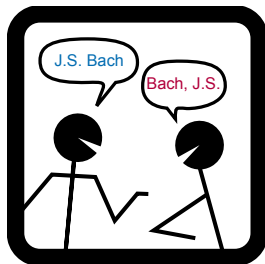


Edit Lenses



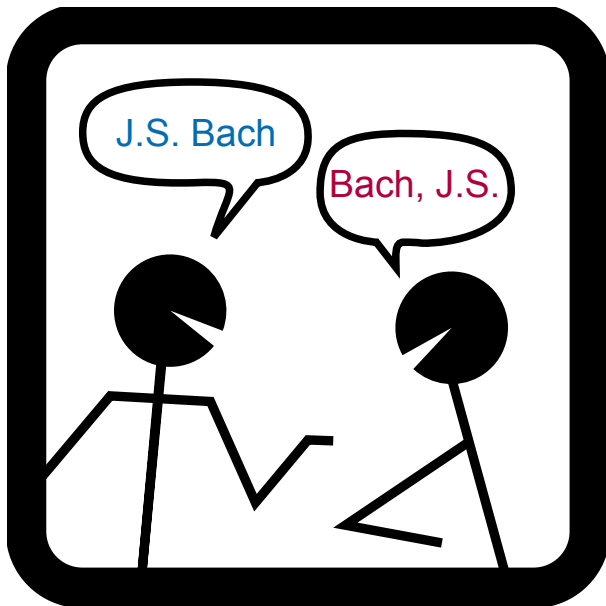
Martin Hofmann

Benjamin Pierce

Daniel Wagner

January 27, 2012
POPL Philadelphia

A brief history of lenses



Isomorphisms [Braun et al, 2003; Brabrand et al, 2007]

(Johann, Bach)

Isomorphisms [Braun et al, 2003; Brabrand et al, 2007]

$$(\text{Johann}, \text{Bach}) \xrightarrow{\lambda(x, y). (y, x)} (\text{Bach}, \text{Johann})$$

Isomorphisms [Braun et al, 2003; Brabrand et al, 2007]

$$(\text{Johann}, \text{Bach}) \xrightarrow{\lambda(x, y). (y, x)} (\text{Bach}, \text{Johann})$$

user action

(Bach, J. S.)

Isomorphisms [Braun et al, 2003; Brabrand et al, 2007]

$$(\text{Johann, Bach}) \xrightarrow{\lambda(x, y). (y, x)} (\text{Bach, Johann})$$

user action

$$(\text{J. S., Bach}) \xleftarrow{\lambda(y, x). (x, y)} (\text{Bach, J. S.})$$

Asymmetric lenses [Foster et al, 2007; Bohannon et al, 2008]

(Johann, Bach, 1685)

Asymmetric lenses [Foster et al, 2007; Bohannon et al, 2008]

$$\lambda(x, y, z). (y, x)$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann)

Asymmetric lenses [Foster et al, 2007; Bohannon et al, 2008]

$$\lambda(x, y, z). (y, x)$$

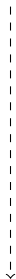
(Johann, Bach, 1685) \longrightarrow (Bach, Johann)

(Bach, J. S.)

Asymmetric lenses [Foster et al, 2007; Bohannon et al, 2008]

$$\lambda(x, y, z). (y, x)$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann)



(J. S., Bach, 1685) \longleftarrow (Bach, J. S.)

$$\lambda(y, x) (-, -, z). (x, y, z)$$

Symmetric lenses [Hofmann et al, 2011]

(Johann, Bach, 1685)

Symmetric lenses [Hofmann et al, 2011]

(Johann, Bach, 1685)

(1685, Air on G)

Symmetric lenses [Hofmann et al, 2011]

$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$

(Johann, Bach, 1685) \longrightarrow

(1685, Air on G)

Symmetric lenses [Hofmann et al, 2011]

$$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann, Air on G)
(1685, Air on G)

Symmetric lenses [Hofmann et al, 2011]

$$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann, Air on G)

(1685, Air on G)



(Bach, J. S., Goldberg Variations)

Symmetric lenses [Hofmann et al, 2011]

$$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann, Air on G)

(1685, Air on G)



\longleftarrow (Bach, J. S., Goldberg Variations)

$$\lambda(y, x, w)(z, -). ((x, y, z), (z, w))$$

Symmetric lenses [Hofmann et al, 2011]

$$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann, Air on G)
(1685, Air on G)

(J. S., Bach, 1685) \longleftarrow (Bach, J. S., Goldberg Variations)

$$\lambda(y, x, w)(z, -). ((x, y, z), (z, w))$$

Symmetric lenses [Hofmann et al, 2011]

$$\lambda(x, y, z) (-, w). ((y, x, w), (z, w))$$

(Johann, Bach, 1685) \longrightarrow (Bach, Johann, Air on G)
(1685, Air on G)

(1685, Goldberg Variations)

(J. S., Bach, 1685) \longleftarrow (Bach, J. S., Goldberg Variations)

$$\lambda(y, x, w)(z, -). ((x, y, z), (z, w))$$

Alignment

(Johann, Bach, 1685) → (Bach, Johann, Air on G)
(Franz, Liszt, 1811) → (Liszt, Franz, Liebesträume)

Alignment

(Johann, Bach, 1685)	→	(Bach, Johann, Air on G)
(Franz, Liszt, 1811)		(Liszt, Franz, Liebesträume)
		(1685, Air on G)
		(1811, Liebesträume)

Alignment

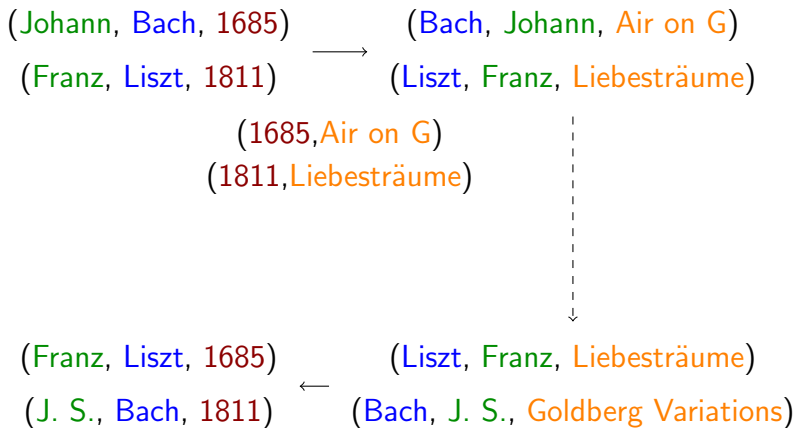
(Johann, Bach, 1685) (Bach, Johann, Air on G)
(Franz, Liszt, 1811) (Liszt, Franz, Liebesträume)

(1685, Air on G)
(1811, Liebesträume)

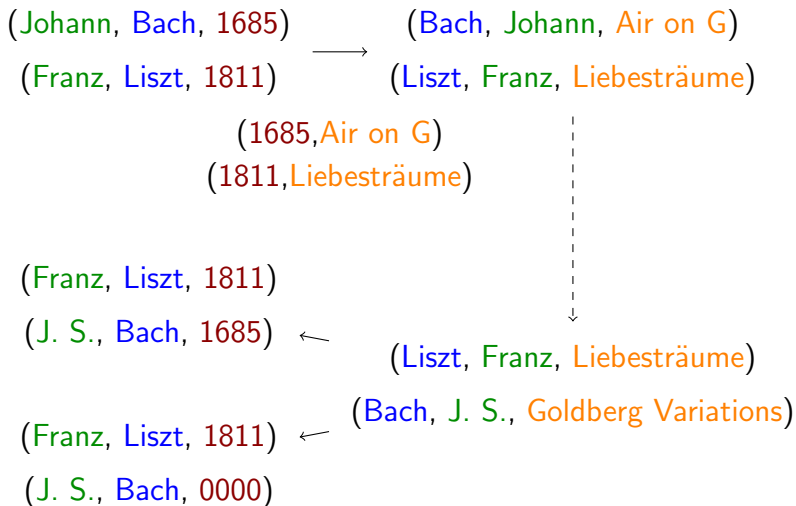


(Liszt, Franz, Liebesträume)
(Bach, J. S., Goldberg Variations)

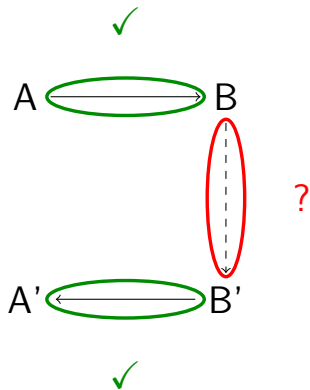
Alignment



Alignment



How to improve?



Contributions

- ▶ Theoretical framework
 - ▶ Model for first-class edits
 - ▶ Formulation of lenses on edits
 - ▶ Adaptation of behavioral laws
 - ▶ Lens syntax
 - ▶ Composition, products, sums
 - ▶ Filtering, mapping, reshaping
 - ▶ Embedding of state-based lenses
 - ▶ Haskell library
-

First-class edits

Edits are a **monoid** M :

$$\mathbf{1}_M \cdot m = m \cdot \mathbf{1}_M = m$$

$$m_1 \cdot (m_2 \cdot m_3) = (m_1 \cdot m_2) \cdot m_3$$

With a **partial monoid action** $\odot \in M \times X \rightarrow X$:

$$\mathbf{1}_M \odot x = x$$

$$(m_1 \cdot m_2) \odot x = m_1 \odot (m_2 \odot x)$$

Editing lists

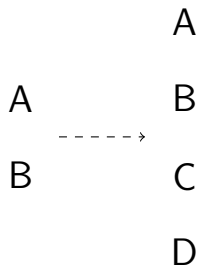
Set ∂X are edits for X .

Define atomic edits E for X^* :

- ▶ $\text{modify}(p, dx)$ where $p \in \mathbb{N}$, $dx \in \partial X$
- ▶ $\text{resize}(i, j, x)$ where $i, j \in \mathbb{N}$, $x \in X$
- ▶ $\text{reorder}(i, f)$ where f permutes $\{0, \dots, i\}$

Take E^* (words of atomic edits) for list edits $\partial(X^*)$.

Edit sequence example



[resize(2, 4, C), modify(3, dx)]

Another example

A	A'
B	-----> B'
C	C'

[modify(0, dx), modify(1, dx'), modify(2, dx'')]

But...

What about this?

[resize(1, 2, A), resize(2, 1, A), resize(1, 2, A), ...]

Make it a monoid

Introduce an operation

$$\cdot \in E^* \times E^* \rightarrow E^*$$

that concatenates, then optimizes.

Respect optimization!

$$(e \cdot e') \odot x = e \odot (e' \odot x)$$

✓ Edits

Lenses

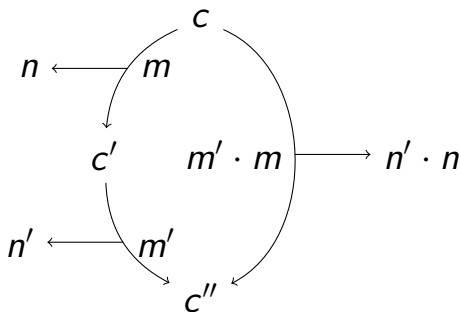
Stateful monoid homomorphisms

Definition

A **stateful monoid homomorphism**

$h : M \times C \rightarrow N \times C$ satisfies

$$\begin{array}{ccc} & C & \\ & | & \\ \mathbf{1}_M & \longrightarrow & \mathbf{1}_N \\ & | & \\ & C & \end{array}$$



Monoid homomorphisms

Definition

A **monoid homomorphism**
 $h : M \rightarrow N$ satisfies

$$n \longleftarrow m$$

$$\mathbf{1}_M \longrightarrow \mathbf{1}_N$$

$$m' \cdot m \longrightarrow n' \cdot n$$

$$n' \longleftarrow m'$$

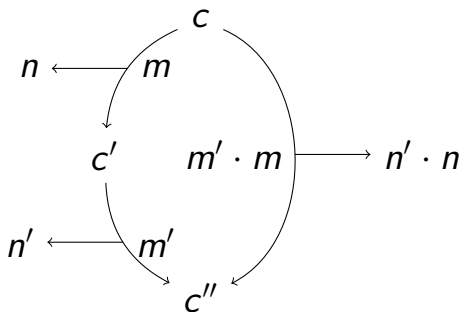
Stateful monoid homomorphisms

Definition

A **stateful monoid homomorphism**

$h : M \times C \rightarrow N \times C$ satisfies

$$\begin{array}{ccc} & C & \\ & | & \\ \mathbf{1}_M & \longrightarrow & \mathbf{1}_N \\ & | & \\ & C & \end{array}$$



Lens definition

Definition

Edit lens $\ell : \langle M, X \rangle \leftrightarrow \langle N, Y \rangle$ has:

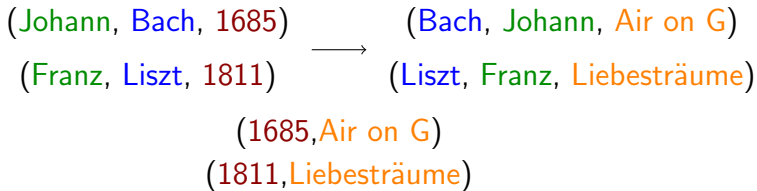
- ▶ a complement set C of private data
- ▶ consistency relation $K \in X \times C \times Y$
- ▶ stateful monoid homomorphisms

$$\Rightarrow : M \times C \rightarrow N \times C$$

$$\Leftarrow : N \times C \rightarrow M \times C$$

that preserve consistency

Consistency



What do we do with *modify*(9999, dx)?

Consistency vs. round-trip laws

Round-trip laws:

“There exists an invariant restored by the lens.”

Consistency relations:

“There exists an invariant restored by the lens,
and that invariant is K .”

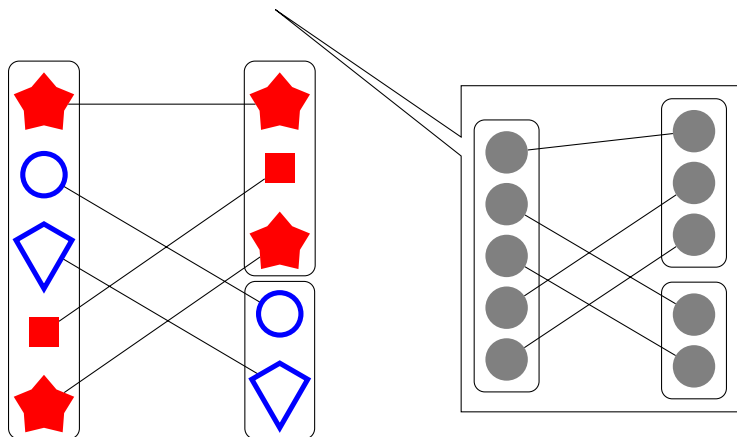
Partition: the code view

$partition \in (X \oplus Y)^* \leftrightarrow X^* \otimes Y^*$		
C	$= \{L, R\}^*$	
$init$	$= \varepsilon$	
K	$= \{(z, \text{map}_{\text{tagof}}(z), (\text{lefts}(z), \text{rights}(z))) \mid z \in (X + Y)^*\}$	
$\Rightarrow_g(\text{mod}(p, dv), c)$	$= (\text{fail}, c)$ when $p > c $	(1)
$\Rightarrow_g(\text{mod}(p, \varepsilon), c)$	$= (\varepsilon, c)$ when $1 \leq p \leq c $	(2)
$\Rightarrow_g(\text{mod}(p, dv\text{dvs}), c)$	$= (d', d', c'')$ where $1 \leq n$ $(d', c') = \Rightarrow_g(\text{mod}(p, \text{dvs}), c)$	(3)
$\Rightarrow_g(\text{mod}(p, \text{switch}_{jk}(\text{dv})), c)$	$= (d_2 d_1 d_0, c[p \mapsto k])$, where $(p_L, p_R) = \text{count}(p, c)$ $d_0 = \text{map}_{\lambda d. \text{tag}(j, d)}(\text{del}'(p_j))$	(4)
	$d_2 = \text{tag}(k, \text{mod}(p_k, \text{dv}))$ $d_1 = \text{map}_{\lambda d. \text{tag}(k, d)}(\text{ins}'(p_k))$	
$\Rightarrow_g(\text{mod}(p, \text{stay}_j(\text{dv})), c)$	$= (\text{tag}(j, \text{mod}(p_j, \text{dv})), c)$, where $(p_L, p_R) = \text{count}(p, c)$	(5)
$\Rightarrow_g(\text{mod}(p, \text{fail}), c)$	$= (\text{fail}, c)$	(6)
$\Rightarrow_g(\text{ins}(i), c)$	$= (\text{left}(\text{ins}(i)), \text{ins}(i) c)$	(7)
$\Rightarrow_g(\text{del}(i), c)$	$= (d_1 d_0, \text{del}(i) c)$, where $c' = \text{reverse}(c)$ $d_0 = \text{left}(\text{del}(n_L - 1))$	(8)
	$(n_L, n_R) = \text{count}(i+1, c')$ $d_1 = \text{right}(\text{del}(n_R - 1))$	
$\Rightarrow_g(\text{reorder}(f), c)$	$= (d_L d_R, c')$, where $h = \text{iso}(c)$ $c' = \text{reorder}(f) c$	(9)
	$h' = \text{iso}(c')$ $(n_L, n_R) = \text{count}(c , c)$	
	$h'' = h'^{-1}; f(c); h$ $f_k(n \neq n_k) = \lambda p. p$	
	$d_L = \text{left}(\text{reorder}(f_L))$ $f_L(n_L) = \text{inl}; h''; \text{out}$	
	$d_R = \text{right}(\text{reorder}(f_R))$ $f_R(n_R) = \text{inr}; h''; \text{out}$	
$\Rightarrow_g(\text{fail}, c)$	$= (\text{fail}, c)$	(10)
$\Leftarrow_g(\varepsilon, c)$	$= (\varepsilon, c)$	(11)
$\Leftarrow_g(\text{dvs}, c)$	$= (d', d', c'')$ when $n > 1$, where $(d, c') = \Leftarrow_g(\text{dvs}, c)$ $(d', c'') = \Leftarrow_g(\text{dv}, c')$	(12)
$\Leftarrow_g(\text{left}(\text{mod}(p, \text{dx})), c)$	$= (\text{stay}_L(\text{mod}(p', \text{dx})), c)$, where $p' = \text{iso}(c)^{-1}(\text{inl}(p))$	(13)
$\Leftarrow_g(\text{left}(\text{reorder}(f)), c)$	$= (\text{reorder}(f'), c)$, where $g(\text{inl}(p)) = \text{inr}(p)$ $f'(n \neq c) = \lambda p. p$	(14)
	$g(\text{inl}(p)) = \text{inl}(f(n_L)(p))$ $f'(c) = h; g; h^{-1}$	
	$(n_L, n_R) = \text{count}(c , c)$ $h = \text{iso}(c)$	
$\Leftarrow_g(\text{left}(\text{ins}(i)), c)$	$= (\text{ins}(i), \text{ins}(i) c)$	(15)
$\Leftarrow_g(\text{left}(\text{del}(0)), c)$	$= (\varepsilon, c)$	(16)
$\Leftarrow_g(\text{left}(\text{del}(i)), c)$	$= (d'' \text{del}'(p), c'')$, where $h = \text{iso}(c)$ $(n_L, n_R) = \text{count}(c , c)$	(17)
	$p = h^{-1}(\text{inl}(n_L))$ $(d'', c'') = \Leftarrow_g(d', c')$	
	$c' = \text{del}'(p) c$ $d' = \text{left}(\text{del}(i-1))$	
	when $0 < i \leq n_L + 1$	
$\Leftarrow_g(\text{left}(\text{del}(i)), c)$	$= (\text{fail}, c)$ otherwise	(18)
$\Leftarrow_g(\text{left}(\text{fail}), c)$	$= (\text{fail}, c)$	(19)
$\Leftarrow_g(\text{right}(\text{dy}), c)$	similar	

Partition: the consistency view

$$K \subset (A + B)^* \times C \times (A^* \times B^*)$$

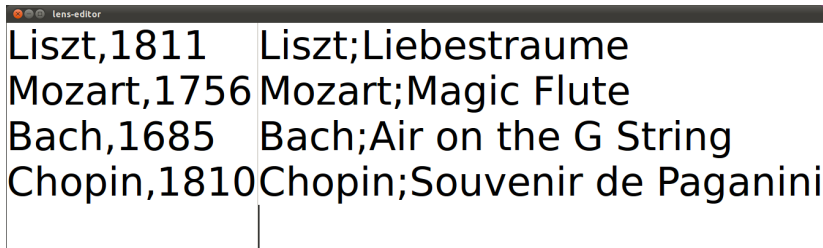
$$K = \{(z, \dots, (\text{lefts}(z), \text{rights}(z)))\}$$



Contributions

- ▶ Theoretical framework
 - ▶ Model for first-class edits
 - ▶ Formulation of lenses on edits
 - ▶ Adaptation of behavioral laws
 - ▶ Lens syntax
 - ▶ Composition, products, sums
 - ▶ Filtering, mapping, reshaping
 - ▶ Embedding of state-based lenses
 - ▶ Haskell library
-

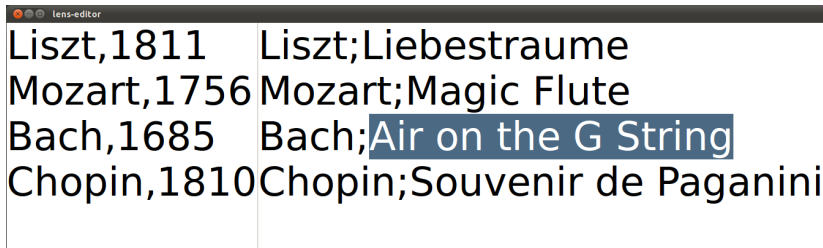
Demo



The image shows a screenshot of a window titled "lens-editor". The window contains a table with four rows of data. Each row consists of a composer's name and year on the left, and the composer's name followed by a semicolon and the title of a work on the right. The text is rendered in a monospaced font.

Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Bach,1685	Bach;Air on the G String
Chopin,1810	Chopin;Souvenir de Paganini

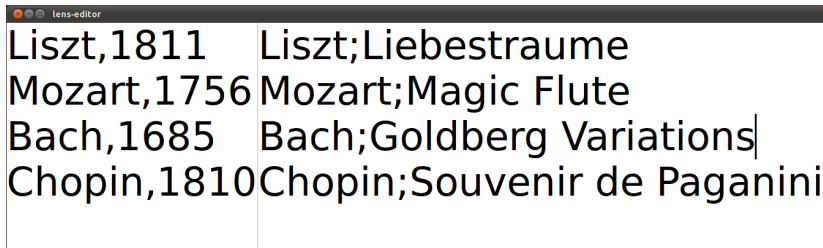
Demo



A screenshot of a window titled "lens-editor" showing a table with two columns. The first column contains composer names and years, and the second column contains the names of musical pieces. The piece "Air on the G String" by Bach is highlighted with a blue background.

Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Bach,1685	Bach;Air on the G String
Chopin,1810	Chopin;Souvenir de Paganini

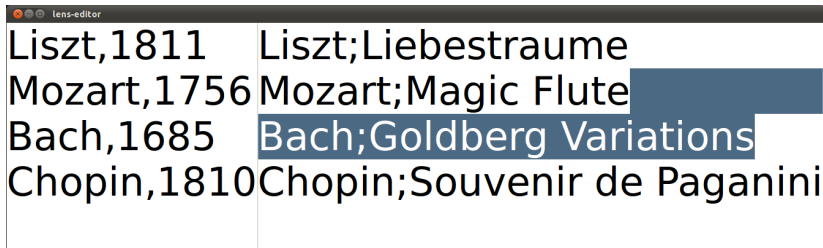
Demo



Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Bach,1685	Bach;Goldberg Variations
Chopin,1810	Chopin;Souvenir de Paganini

[Modify 2 ([], [Delete 0 8, Insert 0 "Goldberg Variations"])]

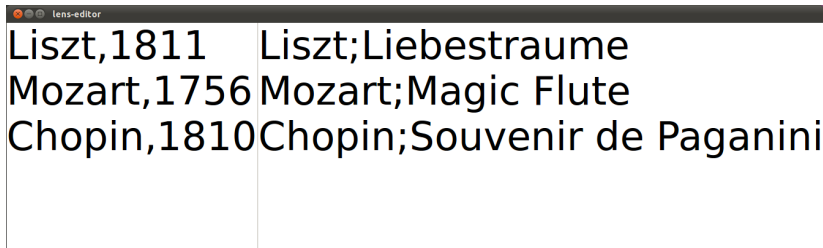
Demo



The image shows a screenshot of a window titled "lens-editor". The window contains a table with two columns. The first column lists composers and their birth years, and the second column lists their works. The text "Bach;Goldberg Variations" is highlighted in blue.

Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Bach,1685	Bach;Goldberg Variations
Chopin,1810	Chopin;Souvenir de Paganini

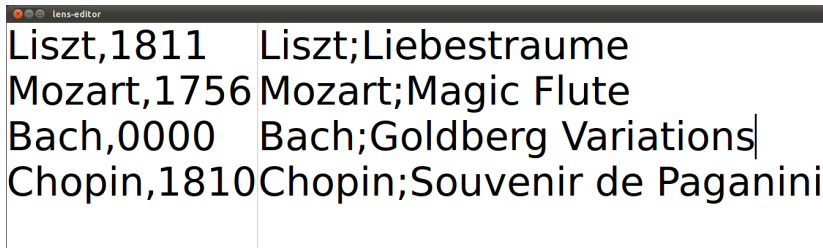
Demo



Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Chopin,1810	Chopin;Souvenir de Paganini

[Rearrange [0,1,3,2], Delete 1]

Demo



The screenshot shows a window titled "lens-editor" containing a table with two columns. The first column lists composers and their birth years, and the second column lists their works. The rows are: Liszt, 1811; Mozart, 1756; Bach, 0000; and Chopin, 1810. The cursor is positioned at the end of the text "Bach;Goldberg Variations|".

Liszt,1811	Liszt;Liebestraume
Mozart,1756	Mozart;Magic Flute
Bach,0000	Bach;Goldberg Variations
Chopin,1810	Chopin;Souvenir de Paganini

[Insert 1,
Modify 3 ([Insert 0 "Bach"], [Insert 0 "Goldberg Variations"]),
Rearrange [0,1,3,2]]

Other approaches to edits

as congruence classes of functions

- ▶ monoids subsume functions
- ▶ can have more intensional representation
- ▶ “Towards an algebraic theory of bidirectional transformations”, Stevens, ICGT 2008

as skeleton structures

- ▶ not specific to containers
- ▶ fundamental objects are less complex
- ▶ “Matching lenses: alignment and view update”, Barbosa, et al., 2010

as a category

- ▶ see paper for partial actions vs. arrows
 - ▶ smaller edits: need not store entire source/target structures in edit
 - ▶ more syntax and combinators available
 - ▶ “From state- to delta-based bidirectional model transformations: The (a)symmetric case”, Diskin, et al., 2011
-

Questions

```
cabal install edit-lenses-demo
```

Whoa, hold up... partial?

Can't we just do nothing?

Consider the following optimization:

$$[\text{resize}(j, k, x)] \cdot [\text{resize}(i, j, x)] = [\text{resize}(i, k, x)]$$

Bad optimization

