A Coalgebraic Approach to Bidirectional Transformations

Faris Abou-Saleh[1] and James McKinna[2]

[1] University of Oxford; `firstname.lastname@@cs.ox.ac.uk`
[2] University of Edinburgh; `firstname.lastname@@ed.ac.uk`

1 Introduction

Bidirectional transformations (bx) are a diverse collection of formalisms for maintaining consistency between two or more related data models, such as (a)symmetric lenses [2] and algebraic bx [3]. In a previous paper [1] we proposed structures called set-bx as a unified framework for studying these formalisms. The main insight was that a bx between data sources A, B could be represented by get and set operations on both A and B – such as $\mathsf{get}_A : MA$ and $\mathsf{set}_A : A \to M()$, for some monad M – satisfying four 'get-set' laws. Crucially, updates to A may affect B, and vice versa; the operations on A and B will not commute in general, so that the states are *entangled*.

There are two important, related issues such a framework needs to address. The first is how to compose two bx's $x : A \Leftrightarrow B$ and $y : B \Leftrightarrow C$, giving a bx $(x \cdot y) : A \Leftrightarrow C$. The second is that this composition is often only associative, and has identities, up to some notion of equivalence of bx which must be identified. We hope to describe partial answers to these questions in the context of set-bx in a companion paper. However, the state monads we often use to describe set-bx, and the corresponding composition and equivalence, also have an interesting interpretation in terms of coalgebras, which we illustrate in this extended abstract.

2 Coalgebraic set-bx

Definition 1. *A set-bx* $h : A \Leftrightarrow B$ *between* A *and* B *with respect to a monad* M *consists of the following operations, satisfying the 'get-set' laws given in [1].*

$$h.get_A : MA \qquad h.set_A : A \to M() \qquad h.get_B : MB \qquad h.set_B : B \to M()$$

In the case that MX is the *side-effect monad* for some state S, functions $f : X \to MY = (Y \times S)^S$ are equivalent to functions $f' : S \to (Y \times S)^X$ by (un)currying. This turns set-bx operations into functions resembling coalgebra structure:

$$h.\mathsf{get}'_A : S \to (A \times S) \quad h.\mathsf{set}'_A : S \to (() \times S)^A \quad \text{(and similarly for } h.\mathsf{get}'_B, h.\mathsf{set}'_B)$$

Simplifying the types of $h.\mathsf{get}'_A, h.\mathsf{get}'_B$ into $S \to A$ and $S \to B$ and removing units $()$, the product of these functions $f := \langle h.\mathsf{get}'_A, h.\mathsf{set}'_A, h.\mathsf{get}'_B, h.\mathsf{set}'_B \rangle$ is the structure of a coalgebra for a suitable functor:

Definition 2. *A* coalgebraic set-bx $\alpha : A \Leftrightarrow B$ *is an* F_{AB} *coalgebra* $\alpha = (S, f : S \to F_{AB}S)$ *where* $F_{AB}S = A \times S^A \times B \times S^B$*, such that the components of* f *(suitably expressed as* $\alpha.\mathsf{get}_A, \ldots, \alpha.\mathsf{set}_B$*) satisfy the 'get-set' laws.*

We now show how to compose two coalgebraic set-bx, writing e.g. $x.\mathsf{get}'_A$ for $\alpha.\mathsf{get}'_A\ x$, and $y.\mathsf{set}'_B\ b$ for $\beta.\mathsf{set}'_B\ y\ b$ (and similarly for γ below).

Definition 3. *Given coalgebraic set-bx* $\alpha : A \Leftrightarrow B$ *and* $\beta : B \Leftrightarrow C$ *with carriers* X, Y *respectively, the composition* $\gamma = (\alpha \cdot \beta) : A \Leftrightarrow C$ *has carrier* Z *given by the pullback of* $f := \alpha.\mathsf{get}'_B : X \to B$ *and* $g := \beta.\mathsf{get}'_B : Y \to B$ *– i.e. (in* Set*),* $Z = \{(x, y) \in (X \times Y) : f(x) = g(y)\}$ *– and structure corresponding to the following operations (*get'_C*,* set'_C *are similar).*

$$(x, y).\mathsf{get}'_A = x.\mathsf{get}'_A \qquad (x, y).\mathsf{set}'_A a = \mathsf{let}\ x' \Leftarrow x.\mathsf{set}'_A a\ \mathsf{in}\ (x', y.\mathsf{set}'_B(x'.\mathsf{get}'_B))$$

Theorem 1. *The composition* $(\alpha \cdot \beta)$ *is a well-defined coalgebraic set-bx. Moreover, coalgebraic set-bx-composition is associative, and has identities, up to equivalence given by coalgebraic bisimulation.*

3 Applications and Future Work

When applied to the set-bx given in [1], our coalgebraic definitions capture standard notions of composition for asymmetric and symmetric lenses, and algebraic bx. In particular, there is a close relationship between symmetric lens composition and equivalence [2], and their coalgebraic counterparts. Rather than relying on ad-hoc definitions of behavioural equivalence and composition for each class of bx, our definition allows standard concepts of coalgebra theory, such as bisimulations and final coalgebras, to be used for reasoning about bx.

Another advantage is that our definitions readily generalise to wider class of bx than previously considered, such as the I/O example in [1]. These bx are allowed to introduce effects, such as non-determinism or I/O, given by some monad N; here, they correspond to coalgebras for the functor $G_{AB}S = NA \times (NS)^A \times NB \times (NS)^B$, which compose in a similar manner to the above definition. Moreover, coalgebraic bisimulation yields a natural notion of behavioural equivalence which may be used for studying these structures.

References

1. Faris Abou-Saleh, James Cheney, Jeremy Gibbons, James McKinna, and Perdita Stevens. Entangled state monads. *Third International Workshop on Bidirectional Transformations (BX 2014), Athens*, 2014.
2. Martin Hofmann, Benjamin C. Pierce, and Daniel Wagner. Symmetric lenses. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Austin, Texas*, January 2011.
3. Perdita Stevens. Bidirectional model transformations in QVT: Semantic issues and open questions. *Journal of Software and Systems Modeling (SoSyM)*, 9(1):7–20, 2010.