Generalizing Lenses

Daniel Wagner

August 19, 2013
There are many fundamentally bidirectional settings that call for generalizations of traditional lenses where a language is possible and helpful.
There are many fundamentally bidirectional settings that call for generalizations of traditional lenses where a language is possible and helpful.
There are many fundamentally bidirectional settings that call for generalizations of traditional lenses where a language is possible and helpful.
Thesis Proposal

There are many fundamentally bidirectional settings that call for generalizations of traditional lenses where a language is possible and helpful.
Overview

Traditional lenses

Symmetry

Edits

Multidirectionality

Logistics
Traditional lenses
Jan
/ \ 
palindrome.jpg gamer.jpg froghead.jpg

May
burrito.jpg withperson.jpg
Abstract model

A lens $\ell \in X \leftrightarrow^a Y$ has components

$$get \in X \to Y$$

$$put \in Y \times X \to X$$
Abstract model

A lens $\ell \in X \leftrightarrow Y$ has components

$$\text{get} \in X \rightarrow Y$$

$$\text{put} \in Y \times X \rightarrow X$$

Synchronizing too often doesn’t hurt.

$$\text{get}(\text{put}(y, x)) = y$$

$$\text{put}(\text{get}(x), x) = x$$
Abstract model

A lens \( \ell \in X \leftrightarrow Y \) has components

\[
\begin{align*}
\text{get} & \in X \to Y \\
\text{put} & \in Y \times X \to X
\end{align*}
\]

Synchronizing too often doesn’t hurt.

\[
\begin{align*}
\text{get}(\text{put}(y, x)) & = y \\
\text{put}(\text{get}(x), x) & = x \\
\text{put}(y', \text{put}(y, x)) & = \text{put}(y', x)
\end{align*}
\]

Not synchronizing often enough doesn’t hurt.
Related work: asymmetric lenses

- Combinators for Bidirectional Tree Transformations (Foster, Greenwald, Moore, Pierce, Schmitt; POPL 2005)
- Relational Lenses: A Language For Updateable Views (Bohannon, Vaughn, and Pierce; PODS 2006)
- Boomerang: Resourceful Lenses for String Data (Bohannon, Foster, Pierce, Pilkiewicz, and Schmitt; POPL 2008)
- Bidirectional Programming Languages (Foster; thesis 2009)
- Bidirectionalizing Graph Transformations (Hidaka, Hu, Inaba, and Kato; ICFP 2010)
- Update Semantics of Relational Views (Bancilhon and Spyratos; 1981)
Symmetry

(in collaboration with Martin Hofmann and Benjamin Pierce)
Jan
  /\  \\
palindrome.jpg [costume,food] gamer.jpg [costume] froghead.jpg [costume]

May
  /\  \\  \\
burrito.jpg [food,onlyface] withperson.jpg [ ]

[costume,food]
[food,onlyface]
[food,onlyface]
[ ]
Abstract model

A lens \( \ell \in X \leftrightarrow Y \) has a set \( C \) and components

\[
\begin{align*}
\text{putr} & \in X \times C \to Y \times C \\
\text{putl} & \in Y \times C \to X \times C
\end{align*}
\]
Abstract model

A lens $\ell \in X \leftrightarrow^s Y$ has a set $C$ and components

$$\text{putr} \in X \times C \rightarrow Y \times C$$

$$\text{putl} \in Y \times C \rightarrow X \times C$$

Synchronizing too often doesn’t hurt.

$$\text{putr}(x, c) = (y, c')$$

$$\frac{\text{putr}(x, c) = (y, c')}{\text{putl}(y, c') = (x, c')}$$
Abstract model

A lens $\ell \in X \leftrightarrow^s Y$ has a set $C$ and components

$$putr \in X \times C \rightarrow Y \times C$$
$$putl \in Y \times C \rightarrow X \times C$$

Synchronizing too often doesn’t hurt.

$$putr(x, c) = (y, c')$$
$$putl(y, c') = (x, c')$$

Not synchronizing often enough doesn’t hurt.

$$putr(x, c) = putr(x, c')$$
Nice property of asymmetric lenses:

\[(k; \ell); m = k; (\ell; m)\]
Twist: equational reasoning

Nice property of asymmetric lenses:

\[(k; \ell); m = k; (\ell; m)\]

Not true for symmetric lenses!
In dissertation

- Observational equivalence
- Point-free programming language
  - Basic (non-recursive) data types
  - Lists, with folds and unfolds
  - Some generalized container operations
- Proof that this generalizes asymmetric lenses
Related work: other symmetric approaches

- Symmetric Constraint Maintainers
  (Meertens; 1998)

- Towards an Algebraic Theory of Bidirectional Transformations
  (Stevens; ICGT 2008)

- Bidirectional Model Transformations in QVT: Semantic Issues and Open Questions
  (Stevens; MoDELS 2007)

- Algebraic Models for Bidirectional Model Synchronization
  (Diskin; MoDELS 2008)

- Supporting Parallel Updates with Bidirectional Model Transformations
  (Xiong, Song, Hu, and Takeichi; ICMT 2009)
Edits

(in collaboration with Martin Hofmann and Benjamin Pierce)
Abstract model

Edit lens $\ell \in (M, X, \cdot) \leftrightarrow (N, Y, \odot)$ has set $C$ and

$$dputr \in M \times C \rightarrow N \times C$$
$$dputl \in N \times C \rightarrow M \times C$$
Abstract model

Edit lens \( \ell \in (M, X, \cdot) \xleftrightarrow{\delta} (N, Y, \odot) \) has set \( C \) and

\[
dputr \in M \times C \to N \times C \\
dputl \in N \times C \to M \times C
\]

Synchronizing too often doesn’t hurt.

\[
dputr(1_M, c) = (1_N, c)
\]
Abstract model

Edit lens $\ell \in (M, X, \cdot) \xleftrightarrow{\delta} (N, Y, \odot)$ has set $C$ and

$$dputr \in M \times C \rightarrow N \times C$$
$$dputl \in N \times C \rightarrow M \times C$$

Synchronizing too often doesn’t hurt.

$$dputr(1_M, c) = (1_N, c)$$

Not synchronizing often enough doesn’t hurt.

$$dputr(m, c) = (n, c')$$
$$dputr(m', c') = (n', c'')$$
$$dputr(mm', c) = (nn', c'')$$
Notable benefits

- All changes reported, so synchronizing less often is less controversial
- Intentional information in edits aids alignment
- Smaller complement in many cases!
- Roundtrip laws are monoid homomorphism laws
Notable benefits

- All changes reported, so synchronizing less often is less controversial
- Intentional information in edits aids alignment
- Smaller complement in many cases!
- Roundtrip laws are monoid homomorphism laws
- Observational equivalence, combinator language, generalizes symmetric lenses
Related work: other edit-based approaches

- Towards an Algebraic Theory of Bidirectional Transformations
  (Stevens; ICGT 2008)
- Matching Lenses: Alignment and View Update
  (Barbosa, Cretin, Foster, Greenberg, and Pierce; ICFP 2010)
- From State- to Delta-based Bidirectional Model Transformations
  (Diskin, Xiong, Czarnecki; TPMT 2010)
- From State- to Delta-based Bidirectional Model Transformations: The Symmetric Case
  (Diskin, Xiong, Czarnecki, Ehrig, Hermann, and Orejas; MoDELS 2011)
- Delta Lenses over Inductive Types
  (Pacheco, Cunha, Hu; ECEASST 2012)
Multidirectionality

(in collaboration with Jen Paykin, Benjamin Pierce, Jeff Vaughan, and Geoff Washburn)
<table>
<thead>
<tr>
<th></th>
<th>costume, food</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>costume</td>
</tr>
<tr>
<td></td>
<td>costume</td>
</tr>
<tr>
<td></td>
<td>food, onlyface</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- costume
- food
- onlyface
<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2012</td>
<td>70000</td>
<td>9000</td>
</tr>
<tr>
<td>3</td>
<td>2013</td>
<td>70000</td>
<td>4000</td>
</tr>
<tr>
<td>4</td>
<td>Total</td>
<td>140000</td>
<td>13000</td>
</tr>
</tbody>
</table>
New interaction mode

<table>
<thead>
<tr>
<th>Year</th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>9000</td>
<td>79000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>4000</td>
<td>74000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>13000</td>
<td>153000</td>
</tr>
</tbody>
</table>

...and this happens behind the scenes, too.
New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>9000</td>
<td>79000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>4000</td>
<td>74000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>13000</td>
<td>166000</td>
</tr>
</tbody>
</table>

... and this happens behind the scenes, too.
New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>9000</td>
<td>79000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>4000</td>
<td>74000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>26000</td>
<td>166000</td>
</tr>
</tbody>
</table>

...and this happens behind the scenes, too.
New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>18000</td>
<td>79000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>8000</td>
<td>74000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>26000</td>
<td>166000</td>
</tr>
</tbody>
</table>

... and this happens behind the scenes, too.
New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>18000</td>
<td>88000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>8000</td>
<td>78000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>26000</td>
<td>166000</td>
</tr>
</tbody>
</table>
New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>18000</td>
<td>88000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>8000</td>
<td>78000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>26000</td>
<td>166000</td>
</tr>
</tbody>
</table>

...and this happens behind the scenes, too.
## New interaction mode

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>18000</td>
<td>88000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>8000</td>
<td>78000</td>
</tr>
<tr>
<td>Total</td>
<td>140000</td>
<td>26000</td>
<td>166000</td>
</tr>
</tbody>
</table>

...and this happens behind the scenes, too.
Straw-man abstract model

For universe $U$, lens $\ell \in \mathcal{M}(N)$ has components

$$\text{put} \in 2^N \to U^N \to U^N$$

$$K \in 2^{U^N}$$
Straw-man abstract model

For universe $U$, lens $\ell \in \mathcal{M}(N)$ has components

$$\text{put} \in 2^N \rightarrow U^N \rightarrow U^N$$

$$K \in 2^{U^N}$$

Inputs are really inputs and consistency is restored.

$$\text{put}(S, f)|_S = f|_S$$

$$\text{put}(S, f) \in K$$
Straw-man abstract model

For universe $U$, lens $\ell \in \mathcal{M}(N)$ has components

$$\text{put} \in 2^N \rightarrow U^N \rightarrow U^N$$

$$K \in 2^{U^N}$$

Inputs are really inputs and consistency is restored.

$$\text{put}(S, f)|_S = f|_S$$

$$\text{put}(S, f) \in K$$

$$\text{put}(\emptyset, f) = f$$

Synchronizing too often doesn’t hurt.
Unsolvable updates

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Equipment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>70000</td>
<td>9000</td>
<td>79000</td>
</tr>
<tr>
<td>2013</td>
<td>70000</td>
<td>4000</td>
<td>74000</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Track sets of names that are always solvable.
Composition intuition

Safe updates: \{X\} or \{Z\}.
Composition intuition

$X \xrightarrow{k} Y \quad Y \xrightarrow{l} Z$

Safe updates:

$\{X\}$ or $\{Z\}$.
Composition intuition

Safe updates: \{X\} or \{Z\}.
Ambiguous updates

\[ O_1 \xrightarrow{k} O_2 \quad I \quad O_3 \xleftarrow{\ell} \]

\[ O_1 \rightarrow I \rightarrow k; \ell \rightarrow O_2 \xrightarrow{\phantom{k}} O_3 \]
Two plans

Observational equivalence is no help.
Remaining questions

- Complete strategies for disambiguation?
- Behavioral specifications for disambiguation?
- How can we extend the static update check?
- What dynamic update checks are possible?
Related work: bidirectional spreadsheets

- Tiresias: The Database Oracle for How-To Queries  
  (Meliou and Suciu; SIGMOD ICMD 2012)
- A Spreadsheet Based on Constraints  
  (Stadelmann; UIST 1993)
- SkyBlue: A Multi-way Local Propagation Constraint Solver for User Interface Construction  
  (Sannella; UIST 1994)
- Expressing Multi-way Dataflow Constraint Systems as a Commutative Monoid Makes Many of their Properties Obvious  
  (Järvi, Haveraaen, Freeman, and Marcus; SIGPLAN WGP 2012)
- A Constraint-Based Spreadsheet for Cooperative Production Planning  
  (Chew and David; KBPPSC 1992)
- How to Use the Spreadsheet Manager  
  (Evans; tech report 1993)
- Interval Constraint Spreadsheets for Financial Planning  
  (Hyvőnen; AIAWS 1991)
Logistics
Nailing ambiguity resolution is lynchpin
Extending static and dynamic checks is polish
Bad case: trade black box time for additional ambiguity time
Worst case: biased composition
Why black boxes?

\[ \text{price} = \text{base} + \text{tax} \]
\[ \text{tax} = 0.08 \times \text{base} \]
Why black boxes?

\[ \text{price} = \text{base} + \text{tax} \]
\[ \text{tax} = 0.08 \times \text{base} \]
Why black boxes?

\[
\text{price} = \text{base} + \text{tax} \\
\text{tax} = 0.08 \times \text{base}
\]
Why black boxes?

\[ price = base + tax \]

\[ tax = 0.08 \times base \]
How to progress

- When any plan will do: greedy algorithm
- Assign a cost to each method
- Specification: min-cost set of methods
- Implementation: search (efficient when combining costs is monotonic)