Constant Complements, Reversibility and Universal View Updates

Michael Johnson

(joint work with Bob Rosebrugh)

CT - 2006

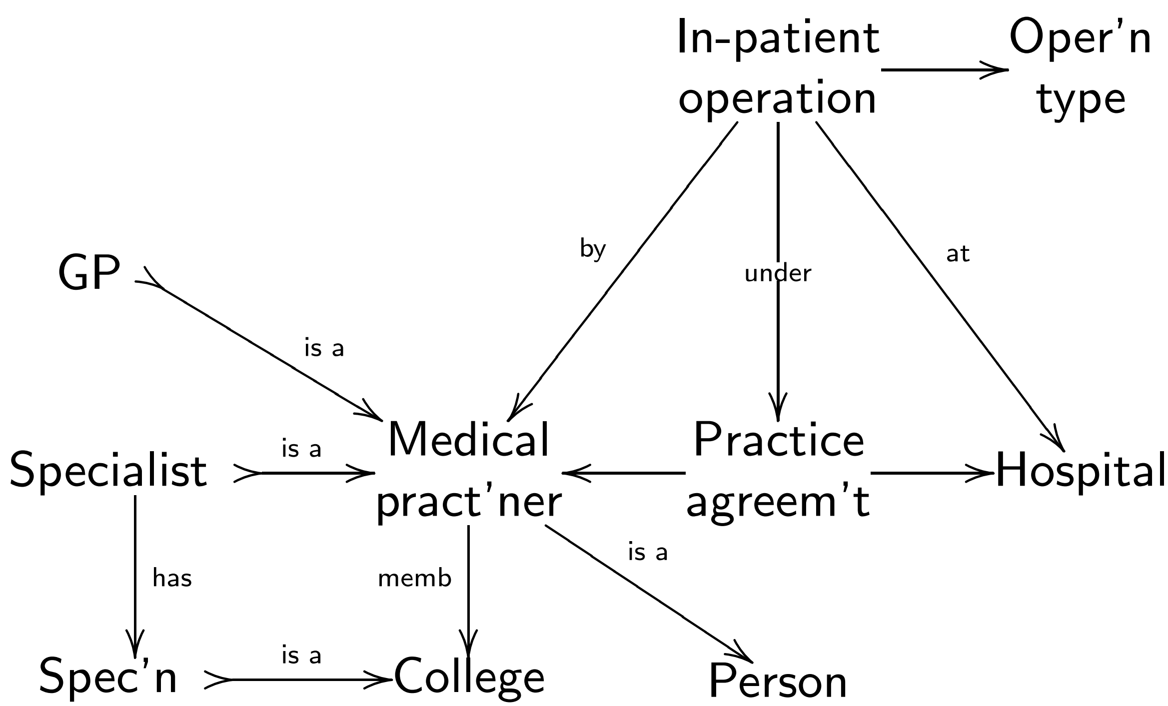
What is . . .

What is a database?	—	sets of tuples (tables)
		(eg address book)
		algebra (multi-sorted)
a database specification?	—	table structure
		theory
a constraint?	—	extra software (triggers)
		commutative diagram

Entities are "things in the world"

Attributes are "values possessed by entities"
(names, addresses, phone numbers . . .)

Example theory (and algebra)



Fragment of a hospital database

Attributes not shown (people have names. . .)

Triangles and square commute

Square is pb; monos by unshown pb squares

Issues

Implementation — well handled commercially

Efficiency — eg query optimization

…

Specification

Security

Views

Interoperation

Specification

ERA models vs Theories

(communication)

Commutative diagrams as constraints

(business rules)

Categorical logic

(constraints inherent in theory)

Leverage

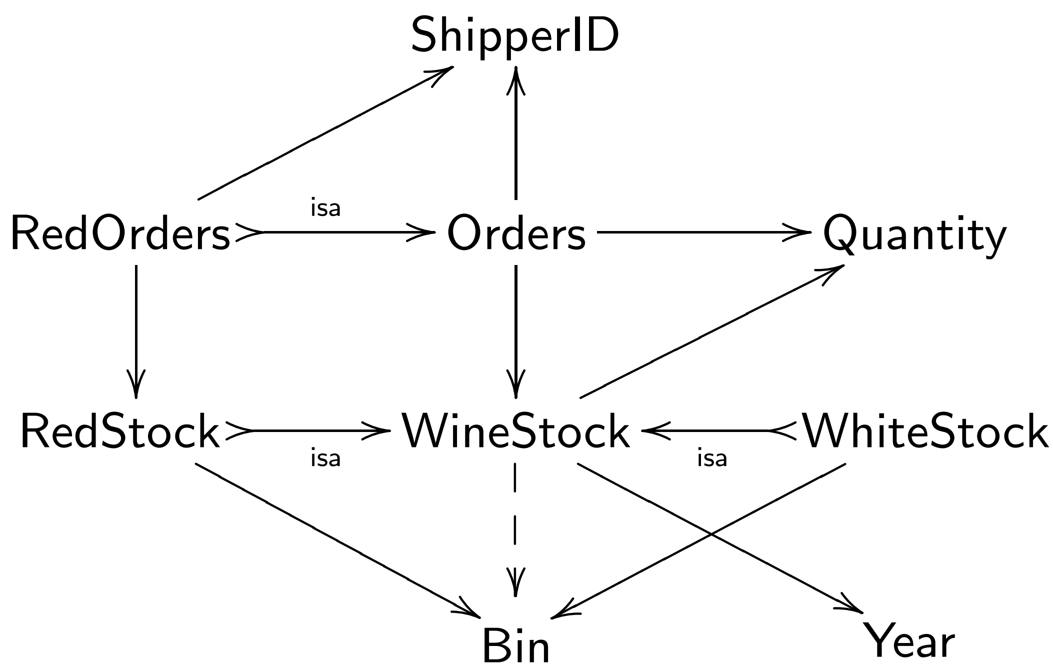
Categorical Specification

An *EA-sketch* $\mathbb{E} = (G, \mathbf{D}, \mathcal{L}, \mathcal{C})$ is a finite limit, finite coproduct sketch with

- a specified empty base cone in $\mathcal{L}$ (vertex is called 1); domain 1 arrows called *elements.*
- *attributes* are vertices of cocones with injections only elements (assume attribute not the domain of arrow); non-attributes called *entities*.
- the graph of G is finite.

An EA sketch is *keyed* if each entity E has a specified monic arrow $k_E : E \rightarrowtail A_E$ to a chosen attribute A_E.

Another example



Graph fragment for a Winery data model

Triangles and square commute

Square is pb; monos from pb

Winestock is a sum

Quantity, ShipperID are attributes

DB state (model, algebra)

- Sketch morphism $\mathbb{E} \longrightarrow \mathbf{Set}_0$ (finite sets)
- Equivalently: finite limit, finite coproduct preserving functor $Q\mathbb{E} \longrightarrow \mathbf{Set}_0$
- More generally, a database state D for an EA sketch $\mathbb{E}$ is a a model of $\mathbb{E}$ in a *lextensive* $\mathbf{S}$ (finite lims and disjoint universal sums)
- *Category of database states of* $\mathbb{E}$ is $\mathrm{Mod}\,(\mathbb{E}, \mathbf{S})$ – morphisms are natural transformations.
- *Update* changes state by deletion and/or insertion
- If $\mathbb{E}$ is keyed then arrows of $\mathrm{Mod}\,(\mathbb{E}, \mathbf{S})$ are monic

Views

A view is . . .

DB theorists

function: states $\longrightarrow$ states (often surjective)

Category theorists

$V : \mathbb{V} \longrightarrow Q\mathbb{E}$ a sketch morphism, hence

$V^* : \mathrm{Mod}\,\mathbb{E} \longrightarrow \mathrm{Mod}\,\mathbb{V}$ a functor

Too abstract? Too far from reality? No!

And what's hard anyway?

View updating

Constant complements

Let $\mathbb{E}$, $\mathbb{V}$ and $\mathbb{C}$ be EA sketches and $\mathbb{V} \xrightarrow{V} Q(\mathbb{E})$ and $\mathbb{C} \xrightarrow{C} Q\mathbb{E}$ be views. We say C is a *complement* of V if the functor

$$\mathrm{Mod}\,(\mathbb{E}) \xrightarrow{\langle V^*, C^* \rangle} \mathrm{Mod}\,(\mathbb{V}) \times \mathrm{Mod}\,(\mathbb{C})$$

is full, faithful and one-one on objects.

Let $\mathbb{V} \xrightarrow{V} Q(\mathbb{E})$ and $\mathbb{C} \xrightarrow{C} Q\mathbb{E}$ be views with C a complement of V and $\alpha : R \longrightarrow V^*D$ be an arrow in $\mathrm{Mod}\,(\mathbb{V})$. We say that α has a *C-constant update* if there is $\hat{\alpha}$ in $\mathrm{Mod}\,(\mathbb{E})$ with $\alpha = V^*\hat{\alpha}$ and $C^*(\hat{\alpha})$ an isomorphism.

(Note $\hat{\alpha}$ is not necessarily cartesian)

Universal view updates

Asking that for a view state delete there should exist a universal lifting to the underlying database state leads to...

Let V be a view schema for $\mathbb{E}$,
D be a state of $\mathbb{E}$,
$T = V^*D$ and $t : T' \rightarrowtail T$.
The delete update t is *propagatable* if there exists a delete $m : D' \rightarrowtail D$ with:
for any state D'' and delete $m'' : D'' \rightarrowtail D$ such that V^*m'' factors as tt' there is a unique delete $m' : D'' \rightarrowtail D'$ such that $V^*m' = t'$.
(propagatable insert is dual)

That is, t is a propagatable delete if it has a cartesian arrow m:

$$
\begin{tikzcd}
D'' \arrow[dr, tail, dashed, "m'"'] \arrow[drr, tail, "m''"] \arrow[dd, dash, dashed] & & \\
& D' \arrow[r, tail, "m"] \arrow[dd, dash, dashed] & D \arrow[dd, dash, dashed] \\
V^*D'' \arrow[dr, tail, "t'"'] \arrow[drr, tail, "V^*m''"] & & \\
& T' \arrow[r, tail, "t"'] & V^*D
\end{tikzcd}
$$

all insert updates for V are propagatable exactly when V^* is a *fibration*.
(remember that for keyed sketches *all* morphisms are monic)

.

Reversibility

Let $\mathbb{V} \xrightarrow{V} Q(\mathbb{E})$ be a view and $\alpha : R \longrightarrow V^*M$ a propagatable deletion in $\mathrm{Mod}\,(\mathbb{V})$. We say that α is *reversible* if its cartesian arrow $\alpha M \xrightarrow{\hat{\alpha}} M$ is also opcartesian. Similarly, a propagatable insertion is reversible if its opcartesian arrow is also cartesian.

Results

- Constant complement updates are universal
- But there exist universal view updates which are not constant compliment
- Constant complement updates are reversible
- Reversible updates are universal (defn)
- But there exist universal view updates which are not reversible

For related articles see:

`www.cs.mq.edu.au/~mike`

`www.mta.ca/~rrosebru`

Remarks on abstraction

"Give me something concrete like a function from states to states, not something abstract like V^* for finite-limit, finite-coproduct preserving functors V ..."

Categorical study of databases

Dampney-Johnson-Monro 1992: An illustrated mathematical foundation for ERA

Rosebrugh-Wood 1992: Relational databases and indexed categories

Baclawski-Simovici-White 1994: A categorical approach to database semantics

Diskin-Cadish 1995–: Algebraic graph-based approach…

Piessens 1995–: Categorical data specifications…

Benson 1996: Stone duality between queries and data

Tuijn-Gyssens 1996: A categorical graph-oriented object data model

Johnson-Rosebrugh-Dampney-Wood 1997–: the Sketch Data Model

Pierce 2006: Lenses and view update translation