Coalgebraic Update Lenses

Danel Ahman[1] and Tarmo Uustalu[2]

[1] Laboratory for Foundations of Computer Science, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9LE, United Kingdom; d.ahman@ed.ac.uk
[2] Institute of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia; tarmo@cs.ioc.ee

O'Connor [6] made the simple but very useful observation with deep consequences that the (very well-behaved) lenses à la Foster et al. [3] are nothing but coalgebras of the array comonads of Power and Shkaravska [7].

The put operation in these lenses is quite rigid in that a whole new view is merged into the source, there is no flexibility for speaking about small changes to the view. We advocate a generalization that is as simple as O'Connor's, but offers also this flexibility. The idea is to introduce updates (or changes, deltas, edits) that can be composed and applied to views. The generalization derives from the work on directed containers of Ahman et al. [1].

A lens in our generalized sense—an update lens—is parameterized by a fixed set S (of views), a monoid $(P, \mathsf{o}, \oplus)$ (of updates) and an action $\downarrow$ of the monoid on the set (describing the outcome of applying any given update on any given view).

These data, sometimes collectively called an act, define a comonad (D, ε, δ) by $D\,X = S \times (P \to X)$.

We define an update lens to be a coalgebra of this comonad. This is the same as having a set X and maps $\mathsf{lkp} : X \to S$ and $\mathsf{upd} : X \times P \to X$ satisfying the conditions

$$\begin{aligned}\mathsf{upd}\,(x, \mathsf{o}) &= x\\ \mathsf{upd}\,(\mathsf{upd}\,(x, p), p') &= \mathsf{upd}\,(x, p \oplus p')\\ \mathsf{lkp}\,(\mathsf{upd}\,(x, p)) &= \mathsf{lkp}\,x \downarrow p\end{aligned}$$

To have an update lens turns out to be equivalent to having a functor R from $\langle\!\langle S, (P, \mathsf{o}, \oplus) \downarrow\rangle\!\rangle$ to **Set**. Here $\langle\!\langle S, (P, \mathsf{o}, \oplus) \downarrow\rangle\!\rangle$ is the category where an object is an element of S, a map between $s, s' : S$ is an element of p such that $s \downarrow p = s'$, the identity on an object s is o and the composition of two maps p, p' is $p \oplus p'$.

An act S, $(P, \mathsf{o}, \oplus)$, $\downarrow$ also defines a monad (T, η, μ) by $T\,X = S \to P \times X$ (a compatible combination of the reader monad for S and the writer monad for $(P, \mathsf{o}, \oplus)$) that we have elsewhere [2] called the update monad. The algebras of this monad and update lenses model resp. comodel the same Lawvere theory.

Ordinary lenses for S are canonically related to update lenses for the act $(S, (P, \mathsf{o}, \oplus), \downarrow)$ where $(P, \mathsf{o}, \oplus)$ is the free monoid on the "overwrite" semigroup structure on S.

The algebraic treatment of ordinary lenses by Johnson et al. [5], compared to O'Connor's coalgebraic account by Gibbons and Johnson [4], extends to update lenses. The action $\downarrow$ defines a lifting of the writer monad for $(P, \mathsf{o}, \oplus)$ to category $\mathbf{Set}/S$. An update lens is essentially the same as an algebra of this lifted monad.

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

References

[1] D. Ahman, J. Chapman, T. Uustalu. When is a container a comonad? *Log. Methods in Comput. Sci.*, to appear. Conference version in L. Birkedal, ed., *Proc. of 15th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS 2012 (Tallinn, March 2012)*, v. 7213 of *Lect. Notes in Comput. Sci.*, pp. 74–88. Springer, 2012.

[2] D. Ahman, T. Uustalu. Update monads: cointerpreting directed containers. In R. Matthes, A. Schubert, eds., *Proc. of 19th Conf. on Types for Proofs and Programs (Toulouse, Apr. 2014)*, *Leibniz Proc. in Informatics*, Schloss Dagstuhl, to appear.

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

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

[5] M. Johnson, R. Rosebrugh, R. J. Wood. Algebras and update strategies. *J. of Univ. Comput. Sci*, v. 16, n. 5, pp. 729748, 2010.

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

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