

Symmetric delta lenses and spans of asymmetric delta lenses

Michael Johnson^a Robert Rosebrugh^b

- a. CoACT, Departments of Mathematics and Computing,
Macquarie University
- b. Department of Mathematics and Computer Science
Mount Allison University

Abstract Bidirectional Transformations provide mechanisms for maintaining synchronization between updatable data sources. Lenses are certain mathematically specified bidirectional transformations. As part of a project to unify the treatment of symmetric lenses (of various kinds) as equivalence classes of spans of asymmetric lenses (of corresponding kinds), we relate symmetric delta lenses with spans of asymmetric delta lenses. Because delta lenses are based on state spaces which are categories rather than sets, there is further structure that needs to be accounted for. One of the main findings in this paper is that the required equivalence relation among spans is compatible with, but coarser than, the one expected. The main result is an isomorphism of categories between a category whose morphisms are equivalence classes of symmetric delta lenses (here called fb-lenses) and the category of spans of delta lenses modulo the new equivalence.

1 Introduction

Synchronization of data among disparate sources has become increasingly important in view of the ubiquity of distributed systems. One approach to this problem is known as Bidirectional Transformations. A bidirectional transformation between two data sources consists of first, a method for determining when data are synchronized, often represented by a relation, and second, further methods for data transformation intended to restore synchronization [18]. Examples include: cache management which requires synchronization of data between caches and main memory, specifications and implementations which need to be kept consistent, and contact lists which need to be synchronized across devices.

Effective use of bidirectional transformations requires precise specification of the consistency relation (the synchronization data) and the consistency restorers. Furthermore, we expect these to satisfy certain basic properties. This amounts to a

mathematical data structure: objects, operations and equations. Examples of bidirectional transformations presented this way are often called lenses.

In some bidirectional transformations there is a master data source and a dependent data source. For instance, the state of a cache can always be derived from main memory, while main memory cannot in general be reconstructed from a cache state. Of course, when the cache state changes (by a write to the cache) the change can be propagated to the main memory (an example of a consistency restoring operation), but the resulting state of the main memory depends on both the new state of the cache and the old state of the main memory. In such bidirectional transformations there is an asymmetric aspect, and the corresponding lenses are called asymmetric.

The lenses that were first introduced [20] were asymmetric in the sense just described. In turn, the lens versions of more general bidirectional transformations are referred to as symmetric lenses.

Notice that the importance of lenses is consistency restoration. When the two sources are consistent there is nothing to be done. When one of the sources changes state consistency needs to be restored, typically by changing the state of the other source. This leads to another distinction among types of lenses. In restoring consistency a lens may use information about the change of state itself, sometimes called the “delta”, or instead it might use only the new state after the change. In the former case the lens is called a delta lens, and in the latter, a set-based lens. (“Set-based” is used because the restoring operation acts from the set of states, paying no attention to the nature of transitions among elements of the set of states.)

In [8], Hofmann et al defined and studied set-based symmetric lenses. Since then, with the study of variants of asymmetric lenses (set-based or otherwise), there has been a need for more definitions of corresponding symmetric variants. This paper is part of a project by the authors to develop a unified theory of symmetric and asymmetric lenses of various kinds. The goal is to make it straightforward to define the symmetric version of any, possibly new, asymmetric lens variant, and conversely. Once an asymmetric lens is defined the unified theory should provide the symmetric version, and vice-versa.

It was already noted in [8] that there were potentially two approaches to defining set-based symmetric lenses. One involved studying various *right* and *left* operations (corresponding to what other authors call *forwards* and *backwards* operations). The other would be based on a pair of asymmetric lenses with a common master source. Such a pair is called a “span” of asymmetric lenses (see the display containing S on page 9). In both approaches an equivalence relation was needed for two reasons: The first need is to ensure that the composition (of symmetric lenses or spans of asymmetric lenses) is associative. The second need is to identify symmetric lenses or spans of asymmetric lenses which are equivalent in their updating actions, although they might differ in “hidden” details such as their *complements* (see [8]) or the common master source. The definition of symmetric lens in [8] is presented in terms of right and left update operations, noting that “in the span presentation there does not seem to be a natural and easy-to-use candidate for ... equivalence”.

In [12] the present authors developed the foundations needed to work with spans of asymmetric lenses of various kinds and proposed an equivalence relation on spans of (well-behaved [20]) set-based asymmetric lenses. This answered the concern noted at the end of the previous paragraph and is a step towards the unified theory. Of course, the work described so far, being entirely set-based, is still far from a unified theory. In this paper we turn to the delta lenses of Diskin et al [2]. We study the

symmetric approach derived from spans of such lenses, and finally we compare it with the symmetric approach to delta lenses (forwards and backwards style) that Diskin et al proposed in [3].

The reader is assumed to be familiar with the most basic notions of category theory as found in, for example, Barr and Wells [1], Mac Lane [17], or other standard texts.

The paper is structured as follows. Section 2 reviews asymmetric delta lenses (based on [2] and referred to here as d-lenses) including several detailed examples and some basic facts about spans of d-lenses. Section 3 introduces symmetric delta lenses (based on [3], and called here fb-lenses after their basic operations called “forwards” and “backwards”). We consider several examples of fb-lenses, define a composite for fb-lenses, and construct an fb-lens from a span of d-lenses and vice versa. As in the work of Hofmann et al, both fb-lenses and spans of d-lenses need to be studied modulo an equivalence relation. Section 4 introduces the two equivalence relations. Section 5 shows that the two equivalence relations do indeed yield two categories whose morphisms are, respectively, (equivalence classes of) fb-lenses and (equivalence classes of) spans of d-lenses. Of course, in order to construct the two categories we also need to show that the equivalence relations we have introduced are congruences. Finally, in Section 6 we show that there is an equivalence between the two categories. Indeed, in this case the equivalence is an isomorphism of categories.

Because of the usefulness of category-based lenses (in particular delta lenses) in applications, the work presented here lays important mathematical foundations. Furthermore the extra mathematical structure provided in the category-based variants has revealed a surprise — an equivalence relation generated by non-trivial lenses is not coarse enough to ensure that two spans of d-lenses with the same fb-behaviour are always identified. The difficulty that arises is illustrated in a short example at the end of the paper which amounts to “twisting” the structures so that no single lens can commute with the lenses on the left side of the spans, and at the same time commute with the lenses on the right side of the spans. The solution, presented as generators of one of the equivalence relations in Section 4, relaxes the requirement that the comparison be itself a lens, and asks that it properly respect the put operations on both sides (rather than having its own put operation commuting with both sides). The resulting requirement is entirely natural.

2 Asymmetric delta lenses

For any category \mathbf{C} , we write $|\mathbf{C}|$ for the set (discrete category) of objects of \mathbf{C} . We write \mathbf{C}^2 for the so-called “arrow category” of \mathbf{C} . An object A of \mathbf{C}^2 is an arrow of \mathbf{C} denoted $A = A_f : A_0 \rightarrow A_1$ (where the subscript f is just a reminder that this is the “function part” of the object A). An arrow in \mathbf{C}^2 from the object A to another object $B = B_f : B_0 \rightarrow B_1$ is a pair of arrows

$$g = (g_0 : A_0 \rightarrow B_0, g_1 : A_1 \rightarrow B_1)$$

satisfying $g_1 A_f = B_f g_0$ so that the following square of arrows commutes:

$$\begin{array}{ccc} A_0 & \xrightarrow{g_0} & B_0 \\ A_f \downarrow & & \downarrow B_f \\ A_1 & \xrightarrow{g_1} & B_1 \end{array}$$

The identity arrow from A to itself is the pair of identity arrows $(\text{id}_{A_0}, \text{id}_{A_1})$ and the composite hg of g with h defined as $h = (h_0, h_1)$ from B to C is just $hg = (h_0g_0, h_1g_1)$.

For a functor $G : \mathbf{S} \rightarrow \mathbf{V}$, we denote by G/\mathbf{V} the so-called ‘‘comma’’ category. The objects of G/\mathbf{V} are formally pairs (S, α) where S is an object of \mathbf{S} and α an arrow of \mathbf{V} whose domain is $G(S)$ and whose codomain is arbitrary, for example $\alpha : G(S) \rightarrow V$. An arrow of G/\mathbf{V} from (S, α) to (S', β) where $\beta : G(S') \rightarrow V'$ is a pair (σ, φ) where $\sigma : S \rightarrow S'$ is an arrow of \mathbf{S} and $\varphi : V \rightarrow V'$ is an arrow of \mathbf{V} and they satisfy $\varphi\alpha = \beta G(\sigma)$, so that the following square of arrows commutes:

$$\begin{array}{ccc} G(S) & \xrightarrow{G(\sigma)} & G(S') \\ \alpha \downarrow & & \downarrow \beta \\ V & \xrightarrow{\varphi} & V' \end{array}$$

The identity arrow from (S, α) to itself is the pair of identity arrows $(\text{id}_S, \text{id}_V)$ and the composite of (σ, φ) with (σ', φ') is defined to be $(\sigma'\sigma, \varphi'\varphi)$.

We now recall the definition of an *asymmetric delta lens* ([2, 11]) which we will usually abbreviate to d-lens.

Definition 1 An asymmetric delta lens (d-lens) from \mathbf{S} to \mathbf{V} is a pair (G, P) where $G : \mathbf{S} \rightarrow \mathbf{V}$ is a functor (the ‘‘Get’’) and $P : |G/\mathbf{V}| \rightarrow |\mathbf{S}^2|$ is a function (the ‘‘Put’’) and the data for $\alpha : G(S) \rightarrow V$ and $\beta : G(S') \rightarrow V'$ satisfy:

- (i) *d-PutInc*: the domain of $P(S, \alpha)$ is S
- (ii) *d-PutId*: $P(S, \text{id}_{G(S)}) = \text{id}_S$
- (iii) *d-PutGet*: $G(P(S, \alpha)) = \alpha$
- (iv) *d-PutPut*: if S' is the codomain of $P(S, \alpha)$ (and hence $G(S') = V$) then $P(S, \beta\alpha) = P(S', \beta)P(S, \alpha)$

The reader will notice that *d-PutId* and *d-PutPut* look like functoriality requirements for P . However, in the definition P is merely a function on the objects of the comma category G/\mathbf{V} , that is, P is not defined on the arrows of that comma category. Nevertheless, because the arrows of the comma category are pairs of arrows, one from \mathbf{S} and one from \mathbf{V} , and because the objects of the comma category are (indexed) arrows of \mathbf{V} , P can be extended to the subcategory of G/\mathbf{V} whose arrows are a pair of an identity arrow from \mathbf{S} and an arrow from \mathbf{V} . Moreover, *d-PutId* and *d-PutPut* make the extension of P functorial on that subcategory. Details are spelled out in [11].

At this point we digress to remark on adjectives commonly used in the lens community. Following [20], lenses satisfying conditions corresponding to *d-PutGet* are called *well-behaved*. A lens satisfying an additional condition analogous to *d-PutPut* is commonly called *very well-behaved*. In this paper all of our asymmetric delta lenses satisfy both of these. We call them simply d-lenses. Further discussion of *PutPut* conditions can be found in [10].

In the rest of this article we are often going to leave out brackets to shorten expressions where the meaning is unambiguous and should be clear. So, for example, the application of the functor G to the object S written $G(S)$ above may be shortened to GS , or $G(P(S, \alpha))$ may be written $GP(S, \alpha)$.

We suggest here some ideas to help orient the reader with the notations used. Both the categories \mathbf{S} and \mathbf{V} represent categories of model states. Objects of \mathbf{S} are model states and an arrow $S \rightarrow S'$ of \mathbf{S} represents a specific transition from the state S which is its domain to the state S' which is its codomain. Such specified transitions are often called “deltas”. Similarly for \mathbf{V} . A functor $G : \mathbf{S} \rightarrow \mathbf{V}$ maps states of \mathbf{S} to states of \mathbf{V} — S is sent to GS . Furthermore, being a functor it maps deltas $S \rightarrow S'$ in \mathbf{S} to deltas $GS \rightarrow GS'$ in \mathbf{V} . An object of G/\mathbf{V} , being a pair $(S, \alpha : GS \rightarrow V)$, encodes both an object of \mathbf{S} and a delta starting from GS . Such a pair is the basic input for a Put operation. The Put operation P itself, in the case of a d-lens, is just a function (not a functor) and it takes such an “anchored delta” $(S, \alpha : GS \rightarrow V)$ in \mathbf{V} to a delta in \mathbf{S} which, by d-PutInc, starts at S . The axioms d-PutId and d-PutPut ensure that the Put operation P respects composition and identities. Finally, the axiom d-PutGet ensures that, as expected, the Put operation on a given delta, say α , results in a delta in \mathbf{S} which is carried by G to α .

For examples of d-lenses, we refer the reader to [2]. We also consider here several examples which we will study further in the context of our constructions below. The third, Example 4, will be considered throughout the paper. The examples are deliberately simple, but non-trivial, and the calculations they embody arise frequently as important aspects of more elaborate real-world examples.

Example 2 We denote by \mathbf{set} the category whose objects are *finite* sets and whose arrows are functions between them. As a category of models, the model states (objects) of \mathbf{set} are each just a single set, which can be thought of as the entity set for a single entity. The arrows are functions which update one state (entity set) to another.

We also consider \mathbf{set}^2 , another category of model states (see the definition of \mathbf{C}^2 at the beginning of this section). Recall that an object (or model state) X of \mathbf{set}^2 is a function $X_f : X_0 \rightarrow X_1$ between (finite) sets X_0 and X_1 . As a model state, the object X has exactly two entity sets, X_0 and X_1 , and a single constraint (perhaps a foreign key) specified by the function X_f . For example, X_0 might be the state of a **Persons** entity, X_1 the state of an **Addresses** entity, and X_f the assignment of a person to an address. An arrow in \mathbf{set}^2 from the object X to another object (model state) Y is a pair of functions $g = (g_0, g_1)$ between corresponding entity sets which are compatible with the respective foreign key or constraint.

We can define two distinct d-lenses from \mathbf{set}^2 to \mathbf{set} as follows.

The first d-lens (G_1, P_1) has as its Get functor, $G_1 : \mathbf{set}^2 \rightarrow \mathbf{set}$, the “codomain” functor which sends an object X with $X_f : X_0 \rightarrow X_1$ of \mathbf{set}^2 to the set $G_1(X) = X_1$ and the arrow $g = (g_0, g_1)$ of \mathbf{set}^2 to the function g_1 . It is easy to verify that G_1 is a functor.

The first Put, P_1 is defined as follows. Consider any finite set Z_1 and any function from $G_1(X) = X_1$ to Z_1 , say $\alpha : X_1 \rightarrow Z_1$. We define $P_1(X, \alpha)$ to be the arrow in (\mathbf{set}^2) whose codomain is the model with function Z_f defined by $Z_f = \alpha X_f : Z_0 = X_0 \rightarrow Z_1$. Then $P_1(X, \alpha)$ from X to Z is defined by the pair of arrows $P_1(X_f, \alpha) = (\text{id}_{X_0}, \alpha)$. Clearly $\alpha X_f = \alpha X_f \text{id}_{X_0}$ so $P_1(X, \alpha)$ is an arrow of \mathbf{set}^2 . The idea of the Put is the following. The function α indicates an update of the entity set X_1 , but we have no information about an update of the entity set X_0 , so we just leave it alone and make the foreign key for the new object Z just the composite of X_f and α . Moreover, this systematic choice guarantees that the d-PutPut axiom will be satisfied.

The other requirements for a d-lens are easily checked for the pair (G_1, P_1) .

The second d-lens (G_0, P_0) has as its Get functor, $G_0 : \mathbf{set}^2 \rightarrow \mathbf{set}$, the “domain”

functor which sends an object X with $X_f : X_0 \rightarrow X_1$ of \mathbf{set}^2 to the set $G_0(X) = X_0$ and the arrow $g = (g_0, g_1)$ of \mathbf{set}^2 to the function g_0 .

The second Put, P_0 has a slightly more complicated definition (and this is why we introduce P_0 after P_1). We begin with any set Z_0 and any function from $G_0(X) = X_0$ to Z_0 in \mathbf{set} , say $\alpha : X_0 \rightarrow Z_0$. The codomain of $P_0(X, \alpha)$ has to be an object of \mathbf{set}^2 whose function has the domain Z_0 . We define Z to be the object whose function is the right hand vertical arrow in the \mathbf{set} expanded pushout of X_f along α :

$$\begin{array}{ccc} X_0 & \xrightarrow{\alpha} & Z_0 \\ X_f \downarrow & & \downarrow Z_f \\ X_1 & \xrightarrow{\alpha'} & Z_1 \end{array}$$

Recall that a *pushout* is also called an “amalgamated sum” and Z_1 is found by first taking the disjoint union of X_1 and Z_0 and then identifying pairs of elements in the two summands that are images of the same element of X_0 under X_f and α . Now we can define $P_0(X, \alpha)$ to be the arrow in (\mathbf{set}^2) from X to Z defined by the pair of functions $P_0(X, \alpha) = (\alpha, \alpha')$. Here the function α gives an update of the entity set X_0 . To define the codomain Z of $P_0(X, \alpha)$ we needed to give a compatible update of X_1 . In principle there are many ways to do this, but taking the pushout is a systematic choice which also guarantees that the axiom d-PutPut will be satisfied.

Again, the other requirements to make (G_0, P_0) a d-lens are easily checked.

Example 3 In our next example of an asymmetric d-lens, \mathbf{set}^2 is the *codomain* of the Get while the domain of the Get is a related category denoted $\mathbf{set}^{\mathbf{cs}}$ (**cs** for *cospan*). The objects of $\mathbf{set}^{\mathbf{cs}}$ can be thought of as model states with three entity sets and two assignments. For example, the entity sets might be **Taxis**, **Persons**, and **Suburbs**. The assignments from **Taxis** to **Suburbs**, and **Persons** to **Suburbs**, give the current locations of **Taxis** and **Persons** in what might be a real-time taxi booking system. Formally an object (model state) X of $\mathbf{set}^{\mathbf{cs}}$ is determined by three sets denoted X_0 , X_1 and X_2 and two functions $X_f : X_0 \rightarrow X_1$ and $X_g : X_2 \rightarrow X_1$. There are no other conditions.

An arrow in $\mathbf{set}^{\mathbf{cs}}$ from the object X to another object (model state) $Y = (Y_0, Y_1, Y_2, Y_f, Y_g)$ is defined to be a triple of functions $h = (h_0, h_1, h_2)$ between corresponding entity sets and which are required to be compatible with the respective arrows. Thus, we require both $h_1 X_f = Y_f h_0$ and $h_1 X_g = Y_g h_2$. That is, the following is a commutative diagram:

$$\begin{array}{ccc} X_0 & \xrightarrow{h_0} & Y_0 \\ X_f \downarrow & & \downarrow Y_f \\ X_1 & \xrightarrow{h_1} & Y_1 \\ X_g \uparrow & & \uparrow Y_g \\ X_2 & \xrightarrow{h_2} & Y_2 \end{array}$$

The d-lens (G_f, P_f) from $\mathbf{set}^{\mathbf{cs}}$ to \mathbf{set}^2 is defined simply as follows. For an object X of $\mathbf{set}^{\mathbf{cs}}$ as above, $G_f(X)$ is the object of \mathbf{set}^2 whose function is X_f . For an arrow h of $\mathbf{set}^{\mathbf{cs}}$ as above, $G_f(h)$ is the arrow of \mathbf{set}^2 defined by the pair (h_0, h_1) . We next define P_f . Suppose that Z is an object of \mathbf{set}^2 given by $Z_f : Z_0 \rightarrow Z_1$, and $h = (h_0, h_1)$ is an arrow from $G_f(X)$ to Z in \mathbf{set}^2 . We define $P_f(X, h)$ to have codomain $Z' = (Z_0, Z_1, X_2, Z_f, h_1 X_g)$ and the $\mathbf{set}^{\mathbf{cs}}$ arrow from X to Z' is $h' = (h_0, h_1, \text{id}_{X_2})$.

It is easy to see that $P_f(X, h)$ is an arrow of $\mathbf{set}^{\mathbf{cs}}$ and that (G_f, P_f) satisfies the required equations. More interesting is to consider the interpretation of the d-lens in terms of model states for Taxis (X_0), Suburbs (X_1), and Persons (X_2). Indeed, G_f is just the “view” that shows only the taxis and suburbs information — it forgets the Persons and Persons to Suburbs information. If the Taxis to Suburbs information in $G_f(X)$ is updated to the \mathbf{set}^2 state Z by h , how should we best update the Persons to Suburbs information in the $\mathbf{set}^{\mathbf{cs}}$ model state? An answer that is sensible and satisfying is not to change the Persons at all (so the Persons update is the identity on X_2), and define Z'_g , as we did, to be the composite $h_1 X_g$. We may have added some new taxis or extended our range by adding some new suburbs, or even “merged” some suburbs in changing the scale of resolution, but the persons are the same and they are in the same (possibly merged) locations.

We note in passing that the functor $G_{dg} : \mathbf{set}^{\mathbf{cs}} \rightarrow \mathbf{set}$ defined on objects by $G_{dg}(X) = X_2$ and in the obvious way on arrows is a candidate for a Get. The reader might wish to contemplate how to define a suitable Put P_{dg} . We will return to this question below.

Example 4 Our final two examples of asymmetric d-lenses, have \mathbf{set}^2 as codomain of one Get and a new category $\mathbf{set}^{\mathbf{sp}}$ (that we define below) as codomain of the other Get. In both cases the domain of the Gets is a category related to $\mathbf{set}^{\mathbf{cs}}$ above, and it is denoted $\mathbf{set}^{\mathbf{scp}}$ (**scp** for *span, composable pair*). The objects of $\mathbf{set}^{\mathbf{scp}}$ can be thought of as model states with four entity sets and three assignments. For example, the entity sets might be Students, Registrations, Courses (offerings) and Instructors. Two assignments are from Registrations to respectively Students and Courses, and the last is from Courses to Instructors. The first two functions record registration of a student in a course offering, and the last gives the instructor for a course offering. Formally an object (model state) X of $\mathbf{set}^{\mathbf{scp}}$ is determined by four sets denoted X_0, X_1, X_2 and X_3 and three functions $X_f : X_1 \rightarrow X_0, X_g : X_1 \rightarrow X_2$ and $X_h : X_2 \rightarrow X_3$. There are no other conditions. The main difference from the previous examples is the direction of the functions, and that changes the lenses we construct.

An arrow of $\mathbf{set}^{\mathbf{scp}}$ from the object X to $Y = (Y_0, Y_1, Y_2, Y_3, Y_f, Y_g, Y_h)$ is a four-tuple $k = (k_0, k_1, k_2, k_3)$ of functions between corresponding entity sets, compatible with the respective assignments. Thus we require $k_0 X_f = Y_f k_1, k_2 X_g = Y_g k_1$ and $k_3 X_h = Y_h k_2$, meaning the following is a commutative diagram:

$$\begin{array}{ccc}
 X_0 & \xrightarrow{k_0} & Y_0 \\
 X_f \uparrow & & \uparrow Y_f \\
 X_1 & \xrightarrow{k_1} & Y_1 \\
 X_g \downarrow & & \downarrow Y_g \\
 X_2 & \xrightarrow{k_2} & Y_2 \\
 X_h \downarrow & & \downarrow Y_h \\
 X_3 & \xrightarrow{k_3} & Y_3
 \end{array}$$

The first d-lens (G_r, P_r) from $\mathbf{set}^{\mathbf{scp}}$ to \mathbf{set}^2 is quite simple. For an object X of $\mathbf{set}^{\mathbf{scp}}$ as above, we define $G_r(X)$ to be the \mathbf{set}^2 object whose function is X_h . So the Get remembers only the instructor assignments. For an arrow k of $\mathbf{set}^{\mathbf{scp}}$ as above, we define $G_r(k)$ to be the arrow of \mathbf{set}^2 defined by the pair (k_2, k_3) . We also need to define P_r . To interpret P_r , suppose that W is a new assignment of

instructors (as an object of \mathbf{set}^2) and $k : G_r(X) \rightarrow W$ is an update from the current assignment (which is $G_r(X)$) to W , for example an insertion of new course offerings and their instructor assignments. There should be no change to students or their course registrations, and that is exactly the effect of $P_r(X, k)$: it incorporates the update k , but leaves the **Students** and **Registrations** entity sets alone. Formally, if W is given by $W_h : W_2 \rightarrow W_3$ (note the indices), and $k = (k_2, k_3)$ is an arrow from $G_r(X)$ to W in \mathbf{set}^2 then $P_r(X, k)$ has codomain $W' = (X_0, X_1, W_2, W_3, X_f, k_2 X_g, W_h)$. The arrow $P_r(X, h)$ is $k' = (\text{id}_{X_0}, \text{id}_{X_1}, k_2, k_3)$. With this in mind, it is easy to check that (G_r, P_r) satisfies the equations for an asymmetric d-lens.

Before describing the second d-lens (G_l, P_l) we need to describe its codomain $\mathbf{set}^{\mathbf{sp}}$ (sp for span). The objects and arrows have data with the same names as for $\mathbf{set}^{\mathbf{cs}}$, but for $X = (X_0, X_1, X_2, X_f, X_g)$ we orient X_f and X_g in opposite directions (so $X_f : X_1 \rightarrow X_0$). An arrow $h = (h_0, h_1, h_2)$ from X to Y satisfies the equations making the squares commute in

$$\begin{array}{ccc}
 X_0 & \xrightarrow{h_0} & Y_0 \\
 X_f \uparrow & & \uparrow Y_f \\
 X_1 & \xrightarrow{h_1} & Y_1 \\
 X_g \downarrow & & \downarrow Y_g \\
 X_2 & \xrightarrow{h_2} & Y_2
 \end{array}$$

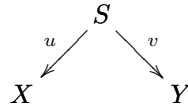
The d-lens (G_l, P_l) is slightly more complicated and we present it formally first. For an object X of $\mathbf{set}^{\mathbf{scp}}$, $G_l(X)$ is the object $(X_0, X_1, X_2, X_f, X_g)$ of $\mathbf{set}^{\mathbf{sp}}$. For an arrow k of $\mathbf{set}^{\mathbf{scp}}$, $G_l(k)$ is the $\mathbf{set}^{\mathbf{sp}}$ arrow defined by the triple (k_0, k_1, k_2) . We interpret an arrow $k = (k_0, k_1, k_2) : G_l(X) \rightarrow Z$ as an update of students, registrations and course offerings (for example by inserting some of each) without specifying new instructors. We need P_l . Suppose that Z is an object of $\mathbf{set}^{\mathbf{sp}}$ given by $(Z_0, Z_1, Z_2, Z_f, Z_g)$, and $k = (k_0, k_1, k_2)$ is an arrow from $G_l(X)$ to Z in $\mathbf{set}^{\mathbf{sp}}$. We define $P_l(X, k)$ to have codomain $Z' = (Z_0, Z_1, Z_2, Z'_3, Z_f, Z_g, Z'_h)$, an object of $\mathbf{set}^{\mathbf{scp}}$, where the following square is a pushout in \mathbf{set} :

$$\begin{array}{ccc}
 X_2 & \xrightarrow{k_2} & Z_2 \\
 X_h \downarrow & & \downarrow Z'_h \\
 X_3 & \xrightarrow{k'_3} & Z'_3
 \end{array}$$

The $\mathbf{set}^{\mathbf{scp}}$ arrow $P_l(X, k)$ from X to Z' is $k' = (k_0, k_1, k_2, k'_3)$. The interpretation of $P_l(X, k)$ is now straightforward. The effect is to freely add undetermined new instructors for any new courses (via Z'_h) and to update the instructors entity set accordingly (via k'_3). This corresponds to the common practice in many universities of assigning courses to “InstructorA”, “InstructorB”, etc, before the corresponding actual instructors are hired (or indeed even determined).

After that extended treatment of examples we now return to the theory of d-lenses. In [12] we studied spans of asymmetric lenses (two lenses whose Gets have a common domain) and we developed the technique to compose spans of lenses in general. For d-lenses, this specializes to Definition 6 below. We first need a small but important proposition, and we remind the reader about notations for spans and cospans.

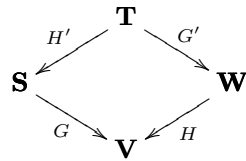
A *span* is a pair of morphisms, with common domain:



Despite the symmetry, such a span is often described as a “span from X to Y ”, and is distinguished from the same two arrows viewed as a span from Y to X . The illustrated span above is often denoted for brevity’s sake $u : X \leftarrow S \rightarrow Y : v$ and, when X , S and Y are understood or easily derived, we sometimes just refer to it as the span u, v . The object S is sometimes called the *head* or *peak* of the span and the arrows u and v are called the *legs* of the span. The objects X and Y are, naturally enough, called the *feet* of the span. Cospans are described and notated in the same way but the arrows u and v are reversed. Finally, if, as sometimes is necessary, a span is drawn upside down, the common domain is still called the head despite being drawn below the feet.

When working with spans, for example to compose them, it is often necessary to calculate a pullback of a cospan. If $u : X \rightarrow Z \leftarrow Y : v$ is a cospan, a pullback of the cospan is the span $v' : X \leftarrow X \times_Z Y \rightarrow Y : u'$. The notation $X \times_Z Y$ is used because the pullback of the cospan of functions in **set**, $u : X \rightarrow Z \leftarrow Y : v$, can always be taken to have as head the set whose elements are the pairs (x, y) of elements from $X \times Y$ satisfying $u(x) = v(y)$. The functions v', u' project (x, y) to x and y , respectively. Similarly, when X, Y, Z are categories and u, v are functors, we will usually assume that a pullback of categories and functors (that is, a pullback in the category **cat** of categories and functors) has been chosen — as can always be done — so that its objects and morphisms are pairs of objects and pairs of morphisms from the categories from which it has been constructed. Thus, in the diagram below, objects of **T** will be pairs of objects (S, W) from the categories **S** and **W** respectively, with the property that $G(S) = H(W)$, and similarly for morphisms of **T**.

Proposition 5 *Let $G : \mathbf{S} \rightarrow \mathbf{V} \leftarrow \mathbf{W} : H$ be a cospan of functors. Suppose that $P : |G/\mathbf{V}| \rightarrow |\mathbf{S}^2|$ is a function which, with G , makes the pair (G, P) a d -lens. Then, in the pullback square in **cat**:*



the functor G' together with $P' : |G'/\mathbf{W}| \rightarrow |\mathbf{T}^2|$ defined by

$$P'((S, W), \beta : G'(S, W) \rightarrow W') = (P(S, H(\beta)), \beta) : (S, W) \rightarrow (S', W')$$

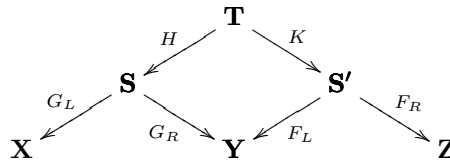
define a d -lens from **T** to **W**.

Proof. Note that P' makes sense since $H(\beta)$ is a morphism $HG'(S, W) \rightarrow H(W')$ but $HG'(S, W) = GH'(S, W) = GS$ so it is in fact a morphism $GS \rightarrow H(W')$. Furthermore we denote the codomain of $P(S, H(\beta))$ by S' so that $G(S') = H(W')$ and thus (S', W') is an object of **T**. The d -PutInc, d -PutId and d -PutGet conditions

on (G', P') are satisfied by construction. The d-PutPut condition follows immediately from d-PutPut for (G, P) . ■

This means we can talk about the “pullback” of a d-lens along an arbitrary functor, in particular along the Get of another d-lens. This is similar to the situations described in [12]. The inverted commas around “pullback” are deliberate because the constructed d-lens may not be an actual pullback in the category of d-lenses (the category whose objects are categories and whose arrows $\mathbf{S} \rightarrow \mathbf{V}$ are d-lenses from \mathbf{S} to \mathbf{V}).

Definition 6 *Suppose that in*



the functors G_L, G_R, F_L , and F_R are the Gets of d-lenses with corresponding Puts P_L, P_R, Q_L , and Q_R , and \mathbf{T} is the pullback of G_R and F_L . For the “pullback” d-lenses with Gets H and K , denote the Puts by P_H and P_K . Then the span composite of the span of d-lenses $(G_L, P_L), (G_R, P_R)$ from \mathbf{X} to \mathbf{Y} with the span of d-lenses $(F_L, Q_L), (F_R, Q_R)$ from \mathbf{Y} to \mathbf{Z} , denoted

$$((G_L, P_L), (G_R, P_R)) \circ ((F_L, Q_L), (F_R, Q_R))$$

is the span of d-lenses from \mathbf{X} to \mathbf{Z} specified as follows. The Gets are $G_L H$ and $F_R K$. The Puts are those for the composite d-lenses $(G_L, P_L)(H, P_H)$ and $(F_R, Q_R)(K, P_K)$.

In a sense, the composite just defined corresponds to the ordinary composite of spans in a category with pullbacks. In the category of categories, the ordinary span composition of the span G_L, G_R and with the span F_L, F_R is the span $G_L H, F_R K$. As usual for such composites, the operation is not associative without introducing an equivalence relation, which we do later in this paper.

3 Symmetric delta lenses

A symmetric delta lens (called an “fb-lens” below) is between categories, say \mathbf{X} and \mathbf{Y} . It consists of a set of synchronizing “corrs” (for “correspondences”), so named because they make explicit intended correspondences between objects of \mathbf{X} and objects of \mathbf{Y} , together with forward and backward “propagation” operations. In the forward direction, given objects X and Y of \mathbf{X} and \mathbf{Y} that are synchronized by a corr r and an arrow x in \mathbf{X} with domain X , the propagation returns an arrow y with domain Y and a corr witnessing the synchronization of the codomains of x and y . This is made precise in the following definition and is based on definitions in [3] and [4]. We denote the domain and codomain of an arrow x by $d_0(x)$ and $d_1(x)$ respectively.

Definition 7 *Let \mathbf{X} and \mathbf{Y} be categories. An fb-lens from \mathbf{X} to \mathbf{Y} is given by a 4-tuple $M = (\delta_{\mathbf{X}}, \delta_{\mathbf{Y}}, f, b) : \mathbf{X} \longleftrightarrow \mathbf{Y}$ specified as follows. The data $\delta_{\mathbf{X}}, \delta_{\mathbf{Y}}$ are functions which come equipped with a common domain $R_{\mathbf{X}\mathbf{Y}}$ and form a span of sets*

$$\delta_{\mathbf{X}} : |\mathbf{X}| \leftarrow R_{\mathbf{X}\mathbf{Y}} \rightarrow |\mathbf{Y}| : \delta_{\mathbf{Y}}$$

An element r of $R_{\mathbf{X}\mathbf{Y}}$ is called a corr. For r in $R_{\mathbf{X}\mathbf{Y}}$, if $\delta_{\mathbf{X}}(r) = X, \delta_{\mathbf{Y}}(r) = Y$ the corr is denoted $r : X \leftrightarrow Y$. The data f and b are operations called forward and backward propagation:

$$f : \text{Arr}(\mathbf{X}) \times_{|\mathbf{X}|} R_{\mathbf{X}\mathbf{Y}} \longrightarrow \text{Arr}(\mathbf{Y}) \times_{|\mathbf{Y}|} R_{\mathbf{X}\mathbf{Y}}$$

$$b : \text{Arr}(\mathbf{Y}) \times_{|\mathbf{Y}|} R_{\mathbf{X}\mathbf{Y}} \longrightarrow \text{Arr}(\mathbf{X}) \times_{|\mathbf{X}|} R_{\mathbf{X}\mathbf{Y}}$$

where the pullbacks ensure that if $f(x, r) = (y, r')$, we have $d_0(x) = \delta_{\mathbf{X}}(r)$, $d_1(y) = \delta_{\mathbf{Y}}(r')$ and similarly for b . We also require that $d_0(y) = \delta_{\mathbf{Y}}(r)$ and $\delta_{\mathbf{X}}(r') = d_1(x)$, and the similar equations for b .

Furthermore, we require that both propagations respect both the identities and composition in \mathbf{X} and \mathbf{Y} , so that we have:

$$r : X \leftrightarrow Y \text{ implies } f(\text{id}_X, r) = (\text{id}_Y, r) \text{ and } b(\text{id}_Y, r) = (\text{id}_X, r)$$

and

$$f(x, r) = (y, r') \text{ and } f(x', r') = (y', r'') \text{ imply } f(x'x, r) = (y'y, r'')$$

and

$$b(y, r) = (x, r') \text{ and } b(y', r') = (x', r'') \text{ imply } b(y'y, r) = (x'x, r'')$$

If $f(x, r) = (y, r')$ and $b(y', r') = (x', r'')$, we display instances of the propagation operations as:

$$\begin{array}{ccc} X & \xleftarrow{r} & Y \\ x \downarrow & \xrightarrow{f} & \downarrow y \\ X' & \xleftarrow{r'} & Y' \end{array} \quad \begin{array}{ccc} X & \xleftarrow{r} & Y \\ x' \downarrow & \xrightarrow{b} & \downarrow y' \\ X' & \xleftarrow{r''} & Y' \end{array}$$

Examples of fb-lenses may also be found in [3], and here we specify some examples which we will study further below.

Example 8 For our first example of an fb-lens M we take $\mathbf{X} = \text{set}^2$ and $\mathbf{Y} = \text{set}$. An interpretation that may help the reader is related to Example 3. We think of an object X of \mathbf{X} as a model state whose function associates elements of the entity set of Taxis to their location Suburbs. An object of \mathbf{Y} is just an entity set of Persons. In this example we take a synchronizing corr r between an \mathbf{X} state X and a \mathbf{Y} state Y to be a function $r_g : Y \longrightarrow X_1$ assigning Persons (or Y) to the Suburbs entity set of X (which is X_1). (The reason for the subscript g will become apparent later.) We take $R_{\mathbf{X}\mathbf{Y}}$ to be the set of all such corrs and define $\delta_{\mathbf{X}}(r) = X$ and $\delta_{\mathbf{Y}}(r) = Y$ when $r : Y \longrightarrow X_1$.

Next we specify forward and backward propagations f and b . The forward propagation is easier to define and interpret. Suppose that X is an \mathbf{X} state specified by $X_f : X_0 \longrightarrow X_1$, the set Y is a \mathbf{Y} state and there is a corr r specified by $r_g : Y \longrightarrow X_1$. Let $h = (h_0, h_1)$ be an (update) arrow from X to Z in \mathbf{X} . We formally define $f(h, r) = (\text{id}_Y, h_1 r_g)$ as in the diagram below:

$$\begin{array}{ccccc} & & X_0 & & \\ & & \downarrow X_f & & \\ Z_0 & \xleftarrow{h_0} & X_1 & \xleftarrow{r_g} & Y \\ Z_f \downarrow & \xleftarrow{h_1} & X_1 & \xleftarrow{\text{id}_Y} & Y \\ & & Z_1 & \xleftarrow{h_1 r_g} & Y \end{array}$$

That is, if we update the \mathbf{X} state, revising **Suburbs** via h_1 , then we leave the **Persons** alone (id_Y), but revise the **corr** (their locations) by composing r_g with h_1 — composing the persons’ locations with the suburbs’ revisions (insertions or merges).

Now we consider the backward propagation. Suppose as before that our \mathbf{Y} state Y is synchronized with an \mathbf{X} state X by the **corr** r with $r_g : Y \rightarrow X_1$, and that Y is updated by a function $y : Y \rightarrow Y'$ (so if y is an injection we are just adding some **Persons**). In the diagram following define the square to be a pushout of sets:

$$\begin{array}{ccc} X_1 & \xleftarrow{r_g} & Y \\ x \downarrow & & \downarrow y \\ X'_1 & \xleftarrow{r'_g} & Y' \end{array}$$

We formally define $\mathbf{b}(y, r) = ((\text{id}_{X_0}, x), r'_g)$. Consider:

$$\begin{array}{ccccc} & & X_0 & & \\ & \swarrow \text{id}_{X_0} & \downarrow X_f & & \\ X_0 & & X_1 & \xleftarrow{r_g} & Y \\ x \swarrow X_f & \swarrow x & & \swarrow y & \\ X'_1 & \xleftarrow{r'_g} & Y' & & \end{array}$$

Note that (id_{X_0}, x) is a \mathbf{set}^2 arrow from X to the object X' specified by the function $xX_f : X_0 \rightarrow X'_1$, and r'_g correctly specifies a **corr** r' from Y' to X' .

The interpretation is that when we update the **Persons** via y , we need to update the **corr** to **Persons**, but we do that by assigning the new **Persons** freely to new (unknown) **Suburbs** (the pushout — recall the discussion of “InstructorA”, “InstructorB”, etc, above). The update of the \mathbf{X} state makes no change to **Taxis** but updates **Suburbs** via x , and the \mathbf{X} state simply augments the **Taxis** to **Suburbs** assignment by composing with x . (Of course, in a real taxi booking system registered persons will, at some future time, specify what actual suburb they are in when they request a taxi, but until then their location (**corr**) is their own unique “unknown” suburb.)

Example 9 The next example of an fb-lens we consider has $\mathbf{X} = \mathbf{set}^{\mathbf{SP}}$ and $\mathbf{Y} = \mathbf{set}^2$. An interpretation that may help the reader is related to Example 4. We think of an object X of \mathbf{X} as a model state whose functions associate elements of the entity set **Registrations** to elements of the entity sets of **Students** and **Courses**. An object of \mathbf{Y} is a model state whose function associates elements of the entity set **Courses** to their instructors. A synchronizing **corr** r between an \mathbf{X} state $X = (X_0, X_1, X_2, X_f, X_g)$ and a \mathbf{Y} state $Y = (Y_2, Y_3, Y_h)$ is determined by an identity function r_1 from X_2 to Y_2 . Thus there either is or is not a **corr** from X to Y and if r is such a **corr** then of course $\delta_{\mathbf{X}}(r) = X$ and $\delta_{\mathbf{Y}}(r) = Y$.

We need to specify forward and backward propagations \mathbf{f} and \mathbf{b} . The backward propagation is easier. Suppose that Y is a \mathbf{Y} state with $Y_h : Y_2 \rightarrow Y_3$, X is an \mathbf{X} state with $X = (X_0, X_1, X_2, X_f, X_g)$ and there is a **corr** r specified by an identity function from X_2 to Y_2 (so $X_2 = Y_2$, but we will keep them separately named). Let Z in \mathbf{Y} be given by $Z_f : Z_2 \rightarrow Z_3$ and let $k = (k_2, k_3)$ be an arrow from Y to Z . We also write k_2 for the function $k_2 : Y_2 = X_2 \rightarrow Z_2$. Note that $(\text{id}_{X_0}, \text{id}_{X_1}, k_2)$ is an \mathbf{X} arrow from X to the object X' specified by $X_f : X_0 \leftarrow X_1 \rightarrow Z_2 : k_2 X_g$, so

there is a corr r' from X' to Z . Now we formally define $\mathbf{b}(h, r) = ((\text{id}_{X_0}, \text{id}_{X_1}, k_2), r')$. The interpretation is that an update of an instructor assignment synchronized with registrations assignments can be updated to a new instructor assignment synchronized with registrations assignments having unchanged registrations and students entity sets.

Now consider the forward propagation. Suppose, with notation above, that our \mathbf{X} state X is synchronized with a \mathbf{Y} state Y by r with $X_2 = Y_2$ and that X is updated by $k = (k_0, k_1, k_2)$ from X to $W = (W_0, W_1, W_2, W_f, W_g)$. Writing also $k_2 : Y_2 \rightarrow W_2$, we let the following be a pushout square in set:

$$\begin{array}{ccc} Y_2 & \xrightarrow{k_2} & W_2 \\ Y_h \downarrow & & \downarrow Y'_h \\ Y_3 & \xrightarrow{k'_3} & Y'_3 \end{array}$$

with the right vertical arrow defining Y' , so that there is a corr r' from W to Y' and we can define $\mathbf{f}(h, r) = ((k_2, k'_3), r')$. The interpretation is that when we update students, registrations and course offerings, we must (as in $(G_l.P_l)$ of Example 4) freely construct a new instructor assignment.

It is easy to define a composition of fb-lenses.

Definition 10 Let $M = (\delta_{\mathbf{X}}^R, \delta_{\mathbf{Y}}^R, \mathbf{f}^R, \mathbf{b}^R)$ and $M' = (\delta_{\mathbf{Y}}^S, \delta_{\mathbf{Z}}^S, \mathbf{f}^S, \mathbf{b}^S)$ be two fb-lenses between \mathbf{X} and \mathbf{Y} and between \mathbf{Y} and \mathbf{Z} respectively. We define the composite fb-lens $M'M = (\delta_{\mathbf{X}}, \delta_{\mathbf{Z}}, \mathbf{f}, \mathbf{b})$ as follows. Let $T_{\mathbf{XZ}}$ be the pullback

$$\begin{array}{ccc} & T_{\mathbf{XZ}} & \\ \delta_1 \swarrow & & \searrow \delta_2 \\ R_{\mathbf{XY}} & & S_{\mathbf{YZ}} \\ \delta_{\mathbf{Y}}^R \searrow & & \swarrow \delta_{\mathbf{Y}}^S \\ & \mathbf{Y} & \end{array}$$

Let $\delta_{\mathbf{X}} = \delta_{\mathbf{X}}^R \delta_1 : T_{\mathbf{XZ}} \rightarrow \mathbf{X}$ and $\delta_{\mathbf{Z}} = \delta_{\mathbf{Z}}^S \delta_2$. The operations for $M'M$ are defined as follows. Denote $\mathbf{f}^R(x, r) = (y, r_f)$, $\mathbf{f}^S(y, s) = (z, s_f)$ and $\mathbf{b}^S(z', s') = (y', s_b)$, $\mathbf{b}^R(y', r') = (x', r_b)$. Then

$$\mathbf{f}(x, (r, s)) = (z, (r_f, s_f)) \text{ and } \mathbf{b}(z', (r', s')) = (x', (r_b, s_b))$$

The diagram

$$\begin{array}{ccccc} X & \xleftarrow{r} & Y & \xleftarrow{s} & Z \\ x \downarrow & \xrightarrow{\mathbf{f}^R} & \downarrow y & \xrightarrow{\mathbf{f}^S} & \downarrow z \\ X' & \xleftarrow{r_f} & Y' & \xleftarrow{s_f} & Z' \end{array}$$

shows that the arities are correct for \mathbf{f} in the forward direction. That is, we have

$$\mathbf{f} : \text{Arr}(\mathbf{X}) \times_{|\mathbf{X}|} T_{\mathbf{XZ}} \rightarrow \text{Arr}(\mathbf{Z}) \times_{|\mathbf{Z}|} T_{\mathbf{XZ}}$$

and similarly

$$\mathbf{b} : \text{Arr}(\mathbf{Z}) \times_{|\mathbf{Z}|} T_{\mathbf{XZ}} \rightarrow \text{Arr}(\mathbf{X}) \times_{|\mathbf{X}|} T_{\mathbf{XZ}}$$

It is easy to show that the \mathbf{f} and \mathbf{b} just defined respect composition and identities in \mathbf{X} and \mathbf{Z} and we record:

Proposition 11 *The composite $M'M$ just defined is an fb-lens from \mathbf{X} to \mathbf{Z} .*

We note that because it is defined using a pullback, this construction of the composite of a pair of fb-lenses is not associative, and when we later define a category of fb-lenses the arrows will be equivalence classes of fb-lenses and the composition will then be associative.

Next we define two constructions relating fb-lenses with spans of d-lenses.

For the first construction, we begin with an fb-lens $M = (\delta_{\mathbf{V}}, \delta_{\mathbf{W}}, f, b)$ from \mathbf{V} to \mathbf{W} with

$$\delta_{\mathbf{V}} : |\mathbf{V}| \leftarrow R \rightarrow |\mathbf{W}| : \delta_{\mathbf{W}}$$

Now construct a span of d-lenses $L_M : \mathbf{V} \leftarrow \mathbf{S} \rightarrow \mathbf{W} : K_M$ from \mathbf{V} to \mathbf{W} . The first step is to define the head \mathbf{S} of the span. The set of objects of \mathbf{S} is the set R of corrs of M . The morphisms of \mathbf{S} are defined as follows:

For objects (corrs) r and r' , $\mathbf{S}(r, r') = \{(v, w) \mid d_0v = \delta_{\mathbf{V}}(r), d_1v = \delta_{\mathbf{V}}(r'), d_0w = \delta_{\mathbf{W}}(r), d_1w = \delta_{\mathbf{W}}(r')\}$ (where we write, as usual, $\mathbf{S}(r, r')$ for the set of arrows of \mathbf{S} from r to r'). Thus an arrow may be thought of as a formal square:

$$\begin{array}{ccc} V & \xleftarrow{r} & W \\ v \downarrow & & \downarrow w \\ V' & \xleftarrow{r'} & W' \end{array}$$

Composition is inherited from composition in \mathbf{V} and \mathbf{W} at boundaries, or more precisely, for $(v, w) \in \mathbf{S}(r, r')$ and $(v', w') \in \mathbf{S}(r', r'')$ we define:

$$(v', w')(v, w) = (v'v, w'w)$$

in $\mathbf{S}(r, r'')$. The identities are pairs of identities. It is easy to see that \mathbf{S} is a category.

Next we define the d-lens L_M to be the pair (G_L, P_L) where we define $G_L : \mathbf{S} \rightarrow \mathbf{V}$ on objects by $\delta_{\mathbf{V}}$, and on arrows by projection, that is $G_L(v, w) = v$. The Put for L_M , $P_L : |G_L/\mathbf{V}| \rightarrow |\mathbf{S}^2|$, is defined on objects $(r, v : G_L(r) \rightarrow V')$ of the category G_L/\mathbf{V} by $P_L(r, v) = (v, \pi_0 f(v, r))$ which is indeed an arrow of \mathbf{S} from r to $\pi_1 f(v, r)$. (As is usual practice, we write π_0 and π_1 for the projection from any pair onto its first and second factors respectively.) We define $K_M = (G_K, P_K)$ similarly.

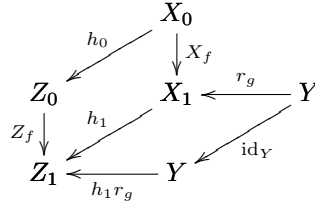
Lemma 12 $L_M = (G_L, P_L)$ and $K_M = (G_K, P_K)$ is a span of d-lenses.

Proof. G_L and G_K are evidently functorial. We need to show that P_L and P_K satisfy (i)-(iv) of Definition 1. These follow immediately from the properties of the fb-lens M . ■

Example 13 We apply the construction of L_M and K_M to Example 8. Recall the fb-lens M from $\mathbf{X} = \mathbf{set}^2$ to $\mathbf{Y} = \mathbf{set}$. An object of the head \mathbf{S} of the constructed span of d-lenses is a corr r from an \mathbf{X} object X to a \mathbf{Y} object Y . Such an r is specified by the object X and a function $r_g : Y \rightarrow X_1$. Thus it is two functions, X_f and r_g with common codomain X_1 . Thus, a corr for M is exactly the same thing as an object of \mathbf{set}^{cs} from Example 3. We exploit this identity of objects of \mathbf{S} and \mathbf{set}^{cs} by writing $r = (X_f, r_g)$. Suppose X' is specified by $X'_0 \rightarrow X'_1$ and r' by $r'_g : Y' \rightarrow X'_1$. Then an arrow of \mathbf{S} from $r = (X_f, r_g)$ to $r' = (X'_f, r'_g)$ is a pair

consisting of an \mathbf{X} arrow $x = (x_0, x_1) : X \rightarrow X'$ in \mathbf{set}^2 , and a \mathbf{Y} arrow $y : Y \rightarrow Y'$ in \mathbf{set} . This makes the definitions of G_L and G_K obvious. For example, on objects we have $G_L(r) = G_L(X_f, r_g) = X$. Notice that we *do not* require that $x_1 r_g = r'_g y$ so \mathbf{S} has “more” arrows than \mathbf{set}^{cs} .

Next we define P_L . Consider an object $(r, h : G_L(r) \rightarrow Z)$ of G_L/\mathbf{X} where Z is specified by $Z_f : Z_0 \rightarrow Z_1$ and $h = (h_0 : X_0 \rightarrow Z_0, h_1 : X_1 \rightarrow Z_1)$. Applying the formula above defines $P_L(r, h)$ to be the arrow (h, id_Y) of \mathbf{S} from r to the object $r' = (Z_f, h_1 r_g)$ of \mathbf{S} . Consider again the first diagram from Example 8



The domain and codomain corrs of $P_L(r, h)$ are (X_f, r_g) and $(Z_f, h_1 r_g)$ respectively, while the arrow between them is (h, id_Y) .

Finally, we define P_K . For an object $(r, y : G_K(r) \rightarrow Y')$ of G_K/\mathbf{Y} , applying the formula to define $P_K(r, y)$ uses the pushout in the definition of $\mathbf{b}(y, r)$ above in Example 8. Thus, $P_K(r, y)$ is the arrow $((\text{id}_{X_0}, x), y)$ of \mathbf{S} from r to $r' = (x X_f, r'_g)$.

Example 14 Returning to Example 9, we recall that there is a corr r with boundaries X and Y if and only if $X_2 = Y_2$, so the objects of the head \mathbf{S} of the constructed span are exactly objects of $\mathbf{set}^{\text{scp}}$. In this case the arrows in \mathbf{S} from r to r' are also the same thing as arrows of $\mathbf{set}^{\text{scp}}$. The two asymmetric lenses we construct are exactly those first seen in Example 4.

For the construction in the other direction, we begin with a span of d-lenses. Let $L = (G_L, P_L)$ where $G_L : \mathbf{S} \rightarrow \mathbf{V}$ and $K = (G_K, P_K)$ where $G_K : \mathbf{S} \rightarrow \mathbf{W}$ be such a span from \mathbf{V} to \mathbf{W} .

Construct the fb-lens $M_{L,K} = (\delta_{\mathbf{V}}, \delta_{\mathbf{W}}, \mathbf{f}, \mathbf{b})$ as follows:

- the corrs are $R_{\mathbf{V}, \mathbf{W}} = |\mathbf{S}|$ with $\delta_{\mathbf{V}} S = G_L S$ and $\delta_{\mathbf{W}} S = G_K S$;
- forward propagation \mathbf{f} for $v : V \rightarrow V'$ and $S : V \leftrightarrow W$ is defined by $\mathbf{f}(v, S) = (w, S')$ where $w = G_K(P_L(S, v))$ and S' is the codomain of $P_L(S, v)$;
- backward propagation \mathbf{b} is defined analogously.

Lemma 15 $M_{L,K}$ is an fb-lens.

Proof. Identity and compositionality for $M_{L,K}$ follow from functoriality of the Gets for L and K and the $d\text{-PutId}$ and $d\text{-PutPut}$ equations in Definition 1. \blacksquare

Example 16 As the reader will have guessed by now, the Put P_{dg} for an asymmetric d-lens (G_{dg}, P_{dg}) where $G_{dg} : \mathbf{set}^{\text{cs}} \rightarrow \mathbf{set}$ involves a pushout. Consider the arrow of

$\mathbf{set}^{\mathbf{cs}}$ below in which the bottom square is a pushout:

$$\begin{array}{ccc}
 X_0 & \xrightarrow{\text{id}_{X_0}} & X_0 \\
 X_f \downarrow & & \downarrow h_1 X_f \\
 X_1 & \xrightarrow{h_1} & Y_1 \\
 X_g \uparrow & & \uparrow Y_g \\
 X_2 & \xrightarrow{h} & Y
 \end{array}$$

We define $P_{dg}(X, h)$ to be the $\mathbf{set}^{\mathbf{cs}}$ arrow in the diagram. That is $P_{dg}(X, h) = (\text{id}_{X_0}, h_1, h)$. It is now easy to check that this makes (G_{dg}, P_{pd}) an asymmetric d-lens, and with (G_f, P_f) we have a span from \mathbf{set}^2 to \mathbf{set} . The forward and backward propagations in the construction applied to this span are exactly those in Example 8.

The two constructions between spans of d-lenses and fb-lenses are closely related. Indeed, one composite of the two constructions is actually the identity.

Proposition 17 *For any fb-lens M , with the notation of the constructions above*

$$M = M_{L_M, K_M}$$

Proof. By inspection, the corrs and δ 's of M_{L_M, K_M} are those of M . Further, it is easy to see that, for example, the forward propagation of M_{L_M, K_M} is identical to that of M . ■

However, the other composite of the constructions above, namely the span of d-lenses $L_{M_{L,K}}, K_{M_{L,K}}$ is not equal to the original span L, K (because the arrows of the original \mathbf{S} have been replaced by the formal squares described above). We have yet to consider the appropriate equivalence for spans of d-lenses, and we do so now. We will see that $L_{M_{L,K}}, K_{M_{L,K}}$ is indeed equivalent to L, K .

4 Two equivalence relations

Our first equivalence relation is on spans of d-lenses from \mathbf{X} to \mathbf{Y} .

Definition 18 *Suppose that in the diagram*

$$\begin{array}{ccccc}
 & & \mathbf{S} & & \\
 & \xleftarrow{(G_L, P_L)} & & \xrightarrow{(G_R, P_R)} & \\
 \mathbf{X} & & \downarrow \Phi & & \mathbf{Y} \\
 & \xleftarrow{(G'_L, P'_L)} & \mathbf{S}' & \xrightarrow{(G'_R, P'_R)} & \\
 & & & &
 \end{array}$$

the top and bottom spans are spans of d-lenses and Φ is a functor. Then Φ is said to satisfy conditions (E) if:

- (1) $G'_L \Phi = G_L$ and $G'_R \Phi = G_R$,
- (2) Φ is surjective on objects, and

(3) whenever $\Phi S = S'$, we have both

$$P'_L(S', G'_L S' \xrightarrow{\alpha} X) = \Phi P_L(S, G_L S \xrightarrow{\alpha} X)$$

and

$$P'_R(S', G'_R S' \xrightarrow{\beta} Y) = \Phi P_R(S, G_R S \xrightarrow{\beta} Y).$$

By (1) any such Φ is a “2-cell” (a morphism of spans) in spans of categories between the two \mathbf{X} to \mathbf{Y} spans G_L, G_R and G'_L, G'_R . Moreover, Φ is required both to be surjective on objects and also to satisfy (3), a condition which expresses a compatibility with the Puts.

Notice that the identity functor on \mathbf{S} satisfies conditions (E), and that the identity functor may be taken as a Φ between the span of asymmetric d-lenses defined in Example 4 and that constructed from the fb-lens described in Example 9.

Example 19 Our considerations in Example 13 provide a less trivial example of a functor satisfying conditions (E). As noted there, $\mathbf{set}^{\mathbf{cs}}$ has the same objects as the category called \mathbf{S} in the example, but fewer arrows. Indeed, because identities and composition in both $\mathbf{set}^{\mathbf{cs}}$ and \mathbf{S} are defined the same way, there is an evidently identity on objects inclusion functor $\Phi : \mathbf{set}^{\mathbf{cs}} \rightarrow \mathbf{S}$. Now $\mathbf{set}^{\mathbf{cs}}$ and \mathbf{S} are the heads of spans of asymmetric d-lenses. Their left legs were called, respectively, (G_f, P_f) and (G_L, P_L) in the examples above, while the right legs were called, respectively, (G_{dg}, P_{dg}) and (G_K, P_K) . Noting that Φ is certainly surjective on objects, we leave to the reader the straightforward verification of the other two conditions (E).

Definition 20 Let \equiv_{S_p} be the equivalence relation on spans of d-lenses from \mathbf{X} to \mathbf{Y} which is generated by functors Φ satisfying (E).

To simplify describing \equiv_{S_p} we now prove some properties of functors satisfying conditions (E).

Lemma 21 A composite of d-lens span morphisms satisfying (E) also satisfies (E).

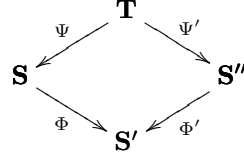
Proof. Consider spans of d-lenses $(G_L, P_L), (G_R, P_R)$ and $(G'_L, P'_L), (G'_R, P'_R)$ as above, and a third such span

$$\mathbf{X} \xleftarrow{(G''_L, P''_L)} \mathbf{S}'' \xrightarrow{(G''_R, P''_R)} \mathbf{Y}.$$

Suppose $\Phi : \mathbf{S} \rightarrow \mathbf{S}'$ and $\Phi' : \mathbf{S}' \rightarrow \mathbf{S}''$ satisfy (E). Properties (1) and (2) for $\Phi' \Phi$ are immediate. We show the P'_L part of property (3) for $\Phi' \Phi$. Suppose $\Phi' \Phi S = S''$ and consider $P''_L(S'', G''_L S'' \xrightarrow{\alpha} X)$. By (E) for Φ and Φ' , since $\Phi'(\Phi(S)) = S''$, we have $P''_L(S'', G''_L S'' \xrightarrow{\alpha} X) = \Phi' P'_L(\Phi(S), G'_L \Phi(S) \xrightarrow{\alpha} X) = \Phi' \Phi P_L(S, G_L S \xrightarrow{\alpha} X)$ as required. ■

Suppose that Φ satisfies (E). When $\Phi S = S'$ it follows that $G_L S = G'_L \Phi S = G'_L S'$, which we will use below. Note that if Φ were the Get of a d-lens (although it need not be) then it would be surjective on arrows by the *d-PutGet* equation, but not necessarily surjective on hom sets.

Lemma 22 *Let $(G_L, P_L), (G_R, P_R), (G'_L, P'_L), (G'_R, P'_R)$ and $(G''_L, P''_L), (G''_R, P''_R)$ be spans of d-lenses as above. Let $\Phi : \mathbf{S} \rightarrow \mathbf{S}' \leftarrow \mathbf{S}'' : \Phi'$ be the functors in a cospan of span morphisms satisfying (E). Let*



be a pullback in **cat**. Then there is a span of d-lenses $\mathbf{X} \xleftarrow{(G'_L, P'_L)} \mathbf{T} \xrightarrow{(G''_R, P''_R)} \mathbf{Y}$ defined by $G'_L = G_L \Psi$ and

$$P'_L((S, S''), G'_L(S, S'')) \xrightarrow{\alpha} X = (S \xrightarrow{P_L(S, \alpha)} W, S'' \xrightarrow{P''_L(S'', \alpha)} W'')$$

and similarly for (G''_R, P''_R) . Moreover, Ψ and Ψ' satisfy (E).

Proof. The first point is that (G'_L, P'_L) and (G''_R, P''_R) actually are d-lenses. We need to know that P'_L is well-defined. Since (S, S'') is an object of the pullback \mathbf{T} , we know that $\Phi(S) = \Phi'(S'') = S'$, say. We want $P'_L((S, S''), G'_L(S, S'')) \xrightarrow{\alpha} X$ to be an arrow of \mathbf{T} , so we need to show that $\Phi(S \xrightarrow{P_L(S, \alpha)} W)$ is equal to $\Phi'(S'' \xrightarrow{P''_L(S'', \alpha)} W'')$. However both are equal to $P'_L(S', \alpha)$ since both Φ and Φ' satisfy (E). Thus furthermore, (W, W'') is an object of \mathbf{T} and using this for *d-PutPut* each of the required d-lens equations is easy to establish.

Next, we show that Ψ and Ψ' satisfy (E). First of all, the Gets commute by definition. Moreover, both Ψ and Ψ' are surjective on objects because Φ and Φ' are so.

It remains to check property (3) for Ψ and Ψ' . We need to show that whenever $\Psi(S, S'') = S$, we have

$$P_L(S, G_L S \xrightarrow{\alpha} U) = \Psi(P'_L((S, S''), G'_L(S, S'')) \xrightarrow{\alpha} U)$$

and this follows immediately from the definitions of Ψ and P'_L . (Notice that for $S = \Psi(S, S'')$ we have $G_L S = G'_L \Phi S = G'_L \Phi'(S'') = G''_L(S'')$ and so $P''_L(S'', G''_L S'' \xrightarrow{\alpha} U)$ is well-defined.) Similarly Ψ' satisfies (3). ■

A “zig-zag” of arrows in a category is any string of arrows in which neighbouring arrows in the string have a common codomain (they are connected head to head) or a common domain (they are connected tail to tail).

Corollary 23 *Zig-zags of span morphisms satisfying (E) can be reduced, using pull-back and composition, to spans of span morphisms satisfying (E).* ■

Thus any proof that two spans of d-lenses are \equiv_{Sp} equivalent can be reduced to a single span Ψ, Ψ' of span morphisms satisfying (E).

The second equivalence relation we introduce is on the set of fb-lenses from \mathbf{X} to \mathbf{Y} . Recall that Diskin et al [4] defined symmetric delta lenses (our fb-lenses),

but they did not consider composing them. Like Hofmann et al [8] they would find that they need to consider equivalence classes of their symmetric delta lenses in order for the appropriate composition to be associative. Also like Hofmann et al, there is a need for an equivalence among their lenses to eliminate artificial differences. In fact, defining an equivalence to restore associativity is easy. Choosing the correct equivalence to eliminate the artificial differences is more delicate. And what do we mean by “artificial differences”? Symmetric lenses of various kinds include hidden data — the complements of Hofmann et al and the corrs of Diskin et al are examples. The hidden data are important for checking and maintaining consistency, but different arrangements of hidden data with the same overall effect should not be counted as different symmetric lenses.

We now introduce such a relation on the set of fb-lenses from \mathbf{X} to \mathbf{Y} .

Definition 24 Let $L = (\delta_{\mathbf{X}}, \delta_{\mathbf{Y}}, f, b)$ and $L' = (\delta'_{\mathbf{X}}, \delta'_{\mathbf{Y}}, f', b')$ be two fb-lenses (from \mathbf{X} to \mathbf{Y}) with corrs $R_{\mathbf{XY}}$ and $R'_{\mathbf{XY}}$. We say $L \equiv_{\text{fb}} L'$ if and only if there is a relation σ from $R_{\mathbf{XY}}$ to $R'_{\mathbf{XY}}$ with the following properties:

1. σ is compatible with the δ 's, i.e. $r\sigma r'$ implies $\delta_{\mathbf{X}}r = \delta'_{\mathbf{X}}r'$ and $\delta_{\mathbf{Y}}r = \delta'_{\mathbf{Y}}r'$
2. σ is total in both directions, i.e. for all r in $R_{\mathbf{XY}}$, there is r' in $R'_{\mathbf{XY}}$ with $r\sigma r'$ and conversely.
3. for all r, r' and for x an arrow of \mathbf{X} , if $r\sigma r'$ and $\delta_{\mathbf{X}}r$ is the domain of x then the first components of $f(x, r)$ and $f'(x, r')$ are equal and the second components are σ related, i.e. $\pi_0 f(x, r) = \pi_0 f'(x, r')$ and $\pi_1 f(x, r)\sigma\pi_1 f'(x, r')$.
4. the corresponding condition for b , i.e. for all r, r' and for y an arrow of \mathbf{Y} , if $r\sigma r'$ and $\delta_{\mathbf{X}}r$ is the domain of x then $\pi_0 b(y, r) = \pi_0 b'(y, r')$ and $\pi_1 b(y, r)\sigma\pi_1 b'(y, r')$.

Lemma 25 The relation \equiv_{fb} is an equivalence relation.

Proof. For reflexivity take σ to be the identity relation; for symmetry take the opposite relation for σ ; for transitivity, the composite relation is easily seen to satisfy conditions 1. to 4. ■

Lemma 26 The equivalence relation \equiv_{fb} is generated by those relations σ from Definition 24 which are surjective functions.

Proof. To see that \equiv_{fb} is generated by such surjections, consider a relation σ witnessing $L \equiv_{\text{fb}} L'$ where $L = (\delta_{\mathbf{X}}, \delta_{\mathbf{Y}}, f, b)$ and $L' = (\delta'_{\mathbf{X}}, \delta'_{\mathbf{Y}}, f', b')$ are fb-lenses (from \mathbf{X} to \mathbf{Y}) with corrs $R_{\mathbf{XY}}$ and $R'_{\mathbf{XY}}$. Let the span tabulating σ be

$$\varphi : R_{\mathbf{XY}} \longleftarrow R_{\sigma} \longrightarrow R'_{\mathbf{XY}} : \varphi'$$

in which φ and φ' are surjective functions since σ is total on both sides. We construct an fb-lens $L_{\sigma} = (\delta_{\mathbf{X}}^{\sigma}, \delta_{\mathbf{Y}}^{\sigma}, f_{\sigma}, b_{\sigma})$ with corrs R_{σ} as follows. The δ 's are composites: $\delta_{\mathbf{X}}^{\sigma} = \delta_{\mathbf{X}}\varphi = \delta'_{\mathbf{X}}\varphi'$, $\delta_{\mathbf{Y}}^{\sigma} = \delta_{\mathbf{Y}}\varphi = \delta'_{\mathbf{Y}}\varphi'$. We next define the operations. First, we define $f_{\sigma}(x, s)$ to have first component $\pi_0 f_{\sigma}(x, s) = \pi_0 f(x, \varphi(s)) = \pi_0 f'(x, \varphi'(s))$. Now $\pi_1 f(x, \varphi(s))$ and $\pi_1 f'(x, \varphi'(s))$ are σ related, so there is an s' in R_{σ} , necessarily unique, such that $\varphi(s') = \pi_1 f(x, \varphi(s))$ and $\varphi'(s') = \pi_1 f'(x, \varphi'(s))$. We define $f_{\sigma}(x, s)$ to have

second component s' . Similarly for b . The operations just defined respect composition and identities in \mathbf{X} and \mathbf{Y} since f and b do.

Notice that the surjective functions φ and φ' now witness $L \equiv_{\text{fb}} L_\sigma \equiv_{\text{fb}} L'$, which completes the proof. \blacksquare

Proposition 27 *Suppose that $M \equiv_{\text{fb}} M'$ are fb-lenses from \mathbf{X} to \mathbf{Y} equivalent by a generator for \equiv_{fb} , i.e. a surjection $\varphi : R_{\mathbf{X}\mathbf{Y}} \rightarrow R'_{\mathbf{X}\mathbf{Y}}$. Then $(L_M, K_M) \equiv_{S_P} (L_{M'}, K_{M'})$ as spans of d-lenses from \mathbf{X} to \mathbf{Y} .*

Proof. We first define $\Phi : \mathbf{S} \rightarrow \mathbf{S}'$ on objects by φ . Notice that Φ is surjective on objects since φ is a surjection. To define Φ on arrows of \mathbf{S} , consider an arrow

$$\begin{array}{ccc} X & \xleftarrow{r} & Y \\ x \downarrow & & \downarrow y \\ X' & \xleftarrow{r'} & Y' \end{array}$$

of \mathbf{S} . Its image under Φ is defined to be the arrow

$$\begin{array}{ccc} X & \xleftarrow{\varphi(r)} & Y \\ x \downarrow & & \downarrow y \\ X' & \xleftarrow{\varphi(r')} & Y' \end{array}$$

which is an arrow of \mathbf{S}' since φ is compatible with the δ s. This Φ is evidently functorial and commutes with the Gets. It remains to show that Φ satisfies condition (3) of (E), that is, whenever $\Phi(r) = r'$, $P'_L(r', G'_L r' \xrightarrow{x} X') = \Phi P_L(r, G_L r \xrightarrow{x} X')$ (with the similar equation holding for P'_K).

Now, when $r' = \Phi(r)$, we have

$$\begin{aligned} P'_L(r', x) &= (x, \pi_0 f'(x, r')) \\ &= (x, \pi_0 f(x, r)) \\ &= \Phi(x, \pi_0 f(x, r)) && \text{definition of } \Phi \\ &= \Phi P_L(r, x) \end{aligned}$$

as required. Similarly for P'_K . \blacksquare

Proposition 28 *Suppose that $(L, K) \equiv_{S_P} (L', K')$ as spans of d-lenses from \mathbf{X} to \mathbf{Y} are made equivalent by a generator for \equiv_{S_P} , i.e. a functor $\Phi : \mathbf{S} \rightarrow \mathbf{S}'$ satisfying conditions (E). Then $M_{L,K} \equiv_{\text{fb}} M_{L',K'}$ as fb-lenses from \mathbf{X} to \mathbf{Y} .*

Proof. Let φ be the object function of Φ . Since Φ commutes with the Gets, φ is compatible with the δ 's. Furthermore, φ is surjective.

We need to show that φ satisfies the remaining two conditions in Definition 24. Suppose $\Phi(S) = S'$ and $G_L(S)$ is the domain of x . Then

$$\begin{aligned}\pi_0 f(x, S) &= G_K P_L(x, S) \\ &= G'_K \Phi P_L(x, S) \\ &= G'_K P'_L(x, S') \\ &= \pi_0 f'(x, S')\end{aligned}$$

and

$$\begin{aligned}\varphi \pi_1 f(x, S) &= \varphi d_1 P_L(x, S) \\ &= d_1 \Phi P_L(x, S) \\ &= d_1 P'_L(x, S') \\ &= \pi_1 f'(x, S').\end{aligned}$$

That establishes the third condition. Similarly for the fourth condition involving \mathbf{b} and \mathbf{b}' . \blacksquare

There is another way to understand conditions (E) which may appeal to some readers, and which confirms that the conditions (E) are not *ad hoc*. We remind the reader that \mathbf{cat}/\mathbf{X} is the “slice” category whose objects are functors with codomain \mathbf{X} and an arrow from $\mathbf{C} \rightarrow \mathbf{X}$ to $\mathbf{C}' \rightarrow \mathbf{X}$ is a functor $\mathbf{C} \rightarrow \mathbf{C}'$ making the evident triangle commute. We recall a result from [11]. We showed there that an (asymmetric) d-lens is the same thing as an algebra (satisfying an additional condition) for a particular semi-monad R_0 on \mathbf{cat}/\mathbf{X} . (Recall that a *semi-monad* is a endofunctor and a “multiplication” natural transformation which together satisfy the associativity condition for a monad. Notice that, unlike a monad, a semi-monad has no specified “unit” natural transformation.)

Viewed as an algebra for the semi-monad R_0 , a d-lens is a pair consisting of a functor part $G : \mathbf{S} \rightarrow \mathbf{X}$ (the object G of \mathbf{cat}/\mathbf{X}) and an algebra structure $P : R_0(G) \rightarrow G$. A span of d-lenses is a pair of algebras whose functor parts have a common domain \mathbf{S} . For two such spans as above, a functor $\Phi : \mathbf{S} \rightarrow \mathbf{S}'$ between their heads satisfying conditions (E) is exactly the same thing as a functor which is simultaneously a surjective on objects algebra homomorphism between the left leg d-lenses and the right leg d-lenses. (We remind the reader that a homomorphism between algebras, say (A, α) and (B, β) for a semi-monad T , or indeed a monad T , on a category \mathbf{C} , is a morphism $f : A \rightarrow B$ such that $\beta T f = f \alpha$.)

Proposition 29 *Suppose that (G_L, P_L) , (G_R, P_R) and (G'_L, P'_L) , (G'_R, P'_R) are spans of d-lenses as in Definition 18. A functor Φ from \mathbf{S} to \mathbf{S}' satisfies conditions (E) if and only if it defines a surjective on objects R_0 semi-monad homomorphism between both (G_L, P_L) and (G'_L, P'_L) , and between (G_R, P_R) and (G'_R, P'_R) .*

Proof. To provide a complete proof we would need to detail the structure of the semi-monad R_0 in [11] and that would take us too far afield. Instead we provide a brief outline.

We begin by noting that the first two of conditions (E) are satisfied if and only if Φ defines a surjective on objects morphism between G_L and G'_L in \mathbf{cat}/\mathbf{X} (respectively between G_R and G'_R in \mathbf{cat}/\mathbf{Y}).

The rest of the proof shows that the R_0 semi-monad algebra structures on G_L and G'_L (respectively on G_R and G'_R) commute with Φ and $R_0(\Phi)$ if and only if Φ satisfies the third of conditions (E). ■

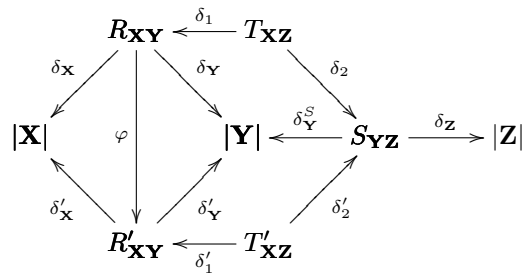
This result could be used to provide different and sometimes shorter proofs of several of the propositions above. For example, Lemma 21 amounts to the easy fact that surjective on objects algebra homomorphisms compose. In other cases, the results in terms of algebra homomorphisms require using more specialized knowledge of categories of algebras. We have restricted ourselves to providing more direct and elementary proofs in terms of conditions (E).

5 Two categories of lenses

The collections of lenses so far discussed do not form categories since their defined composites are not associative. We are going to use the equivalence relations of the previous section to resolve this, but first we show that the equivalence relations respect the composites defined above, that is they are “congruences”.

Proposition 30 *Suppose that $M = (\delta_X, \delta_Y, f^R, b^R)$, $M' = (\delta'_X, \delta'_Y, f^{R'}, b^{R'})$ and $N = (\delta^S_Y, \delta^S_Z, f^S, b^S)$ are fb-lenses (see the diagram below in which R_{XY} , R'_{XY} and S_{YZ} are the corresponding corrs). Further, suppose $\varphi : R_{XY} \rightarrow R'_{XY}$ is a generator of \equiv_{fb} . Thus $M \equiv_{fb} M'$. Then $NM \equiv_{fb} NM'$.*

Proof. The composite NM has as corrs the pullback T_{XZ} as in Definition 10, and similarly NM' has corrs T'_{XZ} .



In order to show that $NM \equiv_{fb} NM'$, we construct $\varphi' : T_{XZ} \rightarrow T'_{XZ}$. This is straightforward using the universal property of the pullback T'_{XZ} , since $\delta'_Y \varphi \delta_1 = \delta^S_Y \delta_2$.

To finish, we need to check that φ' satisfies the four requirements of Definition 24.

Compatibility with δ s is easy when we note that $\varphi \delta_1 = \delta'_1 \varphi'$ and $\delta'_2 \varphi' = \delta_2$.

The function φ' is a total relation in both directions since it is surjective. To see that φ' is surjective, note that any element of T'_{XZ} can be thought of as a pair (r', s) compatible over $|Y|$, and since φ is surjective, there exists an r in R_{XY} , necessarily compatible with s , such that $\varphi'(r, s) = (\varphi(r), s) = (r', s)$.

Let f be the forward propagation of the composite NM , as defined in Definition 10, and let f' be the forward propagation of the composite NM' . Suppose $r' = \varphi(r)$ and thus $(r', s) = \varphi'(r, s)$. We need to show that the first components of $f(x, (r, s))$ and $f'(x, (r', s))$ are equal and that φ' takes the second component of $f(x, (r, s))$ to the second component of $f'(x, (r', s))$.

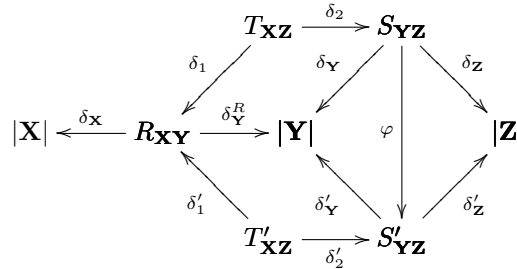
The first component of $f(x, (r, s))$ is $\pi_0 f^S(\pi_0 f^R(x, r), s)$, while the first component of $f'(x, (r', s))$ is $\pi_0 f^S(\pi_0 f^{R'}(x, r'), s)$, and these are equal since φ is a generator of \equiv_{fb} implies $\pi_0 f^R(x, r) = \pi_0 f^{R'}(x, \varphi(r))$.

The second component of $f(x, (r, s))$ is $(\pi_1 f^R(x, r), \pi_1 f^S(\pi_0 f^R(x, r), s))$, while the second component of $f'(x, (r', s))$ is $(\pi_1 f^{R'}(x, r'), \pi_1 f^S(\pi_0 f^{R'}(x, r'), s))$, and φ' of the first equals the second since, as before, $\pi_0 f^R(x, r) = \pi_0 f^{R'}(x, \varphi(r))$.

The same arguments work for \mathbf{b} and \mathbf{b}' . ■

Proposition 31 *Suppose that $M = (\delta_{\mathbf{X}}, \delta_{\mathbf{Y}}^R, f^R, \mathbf{b}^R)$, $N = (\delta_{\mathbf{Y}}, \delta_{\mathbf{Z}}, f^S, \mathbf{b}^S)$ and $N' = (\delta'_{\mathbf{Y}}, \delta'_{\mathbf{Z}}, f^{S'}, \mathbf{b}^{S'})$ are fb-lenses (see the diagram below in which $R_{\mathbf{X}\mathbf{Y}}$, $S_{\mathbf{Y}\mathbf{Z}}$ and $S'_{\mathbf{Y}\mathbf{Z}}$ are the corresponding corrs). Further, suppose $\varphi : S_{\mathbf{Y}\mathbf{Z}} \rightarrow S'_{\mathbf{Y}\mathbf{Z}}$ is a generator of \equiv_{fb} . Thus $N \equiv_{\text{fb}} N'$. Then $NM \equiv_{\text{fb}} N'M$.*

Proof. The composite NM has as corrs the pullback $T_{\mathbf{X}\mathbf{Z}}$ as in Definition 10, and similarly $N'M$ has corrs $T'_{\mathbf{X}\mathbf{Z}}$.



The proof follows the same argument as in the previous proposition. ■

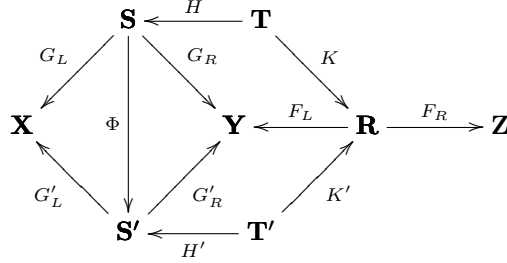
Theorem 32 *Equivalence classes for \equiv_{fb} are the arrows of a category, fbDLens.*

Proof. We first note that Propositions 30 and 31 ensure that composition is well-defined independently of choice of representative. There is an identity fb-lens with obvious structure which acts as an identity for the composition. It remains only to note that associativity follows by standard re-bracketing of iterated pullbacks. The re-bracketing function is the φ for an \equiv_{fb} equivalence. ■

Proposition 33 *Suppose that (G_L, P_L) , (G_R, P_R) , (G'_L, P'_L) , (G'_R, P'_R) , (F_L, Q_L) , and (F_R, Q_R) are d-lenses whose Gets are shown in the diagram below. Further, suppose $\Phi : \mathbf{S} \rightarrow \mathbf{S}'$ is a functor satisfying properties (E). Thus the span (G_L, P_L) , (G_R, P_R) is \equiv_{S_p} to the span (G'_L, P'_L) , (G'_R, P'_R) . Then the two possible span compositions are equivalent, that is*

$$\begin{aligned} & ((G_L, P_L), (G_R, P_R)) \circ ((F_L, Q_L), (F_R, Q_R)) \\ & \equiv_{S_p} ((G'_L, P'_L), (G'_R, P'_R)) \circ ((F_L, Q_L), (F_R, Q_R)) \end{aligned}$$

Proof. The top composite span of d-lenses in the diagram below has head \mathbf{T} , the pullback of G_R and F_L (see Definition 6), similarly the pullback \mathbf{T}' is the head of the bottom span composite.



In order to show the claimed equivalence, we construct a functor $\Phi' : \mathbf{T} \rightarrow \mathbf{T}'$. Since $G'_R \Phi H = F_L K$, the universal property of the pullback \mathbf{T}' can be used to define Φ' .

Since \mathbf{T} and \mathbf{T}' are pullbacks of functors, their objects can be taken to be pairs of objects from \mathbf{S} and \mathbf{R} , respectively \mathbf{S}' and \mathbf{R} . Similarly, their arrows can be taken to be pairs. Also H and K , respectively H' and K' can be taken to be projections. We can now explicitly describe the action of Φ' on an arrow of \mathbf{T} as $\Phi'(t_0, t_1) = (\Phi t_0, t_1)$.

As in Definition 6, we denote the Puts of the lenses whose Gets are H and K by P_H and P_K . Similarly for H' and K' . Denote the composite lens $(G_L, P_L)(H, P_H)$ by (G, P) and similarly let $(G', P') = (G'_L, P'_L)(H', P_{H'})$, $(F, Q) = (F_R, Q_R)(K, P_K)$ and $(F', Q') = (F_R, Q_R)(K', P_{K'})$.

We need to show that Φ' satisfies the conditions (E). By its construction Φ' commutes with the Gets, and is surjective on objects.

It remains to show that whenever $\Phi(S, R) = (S', R')$ (which implies that $R = R'$ and $\Phi(S) = S'$) we have

$$P'((S', R'), G'_L H'(S', R') \xrightarrow{\alpha} X') = \Phi' P((S, R), G_L H(S, R) \xrightarrow{\alpha} X')$$

and

$$Q'((S', R'), F_R K'(S', R') \xrightarrow{\gamma} Z') = \Phi' Q((S, R), F_R K(S, R) \xrightarrow{\gamma} Z')$$

We begin by proving the first of the two equations immediately above. We know that whenever $\Phi(S) = S'$, $P'_L(S', G'_L(S')) \xrightarrow{\alpha} X' = \Phi P_L(S, G_L(S) \xrightarrow{\alpha} X')$. We calculate

$$\begin{aligned} P'((S', R'), G'_L H'(S', R') \xrightarrow{\alpha} X') &= P'((S', R'), G'_L(S') \xrightarrow{\alpha} X') \\ &= P_{H'}((S', R'), P'_L(S', G'_L(S') \xrightarrow{\alpha} X')) \\ &= (P'_L(S', G'_L(S') \xrightarrow{\alpha} X'), Q_L(R', G'_R P'_L(S', G'_L(S') \xrightarrow{\alpha} X'))) \\ &= (\Phi P_L(S, G_L(S) \xrightarrow{\alpha} X'), Q_L(R, G'_R \Phi P_L(S, G_L(S) \xrightarrow{\alpha} X'))) \\ &= \Phi'(P_L(S, G_L(S) \xrightarrow{\alpha} X'), Q_L(R, G_R P_L(S, G_L(S) \xrightarrow{\alpha} X'))) \\ &= \Phi'(P_H((S, R), P_L(S, G_L(S) \xrightarrow{\alpha} X'))) \\ &= \Phi' P((S, R), G_L H(S, R) \xrightarrow{\alpha} X') \end{aligned}$$

The first step is merely that H' is a projection; the second is the definition of P' as the Put of the composite lens whose Get is $G'_L H'$; the third is the definition of

$P_{H'}$ (see Proposition 5); the fourth uses $R' = R$ and the fact stated just before the calculation; the fifth follows since Φ commutes with G_R and G'_R and by the definition of Φ' ; the sixth is the definition of P_H (see Proposition 5); the last is the definition of P as the Put of the composite lens whose Get is $G_L H$.

To establish the second equation, suppose $\Phi'(S, R) = (S', R')$, whence $R = R'$ and $\Phi(S) = S'$, and so because Φ satisfies conditions (E), we have

$$P'_R(S', G'_R(S')) \xrightarrow{\beta} Y' = \Phi P_R(S, G_R(S)) \xrightarrow{\beta} Y'$$

and since $G_R(S) = G'_R \Phi(S) = G'_R(S')$, the right hand side of this can be written as $\Phi P_R(S, G'_R(S')) \xrightarrow{\beta} Y'$. We calculate

$$\begin{aligned} Q'((S', R'), F_R K'(S', R') \xrightarrow{\gamma} Z') &= P_{K'}((S', R'), Q_R(R', F_R(R')) \xrightarrow{\gamma} Z') \\ &= (P'_R(S', F_L Q_R(R', F_R(R')) \xrightarrow{\gamma} Z'), Q_R(R', F_R(R')) \xrightarrow{\gamma} Z'). \end{aligned}$$

Before continuing the calculation, we simplify by defining β by $(G'_R(S') \xrightarrow{\beta} Y') = F_L Q_R(R', F_R(R')) \xrightarrow{\gamma} Z' = F_L Q_R(R, F_R(R)) \xrightarrow{\gamma} Z'$ after noting that $G'_R(S')$ is the domain of $F_L Q_R(R', F_R(R')) \xrightarrow{\gamma} Z'$ since the T' pullback square commutes. Now, continuing the calculation above:

$$\begin{aligned} &= (P'_R(S', G'_R(S')) \xrightarrow{\beta} Y', Q_R(R', F_R(R')) \xrightarrow{\gamma} Z') \\ &= (\Phi P_R(S, G_R(S)) \xrightarrow{\beta} Y', Q_R(R', F_R(R')) \xrightarrow{\gamma} Z') \\ &= (\Phi P_R(S, G'_R(S')) \xrightarrow{\beta} Y', Q_R(R, F_R(R)) \xrightarrow{\gamma} Z') \\ &= \Phi'(P_R(S, F_L Q_R(R, F_R(R)) \xrightarrow{\gamma} Z'), Q_R(R, F_R(R)) \xrightarrow{\gamma} Z') \\ &= \Phi' P_K((S, R), Q_R(R, F_R(R)) \xrightarrow{\gamma} Z') \\ &= \Phi' Q((S, R), F_R K(S, R) \xrightarrow{\gamma} Z') \end{aligned}$$

Starting at the beginning of the calculation (before the simplification) the first step uses that K' is a projection and the definition of Q' as the Put of the composite lens whose Get is $F_R K'$; the second is the definition of $P_{K'}$ (see Proposition 5); the third is the definition of β above; the fourth uses the fact displayed before the calculation; the fifth uses $R = R'$ and the note just before the calculation; the sixth uses the definitions of Φ' and β ; the seventh is the definition of P_K (see Proposition 5); the last is the definition of Q as the Put of the composite lens whose Get is $F_R K$. ■

Like Propositions 30 and 31, there is a reflected version of Proposition 33, showing that equivalent spans of d-lenses when composed on the left with another span of d-lenses are equivalent.

Proposition 34 *In notation analogous to Proposition 33,*

$$\begin{aligned} &((G_L, P_L), (G_R, P_R)) \circ ((F_L, Q_L), (F_R, Q_R)) \\ &\equiv_{Sp} ((G_L, P_L), (G_R, P_R)) \circ ((F'_L, Q'_L), (F'_R, Q'_R)). \end{aligned}$$

Theorem 35 *Equivalence classes for \equiv_{Sp} are the arrows of a category, denoted $SpDLens$.*

Proof. We first note that Proposition 33 and Proposition 34 ensure that composition is well-defined independently of choice of representative. There is a span of identity d-lenses which acts as the identity for the composition. Again, associativity follows by standard re-bracketing of iterated pullbacks of categories. The re-bracketing functor is the Φ for an \equiv_{Sp} equivalence. ■

6 An isomorphism of categories of lenses

Now that we have the categories $fbDLens$ and $SpDLens$, we can extend the constructions of Section 3 to functors on them.

Definition 36 *For the \equiv_{fb} equivalence class $[M]$ of an fb-lens M , let $\mathcal{A}([M])$ be the \equiv_{Sp} equivalence class of, in the notation of Lemma 12, the span L_M, K_M .*

Proposition 37 *\mathcal{A} is the arrow function of a functor, also denoted \mathcal{A} , from $fbDLens$ to $SpDLens$.*

Proof. We need to show that \mathcal{A} preserves identities and composition.

For the former denote by $M_{\mathbf{X}}$ the identity fb-lens on a category \mathbf{X} . We begin by noticing that the category \mathbf{X}_p at the head of the span of d-lenses constructed from $M_{\mathbf{X}}$ has as its objects exactly those of \mathbf{X} . Its arrows from X to X' are arbitrary pairs of \mathbf{X} arrows, both of which are from X to X' . Define the functor Φ from the head \mathbf{X} of the identity span on \mathbf{X} to \mathbf{X}_p by sending an arrow x of \mathbf{X} to the pair of arrows (x, x) . This functor Φ satisfies conditions (E), and so $\mathcal{A}([M_{\mathbf{X}}]) = [\mathbf{X}]$ as required.

Let M and M' be a composable pair of fb-lenses from \mathbf{X} to \mathbf{Y} and \mathbf{Y} to \mathbf{Z} respectively. The composite fb-lens $M'M$ has as corrs compatible pairs of corrs, one from M and one from M' (see Definition 10). The head \mathbf{S} of the span of d-lenses constructed from $M'M$ has as objects compatible pairs of corrs and as arrows from compatible corrs (r_1, r_2) to compatible corrs (r'_1, r'_2) , pairs of arrows, one from \mathbf{X} and one from \mathbf{Z} as shown

$$\begin{array}{ccccc} X & \xleftarrow{r_1} & Y & \xleftarrow{r_2} & Z \\ x \downarrow & & & & \downarrow z \\ X' & \xleftarrow{r'_1} & Y' & \xleftarrow{r'_2} & Z' \end{array}$$

On the other hand, the span composite of the spans constructed from M and M' has as head a category \mathbf{T} whose objects are pairs of compatible corrs from M and M' respectively. The arrows of \mathbf{T} are triples of arrows (x, y, z) as shown

$$\begin{array}{ccccc} X & \xleftarrow{r_1} & Y & \xleftarrow{r_2} & Z \\ x \downarrow & & \downarrow y & & \downarrow z \\ X' & \xleftarrow{r'_1} & Y' & \xleftarrow{r'_2} & Z' \end{array}$$

Define the functor Φ from \mathbf{T} to \mathbf{S} by sending the triple of arrows (x, y, z) to the pair of arrows (x, z) . This functor Φ satisfies conditions (E), and so $\mathcal{A}([M'M]) = \mathcal{A}([M'])\mathcal{A}([M])$ as required. ■

Definition 38 For the \equiv_{Sp} equivalence class $[L, K]$ of a span of d-lenses L, K , let $\mathcal{S}([L, K])$ be the \equiv_{fb} equivalence class of, in the notation of Lemma 15, the fb-lens $M_{L,K}$.

Proposition 39 \mathcal{S} is the arrow function of a functor, also denoted \mathcal{S} , from $SpDLens$ to $fbDLens$.

Proof. We need to show that \mathcal{S} preserves identities and composition.

Unlike the previous proof, the preservation of identities and composition is “on the nose”. That is, the construction applied to the identity gives precisely the identity fb-lens. Moreover, with judicious choice of pullbacks, the construction applied to the composite of two composable spans of d-lenses is the composite of the fb-lenses constructed from each of the spans.

Thus \mathcal{S} preserves the equivalence class of the identity span and

$$\mathcal{S}([L_1, K_1][L_2, K_2]) = \mathcal{S}([L_1, K_1])\mathcal{S}([L_2, K_2]).$$

■

Theorem 40 The functors \mathcal{A} and \mathcal{S} are an isomorphism of categories $SpDLens \cong fbDLens$.

Proof. We need to show that the composites $\mathcal{A}\mathcal{S}$ and $\mathcal{S}\mathcal{A}$ are identities. Recall first that both \mathcal{A} and \mathcal{S} have identity functions as object functions. Considering the arrows, Proposition 17 shows that $\mathcal{S}\mathcal{A}$ is the identity functor. We now consider $\mathcal{A}\mathcal{S}$.

For a span L, K of d-lenses between \mathbf{X} and \mathbf{Y} , using the notation of Lemmas 15 and 12, $\mathcal{A}\mathcal{S}([L, K]) = [L_{M_{L,K}}, K_{M_{L,K}}]$, so we consider the span $L_{M_{L,K}}, K_{M_{L,K}}$ of d-lenses whose Gets and Puts we denote by F_L, Q_L and F_K, Q_K respectively. The head of the span is a category we denote $\mathbf{S}_{L,K}$ whose objects are the same as the objects of \mathbf{S} , the head of the span L, K . We define an identity on objects functor $\Phi : \mathbf{S} \rightarrow \mathbf{S}_{L,K}$ on arrows by $\Phi(s) = (G_L(s), G_K(s))$ (recalling that arrows of $\mathbf{S}_{L,K}$ are pairs of arrows from \mathbf{X} and \mathbf{Y} , respectively). We finish by showing that Φ satisfies conditions (E), and so witnesses $\mathcal{A}\mathcal{S}([L, K]) = [L, K]$.

It remains to show that Φ satisfies conditions (E). Being identity on objects, Φ is certainly surjective on objects, and it commutes with the Gets by its construction. For condition (3), given an object S' of $\mathbf{S}_{L,K}$, an object S of \mathbf{S} such that $\Phi S = S'$, and an arrow $\alpha : G_L(S) = F_L(S') \rightarrow X'$ in \mathbf{X} , we have $P_L(S, \alpha)$ an arrow of \mathbf{S} . We need to show that $\Phi P_L(S, \alpha) = Q_L(S', \alpha)$. Since $\Phi S = S'$ we have $S = S'$. Now $Q_L(S', \alpha) = Q_L(S, \alpha) = (\alpha, \pi_0 f(\alpha, S))$, for the forward propagation f of $M_{L,K}$ constructed as in Lemma 15. By that construction $\pi_0 f(\alpha, S) = G_K(P_L(S, \alpha))$. But $\Phi P_L(S, \alpha) = (G_L(P_L(S, \alpha)), G_K(P_L(S, \alpha))) = (\alpha, \pi_0 f(\alpha, S)) = Q_L(S', \alpha)$.

Thus, since $\mathcal{A}\mathcal{S}([L, K]) = [L, K]$, $\mathcal{A}\mathcal{S}$ is the identity. ■

7 Related work

The present work owes its origins to the articles [2], [3] and [4] of Diskin and various co-authors. Delta lenses based on categories in both asymmetric and symmetric versions were first considered in those articles. We have somewhat simplified the description of symmetric delta lenses (here called fb lenses) from the earlier work. Moreover we have introduced an associative composition via the necessary equivalence relations to ensure that we have a category of fb lenses.

In earlier work [11], we proved that asymmetric delta lenses compose (essentially by composing the Gets and iterating the Puts). We also studied the category based asymmetric lenses we had introduced and called c-lenses in [16]. In [11] we showed that a c-lens is a special case of a d-lens, that the composition of c-lenses is as for d-lenses, and finally that d-lenses are strictly more general than c-lenses.

As noted in Section 4, we also showed in [11] that an asymmetric delta lens is an algebra for a certain semi-monad. This is similar to earlier results showing that asymmetric set-based lenses are algebras for a monad [15] (as well as being coalgebras for a comonad [6]), and so too are c-lenses [16], albeit for another monad.

This paper is a continuation of the project of unification begun in [12]. The goal there was to make precise the relationship between spans of the set-based asymmetric lenses of [5] and the set-based symmetric lenses of [8]. As in the current paper, both sides of this relationship needed to be considered up to appropriate equivalence relations. Hofmann et al [8] had already developed an equivalence relation among set-based symmetric lenses. Their equivalence relation was defined (in part) operationally using a bisimulation style.

The present authors have approached equivalence relations for lenses in an algebraic, rather than an operational, manner. In the algebraic context there is no requirement for an initialisation (the “missing” elements of [8] used to begin an “edit session”). Lenses are algebraic structures up to algebraic equivalence. If two lenses are algebraically equivalent (Definition 24), then choosing any two corresponding elements as initial (“missing”) elements, the two lenses are operationally equivalent meaning that any edit session carried out in the two lenses will have corresponding outcomes. Of course the converse is not in general true — algebraic equivalence is stronger than operational equivalence because, from an a priori choice of “missing” elements, operational equivalence only depends upon reachable elements. Some of the algebraic structure might be inequivalent, but unexplored. Our equivalence relation on set-based symmetric lenses was based on the equivalence relation of Definition 3.2 in [8], and see also Theorem 3.9 of [9].

On the other side, we needed to define an equivalence relation among spans of set-based asymmetric lenses. Our goal was to find the finest equivalence relation that would support composition of spans, ensure that that composition is associative, and identify lenses which are equivalent in their updating actions, although they might differ in details hidden in the heads of their spans. Such an equivalence needed to include, and be coarser than, span equivalence (an isomorphism between the heads of the spans commuting with the legs of the spans). Furthermore, pre-composing the legs of a span with a non-trivial set-based asymmetric lens gives a new span which differs from the first only in the hidden details — the head would be different but the updating actions at the extremities would be the same — so such pairs of spans should also be equivalent. In [12] we were able to show that the equivalence generated by such non-trivial set-based asymmetric lenses (commuting as lenses with the legs of the spans) worked well, and that result was very satisfying because it demonstrated, as so

much work in the theory of lenses does, that lenses and bidirectional transformations more generally are valuable generalisations of isomorphisms.

The recent thesis of Wagner [23] includes an extensive discussion of work on delta lenses as they relate to the edit lenses studied there and introduced earlier in [9]. As we recently showed in [14], there is a category of spans of asymmetric edit lenses closely related to the category of edit lenses. (The asymmetric edit lenses are a new concept introduced in the article.) This result furthers the unification project.

Edit lenses provide a compact representation of updates with common (often even polymorphic) structure by taking updates from a monoid acting partially on the states. It turns out that edit lenses are not more general than delta lenses. Indeed, edit lenses can themselves be represented as symmetric delta lenses [14].

Other important work related to delta lenses has appeared in [22], [21], [19] and, relating delta lenses to triple graph grammars, [7].

8 Conclusions

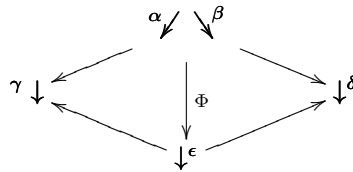
Because asymmetric delta lenses and symmetric delta lenses are so useful in applications, it is important that we understand them well and provide a firm mathematical foundation. This paper provides such a foundation by formalizing fb-lenses and their composition, including determining an appropriate equivalence relation on fb-lenses as is required for the composition to be well-defined and associative. Furthermore the resulting category fbDLens of fb-lenses is equivalent, indeed isomorphic, to the category SpDLens whose arrows are equivalence classes of spans of d-lenses.

This last result, the isomorphism between fbDLens and SpDLens , furthers the program to unify the treatment of symmetric lenses of type X as equivalence classes of spans of asymmetric lenses of type X , carrying that program for the first time into category-based lenses. (And that extension came with a surprise — see below.)

Naturally a unified treatment needs to be tested extensively on a wide range of lens types, and more work remains. We consider two more cases in the forthcoming [14]. The present paper is an important step in the program, and adds to the hope that the unification is close at hand. Indeed, with this work the program encounters the important category-based lenses for the first time and that substantially widens the base of unified examples.

We end with a distilled example. It shows in a simplified way why the equivalence used here, based on conditions (E), needs to be coarser than an equivalence generated by lenses commuting with the spans though it remains compatible with the earlier work. Thus it is also a coarser equivalence relation than might have been expected based on [12].

The figure below presents two spans of d-lenses. The categories at the head and feet of the spans have been shown explicitly. In three cases the category has a single non-identity morphism called variously γ , δ and ϵ while in the fourth case the category has two distinct nonidentity morphisms denoted α and β . In all cases objects and identity morphisms have not been shown. In three cases there are just two objects, while in the fourth case there are three, with a single object serving as the domain of both α and β .



The arrows displayed in both spans represent (the Gets of) d-lenses. In the lower span the d-lenses are simply identity d-lenses (the Gets are isomorphisms sending ϵ to γ in the left hand leg, and to δ in the right hand leg). Both of the Puts are then determined. The upper span is made up of two non-identity d-lenses. In both cases the Gets send both α and β to the one non-identity morphism (γ in the left hand leg and δ in the right hand leg). We specify the Puts for the upper span (eliding reference to objects in the Puts' parameters since they can be easily deduced): $P_L(\gamma) = \alpha$ and $P_R(\delta) = \beta$ for the left and right Puts respectively.

Notice that Φ , the functor that sends both α and β to ϵ , satisfies conditions (E) showing, as expected, that the two spans are equivalent. After all, if one traces through the forward and backward behaviours across the two spans the results at the extremities are in all cases the same, though the intermediate results at the heads of the spans differ. However, Φ cannot be the Get of a lens which commutes with the other four d-lenses. Indeed, to commute with the left hand lenses would require $P_\Phi(\epsilon) = \alpha$ while to commute with the right hand lenses would require $P_\Phi(\epsilon) = \beta$, but $\alpha \neq \beta$.

In fact, a similar example with sets in place of categories shows that the equivalence relation between spans used in [12] (which is generated by well-behaved set-based asymmetric lenses) is finer than might be desired. The authors have recently shown how that paper can be modified to use an equivalence relation based on conditions (E). Using that equivalence relation, the main result of [12] becomes an isomorphism (as in this paper) rather than a retraction.

References

- [1] Barr, M. and Wells, C. (1995) *Category theory for computing science*. Prentice-Hall, second edition, or free online at <http://www.math.mcgill.ca/triples/Barr-Wells-ctcs.pdf>
- [2] Diskin, Z., Xiong, Y., Czarnecki, K. (2011), From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case, *Journal of Object Technology* **10**, 6:1–25, doi:10.5381/jot.2011.10.1.a6
- [3] Diskin, Z., Xiong, Y., Czarnecki, K., Ehrig, H., Hermann, F., and Orejas, F. (2011), From State- to Delta-Based Bidirectional Model Transformations: the Symmetric Case, Springer LNCS **6981**, 304–318, doi:10.1007/978-3-642-24485-8_22
- [4] Diskin, Z., and Maibaum, T. (2012) Category Theory and Model-Driven Engineering: From Formal Semantics to Design Patterns and Beyond. 7th ACCAT Workshop on Applied and Computational Category Theory, 1–21.
- [5] Foster, J., Greenwald, M., Moore, J., Pierce, B. and Schmitt, A. (2005) Combinators for bi-directional tree transformations: A linguistic approach to the view

- update problem. Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 233–246.
- [6] Gibbons, J. and Johnson, M. (2012) Relating algebraic and coalgebraic descriptions of lenses. *Electronic Communications of the EASST*, **49**, 1–16.
 - [7] Hermann, F., Ehrig, H., Orejas, F., Czarnecki, K., Diskin, Z. and Xiong, Y. (2011) Correctness of model synchronization based on triple graph grammars. Springer LNCS **6981**, 668–682, DOI:10.1007/978-3-642-24485-8_49
 - [8] Hofmann, M., Pierce, B., and Wagner, D. (2011) Symmetric Lenses. In ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), 371–384.
 - [9] Hofmann, M., Pierce, B., and Wagner, D. (2012) Edit Lenses. In ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), 495–508.
 - [10] Johnson M. and Rosebrugh, R. (2012) Lens put-put laws: monotonic and mixed. *Electronic Communications of the EASST*, **49**, 1–13.
 - [11] Johnson, M. and Rosebrugh, R. (2013) Delta lenses and opfibrations. *Electronic Communications of the EASST*, **57**, 1–18.
 - [12] Johnson, M. and Rosebrugh, R. (2014) Spans of lenses. *CEUR Proceedings*, **1133**, 112–118.
 - [13] Johnson M. and Rosebrugh, R. (2015) Spans of delta lenses. *CEUR Proceedings*, **1396**, 1–15.
 - [14] Johnson M. and Rosebrugh, R. (2016) Unifying Set-Based, Delta-Based and Edit-Based Lenses. *CEUR Proceedings*, **1571**, 1–13.
 - [15] Johnson, M., Rosebrugh, R. and Wood, R. J. (2010) Algebras and Update Strategies. *J.UCS* **16**, 729–748, DOI:10.3217/jucs-016-05-0729
 - [16] Johnson, M., Rosebrugh, R. and Wood, R. J. (2012) Lenses, fibrations and universal translations. *Mathematical Structures in Computer Science*. **22**, 25–42, DOI:10.1017/S0960129511000442
 - [17] Mac Lane, S. (1998) *Categories for the Working Mathematician*. Springer, 2nd edition.
 - [18] Meertens, L. (1998) Designing Constraint Maintainers for User Interaction. Preprint, 79 pages, <ftp://ftp.kestrel.edu/pub/papers/meertens/dcm.ps>, accessed 2016-08-22.
 - [19] Orejas, F., Boronat, A., Ehrig, H., Hermann, F. and Schölzel, H. (2013) On Propagation-Based Concurrent Model Synchronization. *Electronic Communications of the EASST*, **57**, 1–19.
 - [20] Pierce, B. and Schmitt, A. (2003) Lenses and view update translation. Preprint, 7 pages. <http://www.cis.upenn.edu/~bcpierce/papers/dblenses.pdf>, accessed 2016-08-22.
 - [21] Pacheco, H., Cunha, A., and Hu, Z. (2012) Delta lenses over inductive types. *Electronic Communications of the EASST*, **57**, 1–18.
 - [22] Stevens, P. (2008) Towards an algebraic theory of bidirectional transformations. In *Graph Transformations: 4th International Conference, ICGT*, Springer LNCS **5214**, 1–17, DOI:10.1007/978-3-540-87405-8_1

- [23] Wagner, D. (2014) Symmetric edit lenses: A new foundation for bidirectional languages. Ph.D Thesis, University of Pennsylvania.

About the authors



Michael Johnson is a Professor in the Departments of Mathematics and Computing at Macquarie University. His research interests include theory of database systems and higher dimensional category theory. Contact him at michael.johnson@mq.edu.au



Robert Rosebrugh is a Professor of Mathematics and Computer Science at Mount Allison University. His research interests include theory of database systems and completely distributive categories. Contact him at rrosebrugh@mta.ca

Acknowledgments This paper has benefited from valuable suggestions by anonymous referees. The authors are grateful for the careful and insightful refereeing. In addition, the authors acknowledge with gratitude the support of NSERC Canada and the Australian Research Council. The paper is a revised and substantially expanded version of the conference article [13].