A Categorical Approach to Knowledge Management

Computational Category Theory Workshop

29 September 2015

National Institute of Standards and Technology

National Institute of Standards and Technology
U.S. Department of Commerce

Dr. Ralph L. Wojtowicz

Shepherd University
Shepherdstown, WV
rwojtowi@shepherd.edu

Shepherd UNIVERSITY

Baker Mountain Research Corporation
Yellow Spring, WV
ralphw@bakermountain.org

BAKER MOUNTAIN
Science Technology Service

Personal Background and Perspective

- Employment
  - Shepherd University. 2011–present
  - Baker Mountain Research Corporation. 2011–present
  - Metron, Inc. 2004–2011
  - University of Dallas. 2001–2004
- Education
  - Ph.D. Mathematics, University of Illinois (category theory and dynamic systems)
  - M.S. Aeronautical Engineering, University of Illinois
  - B.S. Mathematics, Aeronautical Engineering. Rensselaer Polytechnic Institute
- Project Experience
  - Consultant: Senior Hadoop Analyst for PNC Financial Services. 2015
  - Consultant: Statistical analysis and model development for Flexible Plan Investments, Bloomfield Hills, MI. 2014–present
  - Established and manages Shepherd Laboratory for Big Data Analytics
  - Co-Investigator with S. Bringsjord (RPI) and J. Hummel (UIUC): *Great Computational Intelligence*. AFOSR. 2011–14
  - PI with N. Yanofsky (CUNY): *Quantum Kan Extensions*. IARPA. 2011–12
  - Analyst. *Wide Aperture Passive Sonar Algorithm Development*. ONR. 2010
  - Technical Lead. *Exposing and Influencing Enemy Networks*. ONR. 2009–10
  - PI: *Robust Decision Making*. AFOSR. 2008–2010
  - Analyst: *TradeNet Integration into Global Trader*. ONI. 2009
  - PI: *Reasoning Under Uncertainty*. Phase I–II SBIR. MDA. 2005–8
  - PI: *Measures of Effectiveness Sensitivity Calculator*. ONR. 2006–7

Opportunities for Computational Category Theory

- Sketch theory as a complement to other semantic technologies
  - Modularity and compositionality
  - Knowledge alignment, theory of a sketch, and Carmody-Walters
  - Q-Trees as a reasoning engine
  - Automated inference of and use of context
  - Transformations between sketches, logical theories, and ontologies
  - Limitations of other technologies
- Quantum computing
  - No improvement in exact calculation of products, equalizers, pullbacks or coproducts in Set_f
  - Exact calculation of coequalizers
  - Fast, approximate algorithms
  - Quantum algorithms in other categories
- Uncertainty models
  - Assigning convex sets to logical formulae (i.e., the Eilenberg-Moore category of the Lawvere-Čencov stochastic category)
  - Categories for other uncertainty models. Corresponding logics.
  - Transformations between uncertainty models

Sketches: Historical Timeline

- 1943: Eilenberg and Mac Lane introduce category theory
- 1958: Kan introduces the concept of adjoints
- 1963: Lawvere characterizes quantifiers and other logical operations as adjoints
- 1968: C. Ehresman introduces sketch theory
- 1985: KL-ONE — First implementation of a description logic system
- 1985: Barr and Wells publish *Toposes, Triples and Theories*
- 1989: J. W. Gray publishes Category of Sketches as a Model for Algebraic Semantics
- 1990: Barr and Wells publish *Categories for Computing Science*
- 1995: Carmody and Walters publish algorithm for computing left Kan extensions
- 1999: RDF becomes a W3C recommendation
- 2000: Johnson and Rosebrugh apply sketch data model to database interoperability
- 2000: DARPA begins development of DAML
- 2001: Dampney, Johnson and Rosebrugh apply sketches to view update problem
- 2001: W3C forms the Web-Ontology Working Group
- 2004: RDFS and OWL become W3C recommendations
- 2008: Johnson and Rosebrugh release Easik software
- 2009: OWL2 becomes a W3C recommendation
- 2012: Johnson, Rosebrugh and Wood use sketches to formulate lens concept of view updates

Knowledge Technologies

Facets of Knowledge Models

Storage	Queries
Constraints	Uncertainty
Alignment	Dynamics
Context/Views	Software
Reasoning	Decision-Making
Translations	Human Interface

Knowledge Technologies

- Mathematical Logic (1879)
- Databases + SQL (1968)
- Semantic Web OWL/RDF
  + Description Logic (1999)
- Sketches (1968/2000)
  + Q-Trees (1990)

Sketch Theory: Overview

- Mature, graph-based foundation
- Vertices = classes or relations
- Edges = type information or maps
- Constraints/meta-data specified via graph maps (cones/cocones)
- Sketch maps respect constraints
- Grew from category theory in 1968
- Applied to data modeling since 1989

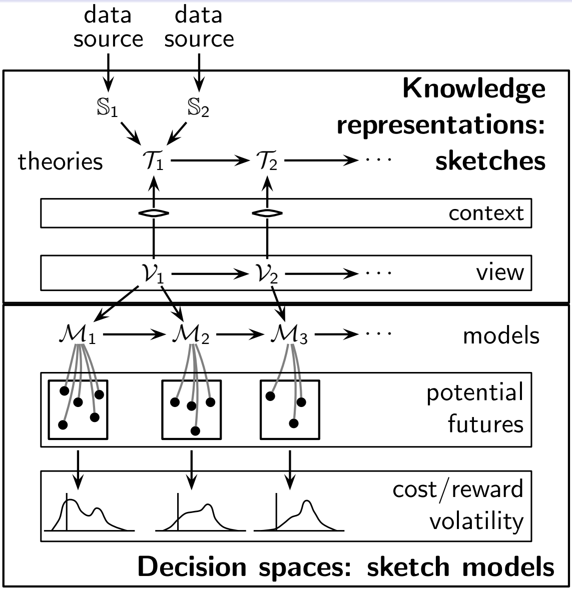
Sketch Theory: Strengths

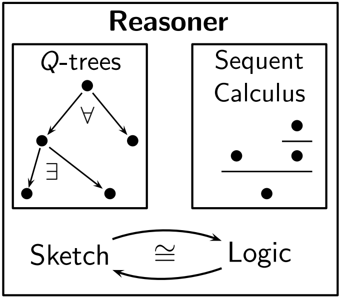
- Visual/graphical modeling
- Modularity: data/concepts/uncertainty
- Combinatory algebra of sketches
- Concise graphical inference and inter-convertibility with 1st order logic
- Derived concepts via CW algorithm
- Rich composable views/context
- Dynamics via sketch maps

Opportunity: Limitations of Knowledge Technologies

- Mathematical Logic
  - Computational complexity of some predicate calculus fragments (e.g., classical logic)
  - Complexity of the syntactic category used for knowledge alignment
  - Challenging to develop a human interface
- Databases + SQL
  - Limited notion of context/view (a single table)
  - Static schema
- Semantic Web OWL/RDF + Description Logic
  - Lack of modularity: meta-data, instance data and uncertainty integrated into a monolithic ontology
  - Limited compositional algebra: (disjoint) unions of ontologies
  - Need for constraint-preserving maps
- Sketch Theory
  - Meager computational infrastructure (e.g., relative to Jena)

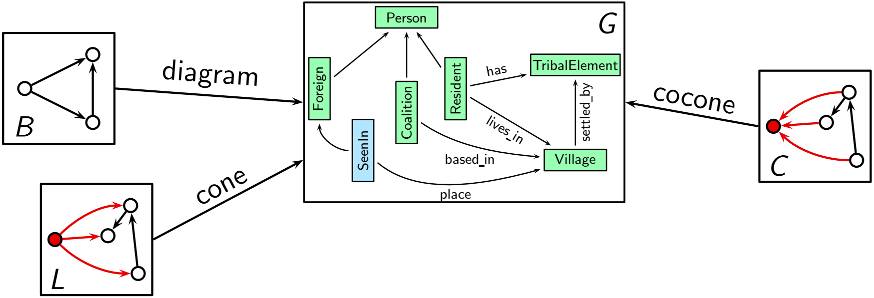
Sketch Conops





Sketch $(G, \mathcal{D}, \mathcal{L}, \mathcal{C})$

- All semantic constraints in a sketch are expressed using graph maps.
- A sketch $(G, \mathcal{D}, \mathcal{L}, \mathcal{C})$ consists of:
  - An underlying graph G
  - A set $\mathcal{D}$ of diagrams $B \to G$
  - A set $\mathcal{L}$ of cones $L \to G$
  - A set $\mathcal{C}$ of cocones $C \to G$



Categorical Semantics of Sketches

- Vertices are interpreted as objects
- Edges are interpreted as morphisms
- Classes of constraints (cones and cocones) are distinguished by the shapes of their base graphs.
- Classes of sketches are distinguished by their classes of constraints.
- Like logics and OWL species, these have different expressive powers.

Small sample of the sketch semantics landscape

Sketch Class	Set	Partial Func.	Stoch. Matrices	Čencov Cat.	Prob. 0 Refl.	Dempster Shafer	Fuzzy Sets	Convex Sets
linear	•	•	•	•	•	•	•	•
Finite Limit	•	•	×	×	×	×	•	•
Finite Coproduct	•	•	•	•	•	•	•	•
Entity-Attribute	•	•	×	×	×	×	•	•
Mixed	•	•	×	×	×	×	•	•

Sketch Maps and Model Maps

- A sketch map $\mathbb{S}_1 \to \mathbb{S}_2$ is a graph map
$$G_1 \longrightarrow G_2$$
that preserves all the constraints of $\mathbb{S}_1$.
$$B \longrightarrow G_1 \longrightarrow G_2$$
- We use sketch maps to formulate the Alignment Problem.
- Given models M_1 and M_2 of a sketch $\mathbb{S}$, a model map $M_1 \to M_2$ is a collection of functions (one for each vertex V of G)
$$M_1(V) \xrightarrow{\tau_V} M_2(V)$$
that are consistent with the edges of G.
- Example:

$$\begin{array}{c} \text{Resident} \\ \Big\downarrow \text{live\_in} \\ \text{Village} \end{array} \qquad \begin{tikzcd} M_1(\text{Resident}) \arrow[r, "\tau"] \arrow[d, "M_1(\text{lives\_in})"'] & M_2(\text{Resident}) \arrow[d, "M_2(\text{lives\_in})"] \\ M_1(\text{Village}) \arrow[r, "\tau"'] & M_2(\text{Village}) \end{tikzcd}$$

- J.W. Gray. The category of sketches as a model for algebraic semantics. 1989.

Sketch Theory: Questions

- EA sketch instance data (models) can be implemented using relational database features such as foreign keys and triggers.
- What technologies support management of distributed models of sketches?
    - Google Megastore, Tenzing or Spanner?
    - Apache Accumulo?
    - Is a graph database more appropriate?

Presentations

- A sketch | first-order theory | ontology is a presentation of knowledge.
- Presentations generate additional knowledge needed for alignment (e.g., 'uncle $=$ brother $\circ$ parent')

Logical theory $\mathbb{T}$	syntactic category $\mathcal{C}_{\mathbb{T}}$
Ontology	rules
Sketch $\mathbb{S}$	theory of a sketch $\mathbb{T}(\mathbb{S})$

- Different presentations may generate 'equivalent' structures.
- Sketches $\mathbb{S}_1$ and $\mathbb{S}_2$ representing common concepts are aligned by finding a sketch $\mathcal{V}$ and sketch maps as shown.

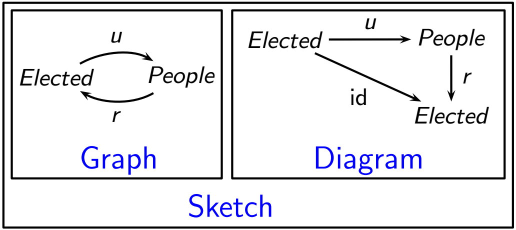
$$\begin{tikzcd}
\mathbb{S}_1 \arrow[d] & & \mathcal{V} \arrow[dll] \arrow[drr] & & \mathbb{S}_2 \arrow[d] \\
\mathcal{T}_1 \arrow[drr] & & & & \mathcal{T}_2 \arrow[dll] \\
& & \mathcal{T} & &
\end{tikzcd}$$

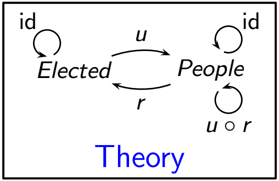
- Theory of a (linear) sketch
  - Carmody-Walters algorithm for computing left Kan extensions: generalizes Todd-Coxeter procedure used in computational group theory
  - Complexity difficult to characterize: can depend on order of constraints

Civics Sketch $\mathbb{S}_1$

First formulation of civics concepts:

- Two classes: People and Elected officials
- People have Elected representatives via r.
- Elected officials are instances of people via u.
- Elected officials represent themselves via a diagram.





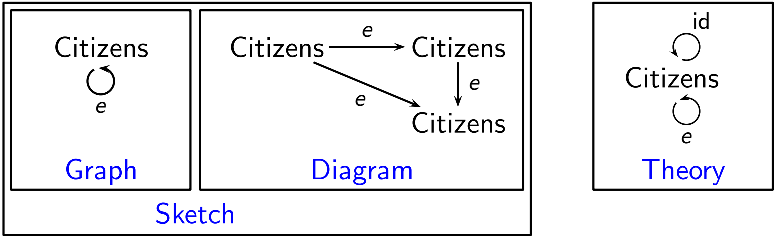
- The diagram truncates the infinite list of composites (property chains).

$$u \circ r \qquad r \circ u \qquad u \circ r \circ u \qquad r \circ u \circ r \qquad \cdots$$

Civics Sketch $\mathbb{S}_2$

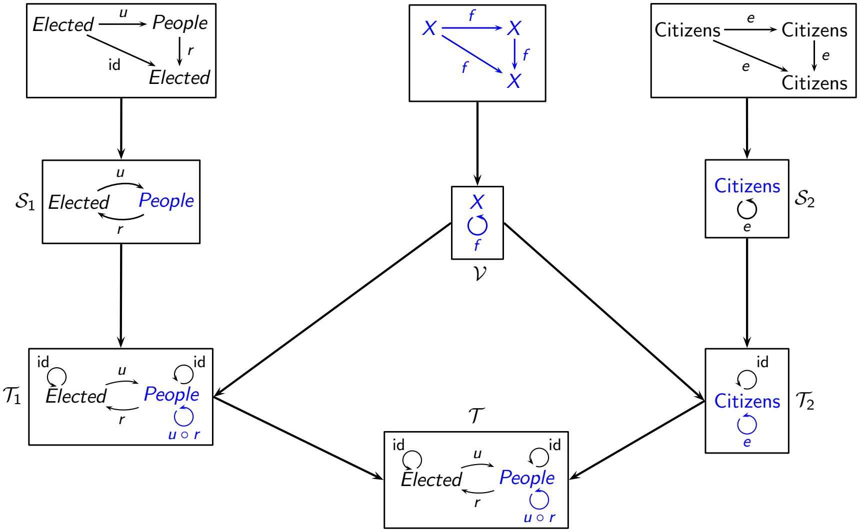
Alternative formulation of the concepts:

- One class: Citizens
- Citizens have elected representatives via *e*.
- Elected officials represent themselves via a diagram.



- Number and names of vertices in $\mathcal{S}_1$ and $\mathcal{S}_2$ differ.
- The edges u and r of $\mathcal{S}_1$ have no corresponding edges in $\mathcal{S}_2$.
- The edge e of $\mathcal{S}_2$ has no corresponding edge in $\mathcal{S}_1$.

Alignment of the Civics Sketches



Sketch Alignment: Questions

- What algorithms are available for computing the theory of a sketch?
  - Carmody-Walters for linear sketches
  - Others?
- To what extent can the sketch alignment problem be automated?
  - Find the appropriate intersection
  - Renaming of vertices and edges
- Can instance data be used to support sketch alignment?

First-Order Civics Theories $\mathbb{T}_1$ and $\mathbb{T}_2$

- $\mathbb{T}_1$
  - **Sorts**: People, Elected
  - **Function symbols**:

$$u : \text{Elected} \longrightarrow \text{People} \qquad r : \text{People} \longrightarrow \text{Elected}$$

  - **Axiom**: elected officials represent themselves

$$\top \vdash_x (r(u(x)) = x)$$

- $\mathbb{T}_2$
  - **Sorts**: Citizens
  - **Function symbols**:

$$e : \text{Citizens} \longrightarrow \text{Citizens}$$

  - **Axiom**: elected officials represent themselves

$$\top \vdash_x (e(e(x)) = e(x))$$

First-Order Logic: Sequent Calculus

Structural Rules[1]

$$(\varphi \vdash_{\vec{x}} \varphi) \qquad \frac{(\varphi \vdash_{\vec{x}} \psi)}{(\varphi[\vec{s}/\vec{x}] \vdash_{\vec{y}} \psi[\vec{s}/\vec{x}])} \qquad \frac{(\varphi \vdash_{\vec{x}} \psi)\ \ (\psi \vdash_{\vec{x}} \chi)}{(\varphi \vdash_{\vec{x}} \chi)}$$

Implication

$$\frac{((\varphi \wedge \psi) \vdash_{\vec{x}} \chi)}{(\varphi \vdash_{\vec{x}} (\psi \Rightarrow \chi))}$$

Equality

$$(\top \vdash_{x} (x = x))$$

$$((\vec{x} = \vec{y}) \wedge \varphi \vdash_{\vec{z}} \varphi[\vec{y}/\vec{x}])$$

Quantification[2]

$$\frac{(\varphi \vdash_{\vec{x},y} \psi)}{((\exists y)\varphi \vdash_{\vec{x}} \psi)} \qquad \frac{(\varphi \vdash_{\vec{x},y} \psi)}{(\varphi \vdash_{\vec{x}} (\forall y)\psi)}$$

Conjunction

$$(\varphi \vdash_{\vec{x}} \top) \qquad ((\varphi \wedge \psi) \vdash_{\vec{x}} \varphi) \qquad ((\varphi \wedge \psi) \vdash_{\vec{x}} \psi) \qquad \frac{(\varphi \vdash_{\vec{x}} \psi)\ \ (\varphi \vdash_{\vec{x}} \chi)}{(\varphi \vdash_{\vec{x}} (\psi \wedge \chi))}$$

Disjunction

$$(\bot \vdash_{\vec{x}} \varphi) \qquad (\varphi \vdash_{\vec{x}} (\varphi \vee \psi)) \qquad (\psi \vdash_{\vec{x}} (\varphi \vee \psi)) \qquad \frac{(\varphi \vdash_{\vec{x}} \chi)\ \ (\psi \vdash_{\vec{x}} \chi)}{((\varphi \vee \psi) \vdash_{\vec{x}} \chi)}$$

Distributive Law[3]

$$((\varphi \wedge (\psi \vee \chi) \vdash_{\vec{x}} (\varphi \wedge \psi) \vee (\varphi \wedge \chi))$$

Frobenius Axiom[3]

$$((\varphi \wedge ((\exists y)\psi) \vdash_{\vec{x}} (\exists y)(\varphi \wedge \psi))$$

Excluded Middle

$$(\top \vdash_{x} (\varphi \vee \neg\varphi))$$

Contexts are suitable for the formulae that occur on both sides of $\vdash$.

1 In the substitution rule, $\vec{y}$ contains all the variables of $\vec{x}$.

2 Bound variables do not also occur free in any sequent.

3 The Distributive Law and Frobenius Axiom are derivable in full, first-order logic.

Soundness

- **Soundness Theorem**: Let $\mathbb{T}$ be a Horn theory and let M be a model of $\mathbb{T}$ in a cartesian category. If $\varphi \vdash_{\vec{x}} \psi$ is provable from $\mathbb{T}$ in Horn logic, then the sequent is satisfied in M.

  Proof: Induction on inference rules using the category properties used to define semantics of terms- and formulae-in-context.
- We can replace Horn and cartesian by any combination of:

Logic	**Category**
Regular	Regular
Coherent	Coherent
First-order	Heyting
Classical first-order	Boolean coherent
Linear	$*$-autonomous
Intuitionistic higher-order	Topos
S4 modal (predicate)	sheaves on a topological space

Completeness

- **Completeness Theorem**: Let $\mathbb{T}$ be a regular theory. If $\varphi \vdash_{\vec{x}} \psi$ is a regular sequent that is satisfied in all models of $\mathbb{T}$ in regular categories $\mathcal{D}$, then it is provable from $\mathbb{T}$ in regular logic.

  Proof: Construct the syntactic category $\mathcal{C}_{\mathbb{T}}$ with a generic model $M_{\mathbb{T}}$

category of models of $\mathbb{T}$ in $\mathcal{D}$	$\cong$	category of regular functors $\mathcal{C}_{\mathbb{T}} \to \mathcal{D}$
$\mathrm{Mod}_{\mathbb{T}}(\mathcal{D})$	$\cong$	$\mathrm{Reg}(\mathcal{C}_{\mathbb{T}}, \mathcal{D})$

- We can replace regular theories and categories by any of:

Logic	**Category**
Cartesian	Cartesian
Coherent	Coherent
First-order	Heyting

- The Completeness Theorem also holds if we replace $\mathcal{D}$ by Set.

Alignment of Logical Theories

- Provable equivalence: applicable to theories over the same signature
- Theories $\mathbb{T}_1$ and $\mathbb{T}_2$ are Morita equivalent if their categories of models $\mathrm{Mod}_{\mathbb{T}}(\mathcal{D})$ (in any category $\mathcal{D}$ of the appropriate class) are equivalent.

$$\mathrm{Mod}_{\mathbb{T}_1}(\mathcal{D}) \cong \mathrm{Mod}_{\mathbb{T}_2}(\mathcal{D})$$

- Theories are Morita equivalent iff their syntactic categories are.

$$\mathcal{C}_{\mathbb{T}_1} \cong \mathcal{C}_{\mathbb{T}_2}$$

- This solves the alignment problem for the civics theories.
- It can be difficult to use in practice.
  - Types are interpreted as equivalence classes of formulae
  - Functions and relations are interpreted as equivalence classes of formuae
  - Syntactic categories are typically infinite, even for simple theories
  - No general algorithm exists
  - Could one develop a lazy algorithm?

Syntactic Categories

- Let $\mathbb{T}$ be a regular theory. There is a regular category $\mathcal{C}_{\mathbb{T}}$ that has a model of $\mathbb{T}$.

objects:	α-equivalence classes of formulae-in-context: $\{\vec{x}.\varphi\}$ where φ is regular over $\mathbb{T}$
morphisms :	$\{\vec{x}.\varphi\} \xrightarrow{[\theta]} \{\vec{y}.\psi\}$ $\theta \vdash_{\vec{x},\vec{y}} \varphi \wedge \psi \qquad \varphi \vdash_{\vec{x}} (\exists \vec{y})\theta \qquad \theta \wedge \theta[\vec{z}/\vec{y}] \vdash_{\vec{x},\vec{y},\vec{z}} (\vec{z} = \vec{y})$
composition:	$\begin{array}{ccc} \{\vec{x}.\varphi\} & \xrightarrow{[\theta]} & \{\vec{y}.\psi\} \\ & \searrow^{[(\exists\vec{y})(\theta\wedge\gamma)]} & \downarrow{[\gamma]} \\ & & \{\vec{z}.\chi\} \end{array}$
identity:	$\{\vec{x}.\varphi\} \xrightarrow{[\varphi\wedge(\vec{x}'=\vec{x})]} \{\vec{x}'.\varphi[\vec{x}'/\vec{x}]\}$

Syntactic Categories (Continued)

- $\mathcal{C}_{\mathbb{T}}$ contains a model of $\mathbb{T}$.

sorts	A	$\{x.\top\}$ for $x : A$
types	1	$\{[].\top\}$
	$A_1 \times \cdots \times A_n$	$\{\vec{x}.\top\}$ for $x_i : A_i$
function symbols	$f : A_1 \times \cdots \times A_n \to B$	$\{\vec{x}.\top\} \xrightarrow{[f(x_1,\ldots,x_n)=y]} \{y.\top\}$ for $x_i : A_i$ and $y : B$
relation symbols	$R \rightarrowtail A_1 \times \cdots \times A_n$	$\{\vec{x}.R(\vec{x})\} \longrightarrow \{\vec{x}.\top\}$

- The axioms of $\mathbb{T}$ are satisfied in this model.

Proof of $(u(x) = u(y)) \vdash_{x,y} (x = y)$ for Civics Theory $\mathbb{T}_1$

1. $(u(x) = u(y)) \vdash_{x,y} (u(x) = u(y))$ .... Id
2. $(u(x) = u(y)) \vdash_{x,y} \top$ .... $\top$
3. $\top \vdash_{x} (r(u(x)) = x)$ .... axiom
4. $\top \vdash_{x,y} (r(u(x)) = x)$ .... Sub (3)
5. $\top \vdash_{x,y} (r(u(y)) = y)$ .... Sub (3)
6. $(x = y) \wedge (r(x) = z) \vdash_{x,y,z} (r(y) = z)$ Eq1
7. $(u(x) = u(y)) \wedge (r(u(x)) = x) \vdash_{x,y,z} (r(u(y)) = x)$ .... Subs (6)
8. $(u(x) = u(y)) \wedge (r(u(x)) = x) \vdash_{x,y} (r(u(y)) = x)$ .... Subs (7)
9. $(x = y) \vdash_{x,y} (y = x)$ .... previous proof
10. $(r(u(y)) = x) \vdash_{x,y} (x = r(u(y)))$ .... Subs (9)
11. $(u(x) = u(y)) \wedge (r(u(x)) = x) \vdash_{x,y} (x = r(u(y)))$ .... Cut (8), (10)
12. $(x = y) \wedge (y = z) \vdash_{x,y,z} (x = z)$ .... previous proof
13. $(x = r(u(y))) \wedge (r(u(y)) = y) \vdash_{x,y,z} (x = y)$ .... Subs (12)
14. $(x = r(u(y))) \wedge (r(u(y)) = y) \vdash_{x,y} (x = y)$ .... Subs (13)
15. $(u(x) = u(y)) \vdash_{x,y} (r(u(x)) = x)$ .... Cut (2), (4)
16. $(u(x) = u(y)) \vdash_{x,y} (u(x) = u(y)) \wedge (r(u(x)) = x)$ .... $\wedge$I (1), (15)
17. $(u(x) = u(y)) \vdash_{x,y} (x = (r(u(y)))$ .... Cut (16), (11)
18. $(u(x) = u(y)) \vdash_{x,y} (r(u(y)) = y)$ .... Cut (2), (5)
19. $(u(x) = u(y)) \vdash_{x,y} (x = r(u(y))) \wedge (r(u(y)) = y)$ .... $\wedge$I (17), (18)
20. $(u(x) = u(y)) \vdash_{x,y} (x = y)$ .... Cut (19), (14)

Prover9 Proof

- Input file:

```
formulas(assumptions).
  all x (r(u(x)) = x).
end_of_list.
formulas(goals).
  all x all y (u(x) = u(y)) -> (x = y).
end_of_list.
```

- Proof:

```
1 (all x r(u(x)) = x) ....................# label(non_clause).  [assumption].
2 (all x all y u(x) = u(y)) -> x = y ................# label(non_clause)
                                                     # label(goal).  [goal].
3 r(u(x)) = x.  ..............................................[clausify(1)].
4 u(x) = u(y).  ...................................................[deny(2)].
5 c2 != c1.  ......................................................[deny(2)].
6 x = y.  .........................[para(4(a,1),3(a,1,1)),rewrite([3(2)])].
7 $F.  ...................................................[resolve(6,a,5,a)].
```

- The shorter proof by contradiction uses classical first-order logic.
- First-order horn logic has lower computational complexity in general.

Q-Sequences and Q-Trees (Freyd-Scedrov 1990)

- P. Freyd and A. Scedrov. *Categories, Allegories*. 1990
- A Q-sequence $\mathcal{Q} = (A, a, Q)$ in a category $\mathcal{D}$ consists of lists of
  - objects $A_0, \ldots, A_n$
  - morphisms $a_i : A_i \to A_{i+1}$ for $0 \leq i < n$
  - quantifiers $Q_0, \ldots, Q_n$

$$\boxed{A_0 \overset{Q_0}{\Big|} \quad A_1 \overset{Q_1}{\Big|} \quad \cdots \quad \overset{Q_{n-1}}{\Big|} \quad A_n \overset{Q_n}{\Big|}}$$

- $\sigma\,\mathcal{Q}$ is: $\boxed{A_1 \overset{Q_1}{\Big|} \quad \cdots \quad \overset{Q_{n-1}}{\Big|} \quad A_n \overset{Q_n}{\Big|}}$
- A morphism $A_0 \xrightarrow{f_0} B$ satisfies $\mathcal{Q}$ if one of the following holds:
  - $n = 0$ and $Q_0 = \forall$
  - $n > 0$, $Q_0 = \forall$, and for every commutative triangle
$$\begin{tikzcd} A_0 \arrow[r, "a_0"] \arrow[dr, "f_0"'] & A_1 \arrow[d, dotted, "f_1"] \\ & B \end{tikzcd}$$
    , the morphism $A_1 \xrightarrow{f_1} B$ satisfies the Q-sequence $\sigma\,\mathcal{Q}$
  - $n > 0$, $Q_0 = \exists$, and there exists a commutative triangle
$$\begin{tikzcd} A_0 \arrow[r, "a_0"] \arrow[dr, "f_0"'] & A_1 \arrow[d, dotted, "f_1"] \\ & B \end{tikzcd}$$
    for which $A_1 \xrightarrow{f_1} B$ satisfies the Q-sequence $\sigma\,\mathcal{Q}$
- Q-trees generalize Q-sequences by allowing branching.

Sketch Inference

In civics sketch $\mathbb{S}_1$, we may conclude that Elected is a subclass of People.

$$\begin{tikzcd} \mathit{Elected} \arrow[r, bend left, "u"] & \mathit{People} \arrow[l, bend left, "r"] \end{tikzcd}$$

Graph

$$\begin{tikzcd} \mathit{Elected} \arrow[r, "u"] \arrow[dr, "\mathrm{id}"'] & \mathit{People} \arrow[d, "r"] \\ & \mathit{Elected} \end{tikzcd}$$

Diagram

$$\begin{tikzcd} Z \arrow[r, bend left, "x"] \arrow[r, bend right, "y"'] \arrow[r, phantom, "+"] & \text{Elected} \end{tikzcd}$$

$\forall$

$$\begin{tikzcd} Z \arrow[r, bend left, "x"] \arrow[r, bend right, "y"'] \arrow[r, phantom, "+"] & \text{Elected} \arrow[r, "u"] & \text{People} \end{tikzcd}$$

$\exists$

$$\begin{tikzcd} Z \arrow[r, bend left, "x"] \arrow[r, bend right, "y"'] \arrow[r, phantom, "+"] & \text{Elected} \arrow[r, "u"] & \text{People} \arrow[r, "r"] & \text{Elected} \end{tikzcd}$$

$\forall$

$$\begin{tikzcd} Z \arrow[r, bend left, "x"] \arrow[r, bend right, "y"'] \arrow[r, phantom, "+"] & \text{Elected} \arrow[r, "u"] \arrow[rr, bend right, "\mathrm{id}"'] & \text{People} \arrow[r, "r"] & \text{Elected} \end{tikzcd}$$

$\exists$

$$\begin{tikzcd} Z \arrow[r, bend left, "x"] \arrow[r, bend right, "y"'] & \text{Elected} \arrow[r, "u"] \arrow[rr, bend right, "\mathrm{id}"'] & \text{People} \arrow[r, "r"] & \text{Elected} \end{tikzcd}$$

$\forall$

Sketch Reasoning: Questions

- Categories, Allegories: 1.3(10)1. *Any elementary predicate in category theory is given by a finitely presented Q-tree with a free category as root.*
- What algorithms have been developed for Q-tree inference?
- Is there a correspondence between classes of logics (Horn, regular, etc.) and classes of Q-trees?
- Is there a correspondence between classes of sketches (linear, finite limit, etc.) and classes of Q-trees?
- τ categories as a guide to implementing cartesian categories?

Transforming Sketches into First-Order Theories

- Sketches are related to first-order logical theories by theorems of the form: Given any sketch $\mathbb{S}$ of class X, there is a logical theory $\mathbb{T}$ of class Y for which $\mathbb{S}$ and $\mathbb{T}$ have equivalent classes of models.
- D2.2 of Johnstone's *Sketches of an Elephant: A Topos Theory Compendium* gives explicit constructions of $\mathbb{T}$ from $\mathbb{S}$ and conversely.

Class of Sketches	Fragment of Predicate Calculus	Logical Connectives
finite limit	cartesian	$=, \top, \wedge, \exists^*$
regular	regular	$=, \top, \wedge, \exists$
coherent	coherent	$=, \top, \wedge, \exists, \perp, \vee$
geometric	geometric	$=, \top, \wedge, \exists, \perp, \bigvee$
σ-coherent	σ-coherent	$=, \top, \wedge, \exists, \perp, \bigvee_{i=1}^{\infty}$
finitary	σ-coherent	

* In cartesian logic, only certain existentially quantified formulae are allowed.

Example: Transforming the Civics Sketches to Theories

- General construction (D2.2 of *Sketches of an Elephant* by P.T. Johnstone)
  - Vertices become sorts
  - Edges become function symbols
  - No relation symbols
  - Diagrams become axioms
  - Cones and cocones induce axiom schema
- $\mathbb{S}_1$ induces $\mathbb{T}_1$ and $\mathbb{S}_2$ induces $\mathbb{T}_2$
- Add a finite limit constraint to $\mathbb{S}_1$

$$\begin{tikzcd} & 1 \arrow[d] \\ 0 \arrow[r] & 2 \end{tikzcd}$$

base

$$\xrightarrow{\varphi}$$

$$\begin{tikzcd} \text{Elected} \arrow[r, "\mathrm{id}"] \arrow[d, "\mathrm{id}"'] \arrow[dr, "u"] & \text{Elected} \arrow[d, "u"] \\ \text{Elected} \arrow[r, "u"'] & \text{Person} \end{tikzcd}$$

All induced sequents are derivable in $\mathbb{T}_1$

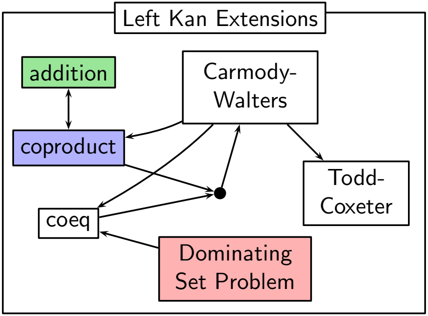
$$\top \vdash_x \big(u(x) = u(x)\big)$$
$$\top \vdash_x \big(u(x) = u(x)\big)$$
$$\big((x = y) \wedge (u(x) = u(y)) \wedge (x = y)\big) \vdash_{x,y} (x = y)$$
$$\big((u(x) = y) \wedge (u(x') = y)\big) \vdash_{x,x',y} \exists x_0\big((x_0 = x) \wedge (u(x_0) = y) \wedge (x_0 = x')\big)$$

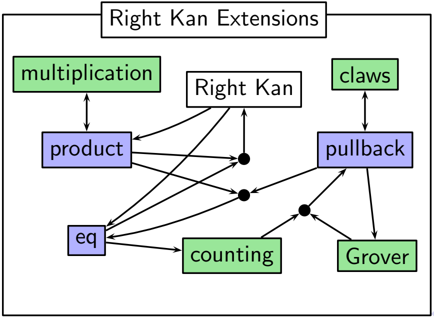
Sketch Translations: Questions

- The proof in 2.2.1 of Johnstone's *Sketches of an Elephant* of the existence of a Morita equivalent sketch for a logical theory (both of suitable classes) is not a direct construction.
- Is there an explicit (finite) construction?
- What classes of sketches correspond to OWL dialects?
- How could such mappings be used to solve the ontology alignment problem?
  - transform ontologies to sketches + instance data
  - align the sketches
  - transform back to ontologies (if necessary)

Quantum Kan Extensions

- Quantum Kan Extensions. IARPA seedling with N. Yanofsky (CUNY)
- What tasks will quantum computers be able to perform better than classical machines?
- Research findings (for Kan extensions in Set_f)
  - Known quantum algorithms
  - Exact quantum algorithms do not improve upon classical complexity
  - NP Complete problem
  - Focus on approximate algorithms, coequalizers and Kan extensions in categories with additional algebraic structure.





Kan Extensions: Definitions

- Given:
  - Categories $\mathcal{A}$ and $\mathcal{B}$ (presented as directed graphs with commutativity constraints)
  - A functor $F : \mathcal{A} \to \mathcal{B}$ (assigning a $\mathcal{B}$-path to each $\mathcal{A}$-edge)
  - An action $X : \mathcal{A} \to \mathbf{Sets}$ (assigning a set to each $\mathcal{A}$-vertex and a function to each $\mathcal{A}$-edge)
- A left Kan extension of X along F consists of:
  - An action $L : \mathcal{B} \to \mathbf{Sets}$
  - A natural transformation $\epsilon_A : X(A) \to L(F(A))$

$$\begin{tikzcd} \mathcal{A} \arrow[rr, "F"] \arrow[dr, "X"'] & \arrow[d, phantom, "\overset{\epsilon}{\nearrow}"] & \mathcal{B} \arrow[dl, dotted, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

- These ingredients satisfy a *universal mapping property*.
- A right Kan extension has ϵ going the other way $\epsilon_A : L(F(A)) \to X(A)$.
- Mac Lane: "All concepts are Kan extensions."

Kan Extensions: Examples

Right Kan Extensions	Left Kan Extensions
products	coproducts
equalizers	coequalizers
fixed points	orbits
greatest lower bound	least upper bound
intersection	union
conjunction $\wedge$	implication $\Rightarrow$
existential quantification $\exists$	universal quantification $\forall$
left adjoints	right adjoints
limits	colimits
ends	coends
claws	coset enumeration

- Right Kan extensions can be calculated from products and equalizers.
- Left Kan extensions can be calculated from coproducts and coequalizers.

Quantum Algorithms for Pullbacks

- Classical complexity: $O(N \log X)$ ($N = \max\{X, Y\}$) to find **all** claws
  - $O(X \log X)$ comparisons to sort the values $f(x)$
  - For each y, $O(\log X)$ comparisons to search for x with $f(x) = g(y)$
- Buhrman, Dürr, Heiligman, Høyer, Magniez, Santha and de Wolf. Quantum Algorithms for Element Distinctness. 2005.
  - $O\left(X^{1/2} Y^{1/4} \log X\right)$ comparisons to (with high probability) find **a** claw (if $X \leq Y \leq X^2$) and $O(Y \log X)$ if $Y > X^2$
  - Theorem: Quantum computers cannot improve upon classical complexity of exact pullback calculations.
    - Assume we have a quantum algorithm that calculates pullbacks.
    - Given $f : X \to Y$, form the pullback P of f with itself.
    - $P - X$ counts the number of *collisions* (i.e., remove the diagonal).
    - Consequently, an efficient pullback algorithm gives an an efficient algorithm for exactly counting the number of collisions.
    - This contradicts Theorem 6.1 of the reference cited above.

Quantum Algorithms for Equalizers

- Classical complexity of finding (E, e) is $O(X)$.
- Theorem: Quantum computers can not improve upon the classical complexity of exact equalizer calculations.
  - Algorithms for equalizers and products give an algorithm for pullbacks.

$$\begin{tikzcd} & & X \arrow[dr, "f"] & \\ E \arrow[r, "e"] & X \times Y \arrow[ur, "\pi_1"] \arrow[dr, "\pi_2"'] & & Z \\ & & Y \arrow[ur, "g"'] & \end{tikzcd}$$

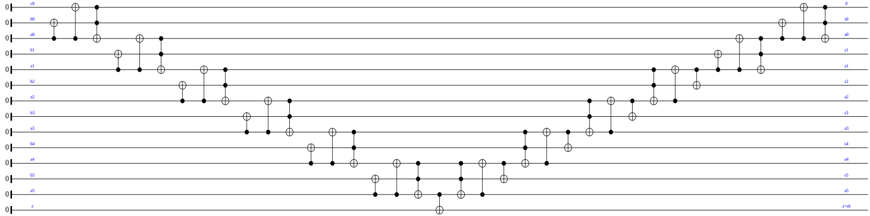
- Approximate quantum algorithm:
  - Brassard, Høyer, Mosca and Tapp. Quantum Amplitude Amplification and Estimation. 2000.
    - **Thm. 18**. `Approx_Count` with $\frac{1}{3X} < \varepsilon \leq 1$ outputs $\widetilde{E}$ with $|\widetilde{E} - E| < \varepsilon E$ with probability $2/3$ and uses an expected number of evaluations of f in the order of $\sqrt{X/\phi} + \sqrt{E(X-E)}/\phi$ where $\phi = \lfloor \varepsilon E \rfloor + 1$.
  - Childs and Eisenberg. Quantum Algorithms for Subset Finding. 2003.
    - **Thm. 1**. The query complexity of $\widetilde{E}$-subset finding is $O(X^{\widetilde{E}/(\widetilde{E}+1)})$.
  - What if `Approx_Count` miscounts?

Quantum Algorithms for Coproducts

- Adaptation of classical algorithms
  - Vedral, Barenco and Ekert. Quantum Networks for Elementary Arithmetic Operations. 1996
    - $O(n)$ depth and $O(n)$ ancillary qubits
  - Draper, Kutin, Rains and Svore. A Logarithmic-Depth Quantum Carry-Lookahead Adder. 2008
    - $O(\log(n))$ depth and $O(n)$ ancillary qubits
  - Cuccaro, Draper, Kutin and Moulton. A New Quantum Ripple-Carry Addition Circuit. 2008
    - $O(n)$ depth and 1 ancillary qubit
- Approximate Fourier transform
  - Draper. Addition on a Quantum Computer. 2000
  - Barenco, Ekert, Suominen and Törmä. Approximate Quantum Fourier Transform and Decoherence. 2008

Coproducts: Implementation (Quipper)

- Cuccaro, Draper, Kutin and Moulton. A new quantum ripple-carry addition circut. 2008. arXiv:quant-ph/0410184v1
- We can implement this in Quipper. Here is a 6-bit circuit:



```
maj :: (Qubit,Qubit,Qubit)->Circ(Qubit,Qubit,Qubit)
maj (q1, q2, q3) = do
    q2 <- qnot q2 `controlled` q3
    q1 <- qnot q1 `controlled` q3
    q3 <- qnot q3 `controlled` [q1, q2]
    return (q1, q2, q3)
```

```
uma :: (Qubit,Qubit,Qubit)->Circ(Qubit,Qubit,Qubit)
uma (q1, q2, q3) = do
    q3 <- qnot q3 `controlled` [q1,q2]
    q1 <- qnot q1 `controlled` q3
    q2 <- qnot q2 `controlled` q1
    return (q1, q2, q3)
```

- This implementation is distinct from that in the `Libraries.Arith` Quipper module which has many more ancillary qubits.
- Note: See quantum programming language references by P. Selinger

The Todd-Coxeter Coset Enumeration Algorithm

- Dehn (1911): Find an algorithm to decide whether, in a finitely-presented group, a word in the generators represents the identity element.
- Todd-Coxeter (1936): Algorithm for enumerating cosets of $H \leq G$.
  - Haselgrove (1953) gave the first computer implementation.
  - Now implemented in many computer algebra systems.
- Novikov, Boone and Britton (1955–1963): The word problem is unsolvable (in finite time by any Turing machine).
- Cannon, Dimino, Havas and Watson. Implementation and Analysis of the Todd-Coxeter Algorithm (1973).
  - Given group G and integer m, there is a presentation of G for which Todd-Coxeter will generate at least m cosets.
  - The number of cosets generated by Todd-Coxeter can vary with the order of the relations in the presentation.
- Carmody-Walters (1995): Left Kan extension algorithm for finitely-presented groups generalizes Todd-Coxeter.

Calculation of Left Kan Extensions when $\mathcal{C} = \textbf{Sets}$

$$L(B) = \left(\sum_{A \in \mathcal{A}} \mathcal{B}(F(A),\, B) \times X(A) \right) \Big/ \sim \qquad L(\beta)[b,\, x] = [\beta \circ b,\, x]$$

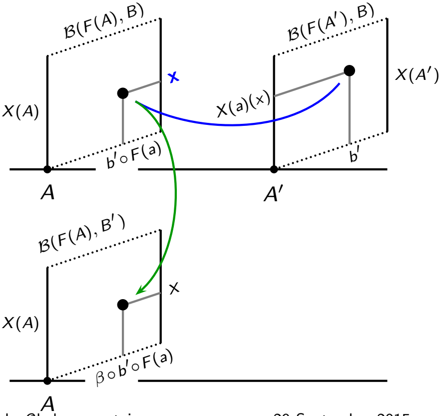
$$A \xrightarrow{a} A'$$

$$\begin{tikzcd} F(A) \arrow[r, "F(a)"] \arrow[dr, "b"'] & F(A') \arrow[d, "b'"] \\ & B \arrow[d, "\beta"] \\ & B' \end{tikzcd}$$

$$X(A) \xrightarrow{X(a)} X(A')$$

$$\begin{tikzcd} L(B) \arrow[d, "L(\beta)"'] \\ L(B') \end{tikzcd}$$

$$(b' \circ F(a),\, x) \sim (b',\, X(a)(x))$$



$$\begin{tikzcd} {[b,\, x]} \arrow[d, mapsto] \\ {[\beta \circ b,\, x]} \end{tikzcd}$$

Carmody-Walters Algorithm: Sample Calculation

Setup	Generators	Relation	Category

Setup

$$\begin{tikzcd} {|\mathcal{B}|} \arrow[r, mapsto] \arrow[dr, "X"'] & \mathcal{B} \arrow[d, dotted, "L"] \\ & \textbf{Sets} \end{tikzcd}$$

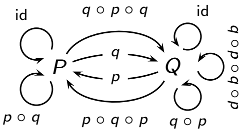
Generators

$$\begin{tikzcd} P \arrow[r, bend left, "q"] & Q \arrow[l, bend left, "p"] \end{tikzcd}$$

Relation

$$\begin{tikzcd} P \arrow[rr, "\mathrm{id}"] \arrow[d, "q"'] & & P \\ Q \arrow[r, "p"'] & P \arrow[r, "q"'] & Q \arrow[u, "p"'] \end{tikzcd}$$

Category



ϵ-tables

$X(P) \to L(P)$	
1	1

$X(Q) \to L(Q)$	
1	1

$L(P)$ = arrows into P

$L(Q)$ = arrows into Q

L-tables

$L(P) \to L(Q)$	
1	2
2	3
3	4
4	5

$L(Q) \to L(P)$	
1	2
2	3
3	4
4	5
5	6

Relation-table

$L(P) \to$	$L(Q) \to$	$L(P) \to$	$L(Q) \to$	$L(P)$	$L(P)$
1	2	3	4	5	1
2	3	4	5	6	2
3	4	1	2	3	3
4	5	2	3	4	4

Uncertainty

- An incomplete list of mathematical models of uncertainty:
  - Probabilities, fuzzy sets, rough sets, vague sets, convex sets, intervals, upper and lower probabilities, sets of probabilities, higher-order probabilities, imprecise probabilities, fuzzy measures, inner measures, outer measures, hints, boolean opinions of experts, probabilistic opinions of experts, Dempster-Shafer belief functions, Spohnian disbelief functions, plausibility functions, ranking functions, possibility functions, propositional logic, predicate logic, higher-order logic, linear logic, intuitionistic logic, modal logics, temporal logics, default logic, relative likelihoods, likelihood logic, conditional logic, Bayesian networks, credal networks, neural networks, gambles, . . .
- Program:
  - Find category-theoretic formulations
  - Derive mappings
  - Establish logical properties
- D. Scott and P. Krauss. Assigning Probabilities to Logical Formulas. 1966
- "Apparently what is needed is a new interpretation of if-then statements"
  J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. 1988

Mes_Π

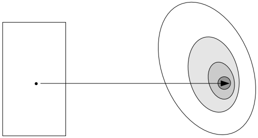
- Lawvere (1960s, unpublished), Čencov (1982), Giry (1985)
- Mes_Π = Stochastic category or category of statistical decisions
- Appears as the semantic category for the probabilistic functional programming language λ_O. Pfenning, Park, and Thrun. A Probabilistic Language Based on Sampling Functions. 2004
- (Π, η, μ) is a monad on Mes with

$$\eta_{\mathcal{X}} = \text{Dirac measure} \quad \text{and} \quad \mu_{\mathcal{X}}(\varpi)(A) = \int_X ev_A \, d\varpi$$

- Mes_Π is the Kleisli category of the monad.

$$\begin{tikzcd}
 & & \Pi^2(\mathcal{Z}) \arrow[d, "\mu_{\mathcal{Z}}"] \\
 & \Pi(\mathcal{Y}) \arrow[ur, "\Pi(g)"] & \Pi(\mathcal{Z}) \\
\mathcal{X} \arrow[ur, "f"] & \mathcal{Y} \arrow[ur, "g"'] & \mathcal{Z}
\end{tikzcd}$$

$$\begin{tikzcd}
 & \text{convex sets} \arrow[dr] & \\
\mathsf{Mes} \arrow[ur] \arrow[dr] & & \mathsf{Mes} \\
 & \mathsf{Mes}_\Pi \arrow[ur] \arrow[uu, dotted] &
\end{tikzcd}$$

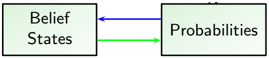


Dempster-Shafer Theory

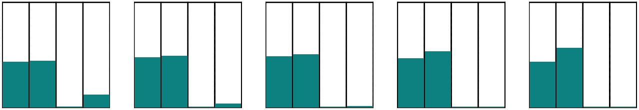
- Adapt Wendt's categogory M0RP of measure-zero reflecting maps between probability spaces.
- Define the category P0R of plausibility-zero reflecting maps between Dempster-Shapfer spaces.
- Obtain a faithful D : M0RP $\rightarrow$ P0R and a faithful V : P0R $\rightarrow$ M0RP (via the Voorbraak map) satisfying the following with V right adjoint to D.

$$
\begin{tikzcd}
 & \mathrm{P0R} \arrow[d, "V"] \\
\mathrm{M0RP} \arrow[ur, "D"] \arrow[r, "1"'] & \mathrm{M0RP}
\end{tikzcd}
$$

Belief States and Bayesian Belief States

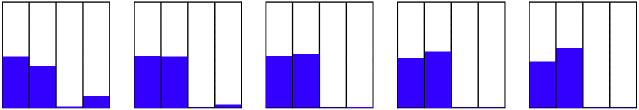


- Voorbraak map
  - Consistent with Dempster's and Bayes' Rules
  - Unnormalized version used by MIT LL and IDA



$\mathfrak{v}(m)$ $\mathfrak{v}(m \oplus m)$ $\mathfrak{v}(m \oplus m \oplus m)$ $\mathfrak{v}(\oplus_{10} m)$ $\mathfrak{v}(\oplus_{20} m)$

- Pignistic map
  - Inconsistent with Dempster's and Bayes' Rules
  - One method used by Rayetheon and Northrup-Grumman



$\mathfrak{p}(m)$ $\mathfrak{p}(m \oplus m)$ $\mathfrak{p}(m \oplus m \oplus m)$ $\mathfrak{p}(\oplus_{10} m)$ $\mathfrak{p}(\oplus_{20} m)$

Other Potential Research Directions

- Context inference and ranking
- Computational linguistics with the Lambek calculus
- Information fusion using the stochastic category

Challenge: Construct Views Tailored to Contexts

- Research area with narrower scope: context-sensitive Internet search
  - Google patent for "methods, systems and apparatus including computer program products, in which context can be used to rank search results" (USPTO 8,209,331 — 2012)
  - Yandex personalized web search challenge: www.kaggle.com
- User-selected context in IBM's Watson
- Techniques to infer context from activities and rank data elements
  - Variable-length hidden Markov model
  - Parametric models of users
  - RankNet, LambdaRank, RankSVM
- Performance metrics used for context-sensitive rankings
  - Normalized discounted cumulative gain (scoring in Kaggle competition)
  - Kendall's τ comparison of rankings
  - Jaccard distance between top N rankings and target

Papers

- R. L. Wojtowicz. *On Categories of Cohesive, Active Sets and Other Dynamic Systems*. Dissertation. University of Illinois at Urbana-Champaign. 2002.
- R. L. Wojtowicz. Symbolic Dynamics and Chaos Defined by Right Adjointness. *CASYS'03- Sixth International Conference on Computing Anticipatory Systems (Liege, Belgium)*. D. Dubois, Editor. American Institute of Physics Conference Proceedings. (718):268-281. 2004. www.adjoint-functors.net/aipcasys2.pdf
- R. L. Wojtowicz. *Categorical Logic as a Foundation for Reasoning Under Uncertainty* SBIR Phase I Final Report. 2005.
- R. L. Wojtowicz. Non-Classical Logic and Network Analysis. IEEE Conference on Information Fusion. Seattle, WA. July, 2009.
- R. L. Wojtowicz. On Transformations Between Belief States. In *Soft Methods for Handling Variability and Imprecision*. Springer-Verlag. pp. 313–320. 2008
- R. L. Wojtowicz, S. Bringsjord and J. Hummel. Dynamic Semantics of τN-Theories. Turing Centenary Conference. Cambridge, UK. 2012.
- R. L. Wojtowicz and N. S. Yanofsky. *Quantum Kan Extensions and Applications*. IARPA contract D11PC20232 Final Report. 2013
- R. L. Wojtowicz. Sketches, Views and Pattern-Based Reasoning. STIDS 2013. George Mason University. November 2013.
- R. L. Wojtowicz. Sketch Theory as a Framework for Knowledge Management. to appear in *Innovation in Systems and Software Engineering*. Springer-Verlag. 2015