Bidirectional Transformations are Proof-Relevant Bisimulations (Extended Abstract)

James McKinna[‡]

[‡]School of Informatics, University of Edinburgh

[‡]firstname.lastname@ed.ac.uk

Abstract

Bidirectional transformations (bx) are a diverse collection of formalisms for maintaining consistency between related data models. While much existing work has described extensional, state-based formalisms, in recent years attention has turned to incorporating intensional data about edits (based on monoid actions), or more generally, deltas (based on categories), describing model updates.

We develop a *proof-relevant* interpretation of such bx, and indicate how to organise their underlying data into a bicategory; none other than that of proof-relevant *bisimulations* between model spaces, with transitions given by updates. Composition of bx is given by a tensor product construction (not previously known to us from the bisimulation literature), inducing a forgetful homomorphism to the underlying bicategory of model spaces and consistency relations. Well-known properties of bx such as hippocraticness ('GetPut' for lenses) or overwriteability ('PutPut') give rise to *full* sub-bicategories; that is, such bx are closed under composition.

Throughout we exploit the interplay between bicategorical and type-theoretic structure: we thus obtain a characterisation of the *alignment problem* in type-theoretic terms, as well as a number of suggestive directions for future investigation.

Keywords bidirectional transformation, lenses, consistency maintenance, dependent types, proof-relevance, bisimulation relation, bicategories

Bidirectional transformations (Bx)

Bx are a diverse collection of formalisms for maintaining consistency between related data models: in databases (as solutions to the so-called *view-update* problem [3]); in HCI (capturing the basic dynamics of MVC-style interfaces [14]); in programming languages via the richly-varied notion of *lens* [1,4,7–10]; and in model-driven development (MDD) of software, via the use of triple-graph grammar (TGG) formalisms and tools [16], and the OMG standard QVT-R [15], which supports a rich specification language for consistency relations between models (in *checkonly* mode), as well as *forward* and *backward* modes for consistency restoration. We start from Stevens' (symmetric) relational account of bx [17], previously introduced as *consistency maintainers* by Meertens [14]:

DEFINITION 1. *A* ***bidirectional transformation*** *is specified by:*

- *a* ***consistency relation*** $R \subseteq A \times B$
- *forward and backward consistency* ***restorers****,*
  $\triangleright_R \ : A \times B \longrightarrow B \qquad \triangleleft_R \ : A \times B \longrightarrow A$
- *such that:*
  - ***correctness****:* $a\,R\,(a \triangleright_R b)$, $(a \triangleleft_R b)\,R\,b$
    i.e. consistency is indeed restored
  - ***hippocraticness****:* $a\,R\,b \Longrightarrow a \triangleright_R b = b, \quad a \triangleleft_R b = a,$
    i.e. if already consistent, do nothing!

Proof-relevance

The basic picture above leaves implicit the underlying notion of *update*, relying on a scenario in which inputs to $\triangleright_R$ or $\triangleleft_R$ reflect an updated value in the A, respectively B, argument. We now consider *proof-relevant* interpretations, via the now-familiar identification of propositions-as-types in dependent type theory: both updates *and* (proofs of) consistency are represented by families of types, with:

- $\partial^A_{a\,a'}$, representing those updates (edits) δ which transform $a : A$ into $a' : A$, symbolically $\delta : a \mapsto a'$; similarly for $\partial^B_{b\,b'}$; for classical state-based formalisms such as asymmetric lenses [8], one may take $\partial^A_{a\,a'}$ to be the trivial singleton family, inhabited everywhere by a dummy witness to each state update $a \mapsto a'$;
- $T_{a\,b}$, representing witnesses t to the proposition '$a R b$'; such families T may be seen as generalising the notion of *lens complement* [13] (itself generalising view-update "with constant complement" [3]; for lenses, the consistency relation is implicit, but recoverable as the *graph relation* of the *get* operation [8]).

This set-up is interpretable in the bicategory [5] **Rel** of *relations*: in type theory, with 0-cells given by types A, B, 1-cells given by relations T, identity and composition given by the *identity* type, and relational composition $a\,(R \otimes S)\,c =_{\text{def}} \Sigma_{b:B} R_{a\,b} \times S_{b\,c}$ with 2-cells the *proof-relevant* inclusions $R \subseteq_p S =_{\text{def}} p : \Pi_{a:A\,b:B}\, a\,R\,b \Rightarrow a\,S\,b$.

Bx are bisimulations

Now, forward consistency restoration specifies that: given a, b T-consistent, and an A-update $\delta : a \mapsto a'$, there should exist a B-update $\delta' : b \mapsto b'$ (vice versa in the backward direction). Observing that proof-relevant inclusions serve as constructive witnesses to such existence, we have (with ∂^{A° the *opposite* relation to ∂^A):

$$\partial^{A^\circ} \otimes T \subseteq_{\triangleright_T} T \otimes \partial^{B^\circ} \qquad \partial^{B^\circ} \otimes T \subseteq_{\triangleleft_T} T \otimes \partial^{A^\circ}$$

These two inclusions encode algebraically the usual diagrammatic properties defining a *bisimulation* between two labelled transition systems, thus (very compactly!) justifying the claim of the title, that the forward and backward transformations witness T as a proof-relevant bisimulation between the model spaces A, B, with updates ∂^A, ∂^B defining the transitions between model states. We write: $\mathbb{T} =_{\text{def}} (T, \triangleright_T, \triangleleft_T) : \mathbb{A} \rightleftharpoons_T \mathbb{B}$, where $\mathbb{A} =_{\text{def}} (A, \partial^A), \mathbb{B} =_{\text{def}} (B, \partial^B)$.

Why Bicategories?

The above structure yields a *bicategory* **Bisim**, with 0-cells given by the model spaces $\mathbb{A}$, 1-cells given by bx, as proof-relevant bisimulations $\mathbb{T}{:}\mathbb{A} \rightleftharpoons_T \mathbb{B}$, and 2-cells given by proof-relevant *equivalences* $T \subseteq_p T' \subseteq_q T$ between consistency relations T, T'.

There is a (forgetful) homomorphism of bicategories between **Bisim**, and **Rel**: on 0-cells, it forgets the update structure, on 1-cells, it maps a bx $\mathbb{T}$ to its underlying consistency relation T, and on 2-cells, an equivalence (p, q) maps to p.

This homomorphism depends on a definition of composition $(\otimes)$ between bx/bisimulations which is new, as far as we are aware (but which generalises existing definitions of lens composition):

$$\mathbb{S} \otimes \mathbb{T} =_{\text{def}} (S \otimes T, \triangleright_S \otimes \triangleright_T, \triangleleft_T \otimes \triangleleft_S : \mathbb{A} \rightleftharpoons_{S \otimes T} \mathbb{C})$$

with $(\triangleright_S \otimes \triangleright_T)(a', \delta_a, (b, s, t)) = (c', \delta_c, (b', s', t'))$ where $(b', \delta_b, s') =_{\text{def}} \triangleright_S\ (a', \delta_a, s)$ and $(c', \delta_c, t') =_{\text{def}} \triangleright_T\ (b', \delta_b, t)$.

Additional properties

Familiar bx properties give rise to *full* sub-bicategories of **Bisim**; that is, such bx are closed under composition $\otimes$, and correspond to having additional structure on model spaces, and to strong, intensional constraints on the interaction between consistency restoration and such structure. Hippocracticness, as in Def 1, analogous to ‘GetPut’ for lenses [8], corresponds to model spaces having the structure of *reflexive graphs*, and to consistency restoration preserving such structure on-the-nose. Passing to model spaces as fully-fledged *categories* allows us to consider *overwriteability* for bx [17], analogous to ‘PutPut’ for lenses [8].

Dependently-typed programming languages such as Agda or Idris offer a natural home for such proof-relevant constructions, with dependent types as strong, machine-checkable, correctness specifications: we can, for example, give a type-theoretic characterisation of the *alignment* problem in the bx literature [4, 7], exhibiting its type as that of a (heuristic) *search* problem: to (forward) *align* $a' : A$ with $b : B$ is to compute an inhabitant of $\Sigma_{a:A} \partial^A_{a\,a'} \times T_{a\,b}$.

Related work

Space prevents exhibiting every bx definition in the literature as a 1-cell $\mathbb{A} \rightleftharpoons_T \mathbb{B}$ in **Bisim**. Our generalisation shares some of the same underlying machinery as Diskin et al.’s *symmetric delta lenses* [7], where model spaces are given as categories, but much of the bicategorical structure of **Bisim**, and relationships to other settings which we describe here, is new to us. The reflexive graph structure necessary for hippocraticness has close connections to that explored by Cai et al. in the theory of static differentation of functions [6].

Conclusions and Future Work

Working both type-theoretically, and (bi-)categorically, has thus established, we hope, a unifying mathematical basis for all existing bx formalisms, as well as a starting point for comparison with existing work in the related area of version control systems and patch theory [2, 18], where type-theoretic ideas have also proved fruitful. We further conjecture that interpretations of our constructions in other (enriched) categorical settings may shed light on (geo-)metric accounts of consistency restoration, in terms of a ‘differential geometry of consistency restoration’ [12].

Acknowledgements This is an extended abstract of a full paper to be submitted elsewhere, based on joint work on the Edinburgh/Oxford EPSRC-funded project *A Theory of Least Change for Bidirectional Transformations* (EP/K020218/1 and EP/K020919/1). I am grateful to Perdita Stevens, James Cheney at Edinburgh, and Jeremy Gibbons and Faris Abou-Saleh at Oxford for discussion of these matters; to the audience and reviewers of the 2016 ETAPS Bx workshop, for their feedback on [12, 13]; similarly, on earlier versions of these ideas, to the Scottish Programming Languages Seminar, DReaM group, and LFCS Lab Lunch audiences; and to the BIRS Seminar 13w5115 for the opportunity to sketch their first blackboard incarnation [11]. Most recently, the 74th IFIP Working Group 2.1 meeting provided a very stimulating and challenging environment in which to sharpen my exposition, and my thinking.

References

[1] D. Ahman and T. Uustalu. Coalgebraic update lenses. *ENTCS*, 308:25–48, 2014.

[2] C. Angiuli, E. Morehouse, D. R. Licata, and R. Harper. Homotopical patch theory. In *ICFP*, pp243–256. ACM, 2014.

[3] F. Bancilhon and N. Spyratos. Update semantics of relational views. *TODS*, 6(4):557–575, 1981.

[4] D. M. Barbosa, J. Cretin, N. Foster, M. Greenberg, and B. C. Pierce. Matching lenses: Alignment and view update. In *ICFP*, pp193–204. ACM, 2010.

[5] J. Bénabou. *Reports of the Midwest Category Seminar*, volume 47 of *LNM*, chapter Introduction to bicategories, pp1–77. Springer, 1967.

[6] Y. Cai, P. G. Giarrusso, T. Rendel, and K. Ostermann. A theory of changes for higher-order languages: Incrementalizing λ-calculi by static differentiation. In *PLDI*, pp145–155. ACM, 2014.

[7] Z. Diskin, Y. Xiong, K. Czarnecki, H. Ehrig, F. Hermann, and F. Orejas. From state- to delta-based bidirectional model transformations: the symmetric case. In *MODELS*, *LNCS 6981*. Springer, 2011.

[8] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt. Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. *TOPLAS*, 29(3):17, 2007.

[9] M. Hofmann, B. C. Pierce, and D. Wagner. Symmetric lenses. In *POPL*, pp371–384. ACM, 2011.

[10] M. Hofmann, B. C. Pierce, and D. Wagner. Edit lenses. In *POPL*, pp495–508. ACM, 2012.

[11] J. McKinna. Type theory and algebraic bx. Talk at Banff BIRS Seminar 13w5115, 2013. Video at: `http://videos.birs.ca/2013/13w5115/201312040919-McKinna.mp4`.

[12] J. McKinna. Bidirectional transformations with deltas: A dependently typed approach (talk proposal). In *Bx Workshop, ETAPS*, 2016. `http://ceur-ws.org/Vol-1571/paper_11.pdf`.

[13] J. McKinna. Complements witness consistency. In *Bx Workshop, ETAPS*, 2016. `http://ceur-ws.org/Vol-1571/paper_10.pdf`.

[14] L. Meertens. Designing constraint maintainers for user interaction. Unpublished manuscript, available from `http://www.kestrel.edu/home/people/meertens/`, June 1998.

[15] OMG. MOF2.0 query/view/transformation (QVT) version 1.2. OMG document formal/2015-02-01, 2015. available from www.omg.org.

[16] A. Schürr and F. Klar. 15 years of triple graph grammars. In *ICGT*, *LNCS 5214*, pp411–425. Springer, 2008.

[17] P. Stevens. Bidirectional model transformations in QVT: Semantic issues and open questions. In *MODELS*, *LNCS 4735*, pp1–15. Springer, 2007.

[18] W. Swierstra and A. Löh. The semantics of version control. In *Onward!*, pp43–54, 2014.