Electronic Communications of the EASST
Volume 49 (2012)

Proceedings of the
First International Workshop on
Bidirectional Transformations
(BX 2012)

Observations relating to the equivalences induced on model sets by bidirectional transformations

Perdita Stevens

16 pages

Guest Editors: Frank Hermann, Janis Voigtländer
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Observations relating to the equivalences induced on model sets by bidirectional transformations

Perdita Stevens

Laboratory for Foundations of Computer Science, School of Informatics, University of Edinburgh

Abstract: A bidirectional transformation on a pair of sets of models induces two principal equivalence relations on each set of models. Since a model can be uniquely identified by specifying its equivalence class in each of these relations, they function as a coordinate system for the model sets, with respect to the transformation. We prove some results relating to this observation. Using them we give the implication relationships between various properties of bidirectional transformations. In particular, we characterise the bidirectional transformations that can be decomposed into a pair of lenses working "tail to tail".

Keywords: bidirectional transformation

1 Introduction

The study of bidirectional transformations between model sets is, so far, hampered by lack of sufficiently deep understanding of the structures involved. This paper aims to be one step in the right direction.

A recurring informal theme in work on bidirectional transformations is the idea of looking at different models "through glasses" which make them indistinguishable; for example, if we look at two correct implementations of the same abstract model through glasses which only care about what is relevant to the abstract model, they should not be distinguished. This idea is formalised using equivalence relations on the sets of models which are related by the transformation.

This paper gives a collection of results concerning them, and goes on to exploit these to deduce (non-)implications between various "niceness" conditions on bidirectional transformations. We also give examples: an informal, MDD-realistic one, and some tiny, formal ones to illustrate points and serve as counter-examples to over-optimistic conjectures.

The paper is structured as follows. In Section 2 we introduce notation and terminology, and give a our first couple of examples. Section 3 introduces the main focus of the paper, defining a pair of equivalence relations on each model set related by a bidirectional transformation, giving examples and basic properties. Section 4 exploits them to prove relationships between the various "niceness" conditions on bidirectional transformations, summarised in Figure 1. Especially, we show that a condition called *simply matching*, defined in terms of our equivalences, is achieved precisely when a bidirectional transformation can be re-expressed as a pair of lenses connected "tail to tail". Section 5 discusses related work, and Section 6 concludes.

2 Background

Here we recapitulate definitions found for example in [Ste10].

For purposes of this paper a bidirectional transformation $R : M \leftrightarrow N$ between non-empty sets M and N is given by specifying a *consistency relation*, also by slight abuse of notation called $R \subseteq M \times N$, together a pair of functions

$$\overrightarrow{R} : M \times N \to N$$

$$\overleftarrow{R} : M \times N \to M$$

whose task is to enforce consistency.

We use the term "model" for the structures related by a bidirectional transformation because our motivation is from model-driven development, but this does not in fact entail any intention to restrict what the structures can be: throughout this paper, M and N can be any sets, finite or infinite, including uncountable.

We will be concerned only with bidirectional transformations that only change a model when consistency requires it and, in that case, do enforce consistency: that is, those that are correct and hippocratic.

Definition 1 $R : M \leftrightarrow N$ is correct if

$$\forall m \in M \ \forall n \in N \qquad R(m, \overrightarrow{R}(m,n))$$

$$\forall m \in M \ \forall n \in N \qquad R(\overleftarrow{R}(m,n), n)$$

Definition 2 $R : M \leftrightarrow N$ is hippocratic if for all $m \in M$ and $n \in N$, we have

$$R(m,n) \quad \Rightarrow \quad \overrightarrow{R}(m,n) = n$$

$$R(m,n) \quad \Rightarrow \quad \overleftarrow{R}(m,n) = m$$

From now on, all bidirectional transformations will be assumed to be correct and hippocratic. Please note also that many statements we shall make have exactly dual versions with dual proofs, replacing $\overrightarrow{R}$ with $\overleftarrow{R}$ etc. Generally we state both versions, but prove only one.

Technically we need not include the consistency relation as part of the definition of a correct and hippocratic transformation, because we have immediately from the definitions

Lemma 1 $R(m,n) \Leftrightarrow \overrightarrow{R}(m,n) = n \Leftrightarrow \overleftarrow{R}(m,n) = m.$

However, the consistency relation is for software engineering purposes the primary part of the transformation definition, so it is convenient to include it.

In order to help the reader's intuition we will use the following, informal, running example.

Example 1 *Let the elements of model set M be UML models. Let each element of model set N be a test suite, given as a set of test classes. Suppose the development team wishes to count a UML model m as being consistent with a test suite n iff the following two conditions hold:*

- *for every class in model m, say with name* `Foo`[1]*, there is at least one test class in n, with the naming convention that each such test class is called* `TestFoo`X *for some natural number X;*
- *for every string* `Name` *such that there is at least one test class in n with a name of the form* `TestName`X *for an integer X (there could be several different values of X for the same* `Name`*), there is a class called* `Name` *in m.*[2]

We write $T(m,n)$ if this holds. Notice that each model typically contains a lot of information not included in the other. There is a choice of ways to restore consistency, but one possibility is:

- *$\overrightarrow{T}(m,n)$ deletes any test classes from n whose names are of the form* `TestName`X *but for which there is no class* `Name` *in m. For any class (say called* `Name`*) in m not having any corresponding test class in n, it adds one "skeleton" test class* `TestName0`*.*
- *$\overleftarrow{T}(m,n)$ deletes from m any class for which n contains no test class (along with any other model elements that must be deleted, such as associations to deleted classes). For any test class in n whose name is of the form* `TestName`X *but for which there is no class* `Name` *in m, it adds a new class* `Name` *to m (in a suitable "default" state, e.g., not related to any other class, and having no attributes or operations).*

Appropriately implemented, this bidirectional transformation could (and should) be correct and hippocratic.

A well-known problem is that these definitions alone are too weak to enforce "sensible" behaviour. Unfortunately the most natural extra conditions to impose have the disadvantage that they seem to be too strong in practice. One well-known possibility, here using terminology coined in [Dis08], is:

Definition 3 $R : M \leftrightarrow N$ is *history ignorant* if for all $m,m' \in M$ and $n,n' \in N$, we have

$$\overrightarrow{R}(m, \overrightarrow{R}(m',n)) = \overrightarrow{R}(m,n)$$

$$\overleftarrow{R}(\overleftarrow{R}(m,n'),n) = \overleftarrow{R}(m,n)$$

The terminology for special properties of bidirectional transformations is mildly problematic, partly because different perspectives make different names seem natural. In earlier versions of this paper, what we now call history ignorance was termed strong undoability. This was natural to

[1] For simplicity, in the example we assume that UML class names are strings containing only characters from [A-Za-z], and in particular that there is only one package!
[2] Note that there could be other test classes in n whose names were not of this form, e.g. called `BasicSetUpTest`, `Performance`; these are deemed to be irrelevant for purposes of determining consistency with a given UML model.

us because the notion called *undoability* in [Ste10] demanded the same conditions as above, but only for $m \in M$ and $n \in N$ satisfying $R(m,n)$. That notion makes the bidirectional transformation undoable in the following sense: suppose that (m,n) is a consistent pair of models, and suppose the developer of m modifies it to m', propagates the modification by applying $\overrightarrow{R}$, and then wishes to undo the modification, returning to m. To say that the transformation is undoable is to say that now, applying $\overrightarrow{R}$ will return the situation to exactly what it was in the beginning, the pair (m,n); mistakes can be undone – provided that the initial pair of models was consistent. History ignorance is a stronger form of undoability, in that it demands the same conditions for more model pairs. Indeed, the condition does capture a notion of undoability that is stronger, in the following sense. Suppose our indecisive developer starts from any pair of models (m,n), not necessarily consistent, and behaves as before. The final pair of models, (m,n') say, may of course not be identical to the initial pair (m,n), since $\overrightarrow{R}$ always ensures consistency and the original state may not have been consistent. Still, the developer's mistaken modification to m has no effect in the final situation, in the sense that n' is the very same model that would have been the result of $\overrightarrow{R}(m,n)$; the mistake has been undone, even though the initial state of the pair of models might not have been consistent.

An alternative perspective, though, is that a history ignorant bidirectional transformation allows the developer to be safely ignorant of the history of when modifications to their model have been propagated to the other side: for example, if a developer starts with m' and modifies it to m, it does not matter whether $\overrightarrow{R}$ was ever applied to m'; provided that it is applied to m at the end, the effect on the other side's model will be the same regardless. We use the term history ignorance here for consistency with [Dis08, XSHT11].

Whatever our terminology, the special situation that we have when a bidirectional transformation is history ignorant would clearly be hugely advantageous to the usability of a bidirectional-transformation enabled tool, because the tool could reasonably propagate changes as and when it was convenient to do so without needing the developer's permission. It is unfortunate that realistic bidirectional transformations do not have this property. As we shall see in Section 4, the essential reason why they do not is that the information that is relevant to consistency is usually interdependent with the rest of the information in a model, whereas history ignorance requires the consistency-relevant information to be independent of the rest.

History ignorance is also algebraically a more natural condition than undoability, as we shall see. That it is indeed stronger than undoability is illustrated by the following (minimal) example, this one given formally.

Example 2 *Let* $M = \{a,b,c\} \times \{\mathsf{false}, \mathsf{true}\}$ *and let* $N = \{a,b,c\}$. *For all* $x,y \in \{a,b,c\}$, $\phi \in \{\mathsf{false}, \mathsf{true}\}$, *let:* $R((x,\phi),y)$ *hold iff* $x = y$; $\overrightarrow{R}((x,\phi),y) = x$;

$$\begin{aligned} \overleftarrow{R}((x,\phi),y) &= (y,\neg\phi) && \text{if (as sets) } \{x,y\} = \{a,c\} \\ &= (y,\phi) && \text{otherwise.} \end{aligned}$$

Then R is correct, hippocratic and undoable, but not history ignorant. A counterexample to history ignorance is that $\overleftarrow{R}(\overleftarrow{R}((a,\mathsf{true}),c),b) = \overleftarrow{R}((c,\mathsf{false}),b) = (b,\mathsf{false})$ *whereas* $\overleftarrow{R}((a,\mathsf{true}),b) = (b,\mathsf{true})$. *However, it is easy to check that there is no counterexample to undoability.*

Example 3 *The example begun in Example 1 is not even undoable. Suppose we start with a UML model m containing class* `Foo` *and a test suite n which has a test class* `TestFoo1` *containing some actual test code, and also consistent in every other way so that $T(m,n)$. We modify m in a way which involves deleting* `Foo`*, giving m', and apply $\overrightarrow{T}$. This results in test class* `TestFoo1` *being deleted, and the code which it contained being lost. If we now edit m' back to its original state m, and again apply $\overrightarrow{T}$, the resulting test suite will include a skeleton test class for* `Foo`*, called* `TestFoo0`*, but the lost test code is not restored. That is, $\overrightarrow{T}(m, \overrightarrow{T}(m',n)) \neq n$.*

3 Coordinate grid induced by a bidirectional transformation

Suppose we are given a bidirectional transformation R which is correct and hippocratic (but not necessarily undoable).

Recall the equivalences from [Ste08] (but presented here with slightly different symbols for better readability), defined in terms of R.

Definition 4 The equivalence relations $\sim_F^M$ and $\sim_B^M$ on M, and $\sim_F^N$ and $\sim_B^N$ on N, are defined as follows:

- $m \sim_F^M m' \Leftrightarrow \forall n \in N. \overrightarrow{R}(m,n) = \overrightarrow{R}(m',n)$
- $m \sim_B^M m' \Leftrightarrow \forall n \in N. \overleftarrow{R}(m,n) = \overleftarrow{R}(m',n)$

and dually,

- $n \sim_F^N n' \Leftrightarrow \forall m \in M. \overrightarrow{R}(m,n) = \overrightarrow{R}(m,n')$
- $n \sim_B^N n' \Leftrightarrow \forall m \in M. \overleftarrow{R}(m,n) = \overleftarrow{R}(m,n')$

Notice that it is immediate that these relations are indeed equivalence relations.

Intuitively, $m \sim_F^M m'$ says "m and m' do not differ in any way that is visible on the N side". The reader familiar with lenses will recognise that this generalises $\sim_g$. On the other hand, $m \sim_B^M m'$ says that "the *only* differences between m and m' are those visible on the N side, so that they become indistinguishable after any synchronisation with an element of N". The reader familiar with [BFP+08] will recognise that this generalises $\sim_{\max}$, the *coarsest* equivalence with respect to which a lens is quasi-oblivious. In an even more specialised setting, that of constant complement view updates, it generalises the equivalence defined in Theorem 7.2 of [BS81]. We can import the latter into the symmetric setting and demonstrate the relationship as follows.

Definition 5 The relation $\equiv$ on M is defined as follows:

$$m \equiv m' \Leftrightarrow \exists n \in N. \overleftarrow{R}(m,n) = m'$$

In general, this is not an equivalence relation, even though for consistency with [BS81] we have used the $\equiv$ symbol for it. For history ignorant bidirectional transformations R, however, we see that it is an equivalence relation because it coincides with $\sim_B^M$:

Proposition 1 *Let R be a history ignorant bidirectional transformation. Then $m \sim_B^M m'$ iff $m \equiv m'$.*

We postpone the proof until Section 4, by which point we will have introduced some consequences of our definitions which make it easier.

We will generally suppress the superscripts and just write $\sim_F$, $\sim_B$.

Example 4 *Returning to the informal example of Examples 1, 3, we see that:*

- *$m \sim_F^M m'$ iff the set of names of classes in m is the same as the set of names of classes in m';*
- *$m \sim_B^M m'$ iff m and m' are identical apart from (perhaps) containing different classes in the default state, that is, if one can be edited into the other just by adding and/or deleting default state classes.*
- *$n \sim_F^N n'$ iff n and n' are identical apart from (perhaps) differing in the set of `Names` for which they include a "skeleton" test class called `TestName0`. (In particular, if n contained a skeleton `TestName0`, n' could be equivalent even if it did not contain any test class `TestName0`. However, if n's `TestName0` had been filled in with some test code, then in order to be equivalent, n' would have to have a `TestName0` with identical code.)*
- *$n \sim_B^N n'$ iff the set of class names derived by taking class name `Name` for every test class in n whose name is of the form `TestName`X for some integer X (ignoring any test classes with names not of that form) is the same as the set of names derived from n' using the same procedure.*

Very informally, the fact that $\sim_B^M$ and $\sim_F^N$ are relatively fine – e.g. one feels that it would be unlikely for a randomly chosen m and m' to happen to be such that $m \sim_B^M m'$ – arises from the fact that both models can hold plenty of information not present in the other; it is not likely that m and m' will have only differences which are visible on the N side, because most potential differences between them are invisible there.

If we represent a bidirectional transformation R by giving a table for $\overrightarrow{R}$ and a table for $\overleftarrow{R}$, each with a row for each element of M and a column for each element of N, the equivalences can be read off: $m \sim_F m'$ if the rows for m and m' in the $\overrightarrow{R}$ table are identical, $n \sim_F n'$ if the columns for n and n' in the $\overrightarrow{R}$ table are identical, and similarly for the $\sim_B$ equivalences using the $\overleftarrow{R}$ table.

Example 5 *Let us represent Example 2 in this way.*

$\overrightarrow{R}$	a	b	c
(a,true)	a	a	a
(b,true)	b	b	b
(c,true)	c	c	c
(a,false)	a	a	a
(b,false)	b	b	b
(c,false)	c	c	c

$\overleftarrow{R}$	a	b	c
(a,true)	(a,true)	(b,true)	(c,false)
(b,true)	(a,true)	(b,true)	(c,true)
(c,true)	(a,false)	(b,true)	(c,true)
(a,false)	(a,false)	(b,false)	(c,true)
(b,false)	(a,false)	(b,false)	(c,false)
(c,false)	(a,true)	(b,false)	(c,false)

We read off that $(x,\phi) \sim_F (x',\phi')$ iff $x = x'$, while $m \sim_B m'$ only when $m = m'$. Looking at N, $n \sim_F n'$ always, while $n \sim_B n'$ only when $n = n'$.

Remark 1 *Clearly, that $n \sim_F n'$ always is equivalent to saying that $\overrightarrow{R}$ ignores its second argument, that is, that the transformation is a lens in the sense of [FGM+07].*

We shall use this as the definition.

Definition 6 A bidirectional transformation R is a lens if it is correct and hippocratic and furthermore $\overrightarrow{R}$ ignores its second argument, which we therefore omit for notational convenience.

The relationship between lenses and bidirectional transformations is easy to see and was discussed in Section 4.4 of [Ste10], although in the context of undoability rather than history ignorance. We will not introduce lens notation and laws, but simply state for the benefit of readers familiar with lenses:

Proposition 2 *Let R be a (for once, not necessarily correct and hippocratic) bidirectional transformation in which $\overrightarrow{R}$ ignores its second argument. Then*

1. *R is a well-behaved lens in the sense of [FGM+07] iff it is correct and hippocratic.*
2. *R is a very well-behaved lens in the sense of [FGM+07] iff it is correct, hippocratic and history ignorant.*

The following easy result was proved in [Ste08] and illustrated above.

Proposition 3 *If both $m_1 \sim_F m_2$ and $m_1 \sim_B m_2$ then $m_1 = m_2$.*

This results shows that a model is uniquely determined by specifying its $\sim_F$ and its $\sim_B$ equivalence class. Given a model m, we can think of its $\sim_F$ equivalence class as its x-coordinate and of its $\sim_B$ equivalence class as its y-coordinate. Specifying coordinates in this sense determines at most one model, but some pairs of coordinates specify no model; that is, if a given $\sim_F$ equivalence class and a given $\sim_B$ equivalence class intersect, they do so in a set of just one model, but it is in general possible that they do not intersect. We think of the elements of a model space laid out in a *coordinate grid* in this way; some squares on the grid are empty, but no square contains more than one model.

Let M_F be a transversal[3] of $\sim_F$ and M_B be a transversal of $\sim_B$.

We may identify M with a subset of $M_F \times M_B$: that is, henceforth we notate any element $m \in M$ as (m_F, m_B) where m_F is the unique element of M_F satisfying $m_F \sim_F m$ and m_B is the unique element of M_B satisfying $m_B \sim_B m$. The closure of M, $\overline{M}$, is the whole of $M_F \times M_B$. Similarly for N, we let N_F be a transversal of $\sim_F$ and N_B be a transversal of $\sim_B$.

Remark 2 *We are identifying a model by specifying its equivalence class in each of two equivalence relations. We have a notational choice to make here: do we represent a model m by giving*

[3] Recall that a transversal T for an equivalence relation on a set A is a set $T \subseteq A$ comprising exactly one element of each equivalence class.

representatives of each equivalence class, (m_F, m_B) *or do we use some notation for the equivalence classes themselves,* $([m]_{\sim_F}, [m]_{\sim_B})$*? Each choice has strengths and weaknesses. Since there is, in our setting, generally no canonical choice of representative of an equivalence class, the latter choice appeals; but it has the disadvantage that we must either define new symbols for a version of* R *lifted to equivalence classes, and juggle both versions in our work, or risk the confusion that might result from our abusing notation by not doing so. Instead we choose the former option, despite the disadvantage that it involves an arbitrary choice of transversal. What we gain is that when we write, for example,* m_F*, it does not matter whether we think of* m_F *as partial information concerning a model* $m = (m_F, m_B)$ *or as a model in its own right: it is both.*

It is important to understand, however, that it is not *generally possible to ensure that the elements of* M_F *are all* $\sim_B$ *equivalent; see Lemma 3.*

This way of representing models is useful because it separates relevant from irrelevant information for purposes of applying the components of the bidirectional transformation.

Lemma 2
1. *Whenever* $m \sim_F m'$ *and* $n \sim_B n'$ *we have* $R(m,n) \Leftrightarrow R(m',n')$*; that is, whether* $R((m_F, m_B), (n_F, n_B))$ *is determined from* m_F *and* n_B *alone: it is* $R(m_F, n_B)$*.*
2. *Whenever* $m \sim_F m'$ *and* $n \sim_F n'$ *we have* $\overrightarrow{R}(m,n) = \overrightarrow{R}(m',n')$*; that is, the value of* $\overrightarrow{R}((m_F, m_B), (n_F, n_B))$ *is determined by* m_F *and* n_F *alone: it is* $\overrightarrow{R}(m_F, n_F)$*.*
3. *Whenever* $m \sim_B m'$ *and* $n \sim_B n'$ *we have* $\overleftarrow{R}(m,n) = \overleftarrow{R}(m',n')$*; that is, the value of* $\overleftarrow{R}((m_F, m_B), (n_F, n_B))$ *is determined by* m_B *and* n_B *alone: it is* $\overleftarrow{R}(m_B, n_B)$*.*

Proof. If $R(m,n)$ and $m' \sim_F m$ then since $n = \overrightarrow{R}(m,n) = \overrightarrow{R}(m',n)$ we must have $R(m',n)$ by Remark 1. Similarly, $\overleftarrow{R}(m,n) = m$ iff $R(m,n)$ in which case if $n' \sim_B n$ then also $R(m,n')$. In other words, whether $R((m_F, m_B), (n_F, n_B))$ is determined by the entries m_F and n_B alone. Since m_F itself is an element of M, and n_B itself is an element of N, we have $R(m.n) = R(m_F, n_B)$.

Similarly, if $\overrightarrow{R}((m_F, m_B), (n_F, n_B)) = (n'_F, n'_B)$ then by definition $\overrightarrow{R}((m_F, m''_B), (n_F, n''_B)) = (n'_F, n'_B)$ for any other m''_B, n''_B; in other words the result of $\overrightarrow{R}((m_F, m_B), (n_F, n_B))$ is determined by the entries m_F, n_F alone. Dually, $\overleftarrow{R}((m_F, m_B), (n_F, n_B))$ is determined by m_B, n_B alone. □

Remark 3 *In terms of the symmetric lens formalism of [HPW11],* $M_B \times N_F$ *forms a minimal complement.*

Example 6 *In model set* M *of Example 2, let us pick* $M_F = \{(a, \mathit{true}), (b, \mathit{true}), (c, \mathit{true})\}$ *(the second element of each pair being an arbitrary choice) and* $M_B = M$*. With respect to these choices, the coordinate grid for* M *is:*

(a, true)	(a, true)		
(b, true)		(b, true)	
(c, true)			(c, true)
(a, false)	(a, false)		
(b, false)		(b, false)	
(c, false)			(c, false)
	(a, true)	(b, true)	(c, true)

For N_F we may take any of $\{a\}$, $\{b\}$, $\{c\}$; for N_B we must take $\{a,b,c\}$. The coordinate grid for N is a single column, because the transformation is a lens with N its abstract side.

In terms of the coordinate grid, whether m is consistent with n is determined by the column of m and the row of n. It is natural to ask whether one column can be consistent with more than one row. The answer is yes, but only provided that the rows do not overlap, in the following sense:

Lemma 3 *Suppose $R(m_F, n_B)$ and $R(m_F, z_B)$ where $n_B, z_B \in N_B$ and $n_B \neq z_B$. Then there cannot exist any $n_F \in N_F$ such that both $(n_F, n_B) \in N$ and $(n_F, z_B) \in N$.*

Dually, suppose $R(m_F, n_B)$ and $R(w_F, n_B)$ and $m_F \neq w_F$. Then there cannot exist any $m_B \in M_B$ such that both $(m_F, m_B) \in M$ and $(w_F, m_B) \in M$.

Proof. Suppose that such an n_F did exist. Then $\overrightarrow{R}(m_F, (n_F, n_B)) = (n_F, n_B)$ by hippocraticness and similarly $\overrightarrow{R}(m_F, (n_F, z_B)) = (n_F, z_B)$. But $(n_F, n_B) \sim_F (n_F, z_B)$ so $(n_F, n_B) = (n_F, z_B)$ which is a contradiction.

□

4 Applications of equivalences

In this section we exploit the results of the previous section to investigate the relationships between different special properties of bidirectional transformations.

We know that given $m_1, m_2 \in M$ there can be at most one $m \in M$ such that $m \sim_F m_1$ and $m \sim_B m_2$. An important special case is when there is exactly one: every square in our "coordinate grid" is occupied.

Definition 7 M is *full* with respect to bidirectional transformation R if for any $m_1, m_2 \in M$ there exists $m \in M$ such that $m \sim_F m_1$ and $m \sim_B m_2$.

Intuitively, when we specify a model $m \in M$ by specifying its $\sim_F$ and $\sim_B$ equivalence classes, what we are doing is picking out the information in m which is relevant to the question of whether m is consistent with some model in N. As we have seen, this consistency is completely determined by m's $\sim_F^M$ equivalence class (and dually, for a model $n \in N$ its consistency is determined by its $\sim_B^N$ class). In the simplest practical cases, this consistency-relevant information may actually be identifiable in the model, as being, for example, the set of model elements of particular types contained in the model; however, our set-up also works generally.

What it means for M to be full is that the consistency-relevant information is in in a certain sense independent of the rest of the information needed to specify a model fully; there is no redundancy in the representation. If M is full, and if $m = (m_F, m_B)$ is changed by modifying one of the pieces of information (say m_F is changed to m'_F), it will not be necessary for the other piece of information to be modified as a side-effect, because fullness tells us that (m'_F, m_B) will definitely exist as a model in M. This simplifies the situation in an important way. Contrast the more usual situation we see in Examples 1, 3; the information in $m \in M$ on which consistency depends (the set of class names) is strongly interdependent with the rest of the information in the model. For example, suppose we take m_1 to be a model containing only a class `Foo` in default state (no attributes or other associated information), and m_2 to be a model containing a class `Bar` which is not in default state (say it has an attribute, or a state machine, or whatever). It is clear that there can be no model m such that $m \sim_F m_1$ and $m \sim_B m_2$. In order for $m \sim_F m_1$ to hold, the set of names of classes in m must be the singleton $\{$"Foo"$\}$; in order for $m \sim_B m_2$ to hold, the non-default-state class `Bar` must be in m.

More generally, we may informally consider that the more sparsely the coordinate grid is populated, the more dependency there is between the consistency-relevant information and the rest.

Definition 8 Let $R : M \leftrightarrow N$ be a bidirectional transformation inducing equivalences and a coordinate system as usual. We say that R is *matching* if there is a bijection $f : M_F \rightarrow N_B$ such that $R(m_F, f(m_F))$ for all $m_F \in M_F$. We say that R is *simply matching* if, in addition, $R(m_F, n_B)$ holds *only* when $n_B = f(m_F)$.

Lemma 4 *If N is full, then given $m_F \in M_F$ there exists a unique $n_B \in N_B$ satisfying $R(m_F, n_B)$. Dually if M is full, then given $n_B \in N_B$ there exists a unique $m_F \in M_F$ satisfying $R(m_F, n_B)$. Therefore if both M and N are full, then R is simply matching.*

Proof. Follows directly from Lemma 3. □

Proposition 4 *The following are equivalent:*

1. *$R : M \leftrightarrow N$ is history ignorant.*
2. *M and N are full with respect to R.*
3. *For each $m \in M$ and $n \in N$ we have $\overrightarrow{R}(m,n) \sim_F n$ and $\overleftarrow{R}(m,n) \sim_B m$: that is, $\overrightarrow{R}$ stabilises the coordinate grid columns of N and $\overleftarrow{R}$ stabilises the coordinate grid rows of M.*

Proof. $1 \Leftrightarrow 3$ is immediate from the definitions. Next we show $(1,3) \Rightarrow 2$. Let $m_1, m_2 \in M$. We must construct $m \in M$ such that $m \sim_F m_1$ and $m \sim_B m_2$. Write $m_1 = (m_{1F}, m_{1B})$, $m_2 = (m_{2F}, m_{2B})$. Pick any $(n_F, n_B) \in N$. Then, using 3., for some $w_F \in M_F, z_B \in N_B$ we have

$$\overrightarrow{R}((m_{1F}, m_{1B}), n) = (n_F, z_B)$$

$$\overleftarrow{R}((m_{2F}, m_{2B}), (n_F, z_B)) = (w_F, m_{2B})$$

and by correctness $R(m_{1F}, z_B)$ and $R(w_F, z_B)$. Using 1. and Lemma 4, $m_{1F} = w_F$ so we have constructed $(m_{1F}, m_{2B}) = m \in M$ as required. Dually for N.

Finally we show $2 \Rightarrow 3$. Consider $\overrightarrow{R}(m, (n_F, n_B) = (n'_F, n'_B)$; we must show that $n_F = n'_F$. Since R is correct, $R(m, (n'_F, n'_B))$ which is equivalent to $R(m, n'_B)$. Since N is full, $(n_F, n'_B) \in N$, and since $R(m, n'_B)$, by hippocraticness $\overrightarrow{R}(m, (n_F, n'_B)) = (n_F, n'_B)$. But since $(n_F, n_B) \sim_F (n_F, n'_B)$, $(n'_F, n'_B) = \overrightarrow{R}(m, (n_F, n_B)) = \overrightarrow{R}(m, (n_F, n'_B)) = (n_F, n'_B)$ so $n_F = n'_F$ as required. □

Corollary 1 *If R is history ignorant, then R is simply matching.*

The following example shows that not every undoable transformation is simply matching.

Example 7 *We slightly modify Example 2 so that undoability is retained but simple matching is lost.*

Let $M = N = \{a,b,c\} \times \{\mathsf{false}, \mathsf{true}\}$. *For all* $x,y \in \{a,b,c\}$, $\psi,\phi \in \{\mathsf{false},\mathsf{true}\}$, *let:*
$R((x,\phi),(y,\psi))$ *iff* $x = y$;
$\overrightarrow{R}((x,\phi),(y,\psi)) = (x,\psi)$;

$$\begin{aligned} \overleftarrow{R}((x,\phi),(y,F)) &= (y,\neg\phi) && \textit{if (as sets) } \{x,y\} = \{a,c\} \\ &= (y,\phi) && \textit{otherwise} \\ \overleftarrow{R}((x,\phi),(y,T)) &= (y,\phi) \end{aligned}$$

This transformation is still correct, hippocratic and undoable, but not history ignorant. To see that it is not simply matching, observe that $\{(a,F),(a,T)\}$ *is a* $\sim_F^M$*-equivalence class which is consistent with two* $\sim_B^N$*-classes, viz.* $\{(a,F)\}$ *and* $\{(a,T)\}$. *Our "twist" has ensured that these two classes remain separate, because* $\overleftarrow{R}((c,T),(a,F)) = (a,F) \neq (a,T) = \overleftarrow{R}((c,T),(a,T))$.

Now it is easy to prove Proposition 1 relating Bancilhon and Spyratos' equivalence to our $\sim_B^M$.

Proof. Suppose first that $m \equiv m'$, that is, $\exists n \in N.\overleftarrow{R}(m,n) = m'$. We need to show that $m \sim_B m'$. Since R is history ignorant, by Proposition 4, $m' = \overleftarrow{R}(m,n) \sim_B m$ as required.

Conversely, suppose $m \sim_B m'$; we must show that $\exists n \in N.\overleftarrow{R}(m,n) = m'$. By Corollary 1 there is a unique $n_B \in N_B$ such that $R(m',n_B)$. Then by correctness of R and by Lemma 2, $\overleftarrow{R}(m,n_B) \sim_F m'$. By Proposition 4, $\overleftarrow{R}(m,n_B) \sim_B m \sim_B m'$. Finally by Proposition 3, this implies that $\overleftarrow{R}(m,n_B) = m'$ as required. (Notice that any $n \sim_B^N n_B$ would have done as well.) □

Once we have a simply matching bidirectional transformation R it is natural to relabel the elements of M_F and N_B so that both are identified with a matching transversal P, so that $R((p,m_B),(n_F,p'))$ iff $p = p'$. This is reminiscent of constant complement bidirectional transformations, but it is important to be aware (a) that $M \subseteq P \times M_B$ but equality does not hold in general; (b) that while $\overrightarrow{R}((p,m_B),(n_F,p')) = (n'_F,p)$, we do not necessarily have $n'_F = n_F$. In fact, by Lemma 2 n'_F depends on p and n_F. We will write $\overrightarrow{R}((p,m_B),(n_F,p')) = (f_p(n_F),p)$. Dually, $\overleftarrow{R}((p',m_B),(n_F,p)) = (p,b_p(m_B))$.

The property of being simply matching is an interesting one, being strictly weaker than being history ignorant, satisfied by realistic examples such as Example 1, and yet not universally true. It turns out that there is an alternative characterisation of this class of bidirectional transformations: it comprises those that can be constructed by placing two lenses "tail to tail" in a certain way.

Definition 9 Let $R : M \leftrightarrow N$ be a simply matching bidirectional transformation with matching transversal P. Then we define lenses $R_M : M \leftrightarrow P$ and $R_N : N \leftrightarrow P$ as follows, exploiting the representation of elements of M as elements of $P \times M_B$:

- $R_M(m,p)$ iff $\overrightarrow{R_M}(m) = p$ iff $m = (p,m_B)$ for some $m_B \in M_B$.
- $\overleftarrow{R_M}((p',m_B),p) = (p,b_p(m_B))$.
- Dually, $R_N(n,p)$ iff $\overrightarrow{R_N}(n) = p$ iff $n = (n_F,p)$ for some $n_F \in N_F$, and

- $\overleftarrow{R_N}((n_F, p'), p) = (f_p(n_F), p)$.

The following lemma is obvious:

Lemma 5 *R_M and R_N are indeed correct and hippocratic.*

The following construction of a bidirectional transformation from a pair of lenses has been recorded before, see for example Theorem 6 of [Dis08]; here we shall show that it produces exactly the simply matching bidirectional transformations.

Definition 10 Let $R_M : M \leftrightarrow P$ and $R_N : N \leftrightarrow P$ be any lenses sharing a common view. Then the bidirectional transformation $R = \langle R_M, R_N \rangle : M \leftrightarrow N$ constructed from R_M and R_N is defined thus:

- $R(m,n)$ iff $\overrightarrow{R_M}(m) = \overrightarrow{R_N}(n)$
- $\overrightarrow{R}(m,n) = \overleftarrow{R_N}(n, \overrightarrow{R_M}(m))$
- $\overleftarrow{R}(m,n) = \overleftarrow{R_M}(m, \overrightarrow{R_N}(n))$

Proposition 5 *Given lenses $R_M : M \leftrightarrow P$ and $R_N : N \leftrightarrow P$,*

1. *$\langle R_M, R_N \rangle$ is indeed correct and hippocratic.*
2. *$\langle R_M, R_N \rangle$ is simply matching with common transversal in bijection with P.*

Proof. Let $R = \langle R_M, R_N \rangle$.

1. Correctness: let $n' = \overrightarrow{R}(m,n) = \overleftarrow{R_N}(n, \overrightarrow{R_M}(m))$. We have to show that $R(m,n')$, that is, that $\overrightarrow{R_M}(m) = \overrightarrow{R_N}(n')$. Now $\overrightarrow{R_N}(n') = \overrightarrow{R_N}(\overleftarrow{R_N}(n, \overrightarrow{R_M}(m))) = \overrightarrow{R_M}(m)$ as required. Hippocraticness: suppose $R(m,n)$, that is, $\overrightarrow{R_M}(m) = \overrightarrow{R_N}(n)$, and consider $\overrightarrow{R}(m,n) = \overleftarrow{R_N}(n, \overrightarrow{R_M}(m)) = \overleftarrow{R_N}(n, \overrightarrow{R_N}(n)) = n$ by hippocraticness of R_N.

2. We show that the $\sim_F^M$ equivalence classes are the sets $(\overrightarrow{R_M})^{-1}(\{p\})$ for $p \in P$. Suppose m_1 and m_2 are in $(\overrightarrow{R_M})^{-1}(\{p\})$. Then for any $n \in N$, $\overrightarrow{R}(m_1,n) = \overleftarrow{R_N}(n, \overrightarrow{R_M}(m_1)) = \overleftarrow{R_N}(n,p) = \overleftarrow{R_N}(n, \overrightarrow{R_M}(m_2)) = \overrightarrow{R}(m_2,n)$, so $m_1 \sim_F m_2$ as required. Conversely, suppose that $m_1 \sim_F m_2$ and pick n_1 such that $\overrightarrow{R_N}(n_1) = \overrightarrow{R_M}(m_1) = p_1$. Then since $\overrightarrow{R}(m_1,n_1) = \overrightarrow{R}(m_2,n_1)$ by hypothesis, we have $n_1 = \overleftarrow{R_N}(n_1, \overrightarrow{R_M}(m_1)) = \overleftarrow{R_N}(n_1, \overrightarrow{R_M}(m_2))$ so by correctness of R_N we must have $R_N(n_1, \overrightarrow{R_M}(m_2))$, that is, $\overrightarrow{R_N}(n_1) = \overrightarrow{R_M}(m_2) = p_1$ as required. It follows from the definition of R that R is simply matching.

□

We may summarise Lemma 5 and Proposition 5 in the following (informally-stated) theorem:

Theorem 1 *A bidirectional transformation $R : M \leftrightarrow N$ is simply matching if and only if it can be decomposed into a pair of lenses working "tail to tail". Any such decomposition gives rise to a choice of matching transversal for R, and vice versa.*

Example 2 is simply matching; take $f((a,true)) = a$ etc. Example 1 is also simply matching, for each possible subset of the set of all legal class names determines and is determined by a transversal element (of both M_F and N_B) and $T(m_F, n_B)$ holds iff the sets thus determined by m_F and n_B are identical. Notice that since we have already observed that neither of these examples is history ignorant, and that Example 1 is not even undoable, this shows that simply matching is a strictly weaker condition.

The following tiny example, which is matching but not simply matching, illustrates some of the difficulty inherent in "less nice" bidirectional transformations.

Example 8 *Let $M = N = \{0,1\}$ and let $R(m,n)$ hold iff $mn = 0$. $\overrightarrow{R}$ and $\overleftarrow{R}$ are determined by correctness and hippocraticness:*

R	0	1
0	T	T
1	T	F

$\overrightarrow{R}$	0	1
0	0	1
1	0	0

$\overleftarrow{R}$	0	1
0	0	0
1	1	0

$0 \not\sim_F 1$ because $\overrightarrow{R}(0,1) = 1 \neq \overrightarrow{R}(1,1)$; $0 \not\sim_B 1$ because $\overleftarrow{R}(0,1) = 0 \neq \overleftarrow{R}(1,1)$; thus each element of M forms an equivalence class under each equivalence, and dually for N. Therefore there is only one choice of transversal, and we identify the elements with the equivalence classes.

Considered as a subset of $M_F \times M_B$, M is the diagonal subset $\{(0,0),(1,1)\}$ where $(0,0)$ represents 0 and $(1,1)$ represents 1. That is, the coordinate grids for M and N are both, identically:

$$\begin{array}{c|cc} 1 & & 1 \\ 0 & 0 & \\ \hline & 0 & 1 \end{array}$$

We have here the simplest possible example in which there are two distinct elements of M_F (0 and 1) compatible with one element of N_B (0), and only one of those M_F elements (0) is also compatible with a second, distinct element of N_B (1). In other words, both columns of M's coordinate grid are compatible with the 0 row of N's, and the 0 column of M's grid is also compatible with the 1 row of N's. Note that neither the rows, nor the columns, of the coordinate grids overlap in the sense of Lemma 3.

Clearly R is not undoable. For example, $R(0,1)$, but $\overrightarrow{R}(0, \overrightarrow{R}(1,1)) = \overrightarrow{R}(0,0) = 0 \neq 1$.

Example 9 *Finally, lest the reader suspect that all bidirectional transformations be matching, we record an example which is not.*

Let $M = \{0,1\}$ and $N = \{a,b,c\}$. Let R, $\overrightarrow{R}$ and $\overleftarrow{R}$ be as follows:

R	a	b	c
0	T	T	F
1	F	T	T

$\overrightarrow{R}$	a	b	c
0	a	b	a
1	c	b	c

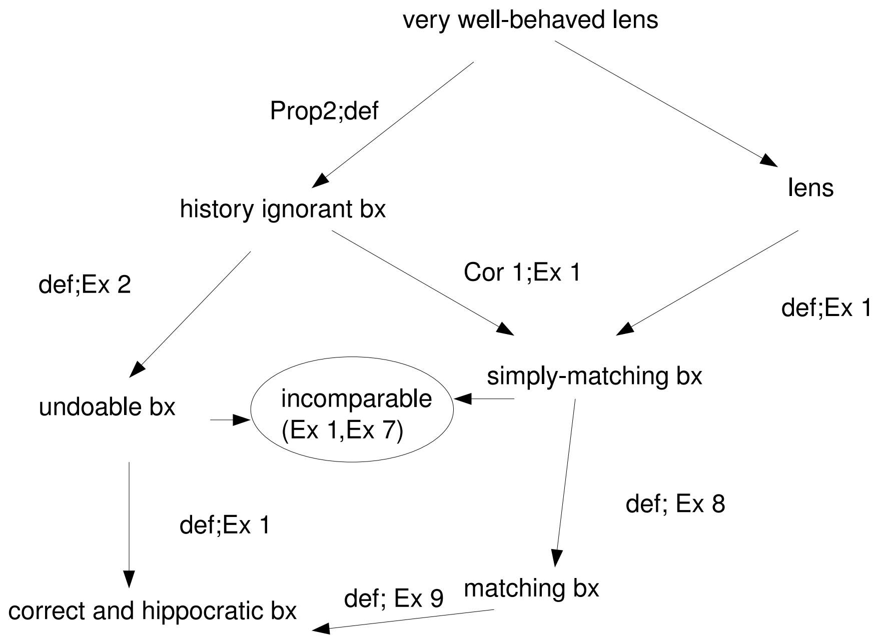
$\overleftarrow{R}$	a	b	c
0	0	0	1
1	0	1	1

We read off that the only non-trivial equivalence is that $a \sim_F^N c$. Taking a as the representative of their $\sim_F^N$ equivalence class, the coordinate grids are

$$\begin{array}{c|cc} 1 & & 1 \\ 0 & 0 & \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|cc} c & c & \\ b & & b \\ a & a & \\ \hline & a & b \end{array}$$

This is not undoable, since although $R(0,b)$, we have $\overleftarrow{R}(\overleftarrow{R}(0,c),b) = \overleftarrow{R}(1,b) = 1$.

Figure 1: Implications between properties of correct hippocratic bidirectional transformations. Labels p;q on arrows indicate that p is for the implication and q for absence of the reverse implication, "def" meaning direct from the definition.



To summarise, Figure 1 gives the implication relationships between different special cases of bidirectional transformations.

5 Related work

Defining equivalence relations on model sets that are induced by bidirectional transformations goes back at least to [BS81] in the special case of constant complement view updates in databases and to [BFP$^+$08] in the less special case of lenses. We have demonstrated the relation of our work to those papers.

A major concern of the model-driven development community, as it has started to consider bidirectional transformations, has been the generalisation to the symmetric case, in which neither model set comprises strict abstractions of the models in the other model set. An early work on properties of such bidirectional transformations was the present author's MODELS'07 paper that led to [Ste10]; Diskin's MODELS'08 paper [Dis08] explored further, including, as already remarked, defining history ignorance and discussing the construction of a bidirectional transformation from a pair of lenses.

Neither of those papers, however, made use of equivalences on model sets comparable to those considered in the earlier database and lens work. These were first generalised to the symmetric case as part of the work in [Ste08], but only a few results relating to them (identified here) were given in that paper.

Later work has included [XSHT11], focusing on the additional problems posed by concurrent updates, and a body of work exemplified by [DXC^{+}11] on delta-based bidirectional transformations. The latter defines several other properties of bidirectional transformations; there is more work to be done to fully elucidate the connections.

6 Conclusion

We have laid out some results relating to the principal equivalence relations induced on model sets by a correct and hippocratic bidirectional transformation between them. By means of these results we have established implications and non-implications between many different conditions that can be placed on bidirectional transformations to ensure that they are "nice".

A topic of conversation among people interested in bidirectional transformations has long been the extent to which the symmetric case, in which we relate models neither of which is a strict abstraction of the other, raises essentially new problems from the asymmetric case studied earlier in the database field. In practice, it is often possible to conceptualise a symmetric bidirectional transformation as a pair of lenses placed "tail to tail", that is, having a common view: the view captures the information which is common to both models, and the lenses manage the synchronisation. Actually writing the transformation this way would have the obvious drawback that it would involve a great deal of duplication between the two lenses. In this paper, we have characterised the bidirectional transformations that could, in principle, be written in this way; they are the simply matching ones. Although we have shown that not all bidirectional transformations are simply matching, we have the impression that a large class of practical examples (such as our Example 1 etc.) are in fact simply matching. A bidirectional transformation language that gave good, usable facilities for expressing and working with simply matching transformations might be practically useful even if it did not permit the expression of non-simply-matching transformations.

We have also begun, but not reported here, an investigation into subspaces and subspace pairs of model sets, which capture the fact that teams of developers can agree to stay within certain subspaces of the model spaces, in which case it is desirable that applying a transformation will not move them outside their agreed zone. In future work we will build on this to investigate the implications of adding an information orderings to a model set, and relate this work to the study of monoids of edits acting on model spaces.

Acknowledgements: The author thanks the referees, the participants in the Bx'12 workshop, and Zinovy Diskin, for very helpful comments, including some that we hope will lead to future work.

Bibliography

[BFP$^+$08] A. Bohannon, J. N. Foster, B. C. Pierce, A. Pilkiewicz, A. Schmitt. Boomerang: Resourceful Lenses for String Data. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), San Francisco, California*. Jan. 2008.

[BS81] F. Bancilhon, N. Spyratos. Update Semantics of Relational Views. *ACM Trans. Database Syst.* 6(4):557–575, 1981.

[Dis08] Z. Diskin. Algebraic Models for Bidirectional Model Synchronization. In *Model Driven Engineering Languages and Systems, 11th International Conference, MODELS 2008, Toulouse, France, September 28 - October 3, 2008. Proceedings*. Lecture Notes in Computer Science 5301, pp. 21–36. Springer, 2008.

[DXC$^+$11] Z. Diskin, Y. Xiong, K. Czarnecki, H. Ehrig, F. Hermann, F. Orejas. From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case. In *Model Driven Engineering Languages and Systems, 14th International Conference, MODELS 2011, Wellington, New Zealand, October 16-21, 2011. Proceedings*. Lecture Notes in Computer Science 6981, pp. 304–318. Springer, 2011.

[FGM$^+$07] 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 Transactions on Programming Languages and Systems* 29(3):17, May 2007.

[HPW11] M. Hofmann, B. C. Pierce, D. Wagner. Symmetric Lenses. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Austin, Texas*. Jan. 2011.

[Ste08] P. Stevens. Towards an algebraic theory of bidirectional transformations. In *Proceedings of the International Conference on Graph Transformations, ICGT'08*. Lecture Notes in Computer Science 5214, pp. 1–17. Springer, September 2008.

[Ste10] P. Stevens. Bidirectional Model Transformations in QVT: Semantic Issues and Open Questions. *Journal of Software and Systems Modeling (SoSyM)* 9(1):7–20, 2010.

[XSHT11] Y. Xiong, H. Song, Z. Hu, M. Takeichi. Synchronizing concurrent model updates based on bidirectional transformation. *Journal of Software and Systems Modeling (SoSyM)*, 2011. Online first.