Bidirectional Mappings

- Most programs work in one direction—from source to target
Most programs work in one direction—from source to target.

But sometimes we want to update the target.
Most programs work in one direction—from source to target.
But sometimes we want to update the target.
And “translate” this update to obtain an appropriately updated source.
The View Update Problem

This is called the view update problem in the database literature.
The View Update Problem In Practice

...but also comes up in other contexts, such as data synchronization.
Linguistic Approach

Lenses: bidirectional programs where each expression denotes both functions.

\[
\text{get : } S \rightarrow T \\
\text{put : } T \times S \rightarrow S
\]
Address Books (Get)

Source string (vCard dialect used in Address Book.app):

BEGIN:VCARD
VERSION:3.0
N:Pierce;Benjamin C.;;;
FN:Benjamin C. Pierce
TEL;type=HOME;type=pref:215 732-4684
TEL;type=CELL:215 266-9001
TEL;type=WORK:215 898-6222
X-ABUID:87B85E7E-AB0F-4819-8647-0BD532019144\:ABPerson
END:VCARD

Target string (simple ASCII format):

Pierce, Benjamin C., 215 732-4684 (h), \
215 266-9001 (c), 215 898-6222 (w)
Source string (SwissProt ASCII):

OS  Solanum melongena (Eggplant) (Aubergine).
OG  Plastid; Apicoplast.
OC  Mammalia; Eutheria.
OX  NCBI_TaxID=9606;
OH  NCBI_TaxID=9481; Callithrix.
OH  NCBI_TaxID=9536; Cercopithecus hamlyni (Owl-faced monkey) (Hamlyn’s monkey).
SwissProt (Get)

Target string (SwissProt XML):

<organism key="1">
  <name type="scientific">Solanum melongena</name>
  <name type="common">Eggplant</name>
  <name type="synonym">Aubergine</name>
  <dbReference type="NCBI Taxonomy" key="1" id="9606"/>
  <lineage>
    <taxon>Mammalia</taxon>
    <taxon>Eutheria</taxon>
  </lineage>
</organism>

...
Source string:

CIS 700 Project - Fall 2005.
@incollection{FJ-afp,
author = {J. Nathan Foster and Dimitrios Vytiniotis},
title = {A Theory of Featherweight {J}ava in 
         I}sabelle/{H}o{L},
booktitle = {The Archive of Formal Proofs},
editor = {Gerwin Klein
         and Tobias Nipkow
         and Lawrence Paulson},
publisher = {http://afp.sf.net},
year = 2006,
month = April,
url = {http://afp.sf.net/entries/FeatherweightJava.shtml},
}
TY - CHAP
ID - FJ-afp
AU - Foster, J. Nathan
AU - Vytiniotis, Dimitrios
ED - Klein, Gerwin
ED - Nipkow, Tobias
ED - Paulson, Lawrence
T1 - A theory of Featherweight Java in Isabelle/HOL
T2 - The Archive of Formal Proofs
PB - http://afp.sf.net
PY - 2006/04/
UR - http://afp.sf.net/entries/FeatherweightJava.shtml
ER -
BibTeX (Lens Definition)

Primitives + regular operators, embedded in $\lambda$-calculus:

```ocaml
let do_types =
  do_type "article" "JOUR"
| do_type "inproceedings" "CONF"
| do_type "misc" "UNPB"
| do_type "inollection" "CHAP"
| do_type "mastersthesis" "THES"
| do_type "manual" "COMP"
| do_type "phdthesis" "THES"
let entry =
  delete_non_entry .
  do_types . newline .
  do_key . newline .
  do_fields . (ws_nl . "}") <-> "ER  

let bibtex2ris = entry* . delete_non_entry
```
BibTeX (Put Direction)

Putting new target:

TY  - CHAP
ID  - FJ-afp
...
T1  - A theory of Featherweight...
ER  -

TY  - JOUR
ID  - Focal-toplas
...
T1  - Combinators for bidirectional...
ER  -

into original source...
...yields new source:

CIS 700 Project - Fall 2005.
@incollection{FJ-afp,
  ... title = {A theory of {F}eatherweight...}, ...
}

@article{Focal-toplas,
  ... title = {Combinators for bidirectional...}, ...
}
Unfortunately, there is a serious problem lurking here.

The *put* component of $l^*$ splits its $T$ and $S$ inputs into sequences of elements

\[ t = t_1 \ . \ t_2 \ . \ t_3 \ldots \]
\[ s = s_1 \ . \ s_2 \ . \ s_3 \ldots \]

then invokes the *put* of $l$ on $t_1$ and $s_1$, on $t_2$ and $s_2$, etc., and then forms a list of the results.

This does not always give us what we want!
A Bad Put

Putting target:

TY - JOUR
ID - Focal-toplas
...
T1 - Combinators for bidirectional...
ER -

TY - CHAP
ID - FJ-afp
...
T1 - A theory of Featherweight...
ER -

into original source...
A Bad Put

...yields mangled source:

CIS 700 Project - Fall 2005.
@article{Focal-toplas,
        ... title = {Combinators for bidirectional...},...
}

@incollection{FJ-afp,
        ... title = {A theory of {F}eatherweight...},...
}
This problem arises whenever lenses are used to deal with list-structured data—sequences where ordering is important and where updates can add, delete, and rearrange elements.

Specifically, it arises when we want to both reorder “chunks” of data in the target and (in the get direction) project away part of each chunk (e.g., the comments associated with each entry).

Our experience writing lenses for a variety of real-world formats suggests that such situations arise frequently in practice.

None of the bidirectional languages proposed in the literature gets this right.
Resourceful Lenses
The Way Forward

We want the *put* function, when it reaches a sequence of reorderable “chunks,” to align chunks from the source and target using a criterion other than their position in the sequence.

- In the example, we want to match up lines that have identical BibTeX keys.

That is, we need to introduce a (weak) notion of *provenance* into our syntax (and semantics!).
Let’s restate what is wrong with the BibTeX example using provenance: entries in the target may be put with entries in the source that are not in their provenance set—e.g.:

TY    - JOUR
ID    - Focal-toplas
... 
T1    - Combinators for bidirectional...
ER    - 

and

CIS 700 Project - Fall 2005.
@incollection{FJ-afp,
  ... title = {A theory of {F}eatherweight...}},
**R-lenses**

**Resourceful lenses** (or *R-lenses*) include annotations for tracking correspondences between pieces of the source and target structures.

Programs identify:

- re-orderable **chunks** in source and target;
- a **key** for each chunk.

The *put* function uses a dictionary of all the source chunks, indexed by key, as a resource for processing target chunks.
let do_key =
    ins (tag "ID") .
    key [^@{},\n ]+ .
    ("," . ws . "\n") <-> ""

let chunk =
    delete_non_entry .
    do_type . newline .
    do_key . newline .
    do_fields . (ws_nl . "}") <-> "ER -\n\n"
let bibtex2ris = (<chunk>)* . delete_non_entry
Extensions and Open Questions

More examples:

► Require more powerful primitives swapping, sorting, permuting, etc.

Richer kinds of “keys”:

► In the current implementation, keys are strings, assembled by concatenating the keys of sub-lenses.

Extend to trees:

► What are good notions of “chunk” and “key” for trees?
► How are they specified?

Can better notions of provenance to put to use:

► In lenses?
► In data synchronizers?
Thank You!

Main collaborators on this work: Aaron Bohannon, Benjamin C. Pierce, Alexandre Pilkiewicz, Alan Schmitt

Other Harmony contributors: Ravi Chugh, Malo Denielou, Michael Greenwald, Owen Gunden, Martin Hofmann, Sanjeev Khanna, Keshav Kunal, Stéphane Lescuyer, Jon Moore, Jeff Vaughan, Zhe Yang

Resources: Papers, slides, (open) source code, and online demos:

http://www.seas.upenn.edu/~harmony/