Mathematical Foundations of Bidirectional Transformations

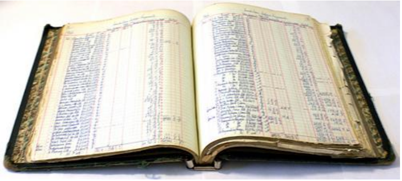
Michael Johnson

Optus Macquarie Cyber Security Hub
Macquarie University Sydney

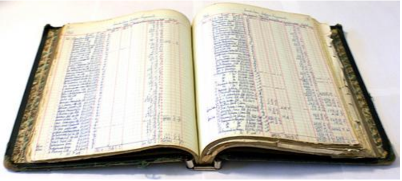
BXSS, Oxford, July 25-29, 2016

$\rightleftarrows$

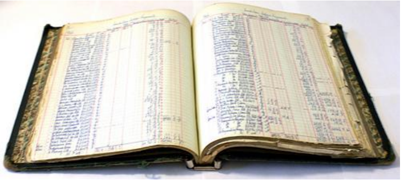
A long time ago



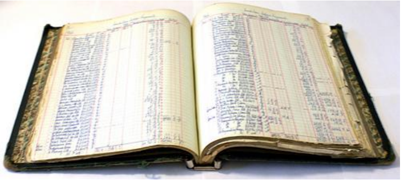
A long time ago

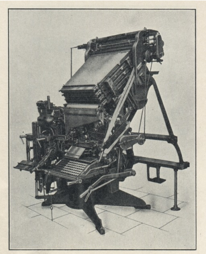


A long time ago



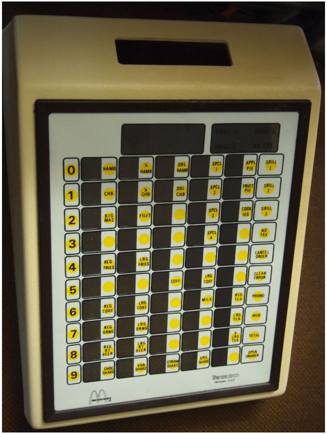
A long time ago



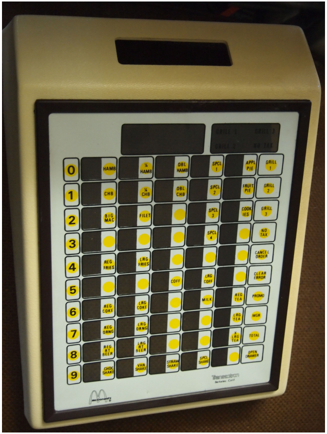


MODEL 6.

Still a long time ago

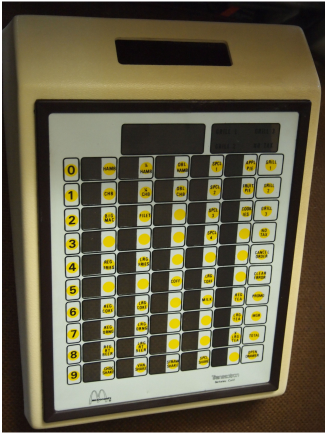


Still a long time ago



PLESSEY

Still a long time ago



PLESSEY

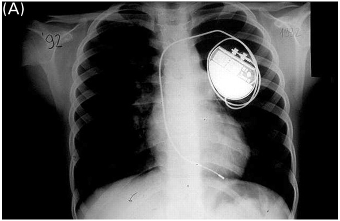
Plessey and Longwick Holdings (Aust): "the first commercial use of client-server technology, peer-to-peer communications, local area network . . . simultaneous backup . . ."

And then . . .

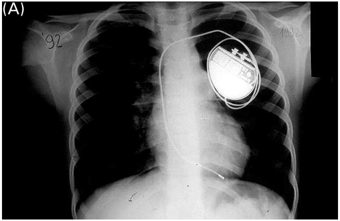

TELECTRONICS
META MV 1202
VVIR • MV
Telectronics
Pacing Systems

And then . . .


TELECTRONICS
META MV 1202
VVIR • MV
Telectronics
Pacing Systems

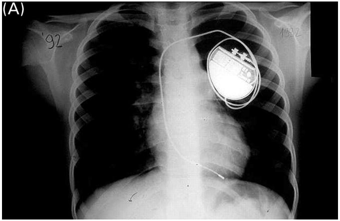

(A)
'92

And then ...



Cochlear™

And then . . .



Cochlear™

Led to information modelling and system integration work joint with CNG (Kit) Dampney, and later Bob Rosebrugh. . .

Schemas and states

- A database schema is an instance of a *data model*
  e.g. relational, ERA, Sketch
- Schemas should have type *and* constraint information;
  - in relational data model: table headings; (foreign) keys
  - in sketch data model: an Entity-Attribute sketch
- Database state for a schema is a snapshot of the currently stored information
  - usually called "the database"
  - Relational model: a set of relations (aka tables)
  - Sketch data model: a model of the sketch

Did I say I would "spiral"?

If you lost it on Monday it's ok — I'm going to start again!

What's a category?

Category/schema/sketch/spec. . .

A conference committee database might capture

- members, authors, articles submitted, their status. . .
- rules such as
  - papers may have several authors and vice versa
  - accepted paper will have author(s) presenting
  - person is author or committee member (not both)

$$
\begin{tikzcd}
 & & & \text{presenter} \arrow[dl] \arrow[dr] & & \\
\text{person} & \text{author} \arrow[l, tail, "\text{is\_a}"'] & \text{authorship} \arrow[l] \arrow[dr] & & \text{acceptedPaper} \arrow[r] \arrow[dl] & 1 \arrow[dl, "\text{true}"] \\
 & \text{commMbr} \arrow[ul, tail, "\text{is\_a}"] & & \text{paper} \arrow[ll, "\text{Reader}"'] \arrow[r, "\text{accept?}"] & \text{bool} &
\end{tikzcd}
$$

ERA Models

Entity-Relationship-Attribute models and related techniques are widely used and relatively simple

- ERA models capture much of the structure inherent in information
- ERA modelling can be an enormous, expensive, and challenging business
- ERA models don't attempt to capture many constraints, logical relations, etc
- ERA models are fundamentally directed graphs

Category Theory

A category is

- A directed graph (multi-, with cycles)
- A binary composition $(\cdot \longrightarrow \cdot \longrightarrow \cdot)$
  - which is associative
  - and has identities

Category Theory

A category is

- A directed graph (multi-, with cycles)
- A binary composition $(\cdot \longrightarrow \cdot \longrightarrow \cdot)$
  - which is associative
  - and has identities

Equivalently

- A directed graph (multi-, with cycles)
- An appropriate set of commutative diagrams (**D**)

Category Theory

A category is

- A directed graph (multi-, with cycles)
- A binary composition $(\cdot \longrightarrow \cdot \longrightarrow \cdot)$
  - which is associative
  - and has identities

Equivalently

- A directed graph (multi-, with cycles)
- An appropriate set of commutative diagrams (**D**)

The semantic power of categories comes largely from

- Universal properties
  - Products
  - Pullbacks
  - Coproducts, etc.

Formally: Entity-Attribute sketches

The foundation for a categorical data model

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

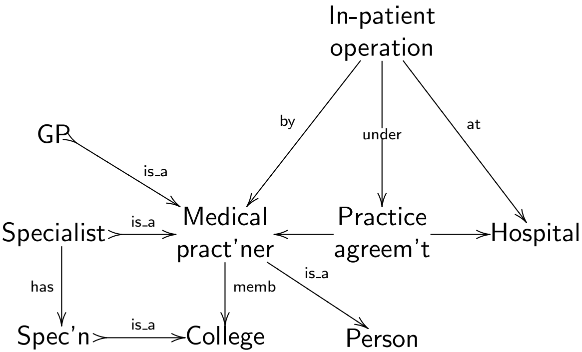
- a specified empty-base cone in $\mathcal{L}$ (vertex is called 1); domain 1 arrows called elements
- attributes: vertices of (discrete) cocones of elements; non-attributes called entities
- the graph G is finite

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

What does this buy for you?

- Constraints (commutative diagrams) are an intrinsic part of the model
- Constraints can involve "derived" entities — limits or coproducts
- Clarity of the entity-attribute distinction
- Elimination of the entity-relation distinction
- Semantics are enhanced
- Triggers are reduced, and it's clear when they're required

Another example: Health Administration



In relational terms: Arrows for foreign keys.
Monos (including primary keys) are pullback constraints
(these, and attributes, are not shown)

Universal algebra (second version)

Models (Algebras/snapshots/...) of a sketch

- A database state D for an EA sketch $\mathbb{E}$ is a model of $\mathbb{E}$ in finite sets, $\mathbf{set}_f$.
- The category of database states of $\mathbb{E}$ is $\mathrm{mod}(\mathbb{E}, \mathbf{set}_f)$, often abbreviated $\mathrm{mod}(\mathbb{E})$;
  — the morphisms are "natural transformations".

Thus, a database state is an assignment of a finite set for each object of the sketch and a function between the corresponding finite sets for each arrow of the sketch
in such a way as to satisfy the specified universal properties and commuting diagrams of the sketch.
The technology here is standard categorical universal algebra.

Equivalently

Write $Q(\mathbb{E})$ for the free lextensive category determined by the EA sketch $\mathbb{E}$.

Note that $Q(\mathbb{E})$ is invariant, and contains the structural queries (hence the choice of the letter Q).

Then, a database state is a finite-limit, finite-coproduct preserving functor

$$Q(\mathbb{E}) \longrightarrow \mathbf{S}$$

and the category of database states, $\mathrm{mod}(\mathbb{E}, \mathbf{S})$, is a full subcategory of the functor category

$$[Q(\mathbb{E}), \mathbf{S}] = \mathbf{S}^{Q(\mathbb{E})}$$

(The morphisms now really *are* natural transformations.)

Views

A view of a database specified by an EA sketch $\mathbb{E}$ is an EA sketch $\mathbb{V}$ and a sketch morphism

$$V : \mathbb{V} \rightarrow Q(\mathbb{E})$$

into the underlying sketch of the invariant representation of the database specified by $\mathbb{E}$.

Note

- A view is itself DB specification ($\mathbb{V}$)
- But view data will be populated by data derived from $\mathbb{E}$ as specified by V.

The view is calculated by $G = V^* : Mod(\mathbb{E}) \longrightarrow Mod(\mathbb{V})$ (perform composition with V).

State spaces

EA sketch models are the states (snapshots) of a database.
So the category of models is, in a sense, the state-space of the database.
If an EA-sketch is keyed then the category of models of the sketch is a partial order.
The components of the natural transformations are all monic.
The order corresponds to the information order.
In relational terms the state space consists of all the usual snapshots of the database with the transitions being given by tuple inclusion.

View updating

$$
\begin{tikzcd}
Mod(\mathbb{E}) \arrow[d, "V^*"'] \\
Mod(\mathbb{V})
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
s \arrow[d, dash, dashed] & \color{red}{s'} \\
v \arrow[r] & v'
\end{tikzcd}
$$

The view update problem:

- There may be no s'
- There may be many s'

A view update solution

If such exists, a pre-opcartesian arrow provides a canonical solution

If preopcartesians exist, and compose, then in this, the database case, V^* is an opfibration.

Explained on the following slides

A view and its view updating provides a classical example of a bidirectional transformation.

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

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

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

Cf "c-lens"

(One of many kinds of bidirectional transformation)

The kind of update just illustrated is called a

universal view update

and note the sense in which such updates are "least change".

c-lens coming soon ...

So: ERA and Category Theory

- ERA modelling is almost category theory
- The slight shift has benefits (not all reported above)
  - Constraint specification and maintenance
  - View updating technology
  - Clarity of treatment of missing information
  - Connections to bidirectional transformation work

I've tried to suppress many technical details above. But some of the real benefits come from the use of an established calculus (whether CT or BX).

Interoperations

Suppose that we have two category theoretically specified systems, with $\mathbb{E}$ and $\mathbb{E}'$ their EA-sketches.
As mentinoned on Monday, a first step in building system interoperations between them is to find a common view

$$W : Q(\mathbb{E}) \longleftarrow \mathbb{W} \longrightarrow Q(\mathbb{E}') : W'$$

Then the "Gets" will be

$$W^* : Mod(\mathbb{E}) \longrightarrow Mod(\mathbb{W}) \longleftarrow Mod(\mathbb{E}') : W'^*$$

But NOTE: In practice these don't need to be the Gets of lenses. We do need universal view updates, but only for the expected and contractually agreed transitions in each of $Mod(\mathbb{E})$ and $Mod(\mathbb{E}')$.

Time for a short pause?

Plenty more coming in three minutes

Revisiting Monday and Monadicity

Theorem

A lens put p is an algebra on g in $\mathbf{set}/V$ for the monad $\Delta\Sigma$ on $\mathbf{set}/V$ if and only if p satisfies PutGet, GetPut and PutPut.

Theorem

V non-empty implies Δ monadic, and hence every such lens (p, g) is isomorphic to a constant complement updating strategy.

$$\begin{tikzcd}
\mathbf{set} \arrow[rr, "K"] \arrow[dr, bend left, "\Delta"] & & \Delta\Sigma\text{-Alg} \arrow[dl, bend left] \\
& \mathbf{set}/V \arrow[ul, bend left, "\Sigma"] \arrow[ur, bend left] &
\end{tikzcd}$$

Note that $KC = \Delta\varepsilon$ where $\varepsilon = \pi_1 : \Sigma\Delta C \longrightarrow C$. So, if K is an equivalence then by essential survectivity every algebra (ie every vwb lens) is isomorphic to one of the form $V \times C \longrightarrow V$ with Put $\pi_{0,2} : V \times V \times C \longrightarrow V \times C$.