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 hippocraticism to the underlying bicategory of model spaces and consistency from the bisimulation literature), inducing a forgetful homomorphism to the underlying bicategory of model spaces and consistency relations.

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, \( \triangleright R : A \times B \rightarrow B \) \( \triangleleft R : A \times B \rightarrow A \)
- such that:
  - correctness: \( a R (a \triangleright R b) \), \( a \triangleleft R b \) \( Rb \)
  - i.e. consistency is indeed restored
  - hippocracy: \( a R b \implies a \triangleleft R b = b \), \( a \triangleright R b = a \)
  - 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:

- \( \delta_a^A \), representing those updates (edits) \( \delta \) which transform \( a : A \) into \( a' : A \); symbolically \( \delta : a \rightarrow a' \); similarly for \( \delta_b^B \); for classical state-based formalisms as asymmetric lenses [8], one may take \( \delta_a^A \) to be the trivial singleton family, inhabited everywhere by a dummy witness to each state update \( a \mapsto a' \);

- \( T_{ab} \), representing witnesses \( t \) to the proposition ‘\( aRb \)’; 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 \( a (R \otimes S) c = \langle \sum p b R(a b), S(p b) \rangle \) with 2-cells the proof-relevant inclusions \( R \subseteq_p S = \sum p : \Pi_{a b . a R b \Rightarrow a S b} \).

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 \( \delta^A \) the opposite relation to \( \delta^A' \)):

\[
\delta^A \otimes T \subseteq_{\delta^A} T \otimes \delta^A \quad \delta^R \otimes T \subseteq_{\delta^R} T \otimes \delta^R
\]

These two inclusions encode algebraically the usual diagrammatic properties defining a bisimulation between two labelled transition systems.
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 $\partial^A$, $\partial^B$ defining the transitions between model states. We write:

$$\text{def} (\langle T, \triangleright_T, \triangleright_T \rangle : A \rightarrow B, T)$$

Why Bicategories?

The above structure yields a bicategory Bisim, with 0-cells given by
the model spaces $A$, 1-cells given by bx, as proof-relevant bisimulations $T : A \rightarrow B$, and 2-cells given by proof-relevant equiva-
cences $T \subseteq \subseteq T$ 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 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):

$$S \otimes T = \text{def} (S \otimes T, \triangleright_S \otimes \triangleright_T, \triangleright_S \otimes \triangleright_T : A \rightarrow B, S \otimes T)$$

with $(\triangleright_S \otimes \triangleright_T (a', \delta_{a'}, (h, s, t))) = (c', \delta_{a'}, (h', s', t'))$ where

$$b' = \text{def} \triangleright_S (a', \delta_{a'}, s)$$

and

$$c' \delta_{a'} t' = \text{def} \triangleright_T (b', \delta_{b'}, t).$$

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, inten-
sional constraints on the interaction between consistency restaura-
tion and such structure. Hippocrasitcness, as in Def 1, analogous to
‘GetPut’ for lenses [8], corresponds to model spaces having the
structure of reflexive graphs, and to consistency restoration pre-
serving 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 charac-
terisation of the alignment problem in the bx literature [4, 7], ex-
hibiting its type as that of a (heuristic) search problem: to (forward)
align $a : A$ with $b : B$ is to compute an inhabitant of $\Sigma_w A, \partial^A \times T_{ab}$.  

Related work

Space prevents exhibiting every bx definition in the literature as a
1-cell $A \rightarrow B$ 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 bicate-
gorical structure of Bisim, and relationships to other settings which
we describe here, is new to us. The reflexive graph structure neces-
sary for hippocrasitcness has close connections to that explored by
Cai et al. in the theory of static differentiation of functions [6].

Conclusions and Future Work

Working both type-theoretically, and (bi-)categorically, has thus es-
tablished, we hope, a unifying mathematical basis for all existing
bx formalisms, as well as a starting point for comparison with exis-
ting 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 geo-
metry of consistency restoration’ [12].

Acknowledgements

This is an extended abstract of a full paper to be submitted elsewhere, based on joint work on the Edin-
burgh/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

[7] Z. Diskin, Y. Xiong, K. Czarnecki, H. Ehrig, F. Hermann, and F. Ore-
edu/home/people/meertens/, June 1998.