Query languages for matrices and $K$-relations

Jan Van den Bussche  
(Hasselt University, Belgium)

joint work with Robert Brijder, Floris Geerts (U.Antwerp), Marc Gyssens, Timmy Weerwag
Relational databases

Database instance: relational structure

- assign domains to attributes
- compatible attributes have same domain
- assign sets of tuples to relation names

Database schema: \( F(\text{name}, \text{friend}), B(\text{name}, \text{year}) \)
Relational algebra

Union $\cup$

Difference $-$

Selection $\sigma_P(A_1,\ldots,A_k)$

\hspace{1cm} e.g. $\sigma_{\text{year} \geq 2000}(B)$

Natural join $\bowtie$

Generalized projection $\pi_f(A_1,\ldots,A_k)$

\hspace{1cm} e.g. $\pi_{\text{year} - 2000}(B)$

Renaming $\rho_{A/B}$
Matrix databases

Data science

\[
A = \begin{pmatrix}
5 & 2 & 0 \\
2 & 1 & 3 \\
\end{pmatrix} \quad B = \begin{pmatrix}
300 \\
250 \\
330 \\
\end{pmatrix}
\]

Matrix schema uses size symbols: \( A(\alpha \times \beta), \ B(\beta \times 1) \)
\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
= \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}
\]
MATLAB

1

diag

diag\left(\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}
MATLANG

1

diag

(conjugate) transpose

matrix multiplication

pointwise functions $f(M_1, \ldots, M_k)$

e.g. $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \circ \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 6 & 4 \end{pmatrix}$ pointwise multiplication
Some simple MATLAB tricks

Let $A$ be the adjacency matrix of a graph on $n$ nodes.

Number of nodes:

$$N = 1(A)^{\ast} \cdot 1(A)$$

Degree vector, duplicated $n$ times:

$$D = A \cdot 1(A) \cdot 1(A)^{\ast}$$

Google matrix:

$$G_{ij} = d \frac{A}{D} + \frac{1 - d}{N}$$
Our proposal

\[
\frac{\text{relational algebra}}{\text{relational databases}} = \frac{\text{MATLANG}}{\text{matrix databases}}
\]

• What is the precise expressive power?

• How does it compare to relational database querying?
Matrix database as relational database

\[ A = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} \quad B = \begin{pmatrix} 300 \\ 250 \\ 330 \end{pmatrix} \]

\[ A = \begin{array}{ccc} \text{row} & \text{col} & K \\ 1 & 1 & 7 \\ 1 & 2 & 8 \\ 1 & 3 & 9 \\ 2 & 1 & 10 \\ 2 & 2 & 11 \\ 2 & 3 & 12 \end{array} \quad B = \begin{array}{cc} \text{row} & K \\ 1 & 300 \\ 2 & 250 \\ 3 & 330 \end{array} \]

\text{*-relations: generalization of the relational database model}

Every tuple is annotated with a value from some fixed semiring \( K \)
(Positive) relational algebra on $K$-relations

Influential paper from 2007 [Green, Garvounarakis, Tannen]

Union: adds annotations

Natural join: multiplies annotations

Selection $\sigma_{A=B}$: sets annotations to 0 for non-qualifying tuples

Projection $\pi_{A_1,\ldots,A_k}$: sums annotations

Renaming

$\mathbf{1}$: sets annotations to 1
Theorem

ARA(3): Annotation-Relation Algebra, width $\leq 3$

Assume $K$ is commutative

Matrix query, expressible in ARA(3) if and only expressible in MATLANG with only $+$ and $\circ$ as pointwise functions

Analogue to classical result by Tarski and Givant:

<table>
<thead>
<tr>
<th>Our result</th>
<th>Tarski and Givant</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix queries</td>
<td>binary-relation queries</td>
</tr>
<tr>
<td>ARA(3)</td>
<td>FO(3)</td>
</tr>
<tr>
<td>MATLANG</td>
<td>classical algebra of binary relations</td>
</tr>
</tbody>
</table>
Expressiveness limitations of MATLANG

Not expressible in MATLANG:

- transitive closure of a graph
- testing for 4-clique
Adding matrix inverse to MATLANG

Akin to solving a system of linear equations

Expressible in MATLANG + inverse:

- PageRank vector of a graph:
  \[ \frac{1 - d}{n} (I - d \frac{A}{D})^{-1} \cdot 1 \]
  (by definition)

- transitive closure: let \( B = A / (n + 1) \)
  \[ \sum_{k=0}^{\infty} B^k = (I - B)^{-1} \]

- number of connected components, testing bipartiteness
Eigenvectors

Eigen-decomposition, another workhorse in data analysis

Diagonizable $A = B \cdot \Lambda \cdot B^{-1}$ where $B$ is a basis of eigenvectors of $A$

$\Lambda$ has the eigenvalues on the diagonal

**Define:** $\text{eigen}(A) := B$, nondeterministic!

**Theorem:** Inverse is expressible in MATLANG + eigen

**Open problem:** Show a graph query that is:

- deterministically expressible in MATLANG + eigen
- not in MATLANG + inverse
References

Brijder, Geerts, Van den Bussche, Weerwag *On the expressive power of query languages for matrices*, ICDT 2018

- Full version in TODS
- Research highlight, SIGMOD Record


Floris Geerts *On the expressive power of linear algebra on graphs*, ICDT 2019

Related work: LaraDB, SQL, in-database machine learning, etc.