Edit Space Lenses

—draft—

Musa Al-hassy and Zinovy Diskin

July 20, 2020

1 Background

In this section, let $\mathcal{A}$ and $\mathcal{B}$ be categories. We will use the notation $\mathsf{Obj}\,\mathcal{X}, \mathsf{Hom}\mathcal{X}, \mathsf{src}\,, \mathsf{tgt}\,, \_ ⨾ \_, \mathsf{Id}$ to refer to the objects of a category $\mathcal{X}$, its homsets, the source object assignment, the target object assignment, the (forward/diagrammatic) composition operation, and for the identity operations; respectively.

Definition 0: Lifting

Let $F : \mathsf{Obj}\,\mathcal{A} \to \mathsf{Obj}\,\mathcal{B}$ be a *function on objects*. Define a **lift**[a] **for** F to be a *family G_A of functions on arrows*, for each object $A : \mathcal{A}$:

$$G_A : \mathsf{Hom}(A, -) \leftarrow_{\mathcal{A}} \mathsf{Hom}(F\,A, -)$$

$$\begin{tikzcd} A \arrow[r, "u"'] \arrow[d, "F"] & A' \arrow[d, "F"] \\ B \arrow[r, "v"] \arrow[ur, phantom, "\Uparrow G_A"] & B' \end{tikzcd}$$

[a]The diagram suggests the name 'lift' since arrows in the bottom category $\mathcal{B}$ are assigned to arrows in the top category $\mathcal{A}$.

This paper adheres to the following *colouring discipline* for diagrams: Items in hand are coloured black, whereas *derived* elements are coloured in blue. Intermediary elements, such as those that are output of one transformation but input to another, will be coloured grey.

Example 0: Common Liftings

- For F the identity function, the identity function $\alpha \mapsto \alpha$ is a lift.
- For $\mathcal{A} = \mathcal{B} = \mathcal{S}et$, the category of sets and functions, and F a bijective map on sets, we may take $G_A(v) = F \fatsemi v \fatsemi F^{-1}$ where $A' = F^{-1}\Big(\{v(b) \mid b \in F(A)\}\Big)$.

  In general, F acts as a form of 'query' on $\mathcal{A}$-objects that yields local, focused, information. Then the lift G allows us to transport the local transformation —e.g., the set comprehension above— to the global setting.
- Recall that a functor F is an *obfibration* if given any source object $A : \mathcal{A}$ and an $\mathcal{B}$-arrow $u : F\,A \to B'$, there is universal $\mathcal{A}$-arrow $G_A(u) : A \to A'$ with $F\,(G_A(u)) = u$.

  Every opfibration is thus a lift as witnessed by the G_A arrows. As such, out notion of lifts generalises the idea of opfibrations by dropping the universality requirement.

Definition 1: Stable Lifting

A lifting is **stable** if it preserves identities.

$$G_A\left(\mathsf{Id}_{F\,A}\right) \quad = \quad \mathsf{Id}_{F\,A} \qquad \text{(STABILITY)}$$

Definition 2: Sectional Lifting

Let $F : \mathcal{A} \to \mathcal{B}$ be a *functor*, then a lifting for its object mapping is **sectional** if F's morphism mapping is a post-inverse to each member of the family:

$$F\,(G_A\,\alpha) \quad = \quad \alpha \qquad \text{(SECTIONAL)}$$

Definition 3: Delta Lens

A **delta lens**[a] is a functor that has a stable sectional lifting.

More explicitly, a delta lens is a tuple $(\mathcal{A}, \mathcal{B}, \mathsf{get}, \mathsf{put})$ consisting of two categories $\mathcal{A}$ and $\mathcal{B}$, along with a functor $\mathsf{get} : \mathcal{A} \to \mathcal{B}$ that has put as its lifting that satisfies both (STABILITY) and (SECTIONAL).

[a] This is also known as a *very-well behaved delta lens* that lacks the PutPut law, which usually does not hold in applications and so we ignore it.

Theorem 0: Delta Lenses form a subcategory "δ-Lens" of $\mathcal{C}at$

The identity functor is clearly a delta lens; it remains to show that the composition of functors that are delta lenses is again a delta lens. Indeed, if F_1 and F_2 are functors with stable sectional liftings G_1 and G_2, respectively, then $F_1 ⨾ F_2$ has $A \mapsto G_{2A} ⨾ G_{1A}$ as a stable sectional lifting: Each property is immediately verified by, in the following diagram, starting with the given black elements then following the G_i's to obtain the required blue elements.

$$\begin{tikzcd}
A \arrow[r, "u"'] \arrow[d, "F_1"] \arrow[dr, phantom, "\Uparrow G_{1A}"] & A' \arrow[d, "F_1"] \\
B \arrow[r, "v", "v"'] \arrow[d, "F_2"] \arrow[dr, phantom, "\Uparrow G_{2A}"] & B' \arrow[d, "F_2"] \\
C \arrow[r, "w"] & C'
\end{tikzcd}$$

2 Asymmetric Edit Delta Lenses

Definition 4: Edit Space

An **edit space** is an (cloven) obfibration[a].

[a]For computing purposes, existence is most useful when taken *constructively*: It is not enough for something to merely exist, but rather a (computable) construction of the thing must be provided for an existence claim to be reasonable. As such, our opfibrations are cloven by default.

Formally, an edit space is a tuple $(\mathcal{A}, \mathcal{M}, \mathsf{control}, \mathsf{apply})$ consisting of:

1. a category of "$\mathcal{A}$ctive processes",
2. a category of "$\mathcal{M}$otions", or '$\mathcal{M}$echanics', —these delimit the behaviour of processes; 'contracts, metamodels, specifications'
3. a functor $\mathsf{control} : \mathcal{A} \to \mathcal{M}$ that indicates how a process is controlled or behaved,
4. a lifting $\mathsf{apply} : \mathsf{Hom}_{\boldsymbol{\mathcal{A}}}(A, -) \leftarrow \mathsf{Hom}_{\mathbb{M}}(\mathsf{control} A, -)$ where we interpret $\mathsf{apply}_A(\phi : \mathsf{control} A \to M)$ as "the execution of the ϕ-update to A's permitted behaviour".
5. such that the lifting produces op-Cartesain arrows:

Given any $v : M \to M'$ and any $A : \mathsf{Obj}\,\mathcal{A}$ with $\mathsf{control}\,A = M$, we have that

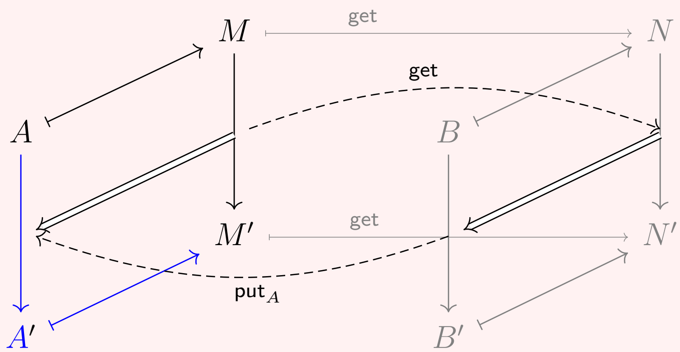
$$\mathsf{control}(\mathsf{apply}_A v) \quad = \quad v \qquad\qquad (\textsc{Over})$$

It is *universal* with this property: For any other $\mathcal{A}$-arrow $u : A \to A'$, each factorisation of $\mathsf{control}\,u$ through v uniquely determines a factorisation of u through $\mathsf{apply}_A v$.

Definition 5: Edit Space Lens

Let $\mathsf{control}_1 : \mathcal{A} \to \mathcal{M}$ and $\mathsf{control}_2 : \mathcal{B} \to \mathcal{N}$ be edit spaces —with liftings apply_1 and apply_2, respectively. Define an **edit δ-lens** to be a functor $\mathsf{get} : \mathcal{M} \to \mathcal{N}$ with a family of *functions* $\mathsf{put}_A : \mathsf{Hom}_{\mathcal{A}}(A, -) \leftarrow \mathsf{Hom}_{\mathcal{B}}\big((\mathsf{control}_1 \fatsemi \mathsf{get} \fatsemi \mathsf{apply}_2)\,A, -\big)$ satisfying (Stability), (EditSectional), and the following commutativity condition; i.e., $\mathsf{apply}_1 = \mathsf{get} \fatsemi \mathsf{apply}_2 \fatsemi \mathsf{put}$.

$$\mathsf{put} \fatsemi \mathsf{control}_1 \fatsemi \mathsf{get} \quad = \quad \mathsf{control}_2 \qquad\qquad (\textsc{EditSectional})$$



Theorem 1: Edit Delta Lenses form a subcategory "$\varepsilon\delta$-Lens" of δ-Lens

Opfibrations are known to form a category; details can be found in [Jac99].

Now to the main theorem.

Given a functor $F : \mathcal{A} \to \mathcal{B}$, define the category $\mathcal{A}/\mathcal{B}$ to have objects the pairs $(A : \mathsf{Obj}\,\mathcal{A}, B : \mathsf{Obj}\,)$ with $F\,A = B$, and to have arrows $(A, B) \to (A', B')$ to be pairs $(u : A \to_{\mathcal{A}} A', v : B \to_{\mathcal{B}} B')$ such that the resulting square commutes; i.e., $F\,A = B, F\,A' = B'$, and $F\,u = v$. Composition and identities are inherited from $\mathcal{A}$ and $\mathcal{B}$, and the required conditions are met due to the functoriality of F.

> **Theorem 2: $\varepsilon\delta$-Lenses are δ-Lenses between certain categories**
>
> More precisely: If get is an $\varepsilon\delta$-lens from $\mathsf{control}_1 : \mathcal{A} \to \mathcal{M}$ to $\mathsf{control}_2 : \mathcal{B} \to \mathcal{N}$, then get induces a δ-lens from $\mathcal{A}/\mathcal{M}$ to $\mathcal{B}/\mathcal{N}$.

Proof. Define the functor $\mathsf{get}' : \mathcal{A}/\mathcal{M} \longrightarrow \mathcal{B}/\mathcal{M}$ and family put' by the following equations:

$$\begin{aligned} \mathsf{get}'(x, y) &= (\mathsf{apply}_2\,\mathsf{get}\,\mathsf{control}_1\,x,\ \mathsf{get}\,y) \\ \mathsf{put}'_{A,M}(x, y) &= (\mathsf{put}_A\,x,\ \mathsf{control}_1\,\mathsf{put}_A\,\mathsf{apply}_2\,y) \end{aligned}$$

Functoriality of get' is inherited from get. It remains to show that this has put' as a stable sectional lifting.

- ◇ Stability follows from the stability of put_A.
- ◇ Sectional:

$$\begin{aligned} & \mathsf{get}'(\mathsf{put}'(x, y)) \\ = & \quad \{\ \text{Definition of } \mathsf{put}'\ \} \\ & \mathsf{get}'(\mathsf{put}_A\,x,\ \mathsf{control}_1\,\mathsf{put}_A\,\mathsf{apply}_2\,y) \\ = & \quad \{\ \text{Definition of } \mathsf{get}'\ \} \\ & ((\mathsf{put} \fatsemi \mathsf{control}_1 \fatsemi \mathsf{get} \fatsemi \mathsf{apply}_2)\,x,\ (\mathsf{apply}_2 \fatsemi \mathsf{put} \fatsemi \mathsf{control}_1 \fatsemi \mathsf{get})\,y) \\ = & \quad \{\ (\textsc{EditSectional})\ \text{axiom, twice}\ \} \\ & ((\mathsf{control}_2 \fatsemi \mathsf{apply}_2)\,x,\ (\mathsf{apply}_2 \fatsemi \mathsf{control}_2)\,y) \\ = & \quad \{\ (\textsc{Over})\ \text{laws}\ \} \\ & ((\mathsf{control}_2 \fatsemi \mathsf{apply}_2)\,x,\ y) \\ = & \quad \{\ ???\ \} \\ & (x, y) \end{aligned}$$

References

[Jac99] B. Jacobs. *Categorical Logic and Type Theory.* Studies in Logic and the Foundations of Mathematics 141. Amsterdam: North Holland, 1999 (cit. on p. 4).