Taking updates seriously

Tarmo Uustalu

NII Shonan Meeting on BX, 25–29 Sept. 2016

My creed

- Delta-based is superior to state-based; state-based is a special case
- State-based lenses must be required to satisfy put-put.
  For weaker forms of overwriting (e.g., stated stored in "wearing memory") delta-based lenses are suitable and they satisfy their own appropriate put-put.
- By deltas I do not mean diffs, but **updates** (or edits) acting on states
- Around delta lenses, there is lots of structure

- This talk deals with asymmetric lenses only, but the symmetric case can be dealt with spans

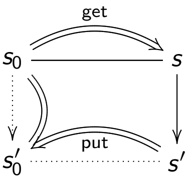
This talk

- From state-based lenses
- ... to (simply and dep. typed) update lenses
- ... to a further generalization
  (versions of Diskin's delta-lenses)
- emphasizing that ...
- ... different types lenses are the same as comonad coalgebras or comonad morphisms for appropriate comonads and
- ... special cases of proof-relevant simulations à la McKinna

State-based lenses (Foster et al.)

- Let S be a set (of view states).
- A *(state-based) lens* for S is a set S_0 (of source states) with maps $\text{get} : S_0 \to S$ and $\text{put} : S_0 \times S \to S_0$ such that

$$\text{get}\,(\text{put}\,(s_0, s)) = s$$
$$s_0 = \text{put}\,(s_0, \text{get}\, s_0)$$
$$\text{put}\,(\text{put}\,(s_0, s), s') = \text{put}\,(s_0, s')$$



- Lenses for S are the same as coalgebras of the *costate comonad* for S, defined by $DX =_{\mathrm{df}} S \times (S \Rightarrow X)$.
- Incidentally, they are also the same as morphisms between the costate comonads for S_0 and S.
- Given a lens $(S_0, \mathrm{get}, \mathrm{put})$, the coalgebra carrier is S_0 and the structure map is

$$\begin{array}{ccc} S_0 & \to & S \times (S \Rightarrow S_0) \\ s_0 & \mapsto & (\mathrm{get}\, s_0, \lambda s.\, \mathrm{put}\,(s_0, s)) \end{array}$$

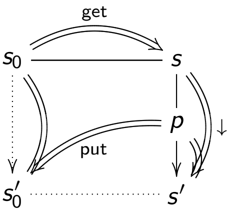
  while the underlying natural transformation of the comonad morphism is

$$\begin{array}{ccc} S_0 \times (S_0 \Rightarrow X) & \to & S \times (S \Rightarrow X) \\ (s_0, v) & \mapsto & (\mathrm{get}\, s_0, \lambda s.\, v\,(\mathrm{put}\,(s_0, s))) \end{array}$$

Update lenses (Ahman, Uustalu, MFPS XXX, 2014)

- Let S be a set (of view states), $(P, \mathrm{o}, \oplus)$ be a monoid (of view updates) and $\downarrow : S \times P \to S$ a right action of S on $(P, \mathrm{o}, \oplus)$ (application of view updates to view states).
- An *update lens* for $(S, (P, \mathrm{o}, \oplus), \downarrow)$ is a set S_0 (of source states) with maps $\mathrm{get} : S_0 \to S$ and $\mathrm{put} : S_0 \times P \to S_0$ (application of view updates to source states) such that

$$\mathrm{get}\,(\mathrm{put}\,(s_0, p)) = \mathrm{get}\, s_0 \downarrow p$$
$$s_0 = \mathrm{put}\,(s_0, \mathrm{o})$$
$$\mathrm{put}\,(\mathrm{put}\,(s_0, p), p') = \mathrm{put}\,(s_0, p \oplus p')$$



- In other words, an update lens is an action put of $(P, \mathrm{o}, \oplus)$ on S_0 together with an action morphism get from put to $\downarrow$.

- Update lenses for S are the same as coalgebras of the *coupdate comonad* for $(S, (P, \mathrm{o}, \oplus), \downarrow)$, defined by $D\,X =_{\mathrm{df}} S \times (P \Rightarrow X)$.
- Right actions of $(P, \mathrm{o}, \oplus)$ on S are in a bijective correspondence with distributive laws of the product comonad $D_0 X = S \times X$ and the exponent comonad $D_1 X = P \Rightarrow X$.
  Coupdate comonads are exactly the compatible compositions of the product and exponent comonads for S resp. $(P, \mathrm{o}, \oplus)$.
- This leads to a number of further characterizations of update lenses, e.g., as pairs of coalgebras, bialgebras (pairs of a coalgebra and an algebra) etc.

Directed containers

- A *directed container* is a set S (of states) and an S-indexed family P of sets (of updates enabled in each state) with
    - $\downarrow : \Pi s : S.\, P\, s \to S$ (application of updates to states),
    - $\mathsf{o} : \Pi\{s : S\}.\, P\, s$ (the trivial update),
    - $\oplus : \Pi\{s : S\}.\, \Pi p : P\, s.\, P\,(s \downarrow p) \to P\, s$ (composition of updates)

  such that

$$
\begin{aligned}
s \downarrow \mathsf{o} &= s \\
s \downarrow (p \oplus p') &= (s \downarrow p) \downarrow p' \\
p \oplus \mathsf{o} &= p \\
\mathsf{o} \oplus p &= p \\
(p \oplus p') \oplus p'' &= p \oplus (p' \oplus p'')
\end{aligned}
$$

- The data and laws are like for a monoid and an action, modulo the dependent typing.
- A directed container defines a comonad by $D\, X = \Sigma s : S.\, (P\, s \Rightarrow X)$.

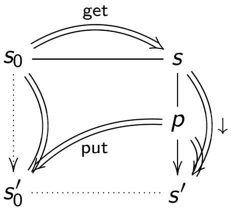
- A directed container is the same as a category.
  - The set of objects is S.
  - The set of maps with domain $s : S$ is $P\,s$.
  - The codomain of map $p : P\,s$ with domain $s : S$ is $s \downarrow p$.
  - The identity map on object s is $\mathsf{o}\,\{s\}$.
  - A map $p : P\,s$ with domain $s : S$ can only be composed with a map $p' : P\,(s \downarrow p)$ with domain $s \downarrow p : S$, which is the codomain of p.

- Morphisms between two directed containers are in a bijection with morphisms between the corresponding comonads.
- While a directed container is the same as a category, we will see that a directed container morphism is <u>not</u> a functor, but something entirely different (a "relative split pre-opcleavage").

Dependently typed update lenses

- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ be a directed container.
- An *dep. typed update lens* for $(S, P, \downarrow, \mathsf{o}, \oplus)$ is a set S_0 (of source states) with maps $\mathsf{get} : S_0 \to S$ and $\mathsf{put} : \Sigma s_0 : S_0.\ P\,(\mathsf{get}\,s_0) \to S_0$ (application of view updates to source states) such that

$$\mathsf{get}\,(\mathsf{put}\,(s_0, p)) = \mathsf{get}\,s_0 \downarrow p$$
$$s_0 = \mathsf{put}\,(s_0, \mathsf{o}\,\{\mathsf{get}\,s_0\})$$
$$\mathsf{put}\,(\mathsf{put}\,(s_0, p), p') = \mathsf{put}\,(s_0, p \oplus p')$$



(Note there is no difference from simply-typed update lenses apart from the dependent typing.)

- It should not come as a surprise that dependently typed update lenses for $(S, P, \downarrow, \mathrm{o}, \oplus)$ are the same as coalgebras of the corresponding comonad D, defined by $DX =_{\mathrm{df}} \Sigma s : S. (P\, s \Rightarrow X)$.

- State-based lenses for S cannot be cast as simply typed update lenses, but are the same as dep. typed update lenses for $(S, P, \downarrow, \mathrm{o}, \oplus)$ defined by

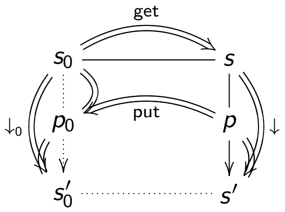
$$\begin{array}{rcll} P\, s & =_{\mathrm{df}} & S & \\ s \downarrow s' & =_{\mathrm{df}} & s' & \\ \mathrm{o}\,\{s\} & =_{\mathrm{df}} & s & \text{—not available in simply typed case} \\ s \oplus s' & =_{\mathrm{df}} & s' & \end{array}$$

- Neither in the case of simply typed nor dep. typed update lenses do we speak about about source updates; view updates apply to both view states and source states.
  This will change on next slide.

A generalization (yet unpublished, but obvious to me now)

- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ be a directed container (for view states, updates and update application).
- An *generalized update lens* for $(S, P, \downarrow, \mathsf{o}, \oplus)$ is another directed container $(S_0, P_0, \downarrow_0, \mathsf{o}_0, \oplus_0)$ (for source states, updates and update application) with maps $\mathsf{get} : S_0 \to S$ and $\mathsf{put} : \Sigma s_0 : S_0.\ P\,(\mathsf{get}\, s_0) \to P_0\, s$ such that

$$\begin{aligned} \mathsf{get}\,(s_0 \downarrow_0 \mathsf{put}\,(s_0, p)) &= \mathsf{get}\, s_0 \downarrow p \\ \mathsf{o}_0\{s_0\} &= \mathsf{put}\,(s_0, \mathsf{o}\,\{\mathsf{get}\, s_0\}) \\ \mathsf{put}\,(s_0, p) \oplus_0 \mathsf{put}\,(s_0 \downarrow_0 \mathsf{put}\,(s_0, p), p') &= \mathsf{put}\,(s_0, p \oplus p') \end{aligned}$$



- Update lenses in this generalized sense are the same as comonad morphisms between the comonads corresponding to the two directed containers.

Takeaway

- Update lenses have a lot of structure around them; this makes that they be characterized in many ways
- They are a realization of McKinna's "lenses as proof-relevant (bi)simulations" doctrine
- Update lenses are coalgebras of a coupdate comonad, generalized update lenses are morphisms between coupdate comonads.

Directed container papers

- Ahman, Chapman, Uustalu, When is a container a comonad? FoSSaCS '12 / LMCS (2014)
- Ahman, Uustalu, Distributive laws for directed containers, Progress in Inform. '13
- Ahman, Uustalu, Update monads: cointerpreting directed containers, TYPES '13 post-proc. (2014)
- Ahman, Uustalu, Coalgebraic update lenses, MFPS XXX (2014)
- Ahman, Uustalu, Directed containers as categories, MSFP '16