Asymmetric lenses, symmetric lenses and spans

Bob Rosebrugh
(with Mike Johnson)

FMCS 2014

Outline

- background: databases, updates and views:
  the view update problem
- asymmetric lenses and view updates
- symmetric lenses and model synchronization
- spans of asymmetric lenses

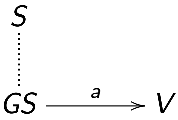
Updates and views

- An update changes database state(s)
- Examples: deletion, insertion, attribute modification
  Either: modification of single state by delete or insert
  or an update process: an endo U of states, $\mathbf{S}$
- A view may limit access e.g. for security
  or present information to user class e.g. clerk
  or specify boundary for database integration
- "Get" view states via $G : \mathbf{S} \longrightarrow \mathbf{V}$

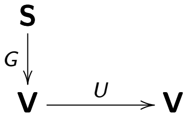
View update problem

When can an update to view state(s) either

- for single (view) state (e.g. formal insertion a):



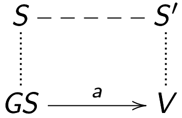
- for an update process (e.g. U):



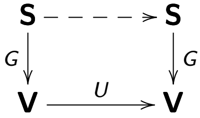
View update problem

When can an update to view state(s) either

- for single (view) state (e.g. insert a):



- for an update process (e.g. U):



propagate (or lift) correctly to full database update?

Abstract view updates

Bancilhon and Spyratos (1982, and others) studied the view update problem. For them:

- database states are an abstract set S
- view states are an abstract set V – the codomain of a surjective view definition mapping $G : S \longrightarrow V$
- a view update is an endo-function $U : V \longrightarrow V$
- a translation T_U of view update U is a database update on S lifting UG through G

$$\begin{array}{ccc} S & \overset{T_U}{\dashrightarrow} & S \\ {\scriptstyle G}\big\downarrow & & \big\downarrow{\scriptstyle G} \\ V & \xrightarrow{\;U\;} & V \end{array}$$

A translation strategy limited to “complemented” (better, “factored”) views follows...

Asymmetric Lenses

(B. Pierce et al, 2005)
Consider a full database state s and view state $G(s)$
When $G(s)$ updated to v, say, want strategy to find
updated full database state $s' = T_U s$ (over v):

$$\begin{array}{ccc} s & \longmapsto\!\!\!\dashrightarrow & s' \\ \downarrow & & \downarrow \\ G(s) & \longmapsto & v \end{array}$$

Idea: provide a process $P : V \times S \longrightarrow S$ called "Put" so that
$P(v, s)$ is the translated state s' after $G(s)$ updated to v
Some equations should follow...

This structure, called a lens, provides translations

Also arose in considering "abstract models of storage"
(where there is a similar update problem)

Asymmetric Lenses

Let **C** be a category with finite limits
An asymmetric lens in **C** is $L = (S, V, G, P)$ with

- S and V objects (... database states/view states)
- $S \xrightarrow{G} V$ aka 'Get' and $V \times S \xrightarrow{P} S$ aka 'Put'

  called well-behaved (wb) if satisfying:

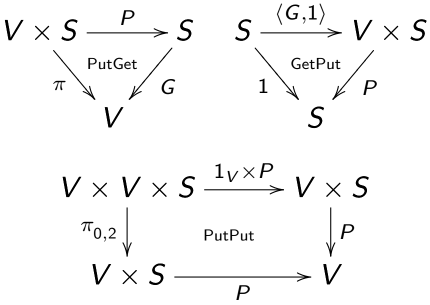
  PutGet: Get of Put is projection: $GP = \pi_0$ (or $GP(v, s) = v$)

  GetPut: Put for non-update is trivial $P\langle G, 1_S \rangle = 1_S$

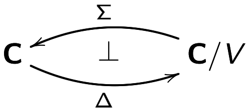
  and very well-behaved (vwb) if also satisfying:

  PutPut: repeated Puts depend only on the last:
  $P(1_V \times P) = P\pi_{0,2}$ (or $P(v', P(v, s)) = P(v', s)$)

the equations diagrammatically



So $\Delta\Sigma G \xrightarrow{P} G$ is in $\mathbf{C}/V$ where



And moreover . . .

Proposition (JRW)

A (vwb) lens has P an algebra structure on G in $\mathbf{C}/V$ for the monad $\Delta\Sigma$ on $\mathbf{C}/V$.

For vwb lenses:

- $\mathbf{C} = \mathbf{set}$, $L = (S, V, G, P)$ recovers B&S results:
  $S \cong V \times C$, G the projection, C 'complement' of V,
  the translation: $T_U(s) := P(UGs, s)$
- $\mathbf{C} = \mathbf{ord}$, recovers results of S. Hegner (2004)
- $\mathbf{C} = \mathbf{cat}$: G a projection and hence fibration and opfibration

Lenses compose

We can compose lenses:
if $L = (S, V, G, P)$ and $M = (V, W, H, Q)$ are lenses in $\mathbf{C}$
then $ML = (S, W, HG, R)$ is a lens, with the Put R defined:

$$W \times S \xrightarrow{1_W \times \langle G, 1_S \rangle} W \times V \times S \xrightarrow{\langle Q, 1_S \rangle} V \times S \xrightarrow{P} S$$

Composites of wb, resp vwb, lenses are wb, resp vwb

There are identity on objects (ioo), *non-full*
functors between asymmetric lens (in $\mathbf{C}$) categories

$$\mathrm{ALens}_v(\mathbf{C}) \longrightarrow \mathrm{ALens}_{\mathrm{w}}(\mathbf{C}) \longrightarrow \mathrm{ALens}(\mathbf{C})$$

Lenses preserved

Suppose $F : \mathbf{C} \longrightarrow \mathbf{D}$ is a finite product preserving functor
For $L = (X, Y, G, P)$ an asymmetric lens in $\mathbf{C}$,
respectively: a well-behaved lens, very well-behaved lens

$FL = (FX, FY, FG, FP)$ is an asymmetric lens in $\mathbf{D}$,
respectively: a well-behaved lens, very well-behaved lens

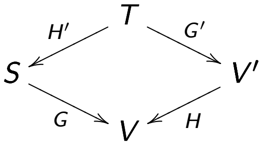
Moreover, F preserves lens composition and we denote:

$$F : \mathrm{ALens}(\mathbf{C}) \longrightarrow \mathrm{ALens}(\mathbf{D})$$

respectively from $\mathrm{ALens}_{\mathrm{w}}(\mathbf{C})$ and $\mathrm{ALens}_{\mathrm{v}}(\mathbf{C})$.

Lenses and pulling back

For $\mathbf{C}$ with pullbacks and an asymmetric lens $L = (X, Y, G, P))$ and $H : V' \longrightarrow V$ in $\mathbf{C}$ pulling back G along H in $\mathbf{C}$ gives the Get for asymmetric lens $L' = (T, V', G', P'))$ with $P' = \langle P(H \times H'), \pi_0 \rangle$



Similarly for well-behaved and very well-behaved lenses

But: $\mathrm{ALens}(\mathbf{C}), \mathrm{ALens}_{\mathrm{w}}(\mathbf{C}), \mathrm{ALens}_{\mathrm{v}}(\mathbf{C})$ may not have pullbacks.

Less abstract lenses

For a view in **cat**, ie $G : \mathbf{S} \longrightarrow \mathbf{V}$

(Insert) updates needing lifts should better be $GS \xrightarrow{a} V$

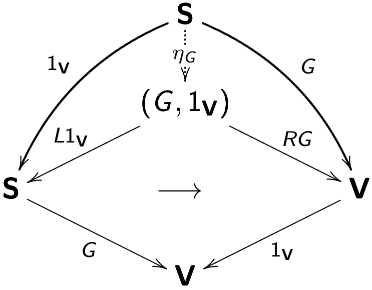
(Contrast simply pairs (S, V) above)

The domain of Put for G is better $(G, 1_{\mathbf{V}})$ than $\mathbf{V} \times \mathbf{S}$

Right comma projection $R(-)$ is functor part of a monad

$$R : \mathbf{cat}/\mathbf{V} \longrightarrow \mathbf{cat}/\mathbf{V}$$

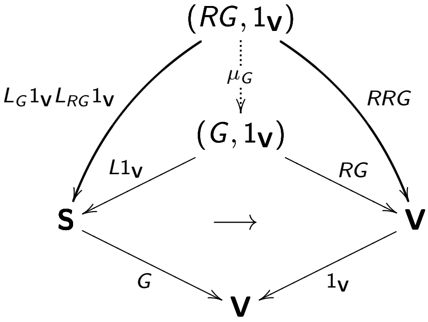
with unit component $G \xrightarrow{\eta_G} RG$ defined by



where $\eta_G = (1_{\mathbf{V}}, G, 1_G) : \mathbf{S} \longrightarrow (G, 1_{\mathbf{V}})$ defined universally

Less abstract lenses

and multiplication $RRG \xrightarrow{\mu_G} RG$ defined by:

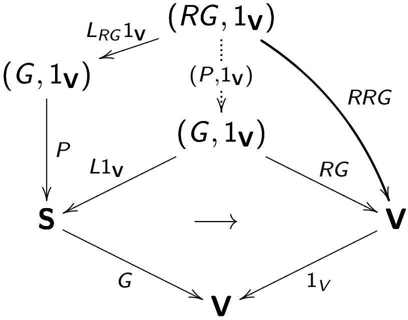


with

$$\mu_G = (L_G 1_{\mathbf{V}} \cdot L_{RG} 1_{\mathbf{V}}, RRG, \beta(\alpha L_{RG} 1_{\mathbf{V}})) : (RG, 1_{\mathbf{V}}) \longrightarrow (G, 1_{\mathbf{V}})$$

An iterate of a P

For $G : \mathbf{S} \longrightarrow \mathbf{V}$ consider a
$P : (G, 1_{\mathbf{V}}) \longrightarrow \mathbf{S}$ satisfying $GP = RG$, so that
$GPL_{RG}1_{\mathbf{V}} = RG \cdot L_{RG}1_{\mathbf{V}} \xrightarrow{\beta} RRG$, define: $(P, 1_{\mathbf{V}})$ by



c-Lenses

Again, for a view in **cat**, $G : \mathbf{S} \longrightarrow \mathbf{V}$
the "Put" for view updates $GS \longrightarrow V$ should be
a process $P : (G, 1_{\mathbf{V}}) \longrightarrow \mathbf{S}$, and we define:

A c-lens in **cat** is $L = (\mathbf{S}, \mathbf{V}, G, P)$
satisfying

c-PutGet: $GP = RG$

c-GetPut: $P\eta_G = 1_{\mathbf{S}}$

c-PutPut: $P\mu_G = P(P, 1_{\mathbf{V}})$

(Could model delete updates $V \longrightarrow GS$, then "Put" s.b.
$P : (1_{\mathbf{V}}, G) \longrightarrow \mathbf{S}$ using LG in the PutGet equation...)

c-Lenses are opfibrations

or diagrammatically:

$$
\begin{array}{ccc}
(G, 1_{\mathbf{V}}) & \xrightarrow{P} & \mathbf{S} \\
& \searrow^{RG} & \downarrow G \\
& & \mathbf{V}
\end{array}
\qquad
\begin{array}{ccc}
\mathbf{S} & \xrightarrow{\eta_G} & (G, 1_{\mathbf{V}}) \\
& \searrow^{1_{\mathbf{S}}} & \downarrow P \\
& & \mathbf{S}
\end{array}
\qquad
\begin{array}{ccc}
(RG, 1_{\mathbf{V}}) & \xrightarrow{(P, 1_{\mathbf{V}})} & (G, 1_{\mathbf{V}}) \\
\downarrow \mu_G & & \downarrow P \\
(G, 1_{\mathbf{V}}) & \xrightarrow[P]{} & \mathbf{S}
\end{array}
$$

Recalling that an algebra structure for the monad

$$\mathbf{cat}/\mathbf{V} \xrightarrow{R} \mathbf{cat}/\mathbf{V}$$

is a split opfibration:

Proposition (JRW)

For a c-lens $L = (\mathbf{S}, \mathbf{V}, G, P)$ in **cat***, P is an algebra structure for R so G is a split opfibration.*

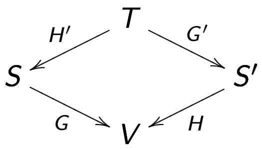
c-Lenses compose

Opfibrations compose, so if $G : \mathbf{S} \longrightarrow \mathbf{V}$ and $G' : \mathbf{V} \longrightarrow \mathbf{W}$ are c-lenses. so is $G'G : \mathbf{S} \longrightarrow \mathbf{W}$

Subcategory of **cat** with arrows c-lenses is denoted ACLens.
Asymmetric lens in **cat** is a c-lens, so $\text{ALens}_v(\mathbf{cat})$ is a subcategory.

Further, opfibrations pull back (along any functor) and a cospan of c-lenses gives span of c-lenses

Interest in spans motivated by *cospan* of views G, H:



giving a *span* of views G', H' (of c-lenses if G, H are)

Another categorical version of lenses

Motivated by similar considerations Z. Diskin and co-authors called updates deltas, made the *set* of deltas the domain of Put (now returning a delta), with axioms similar to c-lenses

An (asymmetric) delta lens (d-lens) in **cat** is $L = (\mathbf{S}, \mathbf{V}, G, P)$ where $G : \mathbf{S} \longrightarrow \mathbf{V}$ is a functor and $P : |(G, 1_{\mathbf{V}})| \longrightarrow |\mathbf{S}^2|$ is a function and the data satisfy:

(i) d-PutInc: the domain of $P(S, \alpha : GS \longrightarrow V)$ is S

(ii) d-PutId: $P(S, 1_{GS} : GS \longrightarrow GS) = 1_S$

(iii) d-PutGet: $GP(S, \alpha : GS \longrightarrow V) = \alpha$

(iv) d-PutPut:

$$P(S, \beta\alpha : GS \longrightarrow V \longrightarrow V') = P(S', \beta : GS' \longrightarrow V')P(S, \alpha : GS \longrightarrow V)$$

where S' is the codomain of $P(S, \alpha : GS \longrightarrow V)$

ADLens

Proposition

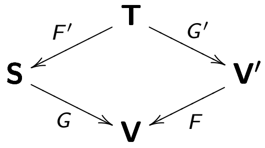
If $L = (\mathbf{S}, \mathbf{V}, G, P)$ *and* $M = (\mathbf{V}, \mathbf{W}, H, Q)$ *are d-lenses then then* $ML = (\mathbf{S}, \mathbf{W}, HG, R)$ *is a d-lens, with* R *as*

$$|(HG, 1_{\mathbf{W}})| \xrightarrow{Q} |(G, 1_{\mathbf{V}})| \xrightarrow{P} |\mathbf{S}|^2$$

Identity functor is Get for a d-lens and unitary for composition. Denote the resulting category ADLens

Proposition

If $L = (\mathbf{S}, \mathbf{V}, G, P)$ *is a d-lens and* $F : \mathbf{V}' \longrightarrow \mathbf{V}$ *is a functor then* G' *in the pullback (in* **cat***) is the Get of a d-lens*



c-Lenses and d-Lenses

For $G : \mathbf{S} \longrightarrow \mathbf{V}$, denote $G_0 = |\mathbf{S}| \longrightarrow \mathbf{S} \xrightarrow{G} \mathbf{V}$ and $R_0 G : (G_0, 1_{\mathbf{V}}) \longrightarrow \mathbf{V}$

Semi-monad (R_0, μ^0) on $\mathbf{cat}/\mathbf{V}$ similar to R, and transformation η^0 to R_0 (from functor sending G to G_0)

Proposition

If $L = (\mathbf{S}, \mathbf{V}, G, P)$ is a d-lens then (G, P_0) is an (R_0, μ^0) algebra satisfying $P_0 \eta^0 G = P_0 \eta_{G_0} = I_{\mathbf{S}}$, and conversely.

Corollary

A c-lens is a d-lens; composition is compatible.

Though not every d-lens is a c-lens

Categories of asymmetric lenses

In summary:

$$
\begin{array}{ccccccc}
\mathsf{ALens_v}(\mathbf{set}) & \longrightarrow & \mathsf{ALens_v}(\mathbf{cat}) & \longrightarrow & \mathsf{ACLens} & \longrightarrow & \mathsf{ADLens} \\
 & \searrow & \nearrow & & & & \\
 & & \mathsf{ALens_v}(\mathbf{ord}) & & & &
\end{array}
$$

All admit the $Sp(U)$ construction which follows...

The $Sp(U)$ Construction

$\mathbf{C}$ with finite limits; $U : \mathbf{A} \longrightarrow \mathbf{C}$ ioo functor reflecting isos
(We are thinking ALens $\longrightarrow \mathbf{C}$)
Assume an operation P on $\mathbf{C}$ cospans

$$B \xrightarrow{g} C \xleftarrow{U(r)} D$$

giving arrows $P(g, r)$ in $\mathbf{A}$ such that
1) there is in $\mathbf{C}$ a pullback:

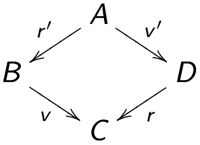
$$\begin{array}{ccccc} & & A & & \\ & {}^{t'}\swarrow & & \searrow^{g'} & \\ B & & & & D \\ & {}_{g}\searrow & & \swarrow_{U(r)} & \\ & & C & & \end{array}$$

with $t' = U(r')$ where $r' = P(g, r)$

And...

The $Sp(U)$ Construction

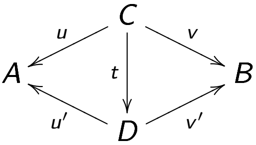
2) If also $g = U(v)$ then for $v' = P(G(r), v)$ the square commutes (in $\mathbf{A}$):



Next, given U and operation P, define category $Sp(U)$:
Objects of $\mathbf{A}$ (or $\mathbf{C}$)
Arrows $\equiv_U$ equiv classes of spans in $\mathbf{A}$ where

The $Sp(U)$ Construction

$\equiv_U$ generated by span morphisms in **A**



with $u = u't$ and $v = v't$ and $G(t)$ split epi.
$Sp(U)$ composition by span composition in **C**

Proposition

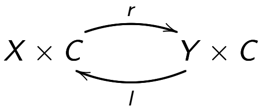
With the data just defined, $Sp(U)$ is a category.

Symmetric lenses

(Hoffman, Pierce and Wagner, 2011)
Idea: Describe re-synchronization for model classes (of states) X, Y having synchronization ("complement") information from C.

Given states x, y synchronized by a complement c and an (updated) state x' of X, determine re-synchronizing complement c' from (x', c) and an updated y' of Y (and vice versa)

So an arrow $r : X \times C \longrightarrow Y \times C$ and vice versa.



Now (x', c', y') is (re)synchronized. Some equations are expected because...

if l applied to (y', c') then the result should be (x', c')

Example (from H,P,W)

The data in states x, y might initially be the following

x :

Schubert	1797-1828
Schumann	1810-1856

y :

Schubert	Austria
Schumann	Germany

with initial complement, "hidden data" (a C state):

c :

1797-1828	Austria
1810-1856	Germany

An edit to x gives new X state x':

Schubert	1797-1828
Schumann	1810-1856
Monteverdi	1567-1643

then applying $r(x', c)$ results in new C and Y states:

c' :

1797-1828	Austria
1810-1856	Germany
1567-1643	?country

y' :

Schubert	Austria
Schumann	Germany
Monteverdi	?country

Symmetric lenses

Let **C** be a category with finite limits.

For objects X, Y in **C**, an rl lens from X to Y, denoted $L = (X, Y, C, r, l)$ with C an object of "complements" and morphisms

$$r : X \times C \longrightarrow Y \times C \quad \text{and} \quad l : Y \times C \longrightarrow X \times C$$

satisfying the equations:

$$\pi_X lr = \pi_X : X \times C \longrightarrow X \qquad \pi_C lr = \pi_C r : X \times C \longrightarrow C \qquad \text{(PutRL)}$$
$$\pi_Y rl = \pi_Y : Y \times C \longrightarrow Y \qquad \pi_C rl = \pi_C l : Y \times C \longrightarrow C \qquad \text{(PutLR)}$$

HPW require an element $m : 1 \longrightarrow C$ where m is for "missing" (called pc-symmetric below)

Symmetric lenses decompose

Remark
For an RL lens $L = (X, Y, C, r, l)$ in $\mathbf{C}$, the equations $rlr = r$ and $lrl = l$ hold.

Suppose that $L = (X, Y, C, r, l)$ is an rl lens in $\mathbf{C}$.

Let $e : S_L \longrightarrow X \times Y \times C$ be an equalizer of $r\pi_{0,2}$ and $\pi_{1,2}$.

If $\mathbf{C} = \mathbf{set}$,

$$S_L = \{(x, y, c) \mid r(x, c) = (y, c)\} = \{(x, y, c) \mid l(y, c) = (x, c)\}$$

Elements of S_L are the "synchronized triples"

Symmetric lenses decompose

For L, S_L as above:

Proposition

There is a span

$$L_l : X \longleftarrow S_L \longrightarrow Y : L_r$$

in $\mathrm{ALens_w}$ *from* X *to* Y *with Gets defined by* $g_l = \pi_X e$, $g_r = \pi_Y e$.
The Put. p_l *for* L_l *(*p_r *similar) is defined by*

$$X \times S_L \xrightarrow{1_X \times e} X \times X \times Y \times C \xrightarrow{\pi_{0,3}} X \times C$$

$$\xrightarrow{\Delta_X \times 1_C} X \times X \times C \xrightarrow{1_X \times r} S_L$$

(The **set** *formula for* p_l *is* $p_l(x', (x, y, c)) = (x', r(x', c))$*.)*

Denote the span (L_l, L_r) by $A(L)$

Recalling $U_w : \mathrm{ALens_w} \longrightarrow \mathbf{C}$, define $\mathrm{SLens}_w = Sp(U_w)$

Symmetric lenses compose

For rl lenses $L_1 = (X, Y, C_1, r_1, l_1)$ and $L_2 = (X, Y, C_2, r_2, l_2)$:

$L_1 \sim L_2$ if exists *well-behaved asymmetric* lens $L = (C_1, C_2, t, p)$ with t split epi and respecting L_1, L_2 operations, which means:

$$r_2(X \times t) = (Y \times t)r_1 \text{ and } l_2(Y \times t) = (X \times t)l_1$$

and

$$r_1(X \times p) = (Y \times p)(r_2 \times C_1) \text{ and } l_1(Y \times p) = (X \times p)(l_2 \times C_1).$$

$\sim$ generates equivalence relation on rl lenses X to Y denoted $\equiv_{rl}$

$\equiv_{rl}$ class of L denoted $[L]_{rl}$.

Symmetric lenses compose

$L = (X, Y, C, r, l)$, $M = (Y, Z, C', r', l')$ rl lenses

Their *rl-composite lens* is $ML = (X, Z, C'', r'', l'', m'')$ where $C'' = C \times C'$ and

$$r'' = \langle \pi_{0,2}, \pi_1 \rangle (r' \times 1_C) \langle \pi_{0,2}, \pi_1 \rangle (r \times 1_{C'}) \quad (l'' \text{ similar})$$

Proposition

For rl lenses L_1, L_2 from X to Y and M_1, M_2 from Y to Z in $\mathbf{C}$, if $L_1 \equiv_{rl} L_2$ and $M_1 \equiv_{rl} M_2$ then $M_1 L_1 \equiv_{rl} M_2 L_2$.

RLLens has objects of **C**; arrows X to Y are $\equiv_{rl}$ classes

Proposition

There is an identity on objects functor

$$\mathbf{A} : \mathrm{RLLens} \longrightarrow \mathrm{SLens}_w$$

defined by $\mathbf{A}([L]_{rl}) = [A(L)]_{U_w}$.

Symmetric lenses from asymmetric

Going the other way... From span of wb asymmetric lenses $L = (S, X, G_X, P_X)$, $M = (S, Y, G_Y, P_Y)$, construct rl lens $S(L, M) = (X, Y, S, r, l)$ where (in **set**)

$$r(x', (x, y, c)) = (G_Y P_X(x', (x, y, c)), P_X(x', (x, y, c))) \quad (l \text{ similar})$$

Proposition

Denote $AS(L, M)$ by $L_l : X \longleftarrow S_L \longrightarrow Y : L_r$. There is iso span morphism
$g : S \longrightarrow S_L$, so $AS(L, M) \equiv_{U_w} (L, M)$,

Categories of symmetric lenses

Proposition

If $L : X \longleftarrow S \longrightarrow Y : M$, $L' : X \longleftarrow S' \longrightarrow Y : M'$ *are* $\equiv_{U_w}$ *equivalent spans of well behaved asymmetric lenses then* $S(L, M) \equiv_{rl} S(L', M')$ *and* $\mathbf{S}([(L, M)]_{\equiv_{U_w}}) = [S(L, M)]_{rl}$ *defines functor* $\mathbf{S} : \mathrm{SLens}_w \longrightarrow \mathrm{RLLens}$.

Theorem

SLens_w *is a retraction of* RLLens *via* $\mathbf{A}$ *and* $\mathbf{S}$.

pc-symmetric lenses

Hofmann, Pierce and Wagner introduced an equivalence relation we denote $\equiv_{pc}$ on their pc-symmetric lenses from X to Y

$\equiv_{pc}$ allows well-defined composition of pc-symmetric lenses giving pcLens

Starting from rl lenses, suitably adding points so that $\equiv_{U_w}$ can be compared, we can show that $\equiv_{pc}$ is in fact coarser than $\equiv_{U_w}$

Symmetric delta lenses (Diskin et al. 2011/12)

For symmetric version of d-lens, again use morphisms for updates:

Let **A** and **B** be small categories.

Given an *update* $a : A \longrightarrow A'$ in **A** *from* state A
where A synchronized with B by "correspondence" $r : A \leftrightarrow B$,
symmetric d-lens should deliver an update $b : B \longrightarrow B'$ in **B** *and*
re-synchronization $r' : A' \leftrightarrow B'$:

$$\begin{array}{ccc} A & \overset{r}{\longleftrightarrow} & B \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle b} \\ A' & \underset{r'}{\leftarrow - - \rightarrow} & B' \end{array}$$

Symmetric delta lenses

A symmetric delta lens (sd-lens) from **A** to **B** is $L = (\delta_{\mathbf{A}}, \delta_{\mathbf{B}}, \mathrm{fP}, \mathrm{bP})$ with a span of sets

$$\delta_{\mathbf{A}} : |\mathbf{A}| \longleftarrow R_{\mathbf{AB}} \longrightarrow |\mathbf{B}| : \delta_{\mathbf{B}}$$

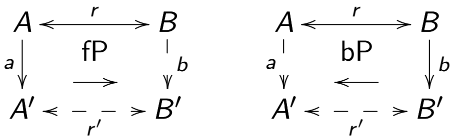
(elements of $R_{\mathbf{AB}}$ called corrs are denoted $r : A \leftrightarrow B$) and forward and backward propagation operations

$$\mathrm{fP} : \mathit{Arr}(\mathbf{A}) \times_{|\mathbf{A}|} R_{\mathbf{AB}} \longrightarrow \mathit{Arr}(\mathbf{B}) \times_{|\mathbf{B}|} R_{\mathbf{AB}}$$

$$\mathrm{bP} : \mathit{Arr}(\mathbf{A}) \times_{|\mathbf{A}|} R_{\mathbf{AB}} \longleftarrow \mathit{Arr}(\mathbf{B}) \times_{|\mathbf{B}|} R_{\mathbf{AB}}$$

Symmetric delta lenses

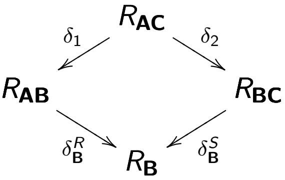
Display instances of propagation operations as:



where $\mathrm{fP}(a, r) = (b, r')$ and $\mathrm{bP}(b, r) = (a, r')$.:
Propagation respects identities: $r : A \leftrightarrow B$ implies $\mathrm{fP}(\mathrm{id}_A, r) = (\mathrm{id}_B, r)$ and $\mathrm{bP}(\mathrm{id}_B, r) = (\mathrm{id}_A, r)$ and composition in **A** and **B**: $\mathrm{fP}(a'a, r) = \mathrm{fP}(a', \pi_1(\mathrm{fP}(a, r)))$, similarly for **B**.

Composing symmetric delta lenses

Let $L = (\delta^R_{\mathbf{A}}, \delta^R_{\mathbf{B}}, \mathrm{fP}^R, \mathrm{bP}^R)$ and $L' = (\delta^S_{\mathbf{B}}, \delta^S_{\mathbf{C}}, \mathrm{fP}^S, \mathrm{bP}^S)$ The composite sd-lens $L'L = (\delta_{\mathbf{A}}, \delta_{\mathbf{C}}, \mathrm{fP}, \mathrm{bP})$ where $\delta_{\mathbf{A}} = \delta^R_{\mathbf{A}}\delta_1$, $\delta_{\mathbf{C}} = \delta^S_{\mathbf{C}}\delta_2$ and $R_{\mathbf{AC}}$ is the pullback in



Define: $\mathrm{fP}(a, (r, s)) = (c, (r_f, s_f))$ and $\mathrm{bP}(c, (r, s)) = (a, (r_b, s_b))$ where $\mathrm{fP}^R(a, r) = (b, r_f), \mathrm{fP}^S(b, s) = (c, s_f)$ and $\mathrm{bP}^S(c, s) = (b, s_b), \mathrm{bP}^R(b, r) = (a, r_b)$.

The construction is used to define a category SDLens

SDLens and spans

Let $L = (\mathbf{S}, \mathbf{V}, G_L, P_L)$, $R = (\mathbf{S}, \mathbf{W}, G_R, P_R)$ be a span of d-lenses

Construct sd-lens $S_{L,R} = (\delta_{\mathbf{V}}, \delta_{\mathbf{W}}, \mathrm{fP}, \mathrm{bP})$ with forward propagation from P_L, G_R.
Conversely, from an sd-lens $M = (\delta_{\mathbf{A}}, \delta_{\mathbf{B}}, \mathrm{fP}, \mathrm{bP})$ we can construct a span $L_M = (\mathbf{S}, \mathbf{A}, G_L, P_L)$, $R_M = (\mathbf{S}, \mathbf{B}, G_K, P_K)$ of d-lenses using the corrs and propagations to define $\mathbf{S}$,

Comparison of SDLens and spans of asymmetric d-lenses remains to be made precise....

Conclusion

- Asymmetric lenses provide solutions to the view update problem in several contexts
- Symmetric lenses describe model synchronization processes also in various contexts
- Symmetric lenses *should* be understood via spans of asymmetric lenses and often arise from cospans

Thanks!