Data Semantics, Sketches and Q-Trees

Category Theory Octoberfest

28 October 2017

Ralph L. Wojtowicz

Shepherd University
Shepherdstown, WV
rwojtowi@shepherd.edu

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

Science Technology Service

Background and Perspective

- Project Experience
  - Consultant: Senior Hadoop Analyst for PNC Financial Services. 2015
  - Consultant: Statistical analysis and model development for Flexible Plan Investments, Bloomfield Hills, MI. 2014–2016
  - Established 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. *Passive Sonar Algorithm Development*. ONR. 2010
  - Technical Lead. *Exposing/Influencing Hidden Networks*. ONR. 2009–10
  - PI: *Robust Decision Making*. AFOSR. 2008–2010
  - Analyst: *TradeNet Integration into Global Trader*. ONI. 2009
  - PI with S. Awodey (CMU): *Categorical Logic as a Foundation for Reasoning Under Uncertainty*. Phase I–II SBIR. MDA. 2005–8

Aspects of Knowledge Technologies

- Mathematical Logic (1879)
  - Availability of automated theorem provers (Prover9, Vampire, ...)
  - High computational complexity of some predicate calculus fragments
  - Complexity of the syntactic category used for knowledge alignment
  - Challenging to develop a human interface
- Databases + SQL (1968)
  - Excellent software infrastructure
  - Limited notion of context/view (a single table), static schema, ...
- Semantic Web OWL/RDF + Description Logic (1999)
  - Excellent software infrastructure (Apache Jena, Protégé, ...)
  - 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 (1968/2000) + Q-Trees (1990)
  - Few software tools (however, see www.mta.ca/∼rrosebru/project/Easik)
  - Mature mathematical framework including sketch and model maps
  - Visual/graphical modeling
  - Deduction system?

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

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

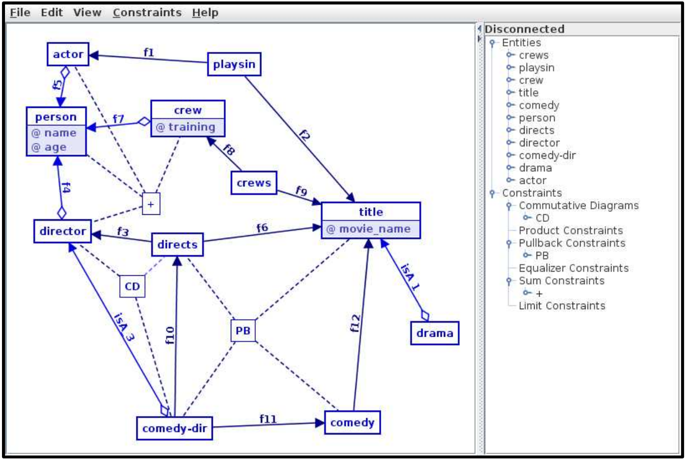
- All semantic constraints in a sketch are expressed using graph maps.
- A sketch $(G, \mathcal{D}, \mathcal{L}, \mathcal{C})$ consists of:

An underlying graph G and sets

$\mathcal{D}$ of diagrams $B \to G$

$\mathcal{L}$ of cones $L \to G$

$\mathcal{C}$ 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	•	•	×	×	×	×	•	•

Questions

- EA sketch instance data (models) can be implemented using relational database features such as foreign keys and triggers.
- What features are required to store instance data for more expressive classes of sketches?
- What technologies support management of large, distributed models of sketches?
- How would relevant algorithms need to be reformulated in a distributed setting?

Presentations

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

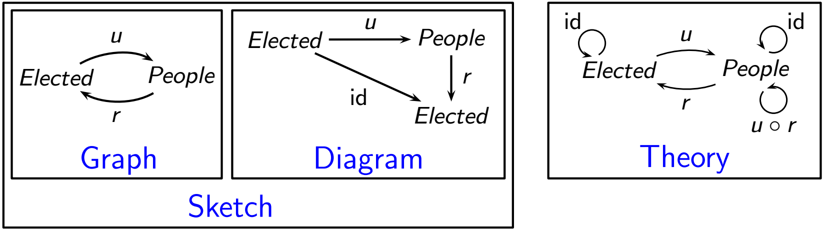
Framework	**Alignment Tool**
Ontology	rules
Sketch $\mathbb{S}$	theory of a sketch $\mathcal{T}(\mathbb{S})$
Logical theory $\mathbb{T}$	syntactic category $\mathcal{C}_{\mathbb{T}}$

- Different presentations may generate ‘equivalent’ structures.
- 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 a civics concept:

- 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.



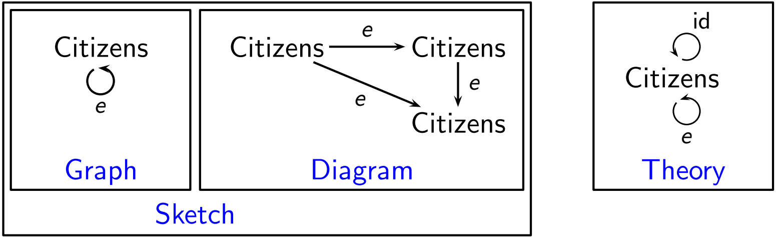
- The diagram truncates the infinite list of composites (property chains).

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

Civics Sketch $\mathbb{S}_2$

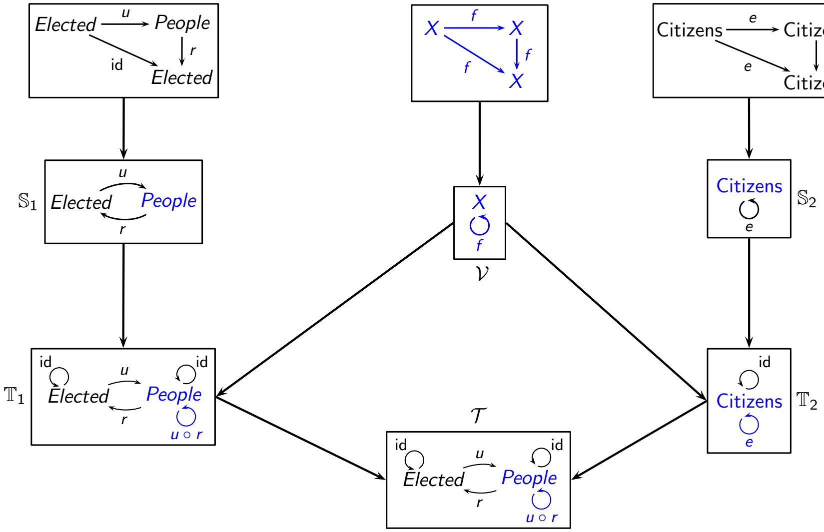
Alternative formulation of the civics concept:

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



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

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?
  - Lazy algorithms?
- To what extent can the sketch alignment problem be automated?
  - Find appropriate intersection(s)/views
  - Rename 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

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

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 $\mathsf{Mod}_{\mathbb{T}}(\mathcal{D})$ (in any category $\mathcal{D}$ of the appropriate class) are equivalent.

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

- 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 provable equivalence classes
  - Syntactic categories are typically infinite, even for simple theories
  - No general algorithm
  - Could one develop a lazy algorithm?

First-Order Logic: Sequent Calculus

Structural Rules[1]

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

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

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

Disjunction

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

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.

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 :	$\mathbb{T}$-provable equivalence classes $[\theta]$ with $\{\vec{x}.\varphi\} \xrightarrow{[\theta]} \{\vec{y}.\psi\}$ $\theta \vdash_{\vec{x},\vec{y}} \varphi \wedge \psi \qquad \varphi \vdash_{\vec{x}} (\exists \vec{y})\,\theta \qquad \theta \wedge \theta[\vec{z}/\vec{y}] \vdash_{\vec{x},\vec{y},\vec{z}} (\vec{z} = \vec{y})$
composition:	$\{\vec{x}.\varphi\} \xrightarrow{[\theta]} \{\vec{y}.\psi\} \xrightarrow{[\gamma]} \{\vec{z}.\chi\}$, composite $\{\vec{x}.\varphi\} \xrightarrow{[(\exists \vec{y})(\theta \wedge \gamma)]} \{\vec{z}.\chi\}$
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	$R \rightarrowtail A_1 \times \cdots \times A_n$	$\{\vec{x}.R(\vec{x})\} \longrightarrow \{\vec{x}.\top\}$

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 categorical properties used to define semantics of terms- and formulae-in-context.

- We can replace Horn and cartesian with other combinations:

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 with:

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

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

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.

Sketch Inference Strategies

How do we show that a property P, that is not an explicit constraint, holds in a sketch?

- Add a constraint for P then show that the resulting sketch is Morita equivalent to the original one.
  - This could change the sketch class (e.g., from linear to finite limit)
- Show that P holds in every model then apply a completeness theorem.
- Translate the sketch into a Morita equivalent theory, then use a sequent calculus.
- Show that P holds in the theory $\mathcal{T}(\mathbb{S})$ of the sketch
  - Express P as a constraint $\mathcal{D}$ then determine if $\mathcal{T}(\mathbb{S})$ satisfies the constraint $\mathcal{D} \to \mathcal{T}(\mathbb{S})$
  - Express P as satisfaction of a Q-tree.
    P may be expressible using different Q-trees.

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

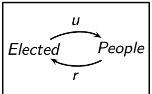
- P. Freyd and A. Scedrov. *Categories, Allegories*. 1990
- A Q-sequence $\mathcal{Q} = (A, a, Q)$ in a category $\mathcal{D}$ consists of lists of
  - objects $A_0, \ldots, A_n$
  - morphisms $a_i : A_i \to A_{i+1}$ for $0 \leq i < n$
  - quantifiers $Q_0, \ldots, Q_n$

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

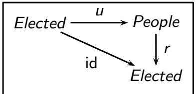
- $\sigma\,\mathcal{Q}$ is: $\boxed{A_1 \overset{Q_1}{|} \quad \cdots \overset{Q_{n-1}}{|} \quad A_n \overset{Q_n}{|}}$
- A morphism $A_0 \xrightarrow{f_0} B$ satisfies $\mathcal{Q}$ if one of the following holds:
  - $n = 0$ and $Q_0 = \forall$
  - $n > 0$, $Q_0 = \forall$, and for every commutative triangle $A_0 \xrightarrow{a_0} A_1$, $A_0 \xrightarrow{f_0} B$, $A_1 \overset{f_1}{\dashrightarrow} B$, the morphism $A_1 \xrightarrow{f_1} B$ satisfies $\sigma\,\mathcal{Q}$
  - $n > 0$, $Q_0 = \exists$, and there exists a commutative triangle $A_0 \xrightarrow{a_0} A_1$, $A_0 \xrightarrow{f_0} B$, $A_1 \overset{f_1}{\dashrightarrow} B$ for which $A_1 \xrightarrow{f_1} B$ satisfies $\sigma\,\mathcal{Q}$
- 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.

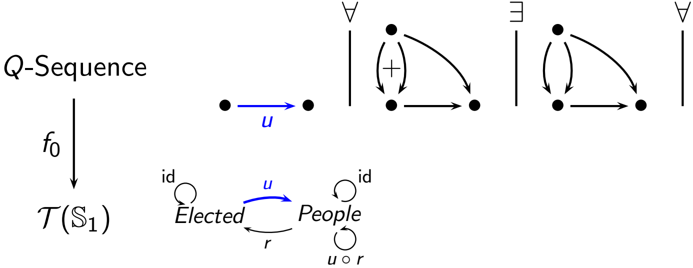


Graph

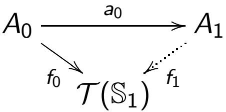


Diagram

- In Cat, the indicated f_0 satisfies the given Q-sequence.



- There are two commutative triangles
- In both cases, f_1 satisfies $\sigma\, \mathcal{Q}$.



Sketch Inference: Questions

- Categories, Allegories 1.398. *Equivalence functors between categories preserve and reflect satisfaction of those Q-trees all of whose functors separate objects.*
    - Morita equivalent sketches (those having equivalent theories) satisfy the same Q-trees.
- Categories, Allegories 1.3(10). *For any elementary property on diagrams preserved and reflected by equivalence functors, there is a finitely presented Q-tree all of whose functors separate objects.*
    - A completeness theorem for sketches?
- What algorithms have been developed for verifying satisfaction of Q-trees?
- Different Q-trees can express the same constraint. Is there a notion of map/equivalence between Q-trees?

Transforming Sketches into First-Order Theories

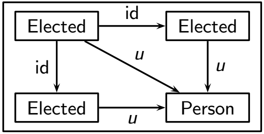
- Sketches are related to first-order logical theories by theorems of the form: Given any sketch $\mathbb{S}$ of class X, there is a logical theory $\mathbb{T}$ of class Y for which $\mathbb{S}$ and $\mathbb{T}$ have equivalent classes of models.
- D2.2 of Johnstone's *Sketches of an Elephant: A Topos 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	$=, \top, \wedge, \exists, \bot, \vee$
geometric	geometric	$=, \top, \wedge, \exists, \bot, \bigvee$
σ-coherent	σ-coherent	$=, \top, \wedge, \exists, \bot, \bigvee_{i=1}^{\infty}$
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 $\mathbb{S}_1$



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

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

$$((x = y) \wedge (u(x) = u(y)) \wedge (x = y)) \vdash_{x,y} (x = y)$$

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

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)