Symmetric Lenses

Martin Hofmann Benjamin Pierce Daniel Wagner

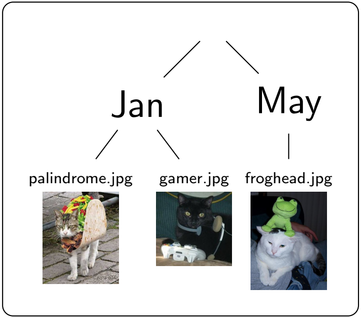
January 27, 2011
POPL Austin

Setup

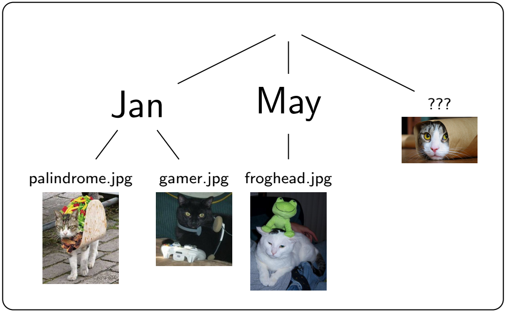
Daniel shares cat pictures with his coworkers, but prefers a different organization scheme than they do.

- At home: tree-structured file-system
- On the web: flat-list picture gallery with tags

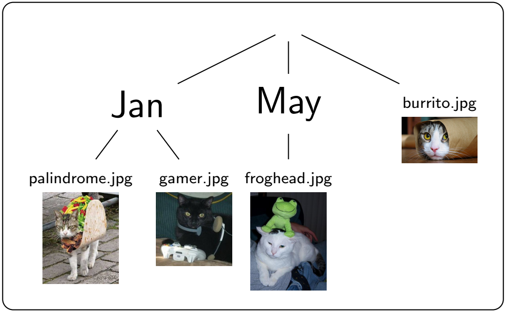
Goal, Part 1: Two Structures



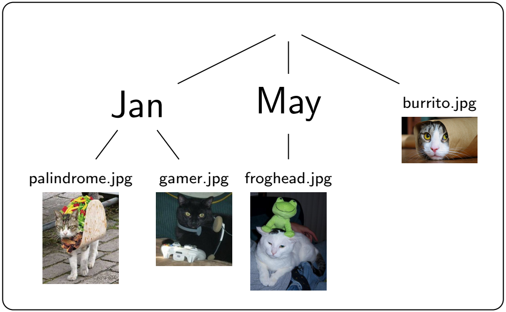
Goal, Part 2: Adding to the Web Gallery



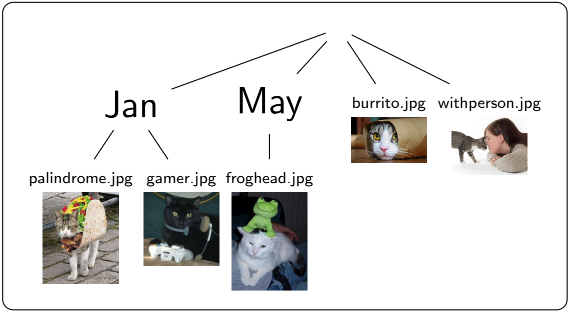
Goal, Part 3: Fixing the Filename



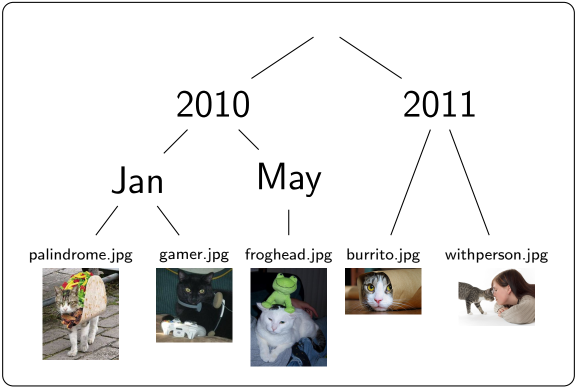
Goal, Part 4: Changing Tags



Goal, Part 5: Adding to the File System



Goal, Part 6: Restructuring

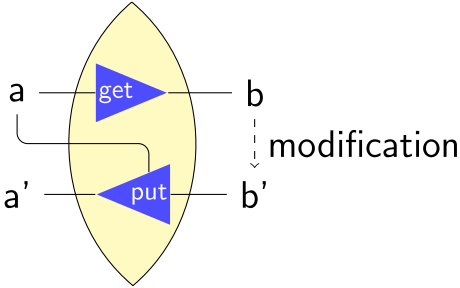


We Get It. . . Use Lenses

Use Lenses, Dummy!

- 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)

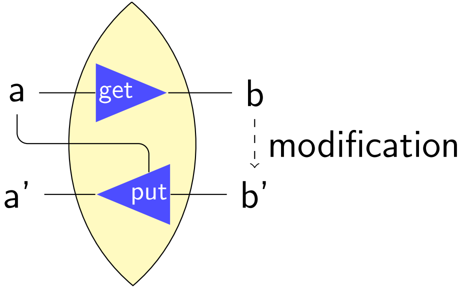
Lenses 101



$$\ell : A \overset{a}{\leftrightarrow} B$$

$$\text{get} : A \to B$$
$$\text{put} : B \times A \to A$$

Lenses 101

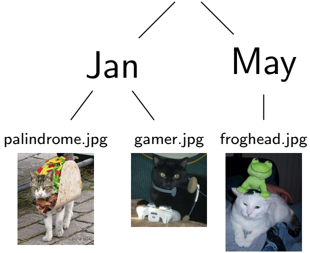


$$\ell : A \stackrel{a}{\leftrightarrow} B$$

$$\text{get} : A \rightarrow B$$

$$\text{put} : B \times A \rightarrow A$$

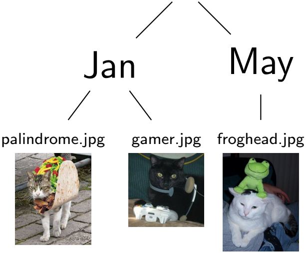
Typing "get"




```
data FS  =  Directory Name [FS]        type Web =
         |  File Name Picture              [(Picture, [Tag])]
```

$$\ell : \mathsf{FS} \overset{?}{\leftrightarrow} \mathsf{Web}$$

Typing "get"



data FS = Directory Name [FS]
 | File Name Picture

type Web =
 [(Picture, [Tag])]

$$\ell : \text{Web} \overset{?}{\leftrightarrow} \text{FS}$$

Formalizing the Oddity: Roundtrip Laws

$$put(get(a), a) = a$$

$$get(put(b, a)) = b$$

Either possibility forbidden!

Well, What About. . .

- 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)

No composition!

Our Contribution

A lens framework with

1. symmetry
2. composition

Our Contribution

A lens framework with

1. symmetry
2. composition
3. ... and other nice combinators

Symmetrizing Lenses

Starting Point: Asymmetric Lenses

$$\ell : A \overset{a}{\leftrightarrow} B$$

$$\begin{array}{lcl} \text{get} & : & A \rightarrow B \\ \text{put} & : & B \times A \rightarrow A \end{array}$$

$$get(put(b, a)) = b$$

$$put(get(a), a) = a$$

L/R Symmetry

$$\ell : A \overset{a}{\leftrightarrow} B$$

$$\begin{aligned} \text{putr} &: A \times B \to B \\ \text{putl} &: B \times A \to A \end{aligned}$$

$$get(put(b, a)) = b$$

$$put(get(a), a) = a$$

Complements

$$\ell : A \overset{a}{\leftrightarrow} B$$

$$\begin{array}{lcl} \text{putr} & : & A \times S_B \rightarrow B \\ \text{putl} & : & B \times S_A \rightarrow A \end{array}$$

$$get(put(b, a)) = b$$

$$put(get(a), a) = a$$

I/O Symmetry

$$\ell : A \overset{a}{\leftrightarrow} B$$

$$\begin{aligned} \text{putr} &: A \times S_B \to B \times S_A \\ \text{putl} &: B \times S_A \to A \times S_B \end{aligned}$$

$$get(put(b, a)) = b$$

$$put(get(a), a) = a$$

Unifying Complements

$$\ell : A \leftrightarrow B$$

$$\begin{array}{rcl} \text{putr} & : & A \times S \rightarrow B \times S \\ \text{putl} & : & B \times S \rightarrow A \times S \end{array}$$

$$get(put(b, a)) = b$$

$$put(get(a), a) = a$$

Updated Lens Laws

$$\ell : A \leftrightarrow B$$

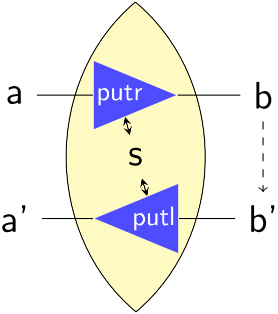
$$\begin{array}{rcl} \text{putr} & : & A \times S \to B \times S \\ \text{putl} & : & B \times S \to A \times S \end{array}$$

$$\frac{putr(a, s) = (b, s')}{putl(b, s') = (a, s')}$$

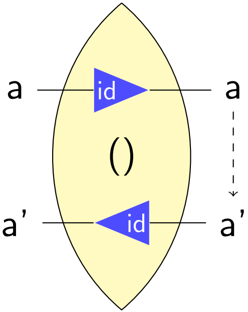
$$\frac{putl(b, s) = (a, s')}{putr(a, s') = (b, s')}$$

Composition

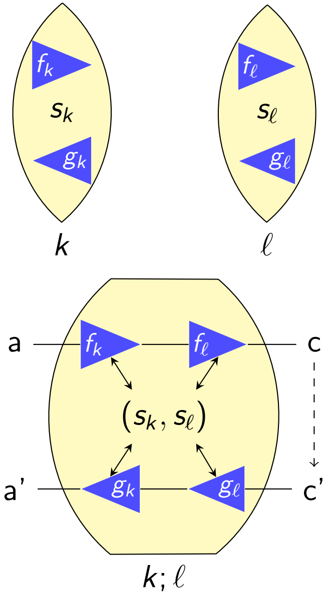
Updated Wiring Diagram



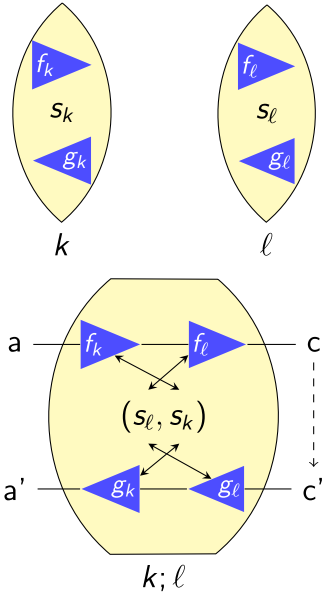
Warm-up: Identity Lens



Composition



Another Composition



Behavioral Equivalence

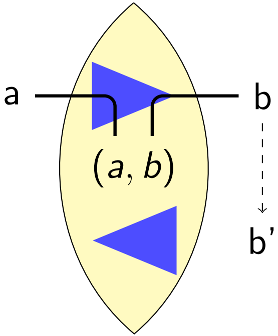
$k \equiv \ell$ when there's a relation $R \subset k.S \times \ell.S$ and:

$$\frac{s_k \; R \; s_\ell \qquad k.putr(a, s_k) = (b_k, s'_k) \qquad \ell.putr(a, s_\ell) = (b_\ell, s'_\ell)}{b_k = b_\ell \wedge s'_k \; R \; s'_\ell}$$

$$\frac{s_k \; R \; s_\ell \qquad k.putl(b, s_k) = (a_k, s'_k) \qquad \ell.putl(b, s_\ell) = (a_\ell, s'_\ell)}{a_k = a_\ell \wedge s'_k \; R \; s'_\ell}$$

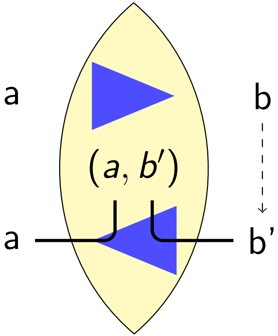
Handy Lenses

Disconnect (putr)



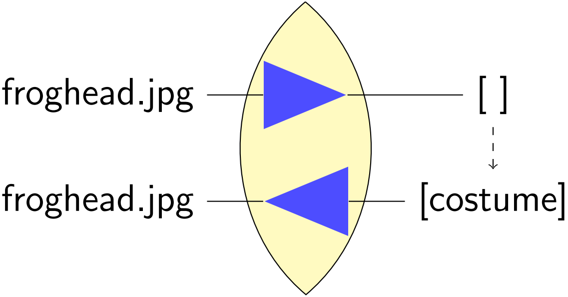
disconnect : $A \leftrightarrow B$

Disconnect (putl)

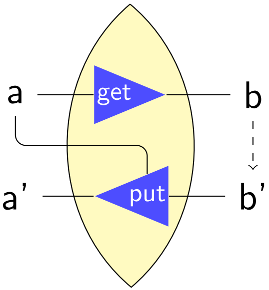


disconnect : $A \leftrightarrow B$

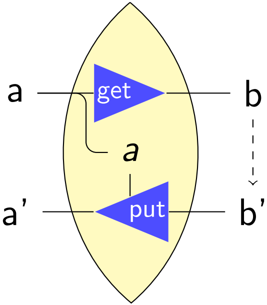
Why disconnect?



Lifting Asymmetric Lenses

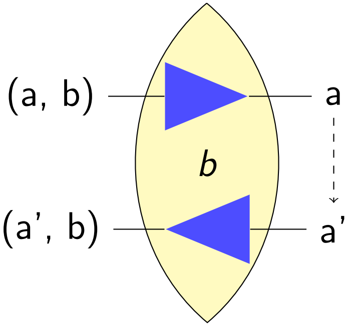


$\ell : A \overset{a}{\leftrightarrow} B$



$\ell^{sym} : A \leftrightarrow B$

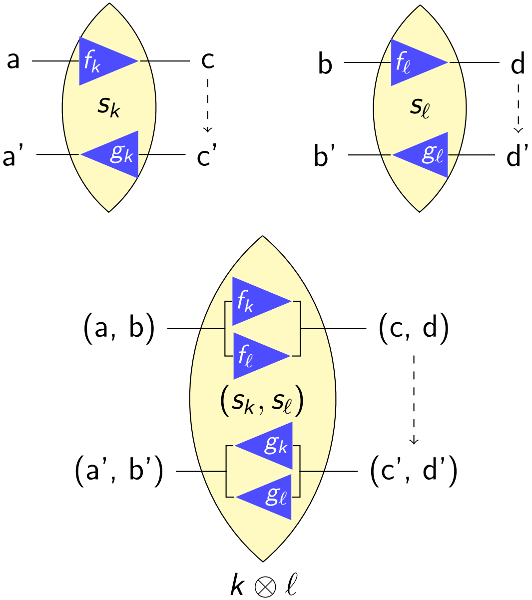
Projection



$\pi_1 : A \times B \leftrightarrow A$

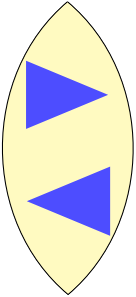
(π_2 is similar)

Tensor Product



Synchronizing Tree Leaves/List Elements

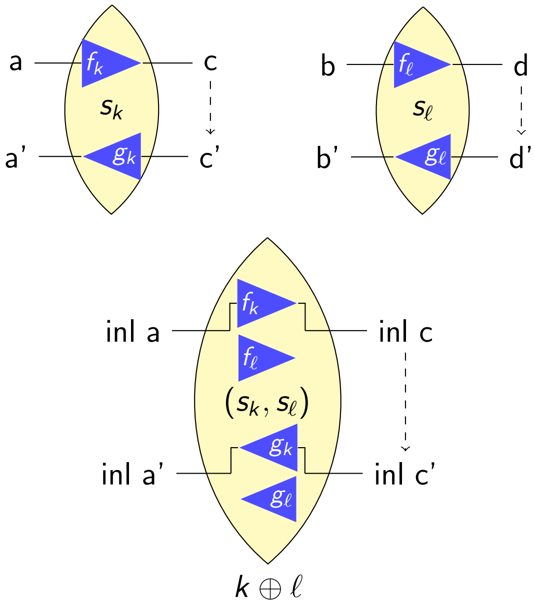
froghead.jpg



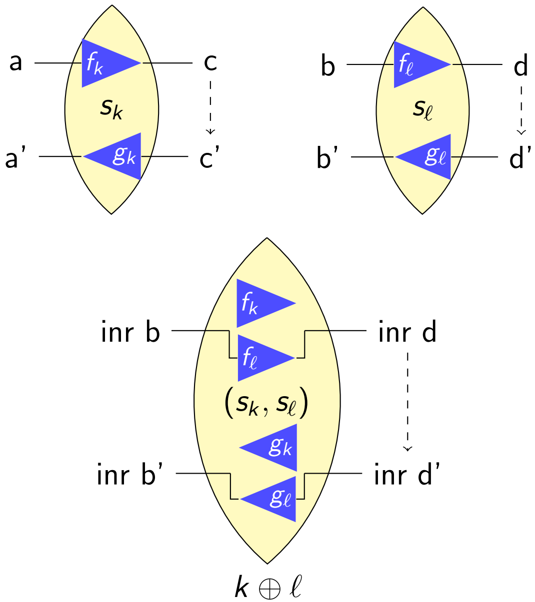
[costume]

disconnect $\otimes$ id

Tensor Sum


a
c
f_k
s_k
a'
c'
g_k
b
d
f_ℓ
s_ℓ
b'
d'
g_ℓ
inl a
inl c
f_k
f_ℓ
(s_k, s_ℓ)
inl a'
inl c'
g_k
g_ℓ
$k \oplus \ell$

Tensor Sum


a
c
f_k
s_k
a'
c'
g_k
b
d
f_ℓ
s_ℓ
b'
d'
g_ℓ
inr b
inr d
f_k
f_ℓ
(s_k, s_ℓ)
g_k
g_ℓ
inr b'
inr d'
$k \oplus \ell$

Lenses for Recursive Data

- $$\frac{\ell \quad : F(A) \leftrightarrow A}{\mathsf{fold}(\ell) : \mu F \leftrightarrow A}$$

- Initial algebra construction for data
- Some technical requirements
- Package up the fold and unfold operations

Useful Folds

- $\text{leaves} : \text{Tree } A \leftrightarrow [A]$
- $\text{concat} : [[A]] \leftrightarrow [A]$
- $\text{partition} : [A \uplus B] \leftrightarrow [A] \times [B]$
- $\text{map} : (A \leftrightarrow B) \rightarrow ([A] \leftrightarrow [B])$

- $\text{pictures} : \text{FS} \leftrightarrow [\text{Name} \times \text{Picture}]$

Final Lens

$$
\begin{array}{c}
\mathsf{FS} \\
\updownarrow \ \mathsf{pictures} \\
[\mathsf{Filename} \times \mathsf{Picture}] \\
\updownarrow \ \mathsf{map}(\mathsf{disconnect} \otimes \mathsf{id}) \\
[[\mathsf{Tag}] \times \mathsf{Picture}]
\end{array}
$$

Conclusion

- Theoretical framework
  - Symmetric and compositional
  - Behavioral equivalence
- Lens language
  - Miscellaneous useful basic lenses
  - Tensor sums and products, projections, injections
  - ADTs via folds and unfolds
  - Mapping for (non-algebraic) containers
- Relationship to asymmetric lenses
  - Embedding of asymmetric lenses
  - Decomposition into asymmetric lens spans