Updated 3/1/2022

Topic 2: Complexity of Query Evaluation
Unit 1: Conjunctive Queries (continued)
Lecture 13

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp22)

https://northeastern-datalab.github.io/cs7240/sp22/

3/1/2022

Pre-class conversations

- Recapitulation of Query containment & homomorphisms
- Feedback on projects & new scribes: after tomorrow

- Today:
  - CQ minimization, beyond CQs

Topic 2: Complexity of Query Evaluation & Reverse Data Management

- **CONTINUED Lecture 11 (Tue 2/22):** 1 Conjunctive Queries
- **Lecture 12 (Fri 2/25):** Conjunctive Queries
- **Lecture 13 (Tue 3/1):** Beyond Conjunctive Queries
- **Lecture 14 (Fri 3/4):** Provenance
- **Lecture 15 (Tue 3/8):** Provenance, Reverse Data Management

  Pointers to relevant concepts & supplementary material:

  - **Unit 1. Conjunctive Queries**: Query evaluation of conjunctive queries (CQs), data vs. query complexity, homomorphisms, constraint satisfaction, query containment, query minimization, absorption: [Kolaitis, Vardi'00], [Vardi'00], [Kolaitis'16], [Koutris'19] L1 & L2
  - **Unit 2. Beyond Conjunctive Queries**: unions of conjunctive queries, bag semantics, nested queries, tree pattern queries: [Kolaitis'16], [Tan+'14], [G.'11], [Martens'17]
  - **Unit 3. Provenance**: [Buneman+02], [Green+07], [Cheney+09], [Green,Tannen'17], [Kepner+16], [Buneman, Tan'18]
  - **Unit 4. Reverse Data Management**: update propagation, resilience: [Buneman+02], [Kimelfeld+12], [Freire+15]

See: https://db.khoury.northeastern.edu/activities/

Matrix Query Languages

Floris Geerts
University of Antwerp
floris.geerts@uantwerp.be

Thomas Muñoz
PUC Chile and IMFD Chile
tfmunoz@uc.cl

Cristian Riveros
PUC Chile and IMFD Chile
cristian.riveros@uc.cl

Jan Van den Bussche
Hasselt University
jan.vandenbussche@uhasselt.be

Domagoj Vrgoč
PUC Chile and IMFD Chile
dvrgoc@ing.puc.cl

ABSTRACT

Linear algebra algorithms often require some sort of iteration or recursion as is illustrated by standard algorithms for Gaussian elimination, matrix inversion, and transitive closure. A key characteristic shared by these algorithms is that they allow looping for a number of steps that is bounded by the matrix dimension. In this paper we extend the matrix query language MATLANG with this type of recursion, and show that this suffices to express classical linear algebra algorithms. We study the expressive power of this language and show that it naturally corresponds to arithmetic circuit families, which are often said to capture linear algebra. Furthermore, we analyze several sub-fragments of our language, and show that their expressive power is closely tied to logical formalisms on semiring-annotated relations.

Expressive Power of Linear Algebra Query Languages

Floris Geerts
University of Antwerp
floris.geerts@uantwerp.be

Thomas Muñoz
PUC Chile and IMFD Chile
tfmunoz@uc.cl

Cristian Riveros
PUC Chile and IMFD Chile
cristian.riveros@uc.cl

Domagoj Vrgoč
PUC Chile and IMFD Chile
dvrgoc@ing.puc.cl

ABSTRACT

Due to the importance of linear algebra and matrix operations in data analytics, there has been a renewed interest in developing query languages that combine both standard relational operations and linear algebra operations. We survey aspects of the matrix query language MATLANG and extensions thereof, and connect matrix query languages to classical query languages and arithmetic circuits.

Outline: T2-1/2: Query Evaluation & Query Equivalence

- T2-1: Conjunctive Queries (CQs)
  - CQ equivalence and containment
  - Graph homomorphisms
  - Homomorphism beyond graphs
  - CQ containment
  - CQ minimization
- T2-2: Equivalence Beyond CQs
  - Union of CQs, and inequalities
  - Union of CQs equivalence under bag semantics
  - Tree pattern queries
  - Nested queries

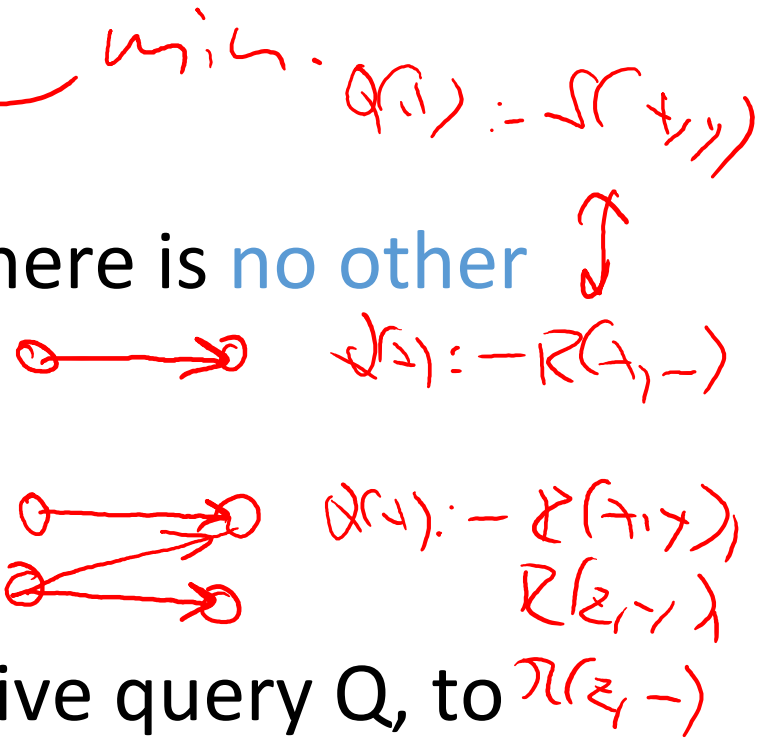
Minimizing Conjunctive Queries

- Goal: minimize the number of joins in a query
- Definition: A conjunctive query Q is minimal if...

?

Minimizing Conjunctive Queries

- Goal: minimize the number of joins in a query
- Definition: A conjunctive query Q is minimal if there is no other conjunctive query Q' such that:
  1. Q ≡ Q'
  2. Q' has fewer atoms than Q
- The task of CQ minimization is, given a conjunctive query Q, to compute a minimal one that is equivalent to Q



Minimizing Conjunctive Queries by Deletion

THEOREM: Given a conjunctive query $Q_1(\mathbf{x})$:- $body_1$

that is logically equivalent to a conjunctive query $Q_2(\mathbf{x})$:- $body_2$

where $|body_2| < |body_1|$, then Q_1 is equivalent to a query

$Q_3(\mathbf{x})$:- $body_3$ such that $body_3 \subseteq body_1$

Intuitively, the above theorem states that to minimize a conjunctive query,

we simply need to remove some atoms from its body

Conjunctive query minimization algorithm

Notice: the order in which we inspect subgoals doesn't matter

Minimize(Q(**x**) :- body)

Repeat {
- Choose an atom α ∈ body; let Q' be the new query after removing α from Q
- If there is a homomorphism from Q to Q', then body := body \ {α}

until no atom can be removed}

1. We trivially know Q⟵Q' (Thus: Q⊆Q')

Q :-E(x,y), E(y,z)
Q':-E(x,y)

2. This forward direction is non-trivial: Q⟶ Q'

Minimization Procedure: Example

a,b,c,d are constants

Q(x) :- R(x,y), R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

Is this query minimal ?

Based upon an example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Minimization Procedure: Example

a,b,c,d are constants

Q(x) :- R(x,y), R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{y⟶'b'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

Is this query minimal ?

Based upon an example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Minimization Procedure: Example

a,b,c,d are constants

Q(x) :- R(x,y), R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{y⟶'b'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{v⟶'c'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), S('a','c','d')

Is this query minimal ?

Based upon an example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Minimization Procedure: Example

a,b,c,d are constants

Q(x) :- R(x,y), R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{y⟶'b'}

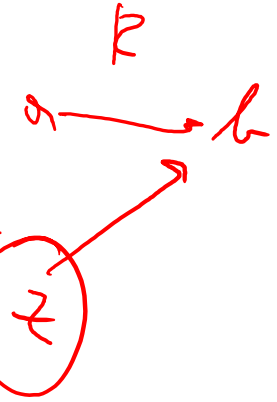
Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{v⟶'c'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), S('a','c','d')

{x⟶'a'}

Q('a') :- R('a','b'), R(u,'c'), S('a','c','d')



Is this query minimal ?

Based upon an example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Minimization Procedure: Example

a,b,c,d are constants

Q(x) :- R(x,y), R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{y⟶'b'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), R(u,v), S('a','c','d')

{v⟶'c'}

Q(x) :- R(x,'b'), R('a','b'), R(u,'c'), S('a','c','d')

Minimal query

{x⟶'a'}

Q('a') :- R('a','b'), R(u,'c'), S('a','c','d')

Actually, we went too far: Mapping x⟶'a' is not valid since x is a head variable!

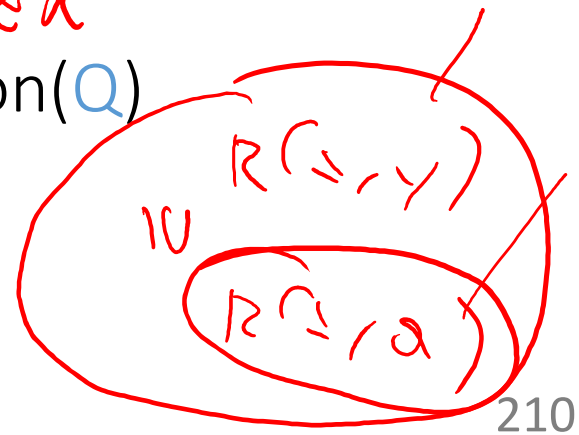
Based upon an example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Uniqueness of Minimal Queries

Natural question: does the order in which we remove atoms from the body of the conjunctive query during minimzation matter?

THEOREM: Consider a conjunctive query Q. Let Q_1 and Q_2 be minimal conjunctive queries such that $Q_1 \equiv Q$ and $Q_2 \equiv Q$. Then, Q_1 and Q_2 are isomorphic (i.e., they are the same up to variable renaming)

Therefore, given a conjunctive query Q, the result of Minimization(Q) is unique (up to variable renaming) and is called the core of Q



Query Minimization for Views

Employee(name, university, manager)

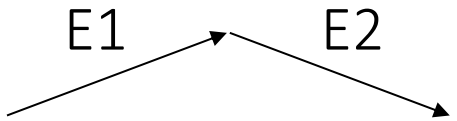
611

name	university	manager
Alice	Northeastern	Bob
Bob	Northeastern	Cecile
Cecile	Northeastern	
...	...	...

NEU employees managed by NEU emp.:

```
CREATE VIEW NeuMentors AS
SELECT DISTINCT E1.name,E1.manager
FROM Employee E1, Employee E2
WHERE E1.manager = E2.name
AND E1.university = 'Northeastern'
AND E2.university= 'Northeastern'
```

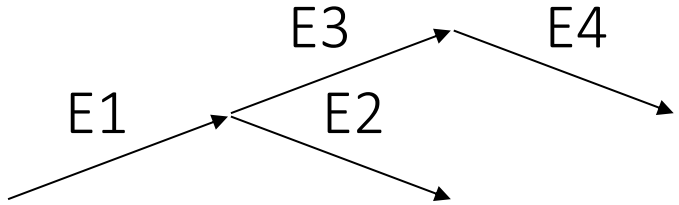
←This query / view is minimal



NEU emp. managed by NEU emp. managed by NEU emp.:

```
SELECT DISTINCT N1.name
FROM NeuMentors N1, NeuMentors N2
WHERE N1.manager = N2.name
```

←This query is minimal



View expansion (when you run a SQL query on a view)

```
SELECT DISTINCT E1.name
FROM Employee E1, Employee E2, Employee E3, Employee E4
WHERE E1.manager = E2.name AND E1.manager = E3.name AND E3.manager = E4.name
AND E1.university = 'Northeastern' AND E2.university = 'Northeastern'
AND E3.university = 'Northeastern' AND E4.university = 'Northeastern'
```

Is this query still minimal? ?

Example adopted from Dan Suciu

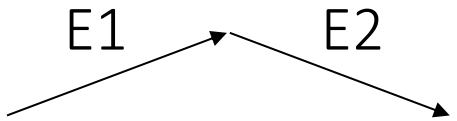
Query Minimization for Views

Employee(name, university, manager)

NEU employees managed by NEU emp.:

```
CREATE VIEW NeuMentors AS
SELECT DISTINCT E1.name,E1.manager
FROM Employee E1, Employee E2
WHERE E1.manager = E2.name
AND E1.university = 'Northeastern'
AND E2.university= 'Northeastern'
```

←This query / view is minimal

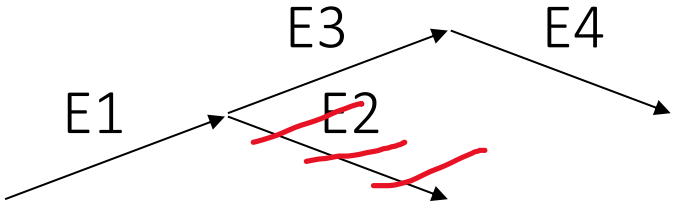


name	university	manager
Alice	Northeastern	Bob
Bob	Northeastern	Cecile
Cecile	Northeastern	
...	...	...

NEU emp. managed by NEU emp. managed by NEU emp.:

```
SELECT DISTINCT N1.name
FROM NeuMentors N1, NeuMentors N2
WHERE N1.manager = N2.name
```

←This query is minimal



View expansion (when you run a SQL query on a view)

```
SELECT DISTINCT E1.name
FROM Employee E1, Employee E2, Employee E3, Employee E4
WHERE E1.manager = E2.name AND E1.manager = E3.name AND E3.manager = E4.name
AND E1.university = 'Northeastern' AND E2.university = 'Northeastern'
AND E3.university = 'Northeastern' AND E4.university = 'Northeastern'
```

E2 is redundant!