charin.fntolist, erase aplist
and how not to do research

Peter Buneman
LFCS 30th anniversary
Once upon a time ...

I spent a lot of time writing programs

But I left the stimulating but tumultuous environment at Edinburgh to work in the US ...
... to work on databases

When relational databases were a theoretical nicety, we had Codasyl:

A Codasyl “set”

“members”

“owner”
An embarrassingly long time ago, when LaTeX had not been invented.

Since FOL is a functional language, we need to adopt a functional view of databases. We regard a database as a collection of functions over various data-types. Figure 1 shows the schema of a (very simple) database containing entities of type EMPLOYEE and of type DEPARTMENT.

* means "stream of"

Schema

Database instance is a set of functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT</td>
<td>EMPLOYEE → DEPARTMENT</td>
</tr>
<tr>
<td>ENAME</td>
<td>EMPLOYEE → CHAR</td>
</tr>
<tr>
<td>SAL</td>
<td>EMPLOYEE → NUM</td>
</tr>
<tr>
<td>MARRIED</td>
<td>EMPLOYEE → BOOL</td>
</tr>
<tr>
<td>DNAME</td>
<td>DEPARTMENT → CHAR</td>
</tr>
</tbody>
</table>

*EMPLOYEE : → *EMPLOYEE
*DEPARTMENT : → *DEPARTMENT
I had developed a taste of lazy and combinatory programming in POP-2. And Backus’ FP had appeared.

charin.fntolist, erase.applist


When Moggi had not spoken the Word, nor had Wadler preached it.

1. Composition. If $f$ and $g$ are such that $f: \alpha \rightarrow \beta$ and $g: \beta \rightarrow \gamma$ then $f \cdot g: \alpha \rightarrow \gamma$.

2. Extension. If $f: \alpha \rightarrow \beta$ then $\cdot f$ operates upon a stream of these types; i.e., $\cdot f: \alpha \rightarrow \ast\beta$.

3. Restriction. If $p$ is a predicate over (i.e., $p: \alpha \rightarrow \text{bool}$) then $[p]: \alpha \rightarrow \ast\beta$.

4. Construction. If $f_1: \alpha \rightarrow \beta_1$, $f_2: \alpha \rightarrow \beta_2$, ..., $f_n: \alpha \rightarrow \beta_n$ then $[f_1, f_2, \ldots, f_n]: \alpha \rightarrow \{\beta_1, \beta_2, \ldots, \beta_n\}$.

CONC maps a pair of streams $[(\alpha, \beta)]$ (whose elements are of the same type) into a single stream $\alpha$; DCONC produces a single stream $\alpha$ by "flattening" an arbitrary stream of streams $\ast\alpha$. The operator $\text{DISTRIB}$ takes a tuple of the form $[(\alpha, \beta)]$ and returns a stream of tuples $\alpha \rightarrow \{\beta, \beta, \ldots, \beta\}$ with the value of type $\beta$ "distributed" over the stream of $\alpha$'s.

FQL got used by people building interfaces to Codasyl DBs. Remember that most database queries are written by programs – not people.

[B., R Frankel, Sigmod 1979. The Functional Data Model D Shipman, Sigmod 1979]
Q1: \(-\rightarrow^*[\text{CHAR}, \text{NUM}] \rightleftharpoons \text{!EMPLOYEE.}\text{MARRIED.}*[\text{DEPT.DNAME}, \text{SAL}]\);
Several years went by.....

Influences from PL theory and LFCS

- Impedance mismatch problem
- Domain theory and partial information in databases
- ML and record polymorphism
- Structural recursion, monads and nested relational algebra (FQL revisited)
- Partially static type systems for semi-structured data
“It’s no secret that ReQL, the RethinkDB query language, is modeled after functional languages like Lisp and Haskell. The functional paradigm is particularly well suited to the needs of a distributed database while being more easily embeddable as a DSL than SQL’s ad hoc syntax. Key to functional programming’s power and simplicity is the anonymous (aka lambda) function.”

```
r.table('users').filter(r.row("age").eq(30)).map(r."name").run();
```
Then I came back to Informatics and joined LFCS

Random thoughts on US vs UK research environment

- US more directed and less forgiving
  - Nothing like the intellectual ferment of the pre-LFCS years
- UK much more supportive of “interdisciplinary research”, but….
  - interdisciplinary research can (like writing programs) be a huge time-waster.
  - You spend most of your time doing boring/marginal stuff, but just occasionally something interesting turns up…
- And sometimes something whacky turns up, like semistructured data, provenance and (about 8 years ago) Data Citation
Now data citation is big business

Large number of organizations: Datacite DataONE, GEOSS, D-Lib Alliance, DCC, COPDES, Force-11, AGU, ESIP, DCMI, CODATA, ICSTI, IASSIST, ICSU

**Force 11**: “Data citations should be accorded the same importance in the scholarly record as citations of other research objects, such as publications.”

**DataCite**: “We believe that you should cite data in just the same way that you can cite other sources of information, such as articles and books.”

**Amsterdam Manifesto**: “Data should be considered citable products of research.”

**Oxford University** (on behalf of EPSRC) “Describe your data ... to enable other researchers to ... cite them”
What is a (conventional) citation?

A collection of “snippets” of information: authors, title, date, etc. and some kind of access mechanism (DOI, URL, ISBN, shelf number etc.)

Not exactly provenance

Self contained, immutable (to within some choice of format)

Needed for a variety of reasons: kudos, currency, authority, recognition, access…

Especially important in curated databases – some kind of mixture of crowd- or expert-sourced data and conventional publication. (IUPHAR – hundreds of contributors, and they want to be acknowledged.)
So what’s the problem

Citations vary with what part of the database is being cited. And the database changes over time.

There is a huge number of “parts” of a database

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>URI/CGI</td>
</tr>
<tr>
<td>RDB</td>
<td>SQL</td>
</tr>
<tr>
<td>XML</td>
<td>XPath/XQuery</td>
</tr>
<tr>
<td>RDF</td>
<td>SPARQL</td>
</tr>
<tr>
<td>File system</td>
<td>set of paths</td>
</tr>
</tbody>
</table>

We cannot expect to put a citation for each “part” into DBLP. We are going to have to generate citations on the fly.
It gets worse

Start of Datacite 400 line XML schema specification for data citation

SELECT /*+ NOPARALLEL bypass_recursive_check */
SP_ALIAS_190,
((CASE SP_ALIAS_191 WHEN 1 THEN 'PROVIDER::ALL_PROV::'
WHEN 0 THEN 'PROVIDER::PROV::'
ELSE NULL END) || SP_ALIAS_190) ALIAS_3553,
SP_ALIAS_194,
SP_ALIAS_191,
SP_ALIAS_192,
SP_ALIAS_193,
SP_ALIAS_205,
D4_AGE_GROUP_ET,
((CASE D4_AGE_GROUP_GID WHEN 1 THEN 'AGE_GROUP::ALL_AGE_GRP::'
WHEN 0 THEN 'AGE_GROUP::PROV::'
ELSE NULL END) || D4_AGE_GROUP_ET) ALIAS_3554,
FROM SP_ALIAS_190
WHERE 1
AND 'OD проекта' = 'All Project'
Another principle/recommendation

Unless we couple the process of generating a citation with the act of extracting the data, the advocacy of data citation is pointless.

The main problem

**Given a database D and a query Q, generate an appropriate citation.**

NB. The citation depends on *both* Q and D
The database problem

Looks hard because any analysis of a query is likely to be hard, if not undecidable, but there's hope.

Key concept is that of a database view – a function that when applied to a database in one schema produces a database in another schema (and model)

It is common for authors/publishers to supply citations for some parts of the database. These can be expressed as views \( V_1 \ldots V_n \).

So given a query \( Q \), a database \( D \) and a schema \( S \), can \( Q \) be factored through a view. That is, is there a \( Q_i \) such that

\[
\forall D \in S. \ Q(S) = Q_i(V_i(D))
\]

If so, the citation for \( V_i \) is the citation for \( Q \).

This is a well-known database problem that comes from optimization. In fact our problem is a bit more subtle because the citation also depends on \( D \), and we have to introduce the notion of a *parameterized* view. But the known machinery can be adapted.
Hierarchical data (files, XPath, some URLs)

A simple pattern-matching language for generating citations in a hierarchy

←
/Root[VersionNumber: $v]/Family[FamilyName: $$f]
/Introduction[Contributor-list: $a]

<table>
<thead>
<tr>
<th>Type</th>
<th>Nonsense mutation</th>
<th>Human</th>
<th>Rare variant identified in attention-deficit hyperactivity disorder (ADHD) patient, premature STOP codon with impaired cell surface expression and cAMP inhibition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amino acid change</td>
<td></td>
<td>Y170X</td>
<td></td>
</tr>
<tr>
<td>Nucleotide accession</td>
<td></td>
<td>NM_005958</td>
<td></td>
</tr>
<tr>
<td>Protein accession</td>
<td></td>
<td>NP_005949</td>
<td></td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Missense mutation</th>
<th>Human</th>
<th>Common variant identified in control population with reduced ERK1/2 activation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amino acid change</td>
<td></td>
<td>A266V</td>
<td></td>
</tr>
<tr>
<td>Nucleotide accession</td>
<td></td>
<td>NM_005958</td>
<td></td>
</tr>
<tr>
<td>Protein accession</td>
<td></td>
<td>NP_005949</td>
<td></td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

**General Comments**

The molecular pharmacology of ovine melatonin receptors has been shown to be different to human recombinant melatonin receptors [49].

**Available Assays**

DiscoverRx

- OPEN ECN PathHunter® eXpress MTNR1A CHO-K1 β-Arrestin GPCR Assay (Cat no. 93-0510E2CP0M)
- PathHunter® CHO-K1 MTNR1A β-Arrestin Cell Line (Cat no. 93-051IC2)

**References**

Show »

**How to cite this page**

Philippe Delagrainge, Marganta L. Dubocovich, James Olcese.


Also works for other kinds of data

The views are spatio-temporal bounding boxes

B., Davidson & Frew *Why Data Citation is a Computational Problem*. CACM. to appear
Thank You, LFCS!

Some of my favorite citations:

BL Cotton Nero A. X

Cotton Otho A. XII


Nature, 171,737-738

Peter Buneman
wget -qO - http://mirror.hmc.edu/ctan/FILES.bname | grep ".bst$" \ | sed 's/.*/\(.*\)/*/1/' | sort -u | wc -l
Executed on 18 November 2011