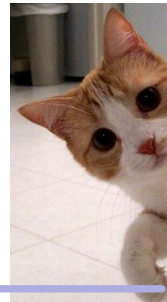
Generalizing Lenses

Daniel Wagner

August 19, 2013



Thesis Proposal

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.

Thesis Proposal

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

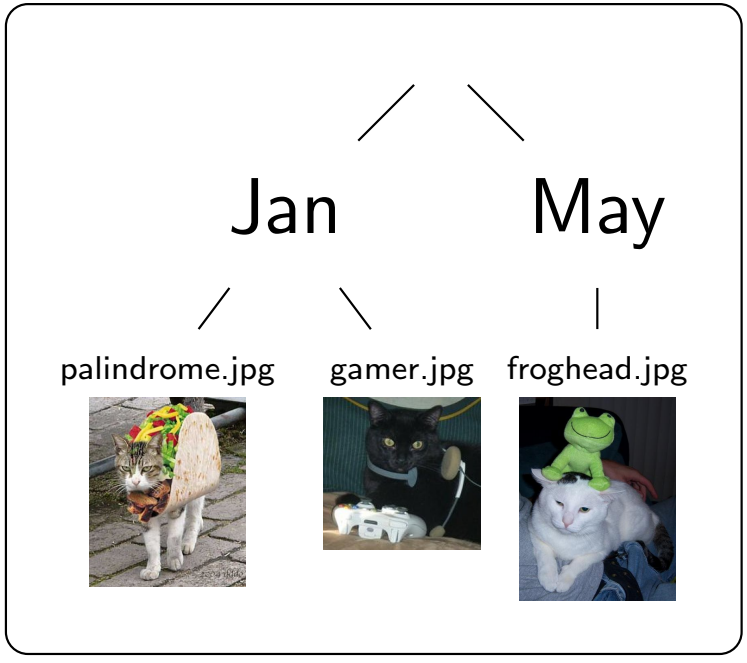
Symmetry

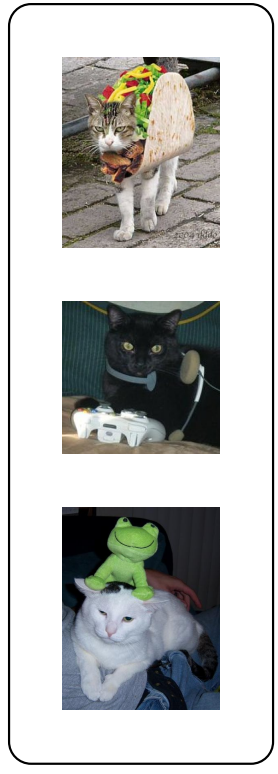
Edits

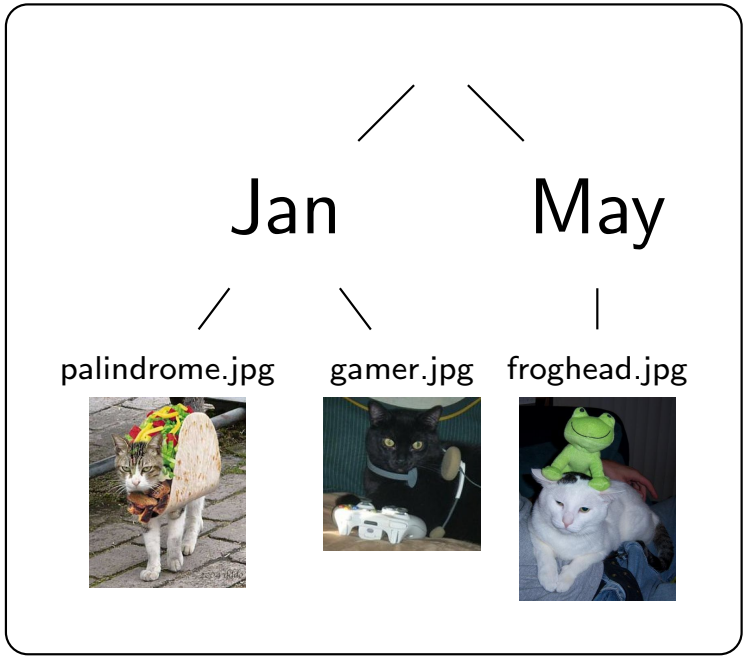
Multidirectionality

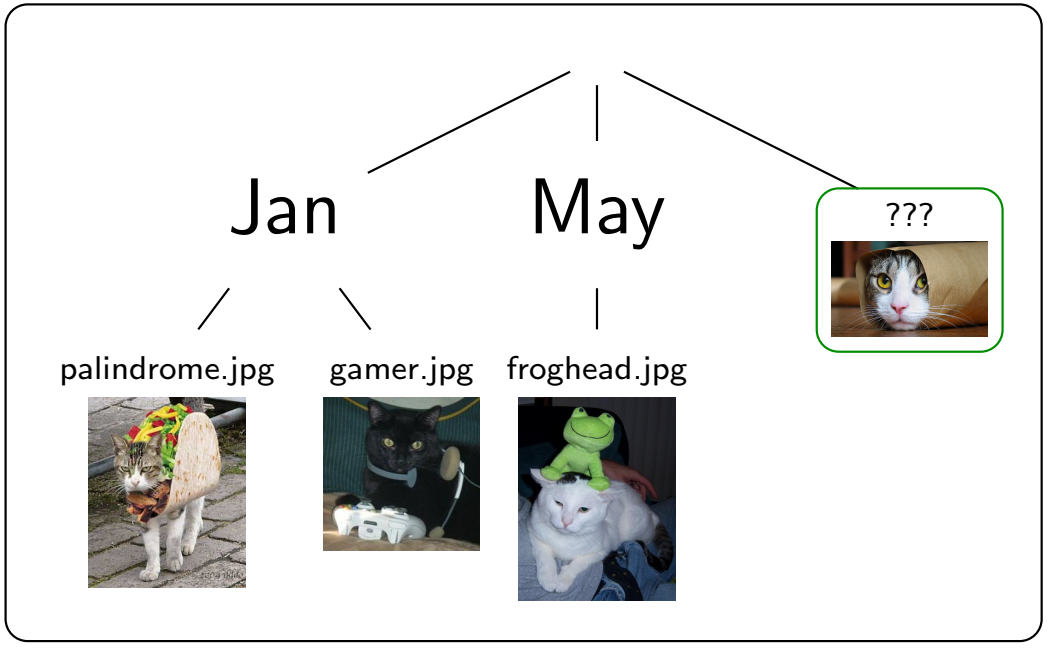
Logistics

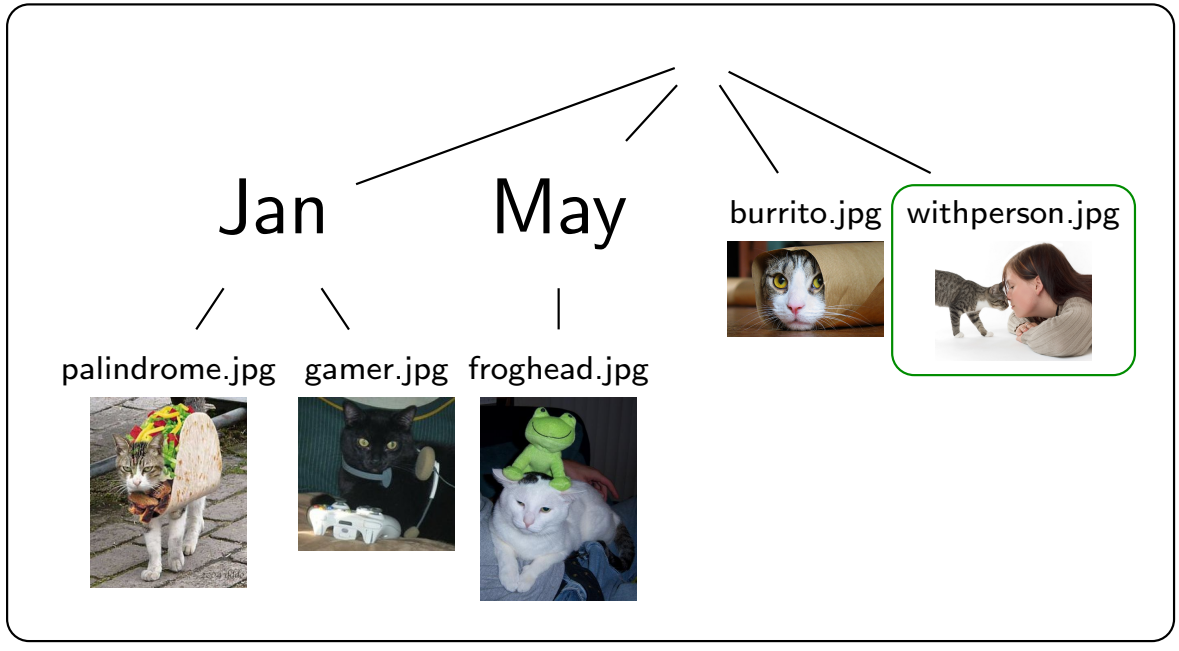
Traditional lenses

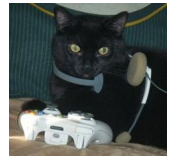

Jan
May
palindrome.jpg
gamer.jpg
froghead.jpg

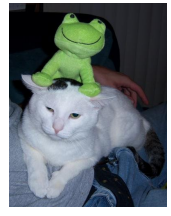


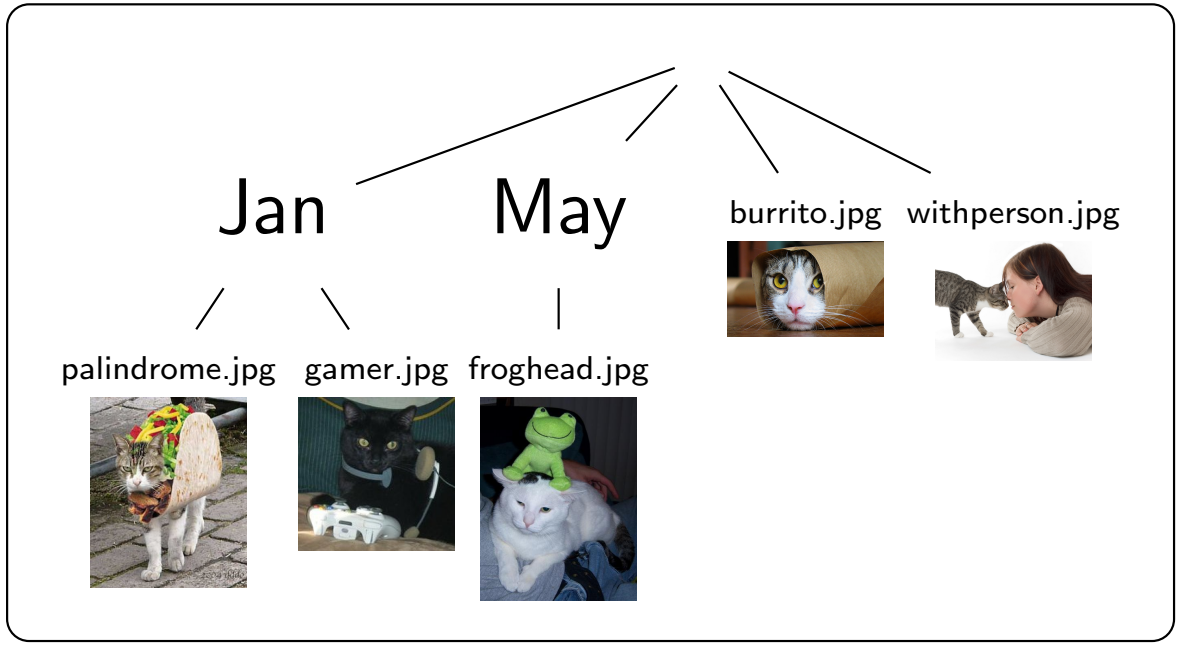

Jan
May
palindrome.jpg
gamer.jpg
froghead.jpg


Jan
May
???
palindrome.jpg
gamer.jpg
froghead.jpg


Jan
May
burrito.jpg
withperson.jpg
palindrome.jpg
gamer.jpg
froghead.jpg






Jan
May
burrito.jpg
withperson.jpg
palindrome.jpg
gamer.jpg
froghead.jpg

Abstract model

A lens $\ell \in X \overset{a}{\leftrightarrow} Y$ has components

$$get \in X \to Y$$
$$put \in Y \times X \to X$$

Abstract model

A lens $\ell \in X \overset{a}{\leftrightarrow} Y$ has components

$$get \in X \to Y$$
$$put \in Y \times X \to X$$

Synchronizing too often doesn't hurt.

$$get(put(y, x)) = y$$
$$put(get(x), x) = x$$

Abstract model

A lens $\ell \in X \overset{a}{\leftrightarrow} Y$ has components

$$get \in X \to Y$$
$$put \in Y \times X \to X$$

Synchronizing too often doesn't hurt.

$$get(put(y, x)) = y$$
$$put(get(x), x) = x$$
$$put(y', put(y, x)) = put(y', x)$$

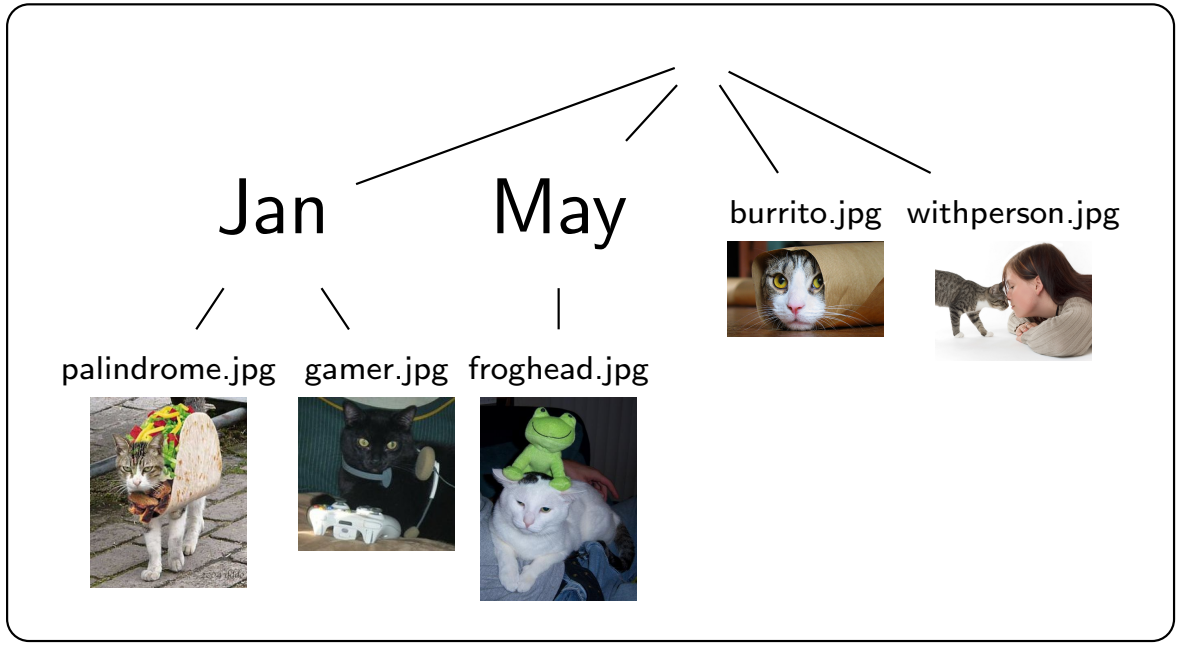
Not synchronizing often enough doesn't hurt.

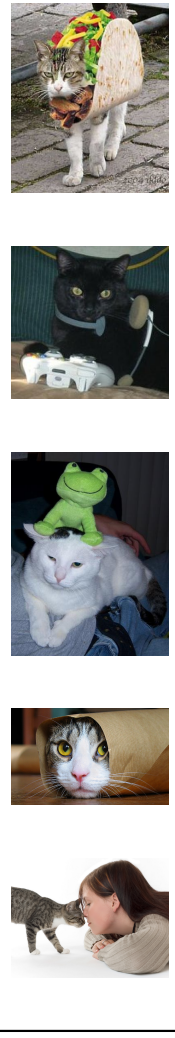
Related work: asymmetric lenses

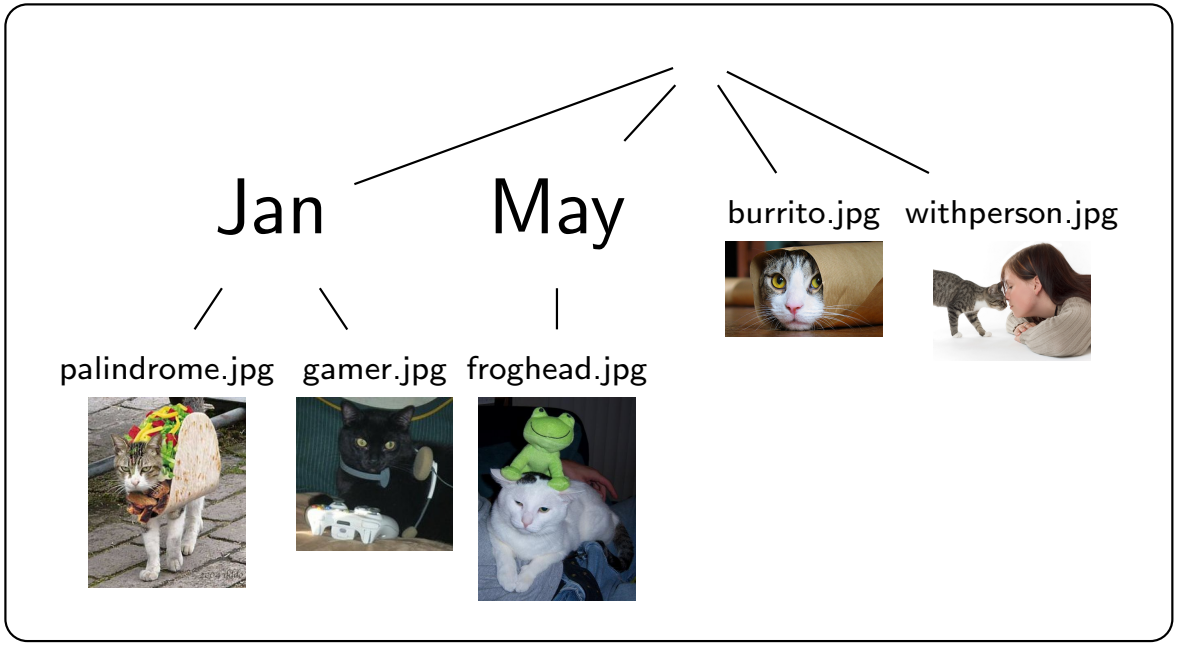
- Combinators for Bidirectional Tree Transformations
  (Foster, Greenwald, Moore, Pierce, Schmitt;
  POPL 2005)
- Relational Lenses: A Language For Updateable Views
  (Bohannon, Vaughn, and Pierce; PODS 2006)
- Boomerang: Resourceful Lenses for String Data
  (Bohannon, Foster, Pierce, Pilkiewicz, and 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

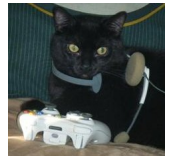
(in collaboration with Martin Hofmann and Benjamin Pierce)

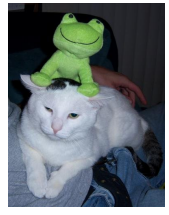

Jan
May
burrito.jpg
withperson.jpg
palindrome.jpg
gamer.jpg
froghead.jpg





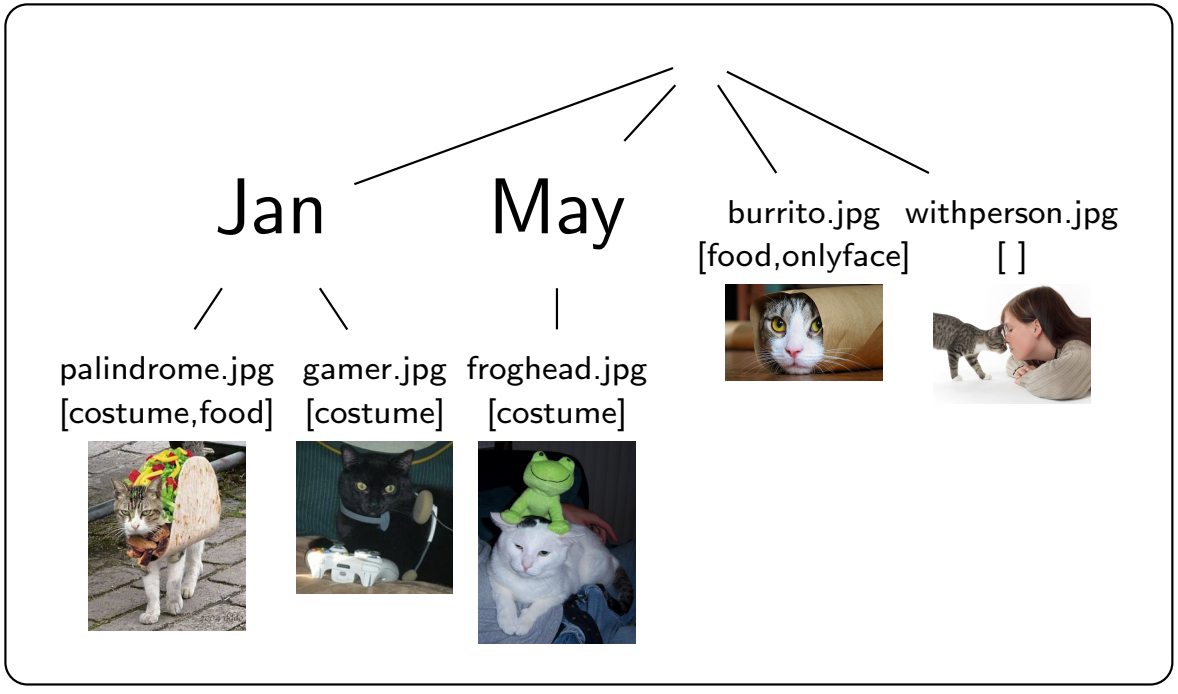
[costume,food]

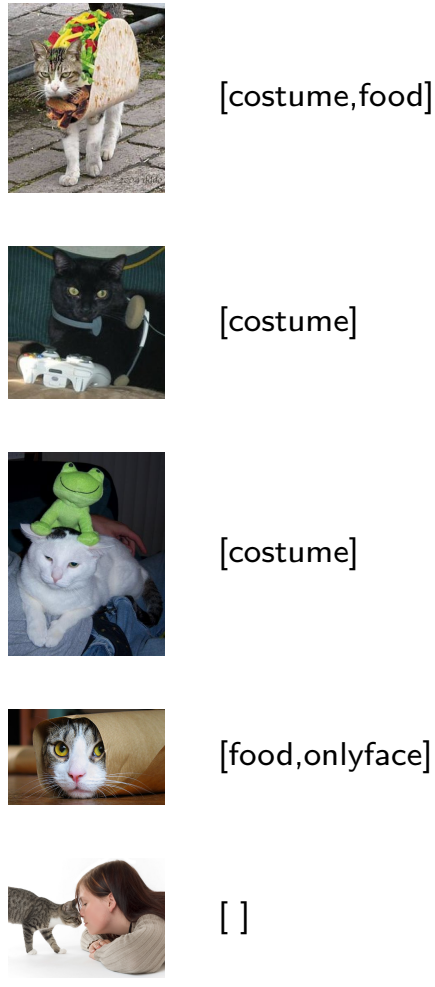

[costume]


[costume]

[food,onlyface]

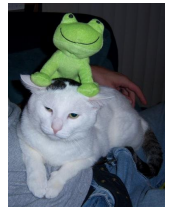
[ ]


Jan
May
burrito.jpg
[food,onlyface]
withperson.jpg
[ ]
palindrome.jpg
[costume,food]
gamer.jpg
[costume]
froghead.jpg
[costume]


[costume,food]
[costume]
[costume]
[food,onlyface]
[ ]

[costume,food]

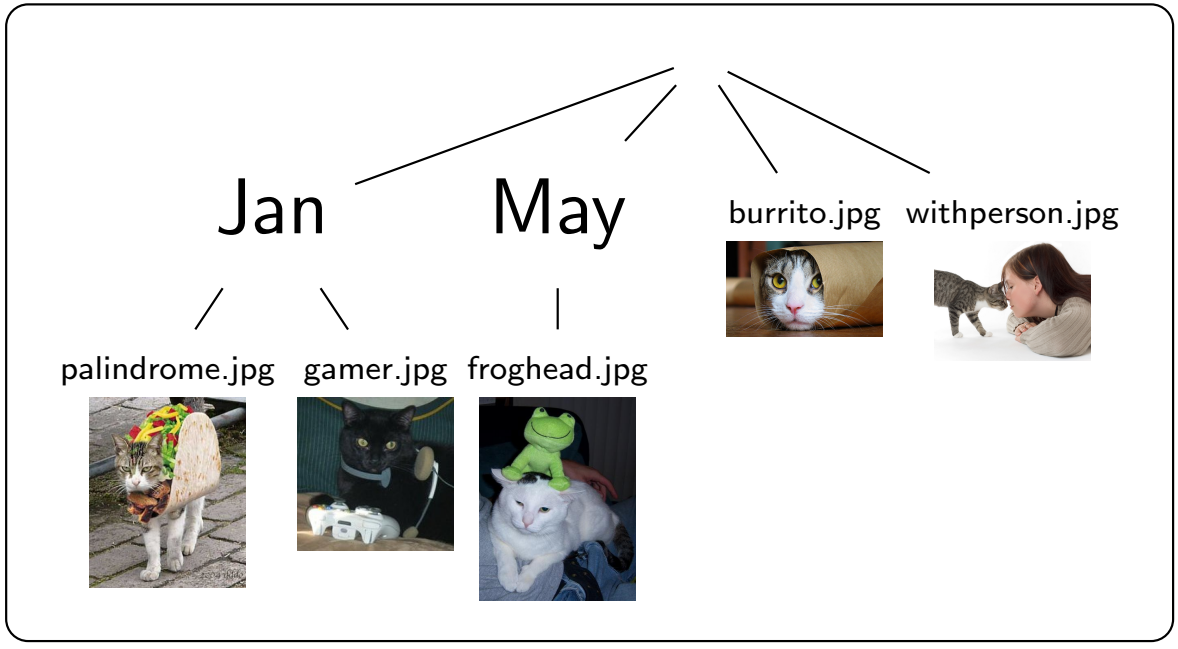
[costume]

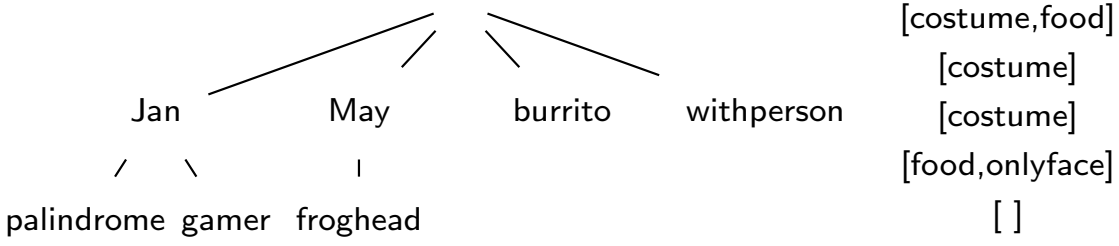


[costume]

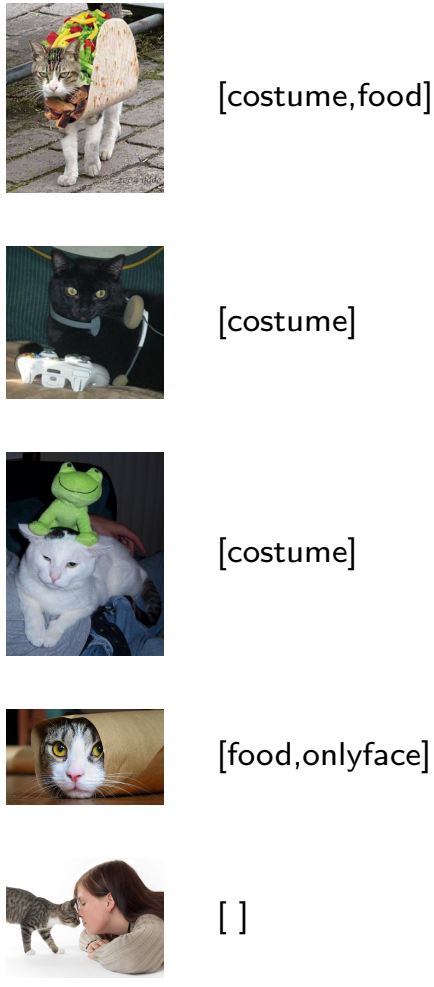
[food,onlyface]

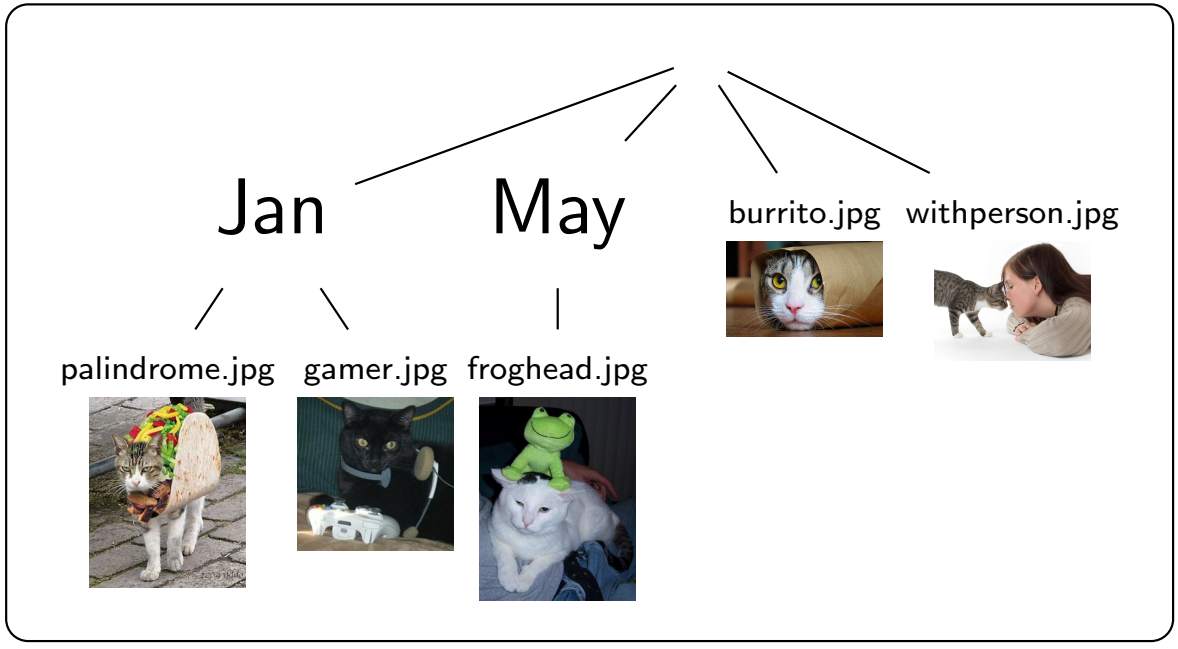
[ ]

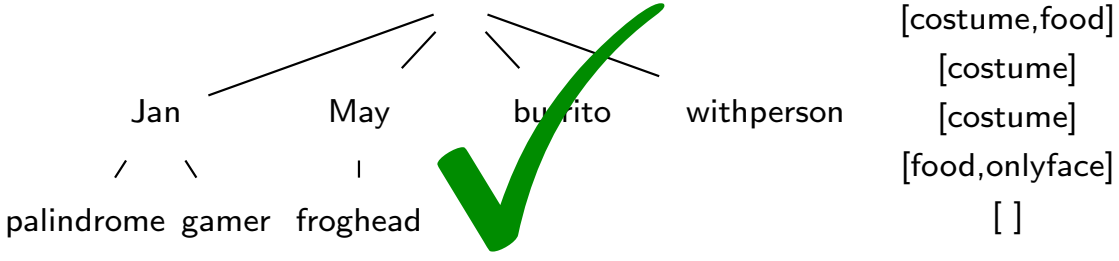




[costume,food]
[costume]
[costume]
[food,onlyface]
[ ]

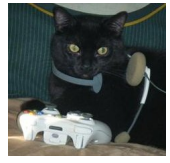




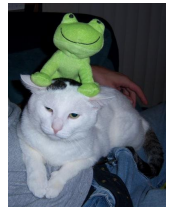


[costume,food]
[costume]
[costume]
[food,onlyface]
[ ]

[costume,food]



[costume]



[costume]

[food,onlyface]

[ ]

Abstract model

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

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

Abstract model

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

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

Synchronizing too often doesn't hurt.

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

Abstract model

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

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

Synchronizing too often doesn't hurt.

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

Not synchronizing often enough doesn't hurt.

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

Twist: equational reasoning

$$\mathsf{A} \overset{a}{\underset{k}{\longleftrightarrow}} \mathsf{B} \overset{a}{\underset{\ell}{\longleftrightarrow}} \mathsf{C} \overset{a}{\underset{m}{\longleftrightarrow}} \mathsf{D}$$

Nice property of asymmetric lenses:

$$(k; \ell); m = k; (\ell; m)$$

Twist: equational reasoning

$$\mathsf{A} \overset{a}{\underset{k}{\longleftrightarrow}} \mathsf{B} \overset{a}{\underset{\ell}{\longleftrightarrow}} \mathsf{C} \overset{a}{\underset{m}{\longleftrightarrow}} \mathsf{D}$$

Nice property of asymmetric lenses:

$$(k; \ell); m = k; (\ell; m)$$

Not true for symmetric lenses!

In dissertation

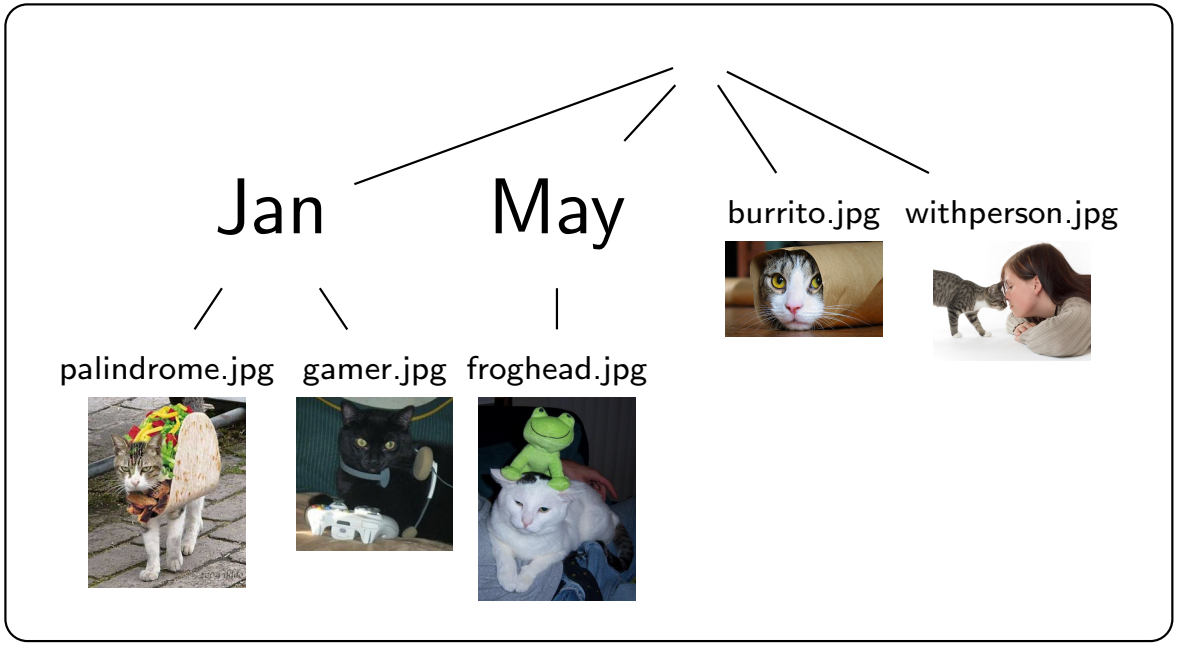
- Observational equivalence
- Point-free programming language
  - Basic (non-recursive) data types
  - Lists, with folds and unfolds
  - Some generalized container operations
- Proof that this generalizes asymmetric lenses

Related work: other symmetric approaches

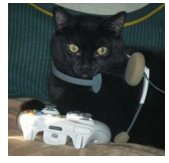
- Symmetric Constraint Maintainers
  (Meertens; 1998)
- Towards an Algebraic Theory of Bidirectional Transformations
  (Stevens; ICGT 2008)
- Bidirectional Model Transformations in QVT: Semantic Issues and Open Questions
  (Stevens; MoDELS 2007)
- Algebraic Models for Bidirectional Model Synchronization
  (Diskin; MoDELS 2008)
- Supporting Parallel Updates with Bidirectional Model Transformations
  (Xiong, Song, Hu, and Takeichi; ICMT 2009)

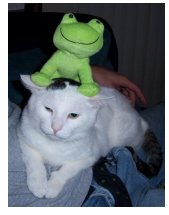
Edits

(in collaboration with Martin Hofmann and Benjamin Pierce)



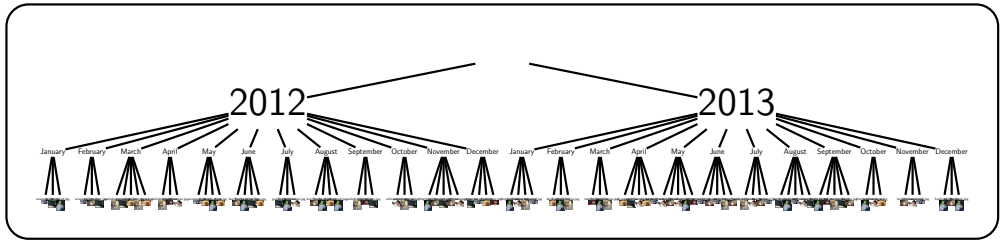
[costume,food]


[costume]


[costume]

[food,onlyface]

[ ]


2012
2013
January
February
March
April
May
June
July
August
September
October
November
December
January
February
March
April
May
June
July
August
September
October
November
December

Abstract model

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

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

Abstract model

Edit lens $\ell \in (M, X, \cdot) \overset{\delta}{\leftrightarrow} (N, Y, \odot)$ 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.

$$dputr(\mathbf{1}_M, c) = (\mathbf{1}_N, c)$$

Abstract model

Edit lens $\ell \in (M, X, \cdot) \overset{\delta}{\leftrightarrow} (N, Y, \odot)$ 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.

$$dputr(\mathbf{1}_M, c) = (\mathbf{1}_N, c)$$

Not synchronizing often enough doesn't hurt.

$$\frac{dputr(m, c) = (n, c') \quad dputr(m', c') = (n', c'')}{dputr(mm', c) = (nn', c'')}$$

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

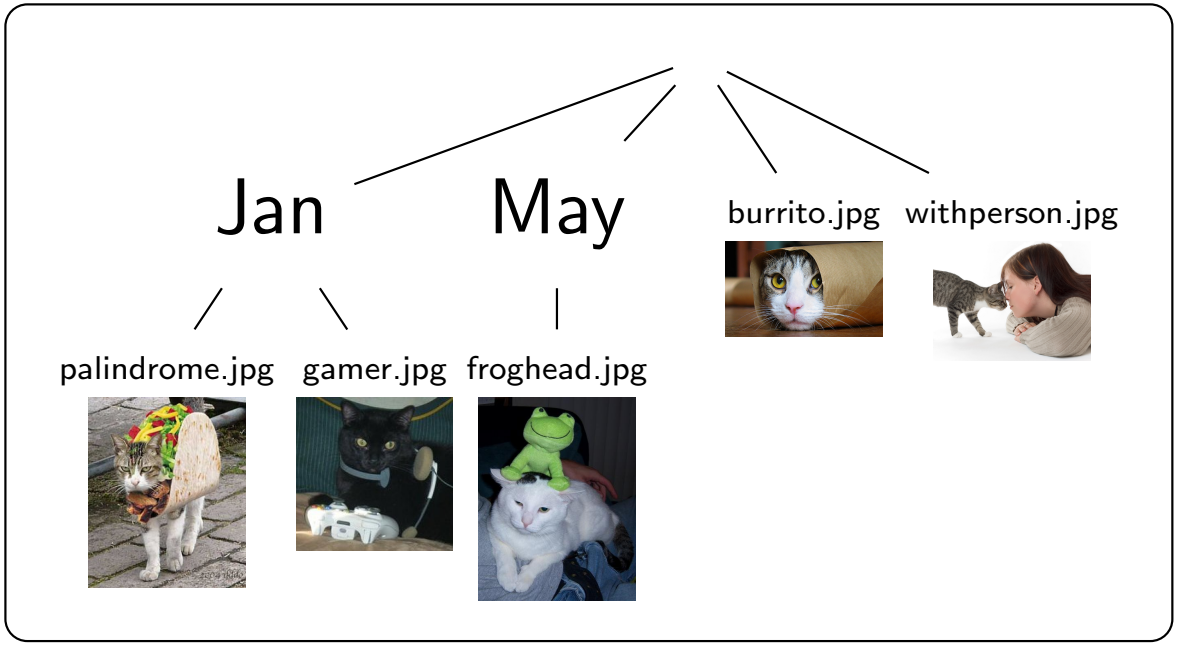
- All changes reported, so synchronizing less often is less controversial
- Intentional information in edits aids alignment
- Smaller complement in many cases!
- Roundtrip laws are monoid homomorphism laws
- Observational equivalence, combinator language, generalizes symmetric lenses

Related work: other edit-based approaches

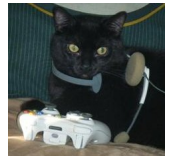
- Towards an Algebraic Theory of Bidirectional Transformations
  (Stevens; ICGT 2008)
- Matching Lenses: Alignment and View Update
  (Barbosa, Cretin, Foster, Greenberg, and Pierce; ICFP 2010)
- From State- to Delta-based Bidirectional Model Transformations
  (Diskin, Xiong, Czarnecki; TPMT 2010)
- From 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)

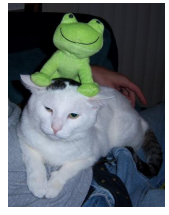
Multidirectionality

(in collaboration with Jen Paykin, Benjamin Pierce, Jeff Vaughan, and Geoff Washburn)



[costume,food]


[costume]


[costume]

[food,onlyface]

[ ]

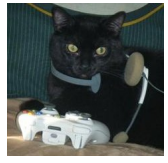
[costume,food]

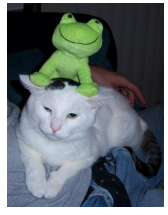
[costume]

[costume]

[food,onlyface]

[ ]





costume

food

onlyface

	A	B	C	D
1		Students	Equipment	Total
2	2012	70000	9000	79000
3	2013	70000	4000	74000
4	Total	140000	13000	153000

New interaction mode

	Students	Equipment	Total
2012	70000	9000	79000
2013	70000	4000	74000
Total	140000	13000	153000

New interaction mode

	Students	Equipment	Total
2012	70000	9000	79000
2013	70000	4000	74000
Total	140000	13000	166000

New interaction mode

	Students	Equipment	Total
2012	70000	9000	79000
2013	70000	4000	74000
Total	140000	26000	166000

New interaction mode

	Students	Equipment	Total
2012	70000	18000	79000
2013	70000	8000	74000
Total	140000	26000	166000

New interaction mode

	Students	Equipment	Total
2012	70000	18000	88000
2013	70000	8000	78000
Total	140000	26000	166000

New interaction mode

	Students	Equipment	Total
2012	70000	18000	88000
2013	70000	8000	78000
Total	140000	26000	166000

New interaction mode

	Students	Equipment	Total
2012	70000	18000	88000
2013	70000	8000	78000
Total	140000	26000	166000

. . . and this happens behind the scenes, too.

Straw-man abstract model

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

$$put \in 2^N \to U^N \to U^N$$

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

Straw-man abstract model

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

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

Inputs are really inputs and consistency is restored.

$$put(S, f)|_S = f|_S$$
$$put(S, f) \in K$$

Straw-man abstract model

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

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

Inputs are really inputs and consistency is restored.

$$put(S, f)|_S = f|_S$$
$$put(S, f) \in K$$
$$put(\emptyset, f) = f$$

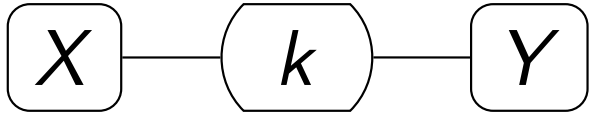
Synchronizing too often doesn't hurt.

Unsolvable updates

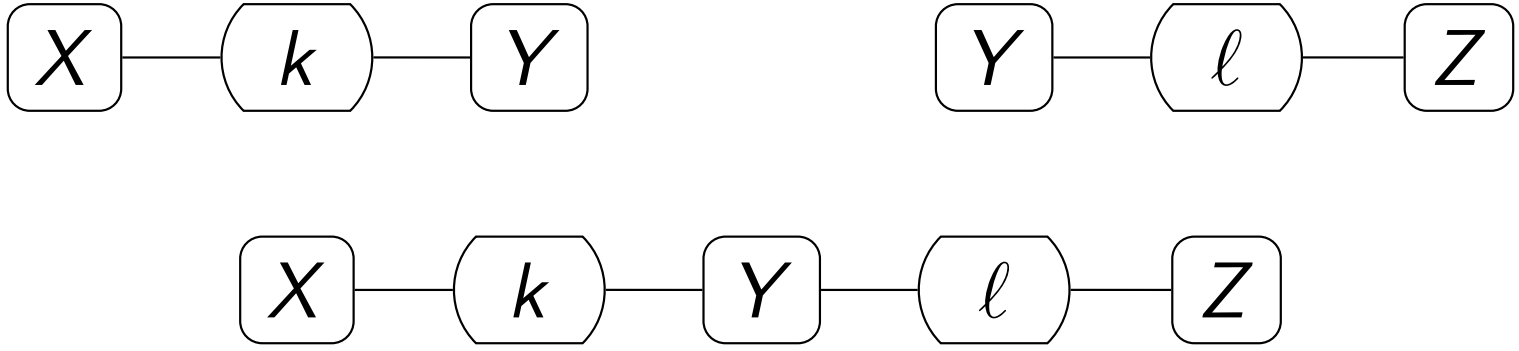
	Students	Equipment	Total
2012	70000	9000	79000
2013	70000	4000	74000
Total	0	0	1

Track sets of names that are always solvable.

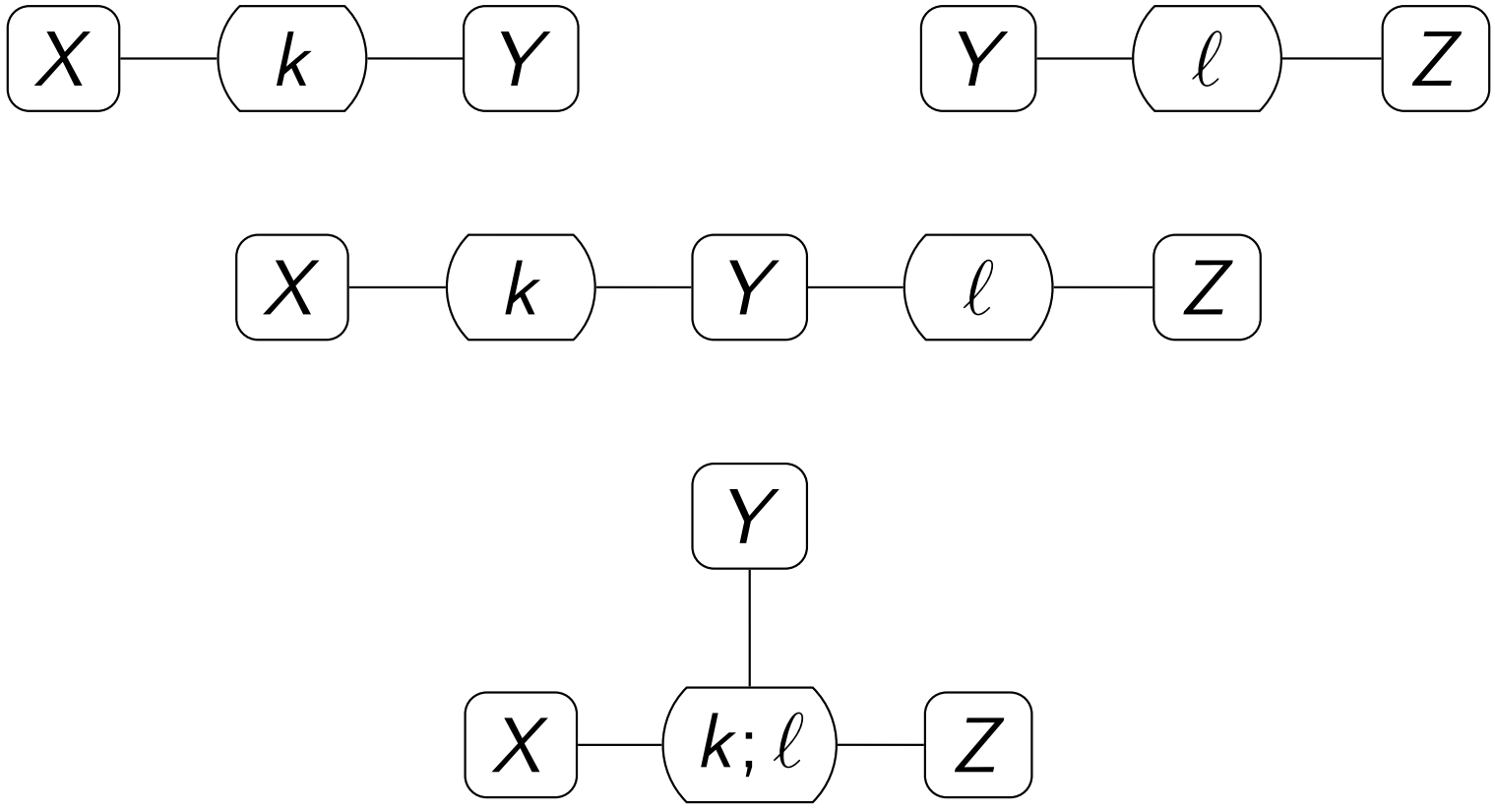
Composition intuition



Composition intuition

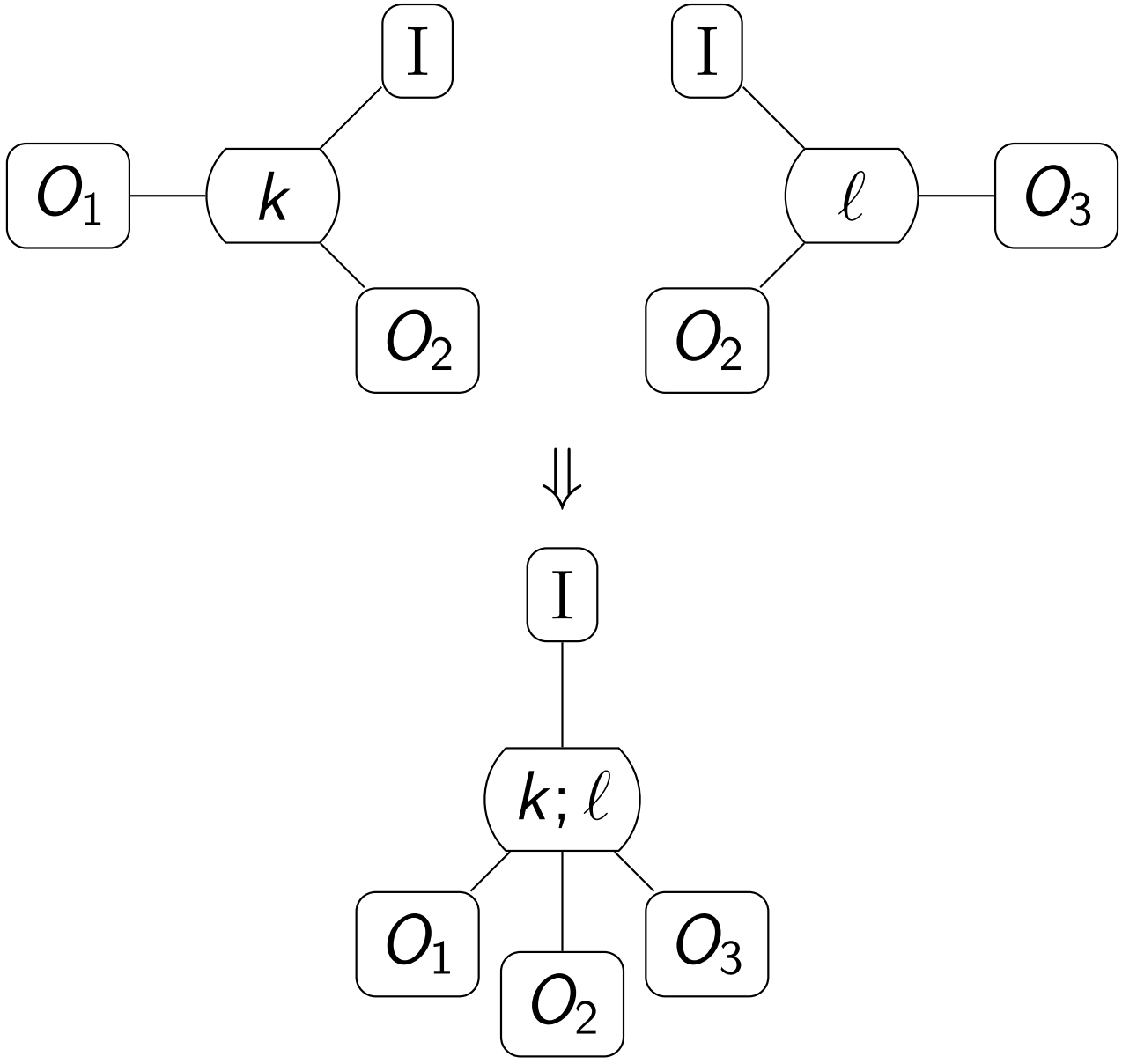


Composition intuition

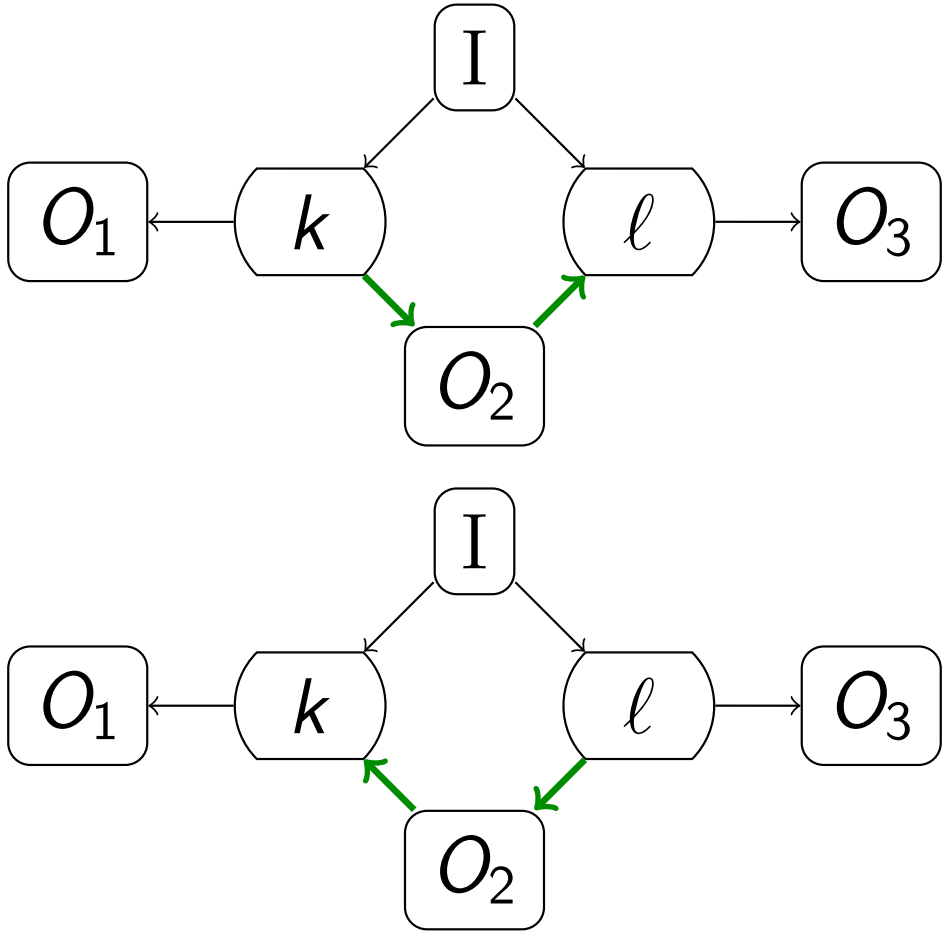


Safe updates: $\{X\}$ or $\{Z\}$.

Ambiguous updates



Two plans



Observational equivalence is no help.

Remaining questions

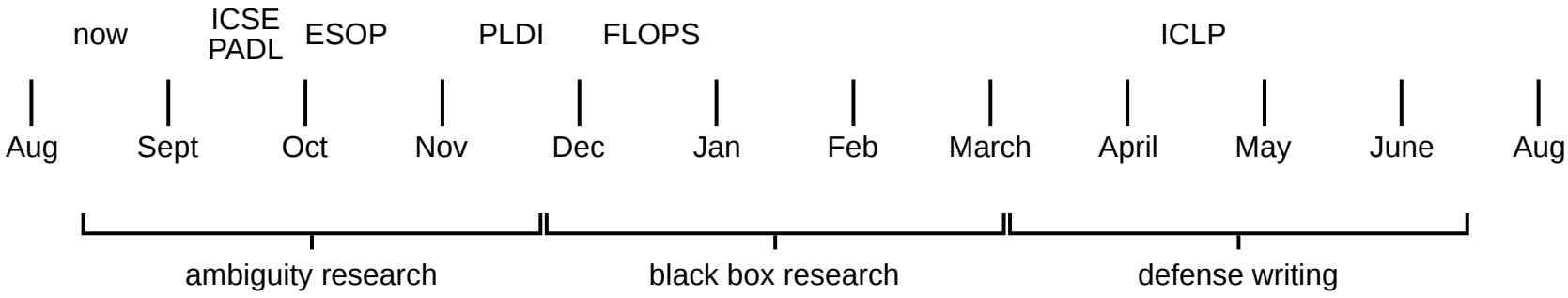
- Complete strategies for disambiguation?
- Behavioral specifications for disambiguation?
- How can we extend the static update check?
- What dynamic update checks are possible?

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

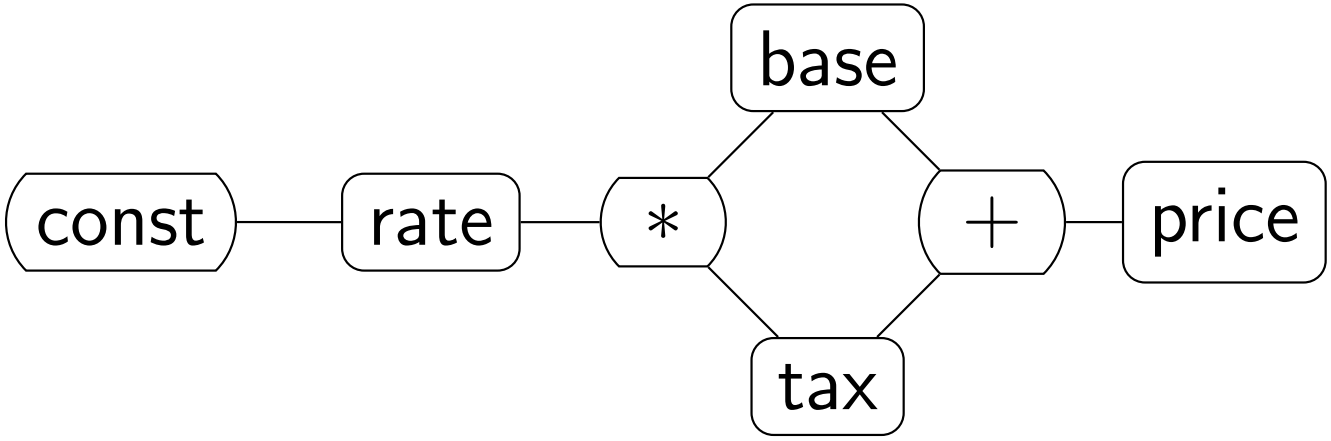
Timeline



- 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

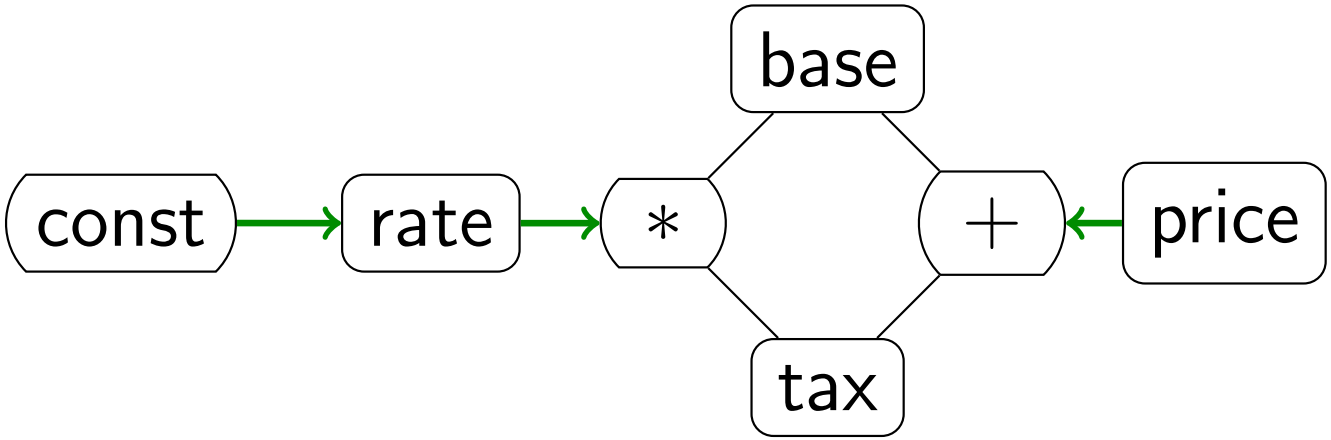
Why black boxes?

$$price = base + tax$$
$$tax = 0.08 * base$$



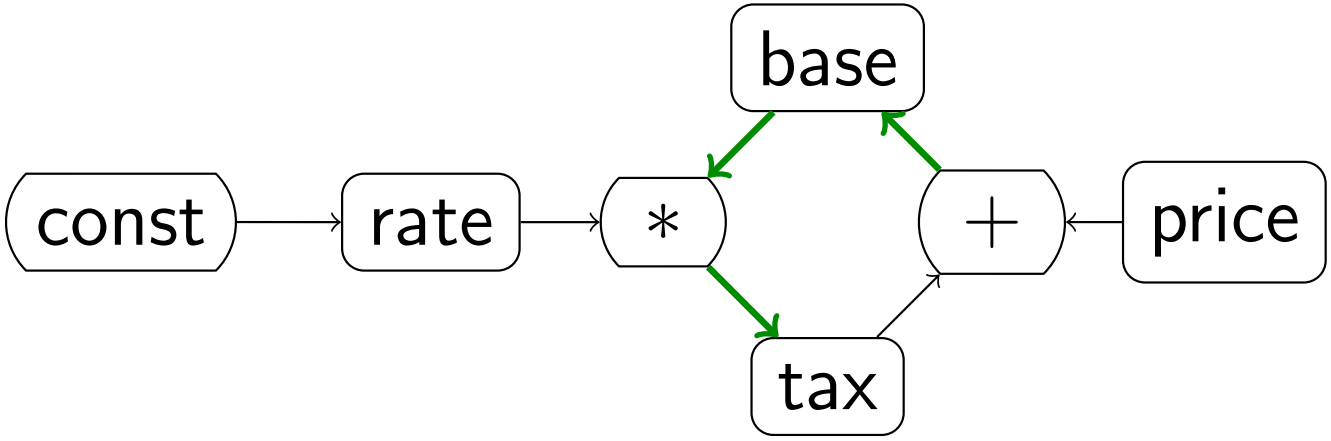
Why black boxes?

$$price = base + tax$$
$$tax = 0.08 * base$$



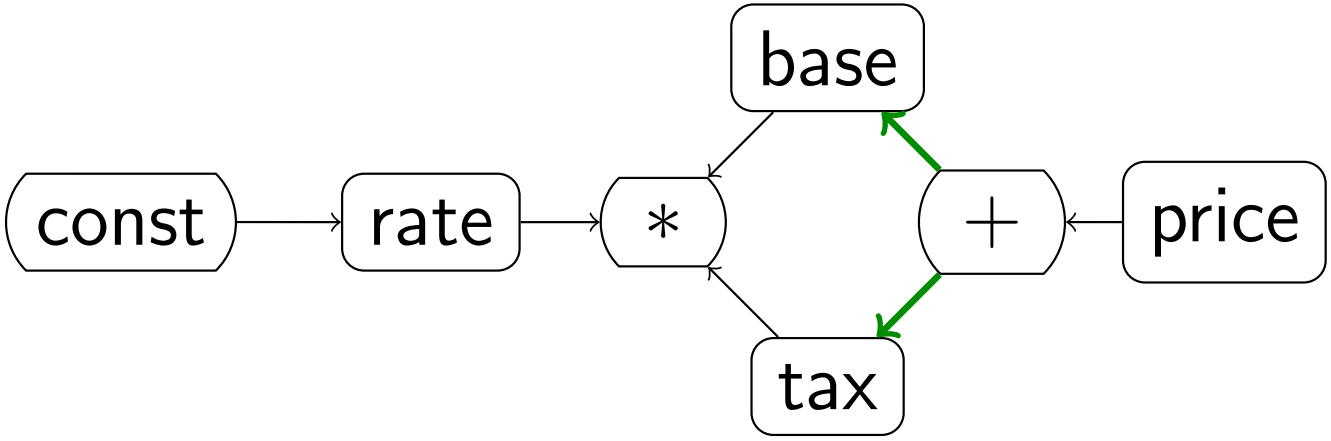
Why black boxes?

$$price = base + tax$$
$$tax = 0.08 * base$$

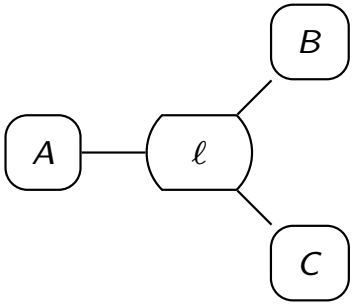


Why black boxes?

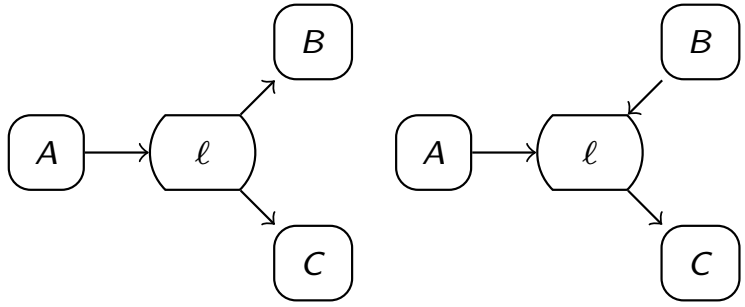
$$price = base + tax$$
$$tax = 0.08 * base$$



How to progress



constraint



sample methods

- 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)