QUANTIFICATION IN DOUBLE-CATEGORICAL DATABASE SCHEMAS

Michael Lambert

November 2026

Virtual Double Categories Workshop 2026

This is a report on two papers:

1. M.L. & Evan Patterson, “Representing Knowledge and Querying Data using Double-Functorial Semantics,” *2024 ACT Proceedings*, 2025
2. M.L. “Quantification in Double-Categorical Database Schemas,” 2026 draft: arXiv:2608.00913v2, under review at *Logical Methods in Computer Science*

A preliminary version of these results were presented at *FMCS* 2026 held at StFx in Nova Scotia.

INTRODUCTORY DISCUSSION: REVIEW OF DOUBLE OLOGS

DEFINITION

A ‘double category of relations’ is a locally posetal cartesian equipment is which every object is *discrete* (i.e. satisfies Carboni and Walter’s Frobenius Axiom).

A **double-categorical olog** is a small and finitely-presented ‘double category of relations’.[1]

A **data instance** on an olog $\mathbb{D}$ is a cartesian functor $I\colon \mathbb{D} \to \mathbb{R}\textbf{el}$.

---

[1] As in M.L. & Evan Patterson, “Representing Knowledge and Querying Data using Double-Functorial Semantics,” *2024 ACT Proceedings*, 2025

CONSEQUENTLY

For any ‘double category of relations’ (hence every double OLOG),

1. every object is a comonoid in a unique way wrt loose structure
2. every proarrow is a lax comonoid homomorphism
3. comonoid structure proarrows have right adjoints making each object a monoid too
4. likewise proarrows are lax monoid homomorphisms

AND

1. get *transfer of variables*, that is, a form of compactness:

$$\frac{X \times Y \nrightarrow Z}{X \nrightarrow Z \times Y}$$

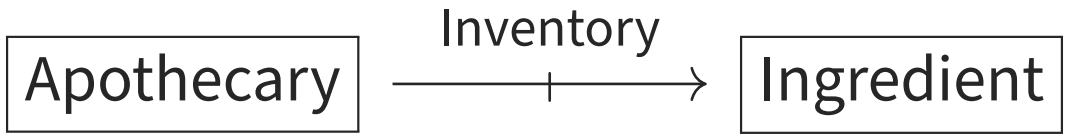
   by taking $p \colon X \times Y \nrightarrow Z$ to

$$X \xrightarrow{\cong} X \times 1 \xrightarrow{1 \times \eta} X \times Y \times Y \xrightarrow{p \times 1} Z \times Y$$

2. duality involution is trivial

TWO QUERY EXAMPLES

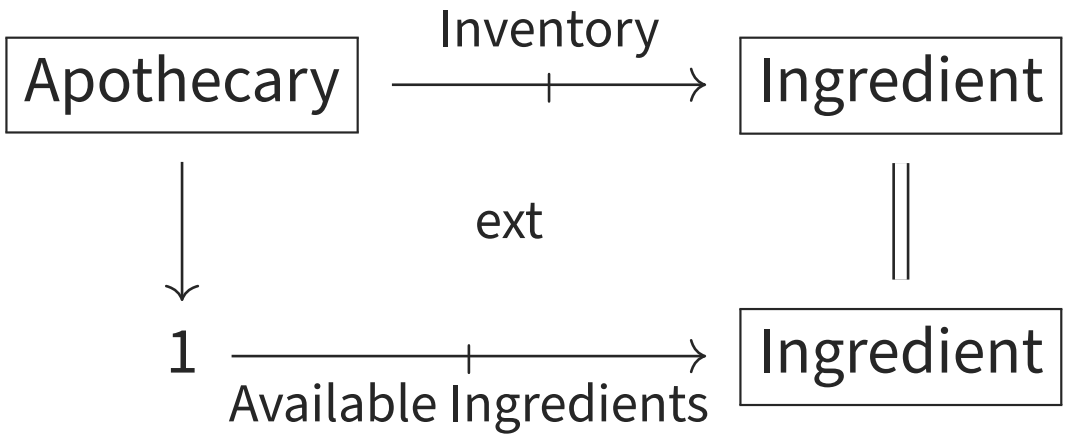
AN OLOG



AN INSTANCE

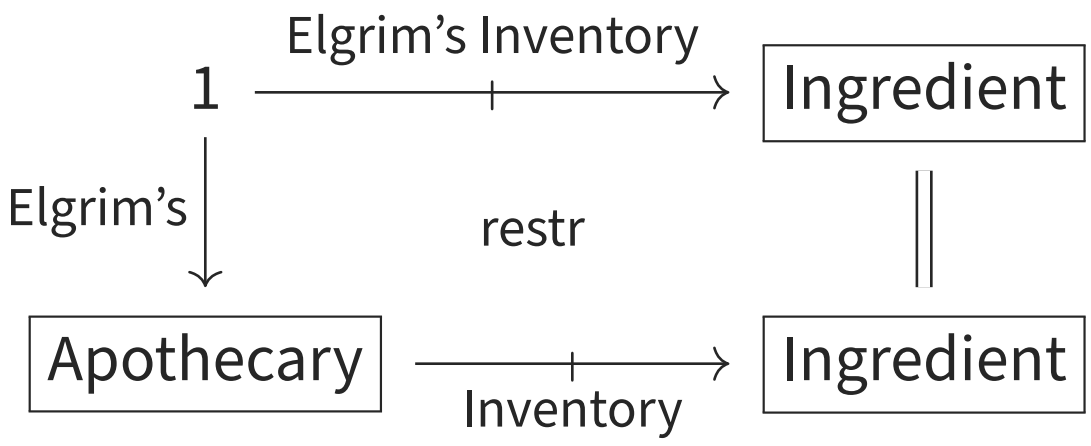
Inventory	
Apothecary	**Ingredient**
Arcadia’s Cauldron	giant’s toe
Arcadia’s Cauldron	wheat
Arcadia’s Cauldron	void salts
Elgrim’s Elixers	giant’s toe
Elgrim’s Elixers	wheat
The White Phial	wheat

SELECT COLUMN



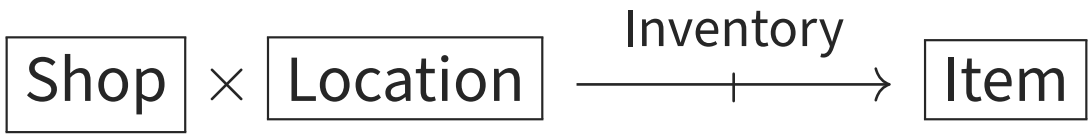
Available Ingredients
giant’s toe
wheat
void salts

FILTER BY VALUE



Elgrim's Inventory
giant's toe
wheat

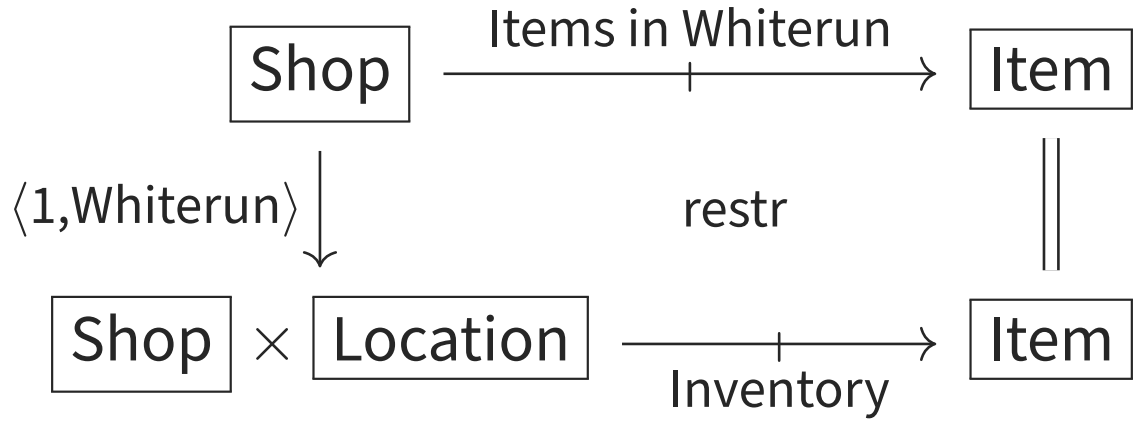
A NEW OLOG



AN NEW INSTANCE

Inventory		
Shop	**Location**	**Item**
Arcadia’s Cauldron	Whiterun	giant’s toe
Warmaiden’s	Whiterun	enchanted dagger
Arcadia’s Cauldron	Whiterun	void salts
Elgrim’s Elixers	Riften	giant’s toe
Elgrim’s Elixers	Riften	wheat
The Pawned Prawn	Riften	restore magicka
The White Phial	Solitude	salt pile

FILTER BY VALUE



RESULT OF FILTER QUERY

Items in Whiterun	
Shop	**Item**
Arcadia’s Cauldron	giant’s toe
Warmaiden’s	enchanted dagger
Arcadia’s Cauldron	void salts

RELATIONAL DIVISION

OVERVIEW

1. In the theory of relational databases, relational division is a more advanced query having the effect of checking inventories against lists and vice versa.[2]
2. Universal quantification in crucial in the query.
3. Our goal is to simplify the syntax and recast the operation using double OLOGs and functorial semantics.
4. Illustrate via an example.

---

[2] Codd, “Relational Completeness of Data Base Sublanguages," *Database Systems*, Prentice-Hall, 1972.

SET-UP AND NOTATION

Start with a relation $\langle d, c \rangle : R \rightarrowtail C \times A$

Set notation

$$r[C] = d(r) \quad r[A] = c(r) \quad \text{where } r \in R$$

and

$$g_R(x) = \{ y \mid xR\,y \}$$

the “orbit of x under R.”

TWO TABLES

R	
C	**A**
1	a
1	b
1	c
2	a
2	b
3	b

S	
B	**D**
a	2
b	1
b	2
a	3

SYNTAX FOR DIVISION QUERY

Codd's syntax is

$$R[A \div B]S = \{r[C] \mid r \in R \text{ and } S[B] \subset g_R(r[C])\}$$

This returns the set of C-entries whose orbits contain all B-entries.

COMPUTE ORBITS

Orbits:

$S[B] = \{a, b\} \subset \{a, b, c\} = g_R(1)$

$S[B] = \{a, b\} = \{a, b\} = g_R(2)$

$S[B] = \{a, b\} \not\subset \{b\} = g_R(3)$

Tables:

R	
C	**A**
1	a
1	b
1	c
2	a
2	b
3	b

S	
B	**D**
a	2
b	1
b	2
a	3

COMPUTE DIVISION

Orbits:

$$S[B] = \{a, b\} \subset \{a, b, c\} = g_R(1)$$

$$S[B] = \{a, b\} = \{a, b\} = g_R(2)$$

$$S[B] = \{a, b\} \not\subset \{b\} = g_R(3)$$

Division:

$$R[A \div B]S =$$

$$\{r[C] \mid r \in R \text{ and } S[B] \subset g_R(r[C])\}$$

Hence

$$R[A \div B]S = \{1, 2\}$$

CONCERTIZE THE EXAMPLE

Inventory	
Apothecary	**Ingredient**
Arcadia’s Cauldron	giant’s toe
Arcadia’s Cauldron	wheat
Arcadia’s Cauldron	void salts
Elgrim’s Elixers	giant’s toe
Elgrim’s Elixers	wheat
The White Phial	wheat

Potion	
Ingredient	**Effect**
giant’s toe	fortify health
wheat	fortify health
wheat	lingering damage magicka
giant’s toe	damage stamina regen

ORBITS & QUERY REDUX

$$g_{\text{Inventory}}(\text{Arcadia's}) = \{\text{giant's toe, wheat, void salts}\}$$

$$g_{\text{Inventory}}(\text{Elgrim's}) = \{\text{giant's toe, wheat}\}$$

$$g_{\text{Inventory}}(\text{Phial}) = \{\text{wheat}\}$$

$$\text{Apothecary}[\text{Ingredient} \div \text{Ingredient}]\text{Potion}$$

$$= \{\text{Apothecary} \mid \{\text{giant's toe, wheat}\} \subset \text{Inventory}(\text{Apothecary})\}$$

$$= \{\text{Arcadia's, Elgrim's}\}$$

COMMENTS

1. The second column of the second table (effect) is superfluous.
2. The query returns the set of shops with all the items in the first column of the second table.
3. This is a *check inventory against shopping list* type of query.
4. This is Codd's own example: suppliers, parts supplied, and parts needed.[3]

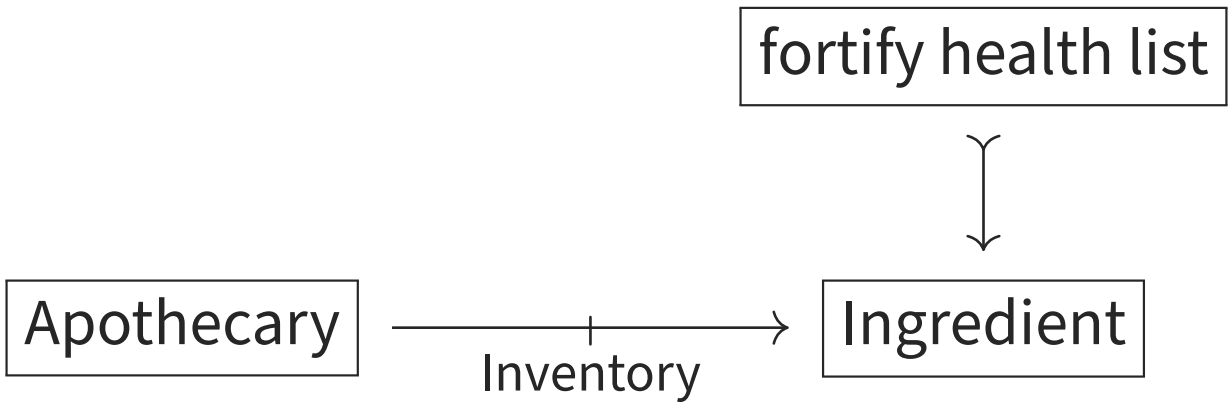
[3] §2.4 of "Relational Completeness [...]"

RECAST RELATIONAL DIVISION

AN OLOG

Fortify health potion = giant’s toe + wheat.

Make a shopping list:

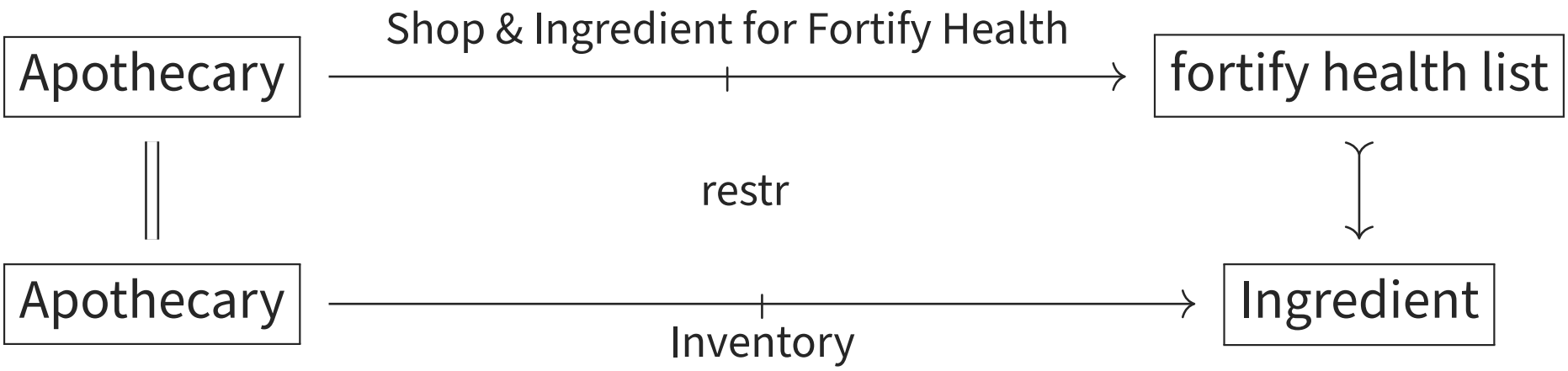


INSTANCE

Inventory	
Apothecary	**Ingredient**
Arcadia’s Cauldron	giant’s toe
Arcadia’s Cauldron	wheat
Arcadia’s Cauldron	void salts
Elgrim’s Elixers	giant’s toe
Elgrim’s Elixers	wheat
The White Phial	wheat

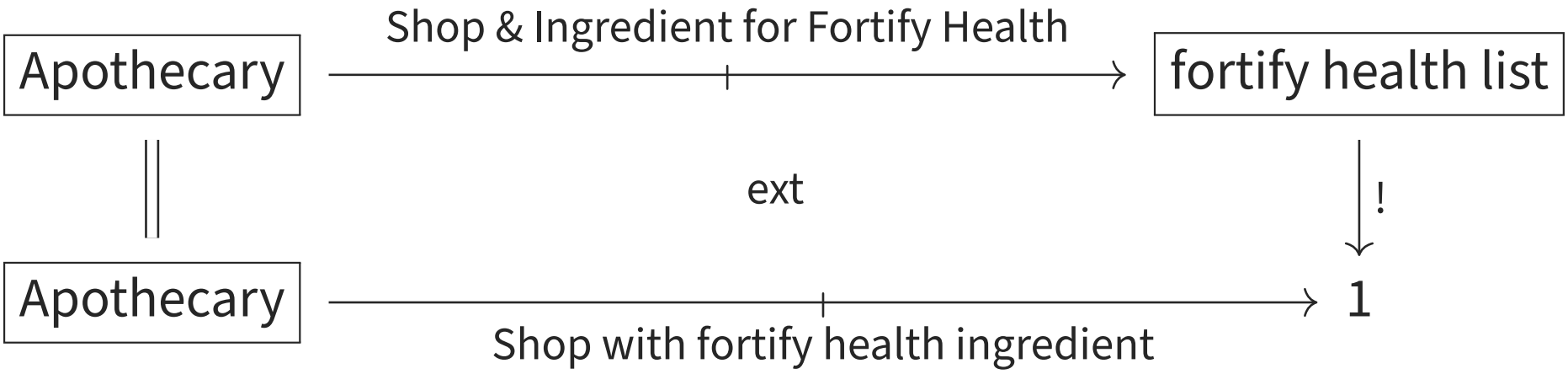
FILTER BY FORTIFY HEALTH INGREDIENTS

Restrict:



SELECT FOR SHOPS

Form extension:



RESULT OF COMPOSITE QUERY

Shop with Fortify Health Ingredient
Arcadia’s Cauldron
Elgrim’s Elixers
The White Phial

COMMENTS

1. This is not what we want!
2. We want a list of shops that have *all* of our ingredients. That is, it's not relational division.
3. We are one-stop shoppers (lazy).
4. Certainly don't want to fight a giant to get his toe.
5. Working in field is tedious and beneath us.
6. Emphasis on *all* above prompts looking at universal quantification.

QUANTIFICATION REVIEW

For any function $f : X \to Y$

$$PX \underset{\exists_f}{\overset{\forall_f}{\leftarrow\!\!-\!\!-\!\!\rightarrow}} PY \qquad\qquad \exists_f \dashv f^* \dashv \forall_f$$

where

$$\exists_f(S) = \{y \in Y \mid f(x) = y \text{ for some } x \in S\}$$

$$\forall_f(S) = \{y \in Y \mid \text{for all } x \in X \text{ if } f(x) = y \text{ then } x \in S\}.$$

SPECIALIZE TO BINARY RELATION & PROJECTION

Fix a relation R on $X \times Y$ and $\pi\colon X \times Y \to Y$

$$\exists_\pi(R) = \{y \in Y \mid R(x, y) \text{ for some } x \in X\}$$

$$\forall_\pi(R) = \{y \in Y \mid R(x, y) \text{ for all } x \in X\}.$$

This is exactly *binding* one of two free variables in a formula/proposition by a quantifier.

APPLICATION I

Inventory = R	
Apothecary = X	**Ingredient = Y**
Arcadia's Cauldron	giant's toe
Arcadia's Cauldron	wheat
Arcadia's Cauldron	void salts
Elgrim's Elixers	giant's toe
Elgrim's Elixers	wheat
The White Phial	wheat

Instance

$$\exists_{\pi}(R) = \{y \mid R(x, y) \text{ for some } x \in X\}$$

$$= \{\text{ingredients someone has}\}$$

$$= \{\text{wheat, giant's toe, void salts}\}$$

$$\forall_{\pi}(R) = \{y \mid R(x, y) \text{ for all } x \in X\}$$

$$= \{\text{ingredients everyone has}\}$$

$$= \{\text{wheat}\}$$

Quantification

APPLICATION II

Inventory = R	
Apothecary = X	**Ingredient = Y**
Arcadia's Cauldron	giant's toe
Arcadia's Cauldron	wheat
Arcadia's Cauldron	void salts
Elgrim's Elixers	giant's toe
Elgrim's Elixers	wheat
The White Phial	wheat

Instance

$$\exists_{\pi}(R^{\dagger}) = \{x \mid R(x, y) \text{ for some } y \in Y\}$$

$$= \{\text{those with an } R\text{-ingredient}\}$$

$$= \{\text{Arcadia's, Elgrim's, Phial}\}$$

$$\forall_{\pi}(R^{\dagger}) = \{x \mid R(x, y) \text{ for all } y \in Y\}$$

$$= \{\text{those with all } R\text{-ingredients}\}$$

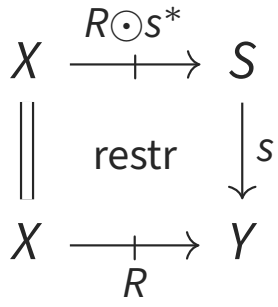
$$= \{\text{Arcadia's}\}$$

Quantification

SHOPPING LIST

Fortify Health Ingredients = S
giant’s toe
wheat

RESTRICT



Restriction in $\mathbb{R}$el

Shop & Fortify Health Ingredient	
Apothecary = X	**Ingredient = S**
Arcadia's Cauldron	giant's toe
Arcadia's Cauldron	wheat
Elgrim's Elixers	giant's toe
Elgrim's Elixers	wheat
The White Phial	wheat

Result of $R \odot s^*$

NOW, QUANTIFY

Shop & Fortify Health Ingredient	
Apothecary = X	**Ingredient = S**
Arcadia's Cauldron	giant's toe
Arcadia's Cauldron	wheat
Elgrim's Elixers	giant's toe
Elgrim's Elixers	wheat
The White Phial	wheat

Restriction in $\mathbb{R}$el

$$\forall_{\pi_X}(R \odot s^*) = \{x \mid R(x, y) \text{ for all } y \in S\}$$

$$= \{\text{shops w/ all } S\text{-ingredients}\}$$

$$= \{\text{Arcadia's, Elgrim's}\}$$

Result of quantification query

ANALYSIS & COMMENTS

That's relational division!

Restriction followed by universal quantification:

$$P(X \times Y) \xrightarrow{(1\times s)^*} P(X \times S) \xrightarrow{\forall_{\pi_X}} PX$$

The question is how to axiomatize this into an OLOG to let functorial semantics take care of the query.

DEFINITION

A $\prod$**-double OLOG** is a double OLOG for which every substitution functor has a right adjoint (satisfying Beck-Chevalley)

$$\mathbb{D}(X,Y) \xrightarrow{(f,g)^*} \mathbb{D}(Z,W) \xrightarrow{\prod_{f,g}} \mathbb{D}(X,Y)$$

for all such f, g.

COMMENTS

1. A $\prod$-double OLOG asks that $\mathbb{D}_1 \to \mathbb{D}_0 \times \mathbb{D}_0$ is a Π-bifibration
2. Not necessarily a cofibration.
3. Universal quantifications are not formed by some obscure coextension or by loose composition with a f_*
4. Not closed structure in Shulman's sense (the underlying bicategory is closed wrt loose composition).
5. Elementary axiomatization possible (counit as universal)

ANALYZE QUANTIFICATION

For the right adjoints we have a natural family of bijections:

$$\mathbb{D}(Z,W)(f_!mg^*,n) \cong \mathbb{D}(X,Y)(m,\prod_{f,g}(n)) \qquad \frac{f_!mg^* \Rightarrow n}{m \Rightarrow \prod_{f,g}(n)}$$

Recognize this as essentially *Kan Quantification* since substitutions are computed by composition with companions and conjoints.[4]

[4] §2 of Lawvere, "Metric Spaces [...]," *TAC Reprints*, 2002.

WHAT'S NEW HERE?

1. Combination of functorial and fibrational semantics. A data instance is a *structure-preserving* double functor

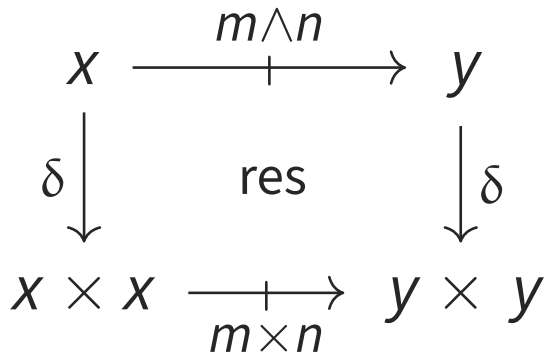
$$I \colon \mathbb{D} \to \mathbb{R}\mathbf{el}$$

   where $\mathbb{D}$ is a Π-double OLOG.
2. *structure-preserving* means cartesian and preserving whatever adjoints are present
3. So, a data instance is also a morphism of hyperdoctrines.
4. Data instance $\approx$ model of a theory

NEW RESULTS: OPTIMIZATION

LOCAL PRODUCTS

Recall: every ‘double category of relations’ has local products:



Local products allow *join* queries in double-categorical database schemas – essentially combining tables across a shared column (expensive!)

LEMMA

Any ‘double category of relations’ satisfies the Frobenius reciprocity Law. That is, the canonical cell

$$\coprod_{f,g}(m \wedge f_! pg^*) \leq (\coprod_{f,g} m) \wedge p$$

is invertible.

OPTIMIZATION EXAMPLE

Belongs to	
Individual	**Class**
Belrand	spellsword
Farcas	warrior
Illia	mage
J’zargo	mage
Marcurio	mage
Teldryn Sero	spellsword

$S = \{\text{Marcurio}, \text{Illia}, \text{Teldryn Sero}\}$ $T = \{\text{spellsword}, \text{warrior}\}$

WHICH CLASSES ARE REPRESENTED?

Computation 1:

$$\exists_{\text{Belongs}}(S \wedge (\text{Belongs}^*(T)))$$

$$= \exists_{\text{Belongs}}(\{\text{Marcurio},\ \text{Illia},\ \text{Teldryn}\} \cap \{\text{Belrand},\ \text{Farcas},\ \text{Teldryn}\})$$

$$= \exists_{\text{Belongs}}(\{\text{Teldryn Sero}\})$$

$$= \{\text{spellsword}\}.$$

This is perhaps the naive way. Note: 3 operations, join leads.

MORE EFFICIENT CALCULATION

By Frobenius reciprocity, this is the same as

$$\exists_{\text{Belongs}}(S) \wedge T = \{\text{spellsword, mage}\} \cap \{\text{spellsword, warrior}\}$$
$$= \{\text{spellsword}\}.$$

The point is cutting down first by projection and joining last.

Frobenius is a property of any double olog; hence this optimization law is provably correct (i.e. type checks & error-free).

A similar analysis can be done for Beck-Chevalley.

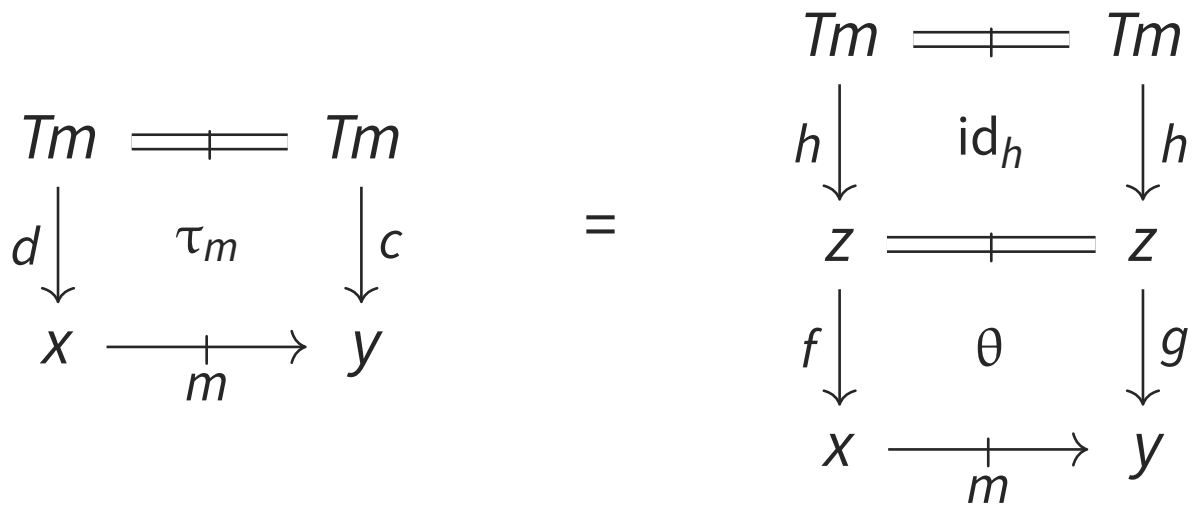
NEW RESULTS: FIRST-ORDER MODAL LOGIC

MOTIVATION: NEGATION

1. Maybe you want to query for everything *not* being on some list.
2. Need negation.
3. Can derive from implication and colimits or just a 0-object.
4. The right adjoint to substitution *should be* some kind of hom under the right conditions.

DEFINITION

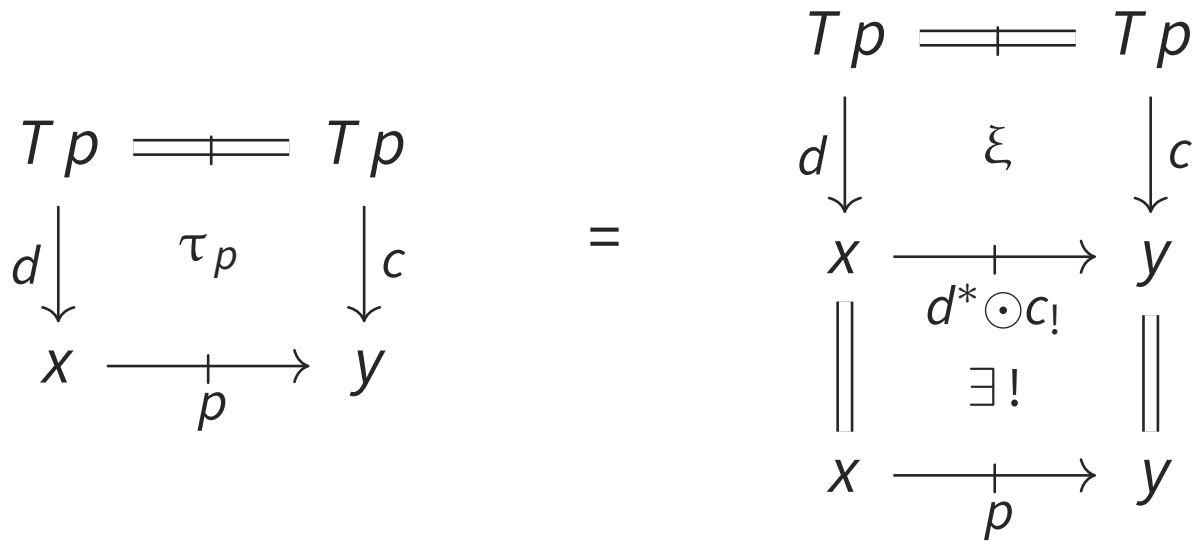
The **tabulator** of a proarrow $m : x \nrightarrow y$ is a cell τ_m having the universal property that there exists a unique morphism h making an equality of diagrams



for each such cell θ.

DEFINITION

The tabulator of a proarrow $p : X \nrightarrow Y$ is **strong** if the canonical morphism



is an isomorphism. A double category has **strong tabulators** if it has tabulators and each is strong.

THEOREM

If $\mathbb{D}$ is a $\prod$-double olog with strong tabulators, then identity proarrows are exponentiable and each hom category $\mathbb{D}(x, y)$ is cartesian closed. The formulas

$$m \wedge n = \coprod_{m} (\mathrm{id}_{Tm} \wedge m^*(n)) \qquad m \supset p = \prod_{m} (\mathrm{id}_{Tm} \supset m^*(p))$$

compute the product and exponential.

COMMENTS

1. This theorem is the centerpiece of the second half of the paper.
2. Technical aspect is showing each identity proarrow is exponentiable.
3. Uses Frobenius reciprocity in an essential way.
4. Every $\prod$-double olog with strong tabulators is thus locally cartesian closed as a fibration.
5. If double olog is also *cocartesian*, we get local disjunction and negation queries, reproducing the usual relational complement in the case of ordinary relations.

THEOREM

If $\mathbb{D}$ is a $\prod$-double olog with strong tabulators that is also cocartesian and satisfies a global distributivity law, then $\mathbb{D}$ models full first-order logic locally.

MODALITY I

Let $\mathbb{D}$ denote a cartesian equipment with strong tabulators and dependent products. The **modal necessity** operator associated to $r : x \nrightarrow y$ is the composite functor

$$\Box_r := \prod_d c^*(-) : \mathbb{D}(y, 1) \to \mathbb{D}(x, 1)$$

where $\langle d, c \rangle : Tr \to x \times y$ is the tabulator of r.
Likewise we have $\diamondsuit_r := \coprod_d c^*(-)$ and other permutations for relational converse.

MODALITY II

Under the foregoing assumptions:

1. $\Diamond \dashv \Box$ (adjoint modality)
2. $\mathrm{id} \leq \Diamond$ and $\Diamond\Diamond \leq \Diamond$
3. $\Box \leq \mathrm{id}$ and $\Box\Box \leq \Box$
4. $\mathrm{id} \leq \Box\Diamond$
5. $\Diamond\Box \leq \mathrm{id}$.

So, each double olog with enough structure models first-order modal logic.

DESCRIPTION LOGIC

Consequently, each double olog with modal operators also models description logic: a fragment of first-order logic with limited quantification over certain relations.

The interpretation of concepts and roles in ordinary set theory is

1. $[\![\exists R.C]\!] = \{x \in X \mid \text{there is } y \in X \text{ such that } (x, y) \in R \text{ and } y \in C\}$
2. $[\![\forall R.C]\!] = \{x \in X \mid \text{for all } y \in X \text{ if } (x, y) \in R \text{ then } y \in C\}$.

which are precisely modal necessity and possibility.

FURTURE WORK / WORK IN PROGRESS

1. Higher-order generalizations?
2. Internal language/logic?
3. What is the role of compactness, actually?
4. For this need twisted functors, loose Yoneda theory, and twisted adjunctions.

THANK YOU!