an *initializable update lens* $(X, \mathsf{lkp}, \mathsf{upd}, \mathsf{create})$ for an act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ to be given by an update lens $(X, \mathsf{lkp}, \mathsf{upd})$ together with an additional map $\mathsf{create} : S \to X$ satisfying

$$\mathsf{lkp}\,(\mathsf{create}\,s) = s \qquad \mathsf{upd}\,(\mathsf{create}\,s, p) = \mathsf{create}\,(s \downarrow p)$$

i.e.,

$$\begin{array}{ccc} S & \xrightarrow{\mathsf{create}} & X \\ & \searrow & \downarrow{\scriptstyle \mathsf{lkp}} \\ & & S \end{array} \qquad \begin{array}{ccc} S \times P & \xrightarrow{\mathsf{create} \times P} & X \times P \\ \downarrow\downarrow & & \downarrow{\scriptstyle \mathsf{upd}} \\ S & \xrightarrow{\mathsf{create}} & X \end{array}$$

A *map* between two initializable update lenses $(X, \mathsf{lkp}, \mathsf{upd}, \mathsf{create})$ and $(X', \mathsf{lkp}', \mathsf{upd}', \mathsf{create}')$ is a map h between $(X, \mathsf{lkp}, \mathsf{upd})$ and $(X', \mathsf{lkp}', \mathsf{upd}')$ that also satisfies

$$h\,(\mathsf{create}\,s) = \mathsf{create}'\,s$$

i.e.,

$$\begin{array}{ccc} S & = & S \\ {\scriptstyle \mathsf{create}}\downarrow & & \downarrow{\scriptstyle \mathsf{create}'} \\ X & \xrightarrow{h} & X' \end{array}$$

Initializable update lenses for $(S, (P, \mathsf{o}, \oplus), \downarrow)$ and maps between them form a category.

Initializable update lenses: alternative descriptions

Recall the definitions of F_0, (T_1, η_1, μ_1) and (D, ε, δ) from Section 3.

It is immediate from the definition that the category of initializable update lenses is the same as

(o) the category of triples of a coalgebra of F_0, algebra of (T_1, η_1, μ_1) and algebra of F_0 on the same carrier, pairwise matched by the distributive law $\downarrow$ of (T_1, η_1, μ_1) over F_0 (used twice) and the distributive law id of F_0 over itself.

Explicitly, the objects of this category are quadruples of a set X, map $\mathsf{lkp} : X \to S$, (T_1, η_1, μ_1)-algebra structure $\mathsf{upd} : X \times P \to X$ and map $\mathsf{create} : S \to X$ satisfying

$$\begin{array}{ccccc}
X \times P & \xrightarrow{\mathsf{upd}} & X & \xrightarrow{\mathsf{lkp}} & S \\
\downarrow{\scriptstyle \mathsf{lkp} \times P} & & & & \| \\
S \times P & & \xrightarrow{\quad\downarrow\quad} & & S
\end{array}
\qquad
\begin{array}{ccccc}
S & \xrightarrow{\mathsf{create}} & X & \xrightarrow{\mathsf{lkp}} & S \\
\| & & & & \| \\
S & & = & & S
\end{array}
\qquad
\begin{array}{ccccc}
X \times P & \xrightarrow{\mathsf{upd}} & X & \xleftarrow{\mathsf{create}} & S \\
\uparrow{\scriptstyle \mathsf{create} \times P} & & & & \| \\
S \times P & & \xrightarrow{\quad\downarrow\quad} & & S
\end{array}$$

(remember that a (T_1, η_1, μ_1)-algebra structure on X is an action of $(P, \mathsf{o}, \oplus)$ on X).

We already know that the category of $\downarrow$-matching bialgebras of F_0 and (T_1, η_1, μ_1) is isomorphic to the category of coalgebras of (D, ε, δ). It is also the case that the distributive laws id of F_0 over itself and $\downarrow$ of (T_1, η_1, μ_1) over F_0 make $[\mathsf{id}, \mathsf{cur}\downarrow]$ a distributive law of F_0 over (D, ε, δ). It follows that the category of initializable update lenses is also isomorphic to

(i) the category of bialgebras of (D, ε, δ) and F_0 matched by the distributive law $\langle \mathsf{id}, \mathsf{cur}\downarrow \rangle$ of F_0 over (D, ε, δ).

Explicitly, the objects of this category are triples of a set X, (T_1, η_1, μ_1)-coalgebra structure $\mathsf{act} : X \to S \times (P \to X)$ and map $\mathsf{create} : S \to X$ satisfying

$$\begin{array}{ccccc}
S & \xrightarrow{\mathsf{create}} & X & \xrightarrow[=\langle \mathsf{lkp}, \mathsf{cur}\ \mathsf{upd} \rangle]{\mathsf{act}} & S \times (P \to X) \\
\| & & & & \uparrow{\scriptstyle S \times (P \to \mathsf{create})} \\
S & & \xrightarrow{\langle \mathsf{id}, \mathsf{cur}\downarrow \rangle} & & S \times (P \to S)
\end{array}$$

But notably we can also pack together upd and create instead of upd and lkp. This is done in two steps.

First we observe that the category of algebras of F_0 is isomorphic to the category of algebras of the free monad (T_0, η_0, μ_0) on F_0, explicitly defined by

$$\begin{aligned}
T_0\, X &= S + X \\
\eta_0\, x &= \mathsf{inr}\, x \\
\mu_0\, (\mathsf{inl}\, s) &= \mathsf{inl}\, s \\
\mu_0\, (\mathsf{inl}\, (\mathsf{inl}\, s)) &= \mathsf{inl}\, s \\
\mu_0\, (\mathsf{inl}\, (\mathsf{inr}\, x)) &= \mathsf{inr}\, x
\end{aligned}$$

The isomorphism assigns to a F_0-algebra (X, create) the (T_0, η_0, μ_0)-algebra $(X, \mathsf{dcreate})$ where $\mathsf{dcreate} = [\mathsf{create}, \mathsf{id}]$.

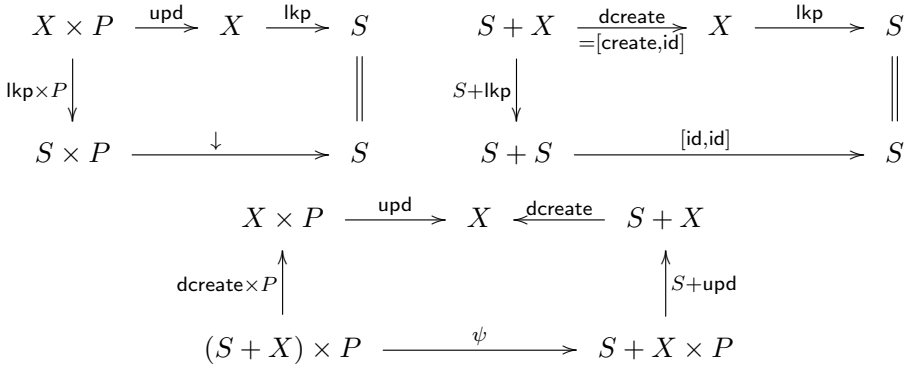
The distributive laws id of F_0 over itself and $\downarrow$ of (T_1, η_1, μ_1) over F_0 induce distributive laws $[\mathsf{id}, \mathsf{id}]$ of (T_0, η_0, μ_0) over F_0 and ϕ of (T_1, η_1, μ_1) over (T_0, η_0, μ_0) where

$$\begin{gathered}\psi : (S + X) \times P \to S + X \times P \\ \psi\,(\mathsf{inl}\, s, p) = \mathsf{inl}\,(s \downarrow p) \\ \psi\,(\mathsf{inr}\, x, p) = \mathsf{inr}\,(x, p)\end{gathered}$$

The category of initializable update lenses is therefore also isomorphic to

(ii) the category of triples of a coalgebra of F_0, algebra of (T_1, η_1, μ_1) and algebra of (T_0, η_0, μ_0) on a common carrier pairwise matched by the distributive laws $\downarrow$, $[\mathsf{id}, \mathsf{id}]$ and ψ.

Explicitly, the objects of this category are quadruples of a set X, map $\mathsf{lkp} : X \to S$, (T_1, η_1, μ_1)-algebra structure $\mathsf{upd} : X \times P \to X$ and (T_0, η_0, μ_0)-algebra structure $\mathsf{dcreate} : S + X \to X$ satisfying



Second, we notice that the distributive law ψ of (T_1, η_1, μ_1) over (T_0, η_0, μ_0) defines a compatible composition (T, η, μ) of the two monads, explicitly

$$\begin{gathered}TX = S + X \times P \\ \eta\, x = \mathsf{inr}\,(x, \mathsf{o}) \\ \mu\,(\mathsf{inl}\, s) = \mathsf{inl}\, s \\ \mu\,(\mathsf{inr}\,(\mathsf{inl}\, s, p)) = \mathsf{inl}\,(s \downarrow p) \\ \mu\,(\mathsf{inr}\,(\mathsf{inr}(x, p), p') = \mathsf{inr}\,(x, p \oplus p')\end{gathered}$$

The category of algebras of (T, η, μ) is isomorphic to the category of ψ-matching pairs of algebras of (T_0, η_0, μ_0) and (T_1, η_1, μ_1). The algebra of (T, η, μ) corresponding to a ψ-matching pair $(X, \mathsf{create}, \mathsf{upd})$ of algebras of (T_0, η_0, μ_0) and (T_1, η_1, μ_1) is $(X, \mathsf{maintain})$ where $\mathsf{maintain} = [\mathsf{create}, \mathsf{upd}]$.

The distributive laws $\downarrow$ of (T_1, η_1, μ_1) over F_0 and $[\mathsf{id}, \mathsf{id}]$ of (T_0, η_0, μ_0) over F_0 determine a distributive law $[\mathsf{id}, \downarrow]$ of (T, η, μ) over F_0. As a result, the category of initializable update lenses is isomorphic to

(iii) the category of bialgebras of F_0 and (T, η, μ) for the distributive law $[\mathsf{id}, \downarrow]$.

Explicitly, the objects of this category are triples of a set X, map $\mathsf{lkp} : X \to S$ and (T, η, μ)-algebra structure $\mathsf{maintain} : S + X \times P \to X$ satisfying

$$\begin{array}{ccccc} S + X \times P & \xrightarrow[=[\mathsf{create},\mathsf{upd}]]{\mathsf{maintain}} & X & \xrightarrow{\mathsf{lkp}} & S \\ {\scriptstyle S+\mathsf{lkp}\times P}\downarrow & & & & \| \\ S + S \times P & & \xrightarrow[[\mathsf{id},\downarrow]]{} & & S \end{array}$$

We see that we can construct initializable update lenses by first giving ourselves the means to initialize and update the source, and only then to view the source.