Updated 3/1/2023

Topic 2: Complexity of Query Evaluation
Unit 1: Conjunctive Queries
Lecture 15

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp23)

https://northeastern-datalab.github.io/cs7240/sp23/

2/28/2023

Pre-class conversations

- Last class summary
- Project ideas

- Today:
  - Homomorphisms and the connections to:
    - Query containment
    - Query minimization
    - Query evaluation
  - Beyond CQs
- Next time
  - Neha on the connection to CSPs (constraint satisfaction problems)

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

Exercise: Find Homomorphisms

q_1: {E(x,y),E(y,z),E(z,w)}

Order of subgoals in the query does not matter (thus written here as sets)

q_2: {E(x,y),E(y,z),E(z,x)}

q_3: {E(x,y),E(y,x)}

What is the containment relation between these queries ?

q_4: {E(x,y),E(y,x),E(y,y)}

q_5: {E(x,x)}

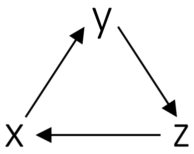
Exercise: Find the Homomorphisms

q_1: {E(x,y),E(y,z),E(z,w)}

$$x \longrightarrow y \longrightarrow z \longrightarrow w$$

Order of subgoals in the query does not matter (thus written here as sets)

q_2: {E(x,y),E(y,z),E(z,x)}

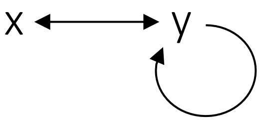


q_3: {E(x,y),E(y,x)}

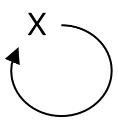
$$x \longleftrightarrow y$$

What is the containment relation between these queries ?

q_4: {E(x,y),E(y,x),E(y,y)}



q_5: {E(x,x)}



Exercise: Find the Homomorphisms

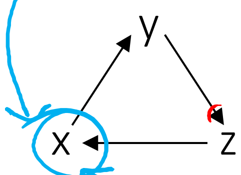
Q

q_1: {E(x,y),E(y,z),E(z,w)}

$x \to y \to z \to w$

q_2: {E(x,y),E(y,z),E(z,x)}

D

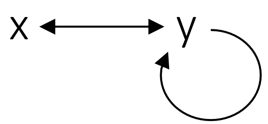


q_3: {E(x,y),E(y,x)}

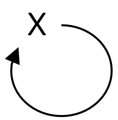
$x \leftrightarrow y$

What is the containment relation between these queries ?

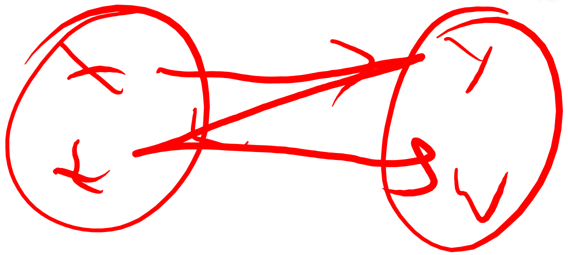
q_4: {E(x,y),E(y,x),E(y,y)}



q_5: {E(x,x)}



Exercise: Find the Homomorphisms



Q

q_1: {E(x,y),E(y,z),E(z,w)}

x ⟶ y ⟶ z ⟶ w

D

q_2: {E(x,y),E(y,z),E(z,x)}

x ⟶ y ⟶ z ⟶ x

q_3: {E(x,y),E(y,x)}

x ⟷ y

What is the containment relation between these queries ?

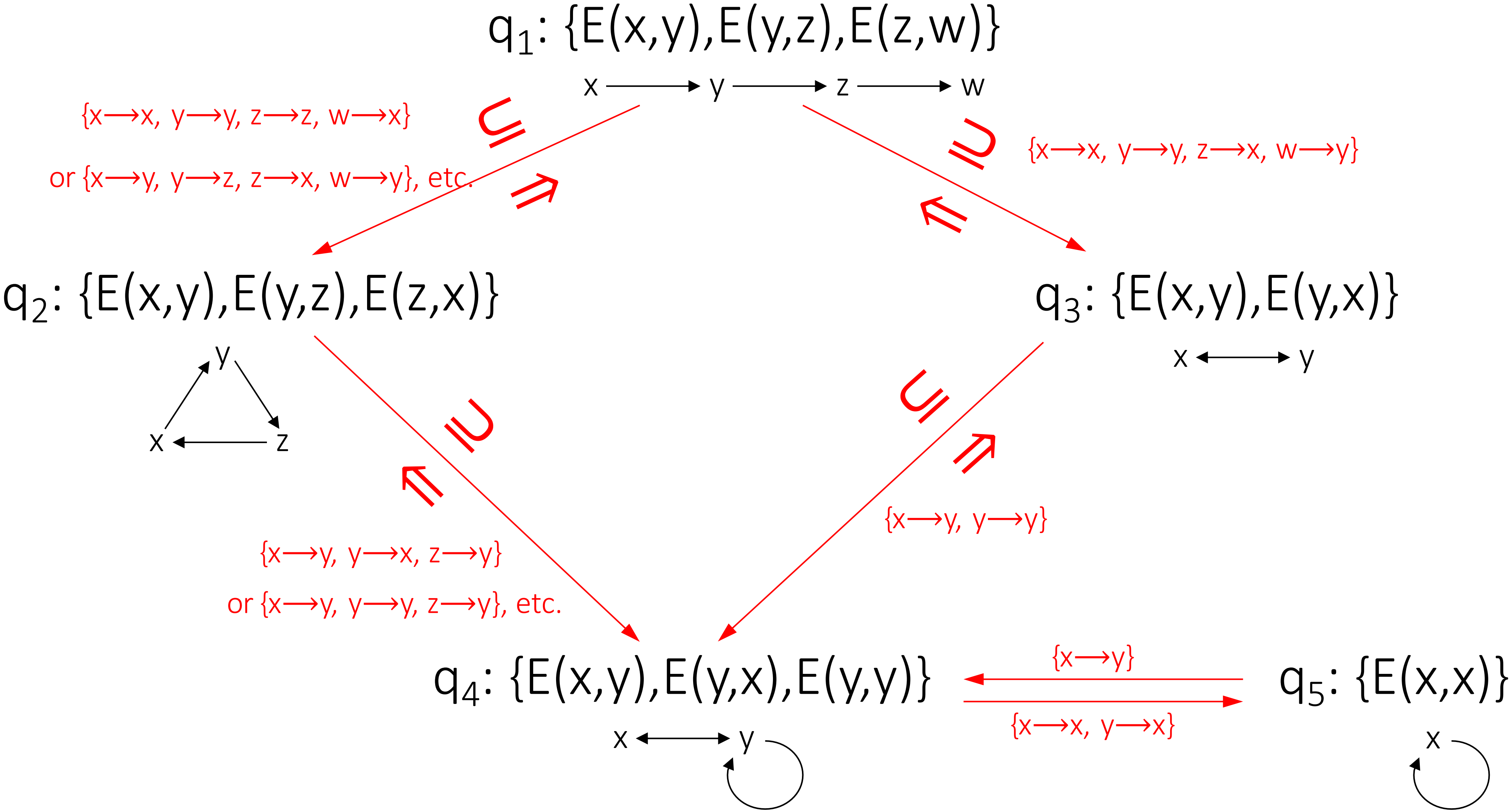
q_4: {E(x,y),E(y,x),E(y,y)}

x ⟷ y, y ⟶ y

q_5: {E(x,x)}

x ⟶ x

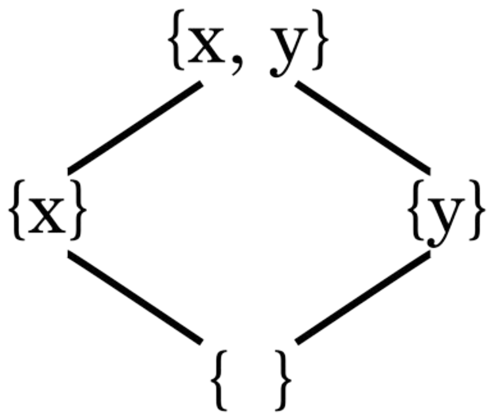
Exercise: Find the Homomorphisms



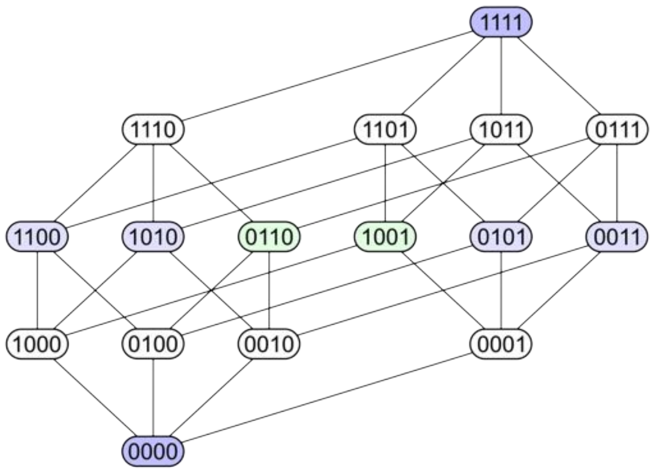
Example by Andreas Pieris: https://homepages.inf.ed.ac.uk/apieris/courses/atfd2020/index.html

Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

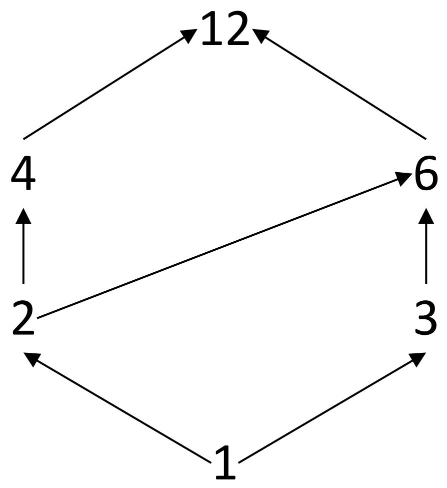
Side-topic: Hasse diagram



The power set of a 2-element set ordered by inclusion



Power set of a 4-element set ordered by inclusion ⊆



Positive integers divisors of 12 ordered by divisibility

Sources of two left figures: https://en.wikipedia.org/wiki/Hasse_diagram

Query Homomorphism Practice

$q_1(x,y)$:- $R(x,u),R(v,u),R(v,y)$ $\qquad$ $var(q_1) = \{x, u, v, y\}$

$q_2(x,y)$:- $R(x,u),R(v,u),R(v,w),R(t,w),R(t,y)$ $\qquad$ $var(q_2) = \{x, u, v, w, t, y\}$

Are these queries equivalent **?**

Query Homomorphism Practice


$q_1(x,y)$:- $R(x,u), R(v,u), R(v,y)$

$q_2(x,y)$:- $R(x,u), R(v,u), R(v,w), R(t,w), R(t,y)$

$\text{var}(q_1) = \{x, u, v, y\}$

$\text{var}(q_2) = \{x, u, v, w, t, y\}$

$q_1 \longrightarrow q_2$ Thus **?**

Which query contains the other?

Query Homomorphism Practice


$q_1(x,y)$:- $R(x,u), R(v,u), R(v,y)$

$q_2(x,y)$:- $R(x,u), R(v,u), R(v,w), R(t,w), R(t,y)$

$\mathrm{var}(q_1) = \{x, u, v, y\}$

$\mathrm{var}(q_2) = \{x, u, v, w, t, y\}$

$q_1 \longrightarrow q_2$ Thus $q_1 \subseteq q_2$!

Query Homomorphism Practice

$q_1(x,y)$:- $R(x,u),R(v,u),R(v,y)$ $\quad$ $var(q_1) = \{x, u, v, y\}$

$q_2(x,y)$:- $R(x,u),R(v,u),R(v,w),R(t,w),R(t,y)$ $\quad$ $var(q_2) = \{x, u, v, w, t, y\}$

Is there any homomorphism $q_2 \longrightarrow q_1$
and thus $q_2 \supseteq q_1$?

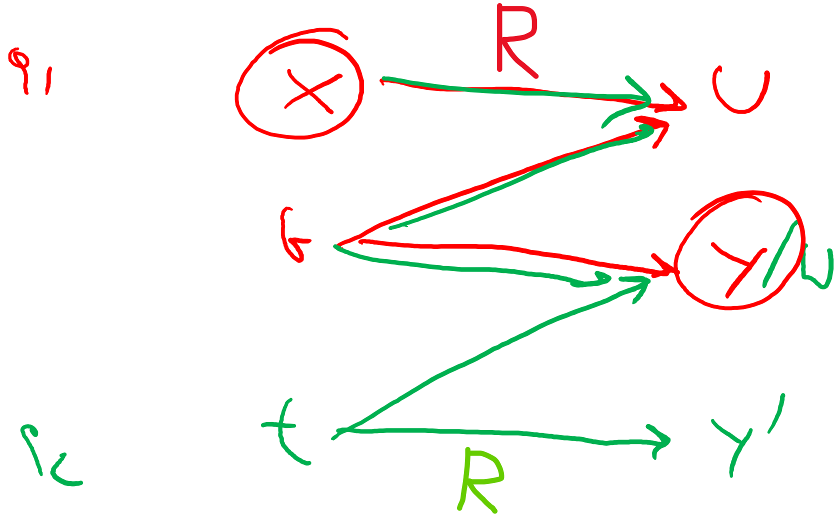
Query Homomorphism Practice

$q_1(x,y)$:- $R(x,u), R(v,u), R(v,y)$

$q_2(x,y)$:- $R(x,u), R(v,u), R(v,w), R(t,w), R(t,y)$

$var(q_1) = \{x, u, v, y\}$

$var(q_2) = \{x, u, v, w, t, y\}$



$q_2 \longrightarrow q_1$

and thus $q_2 \supseteq q_1$

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...

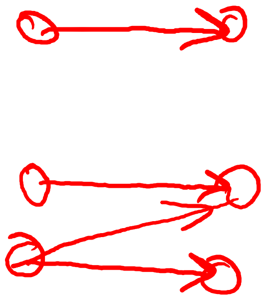
?

Minimizing Conjunctive Queries

- Goal: minimize the number of joins in a query

min. Q(x) :- S(x,y)

- 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



Q(x) :- R(x,-)

Q(x) :- R(x,y)
R(z,y)
R(z,-)

- The task of CQ minimization is, given a conjunctive query Q, to compute a minimal one that is equivalent to Q

Minimizing Conjunctive Queries (CQs) by Deletion

THEOREM: Given a CQ $Q_1(\mathbf{x})$:- $body_1$ that is logically equivalent to a CQ $Q_2(\mathbf{x})$:- $body_2$ where $|body_1| > |body_2|$.
Then Q_1 is equivalent to a CQ $Q_3(\mathbf{x})$:- $body_3$ s.t. $body_1 \supseteq body_3$

Intuitively, the above theorem states that to minimize a CQ, 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

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)

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 ?

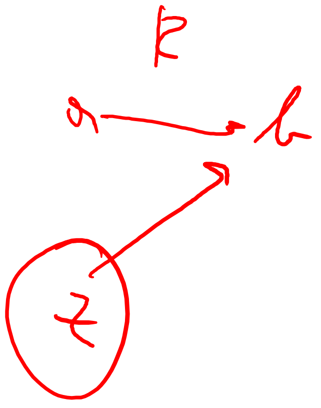
Minimization Procedure: Example

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 ?

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 ?

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!

Uniqueness of Minimal Queries

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

?

Uniqueness of Minimal Queries

Natural question: does the order in which we remove atoms from the body of the conjunctive query during minimization 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)

Church-Rosser

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

R(x,y)

R(x,a)

Query Minimization for Views

Employee(name, university, manager)

611

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

E1 → E2

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

Query Minimization for Views

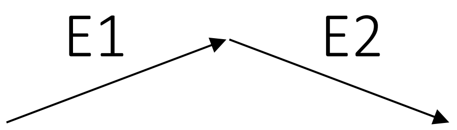
Employee(name, university, manager)

611

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

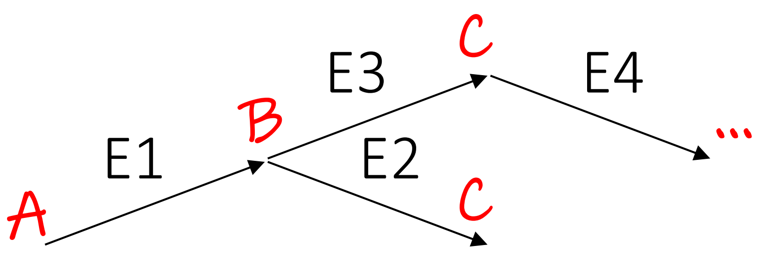


name	university	manager
Alice	Northeastern	Bob
Bob	Northeastern	Cecile
Cecile	Northeastern	
...	...	...

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’
```

Is this query still minimal? ?

Query Minimization for Views

Employee(name, university, manager)

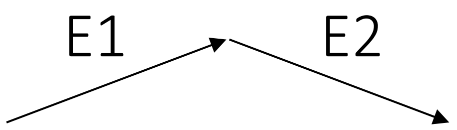
611

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

NEU employees managed by NEU emp.:

```sql
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'
```

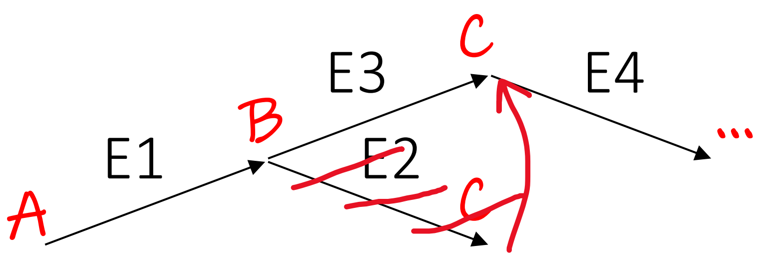
←This query / view is minimal



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

```sql
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!

Example adopted from Dan Suciu

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

Islands of Tractability of CQ Evaluation

- Major Research Program: Identify tractable cases of the combined complexity of conjunctive query evaluation.
- Over the years, this program has been pursued by two different research communities:
  - The Database Theory community
  - The Constraint Satisfaction community
- Explanation: Problems in those community are closely related:

Constraint Satisfaction Problem ≡ Homomorphism Problem ≡ CQ evaluation

[Feder, Vardi 1993] [Chandra, Merlin 1977]

[Kolaitis, Vardi 2000]

Feder, Vardi: Monotone monadic SNP and constraint satisfaction, STOC 1993 https://doi.org/10.1145/167088.167245 / Kolaitis, Vardi: Conjunctive-Query Containment and Constraint Satisfaction, JCSS 2000 https://doi.org/10.1006/jcss.2000.1713 / Chandra, Merlin. "Optimal implementation of conjunctive queries in relational data bases", STOC 1977. https://doi.org/10.1145/800105.803397
Based on Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

Beyond Conjunctive Queries

- What can we say about query languages of intermediate expressive power between conjunctive queries and the full relational calculus?

- Conjunctive queries form the sublanguage of relational algebra obtained by using only cartesian product, projection, and selection with equality conditions.

- The next step would be to consider relational algebra expressions that also involve union.

Beyond Conjunctive Queries

- Definition:
  - A Union of Conjunctive Queries (UCQ) is a query expressible by an expression of the form $q_1 \cup q_2 \cup \ldots \cup q_m$, where each q_i is a conjunctive query.
  - A monotone query is a query expressible by a relational algebra expression which uses only union, cartesian product, projection, and selection (with <u>equality condition only</u>).
- Fact:
  - Monotone queries are precisely the queries expressible by relational calculus expressions using $\wedge$, $\vee$, and $\exists$ only (also assuming restriction to <u>equality</u> here).
  - Every UCQ is a monotone query.
  - Every monotone query is equivalent to a UCQ
    - but this normal form may have exponentially many disjuncts

$(a+b+c)(d+e+f)(g+h+j) = \ldots$ *how big as sum of products* ?

Beyond Conjunctive Queries

- Definition:
  - A Union of Conjunctive Queries (UCQ) is a query expressible by an expression of the form $q_1 \cup q_2 \cup \ldots \cup q_m$, where each q_i is a conjunctive query.
  - A monotone query is a query expressible by a relational algebra expression which uses only union, cartesian product, projection, and selection (with equality condition only).
- Fact:
  - Monotone queries are precisely the queries expressible by relational calculus expressions using $\wedge$, $\vee$, and $\exists$ only (also assuming restriction to equality here).
  - Every UCQ is a monotone query.
  - Every monotone query is equivalent to a UCQ
    - but this normal form may have exponentially many disjuncts

$$(a+b+c)(d+e+f)(g+h+j) = adg + adh + adj + aeg + aeh + \ldots + cfj$$

27 products

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation $E(A,B)$, find paths of length 1 or 2

RA ? (unnamed RA)

DRC ?

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation $E(A,B)$, find paths of length 1 or 2

RA $E \cup \pi_{\$1,\$4}(\sigma_{\$2=\$3}(E \times E))$ (unnamed RA)

DRC ?

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation $E(A,B)$, find paths of length 1 or 2

RA $\quad E \cup \pi_{\$1,\$4}(\sigma_{\$2=\$3}(E \times E))$

DRC $\quad \{(x,y) | E(x,y) \vee \exists z[E(x,z) \wedge E(z,y)]\}$

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation $E(A,B)$, find paths of length 1 or 2

RA $\quad E \cup \pi_{\$1,\$4}(\sigma_{\$2=\$3}(E \times E))$

DRC $\quad \{(x,y) | E(x,y) \vee \exists z[E(x,z) \wedge E(z,y)]\}$

Monotone Query

Assume schema R(A,B), S(A,B), T(B,C), V(B,C)

Is following query monotone ? $(R \cup S) \bowtie (T \cup V)$

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation $E(A,B)$, find paths of length 1 or 2

RA $\quad E \cup \pi_{\$1,\$4}(\sigma_{\$2=\$3}(E \times E))$

DRC $\quad \{(x,y) | E(x,y) \vee \exists z[E(x,z) \wedge E(z,y)]\}$

Monotone Query

Assume schema R(A,B), S(A,B), T(B,C), V(B,C)

Following query is monotone: $\quad (R \cup S) \bowtie (T \cup V)$

Equal to a UCQ? $\quad$?

Unions of CQs and Monotone Queries

Union of Conjunctive Queries (UCQ)

Given edge relation *E*(*A*,*B*), find paths of length 1 or 2

RA $E \cup \pi_{\$1,\$4}(\sigma_{\$2=\$3}(E \times E))$

DRC $\{(x,y) | E(x,y) \vee \exists z[E(x,z) \wedge E(z,y)]\}$

Monotone Query

Assume schema R(A,B), S(A,B), T(B,C), V(B,C)

Following query is monotone: $(R \cup S) \bowtie (T \cup V)$

Equal to following UCQ: $\underbrace{(R \bowtie T)}_{q_1} \cup (R \bowtie_{q_2} V) \cup (S \bowtie T) \cup (S \bowtie V)$

The Containment Problem for Unions of CQs

THEOREM [Sagiv, Yannakakis 1980]

Let $q_1 \cup q_2 \cup \cdots \cup q_\mathrm{m}$ *and* $q'_1 \cup q'_2 \cup \cdots \cup q'_n$ *be two UCQs.*
Then the following are equivalent:

1) $q_1 \cup q_2 \cup \cdots \cup q_\mathrm{m} \subseteq q'_1 \cup q'_2 \cup \cdots \cup q'_n$
2) For every *i* ≤ *m*, there is *j* ≤ *n* such that $q_i \subseteq q'_j$

Proof:

2. ⇒ 1. This direction is obvious.

1. ⇒ 2. Since $D_C[q_i] \vDash q_i$, we have that $D_C[q_i] \vDash q_1 \cup q_2 \cup \ldots \cup q_m$.
Because of containment, $D_C[q_i] \vDash q'_1 \cup q'_2 \cup \ldots \cup q'_n$.
Thus there is some j ≤ n with $D_C[q_i] \vDash q'_j$.
Thus from the CQ homomorphism Theorem $q_i \subseteq q'_j$.

The Complexity of Database Query Languages

	Relational Calculus	CQs	UCQs
Query Evaluation: Data Complexity	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)
Query Evaluation: Combined Compl.	PSPACE-complete	NP-complete	NP-complete
Query Equivalence & Containment	Undecidable	NP-complete	NP-complete

Source: Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

Monotone Queries

- Even though monotone queries have the same expressive power as unions of conjunctive queries, the containment problem for monotone queries has higher complexity than the containment problem for unions of conjunctive queries (syntax/complexity tradeoff)

- Theorem: Sagiv and Yannakakis – 1982

  The containment problem for monotone queries is Π_2^p-complete.

- Note: The prototypical Π_2^p-complete problem is $\forall\exists$SAT, i.e., the restriction of QBF to formulas of the form

$$\forall x_1 \ldots \forall x_m \exists y_1 \ldots \exists y_n \, \phi.$$

Source: Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

The Complexity of Database Query Languages

	Relational Calculus	CQs	UCQs	Monotone queries
Query Evaluation: Data Complexity	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)
Query Evaluation: Combined Compl.	PSPACE-complete	NP-complete	NP-complete	NP-complete
Query Equivalence & Containment	Undecidable	NP-complete	NP-complete	Π_2^p-complete

Source: Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

Conjunctive Queries with Inequalities

- Definition: Conjunctive queries with inequalities form the sublanguage of relational algebra obtained by using only cartesian product, projection, and selection with equality and inequality ($\neq$, $<$, $\leq$) conditions.

- Example: Q(x,y):-- E(x,z), E(z,w),E(w,y), $z \neq w$, $z < y$.

- Theorem: (Klug – 1988, van der Meyden – 1992)
  – The query containment problem for conjunctive queries with inequalities is Π_2^p-complete.
  – The query evaluation problem for conjunctive queries with inequalities in NP-complete.

Source: Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

The Complexity of Database Query Languages

	Relational Calculus	CQs	UCQs	Monotone queries / CQs with inequalities
Query Evaluation: Data Complexity	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)	In LOGSPACE (hence, in P)
Query Evaluation: Combined Compl.	PSPACE-complete	NP-complete	NP-complete	NP-complete
Query Equivalence & Containment	Undecidable	NP-complete	NP-complete	Π_2^p-complete

Source: Phokion Kolaitis' "Logic and Databases" series at Simons Institute, 2016. https://simons.berkeley.edu/talks/logic-and-databases

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

Following slides are literally from Phokion Kolaitis's talk on "Logic and databases" at "Logical structures in Computation Boot Camp", Berkeley 2016:
https://simons.berkeley.edu/talks/logic-and-databases

Logic and Databases

Phokion G. Kolaitis

UC Santa Cruz & IBM Research – Almaden

Lecture 4 – Part 1

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Thematic Roadmap

- ✓ Logic and Database Query Languages
  - Relational Algebra and Relational Calculus
  - Conjunctive queries and their variants
  - Datalog
- ✓ Query Evaluation, Query Containment, Query Equivalence
  - Decidability and Complexity
- ✓ Other Aspects of Conjunctive Query Evaluation
- Alternative Semantics of Queries
  - Bag Databases: Semantics and Conjunctive Query Containment
  - Probabilistic Databases: Semantics and Dichotomy Theorems for Conjunctive Query Evaluation
  - Inconsistent Databases: Semantics and Dichotomy Theorems

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Alternative Semantics

- So far, we have examined logic and databases under classical semantics:
  - The database relations are sets.
  - Tarskian semantics are used to interpret queries definable be first-order formulas.
- Over the years, several different alternative semantics of queries have been investigated. We will discuss three such scenarios:
  - The database relations can be bags (multisets).
  - The databases may be probabilistic.
  - The databases may be inconsistent.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Sets vs. Multisets

Relation EMPLOYEE(name, dept, salary)

- Relational Algebra Expression:

  $$\pi_{salary} (\sigma_{dept = CS} (EMPLOYEE))$$

- SQL query:

  ```
  SELECT   salary
  FROM     EMPLOYEE
  WHERE    dpt = 'CS'
  ```

- SQL returns a bag (multiset) of numbers in which a number may appear several times, provided different faculty had the same salary.
- SQL does not eliminate duplicates, in general, because:
  - Duplicates are important for aggregate queries (e.g., average)
  - Duplicate elimination takes nlogn time.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Relational Algebra Under Bag Semantics

Operation	Multiplicity
Union $R_1 \cup R_2$	$m_1 + m_2$
Intersection $R_1 \cap R_2$	$\min(m_1, m_2)$
Product $R_1 \times R_2$	$m_1 \times m_2$
Projection and Selection	Duplicates are not eliminated

- R_1

A	B
1	2
1	2
2	3

- R_2

B	C
2	4
2	5

- $(R_1 \bowtie R_2)$

A	B	C
1	2	4
1	2	4
1	2	5
1	2	5

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Conjunctive Queries Under Bag Semantics

Chaudhuri & Vardi – 1993

Optimization of ***Real*** Conjunctive Queries

- Called for a re-examination of conjunctive-query optimization under bag semantics.
- In particular, they initiated the study of the containment problem for conjunctive queries under bag semantics.
- This problem has turned out to be *much more challenging* than originally perceived.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

PROBLEMS

Problems worthy
of attack
prove their worth
by hitting back.

in: *Grooks* by Piet Hein (1905-1996)

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Query Containment Under Set Semantics

Class of Queries	**Complexity of Query Containment**
Conjunctive Queries	NP-complete Chandra & Merlin – 1977
Unions of Conjunctive Queries	NP-complete Sagiv & Yannakakis - 1980
Conjunctive Queries with $\neq$, $\leq$, $\geq$	Π_2^p-complete Klug 1988, van der Meyden -1992
First-Order (SQL) queries	Undecidable Trakhtenbrot - 1949

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Bag Semantics vs. Set Semantics

- For bags R_1, R_2:
  $R_1 \subseteq_{BAG} R_2$ if $m(\mathbf{a},R_1) \leq m(\mathbf{a},R_2)$, for every tuple $\mathbf{a}$.
- $Q^{BAG}(D)$: Result of evaluating Q on (bag) database D.
- $Q_1 \subseteq_{BAG} Q_2$ if for every (bag) database D, we have that

$$Q_1^{BAG}(D) \subseteq_{BAG} Q_2^{BAG}(D).$$

Fact:

- $Q_1 \subseteq_{BAG} Q_2$ implies $Q_1 \subseteq Q_2$.
- The converse does **not** always hold.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Bag Semantics vs. Set Semantics

Fact: $Q_1 \subseteq Q_2$ does not imply that $Q_1 \subseteq_{BAG} Q_2$.

Example:

- $Q_1(x)$:- $P(x), T(x)$
- $Q_2(x)$:- $P(x)$

- $Q_1 \subseteq Q_2$ (obvious from the definitions)
- $Q_1 \nsubseteq_{BAG} Q_2$
- Consider the (bag) instance D = {P(a), T(a), T(a)}. Then:
  - $Q_1(D) = \{a,a\}$
  - $Q_2(D) = \{a\}$, so $Q_1(D) \nsubseteq Q_2(D)$.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Query Containment under Bag Semantics

- Chaudhuri & Vardi - 1993 stated that:

  Under bag semantics, the containment problem for conjunctive queries is Π_2^p-hard.

- Problem:
  - What is the exact complexity of the containment problem for conjunctive queries under bag semantics?
  - Is this problem decidable?

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Query Containment Under Bag Semantics

- 23 years have passed since the containment problem for conjunctive queries under bag semantics was raised.
- Several attacks to solve this problem have failed.
- At least two technically flawed PhD theses on this problem have been produced.
- Chaudhuri and Vardi have withdrawn the claimed Π_2^p-hardness of this problem; no one has provided a proof.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Query Containment Under Bag Semantics

- The containment problem for conjunctive queries under bag semantics remains **open** to date.

- However, progress has been made towards the containment problem under bag semantics for the two main extensions of conjunctive queries:
  - Unions of conjunctive queries
  - Conjunctive queries with $\neq$

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

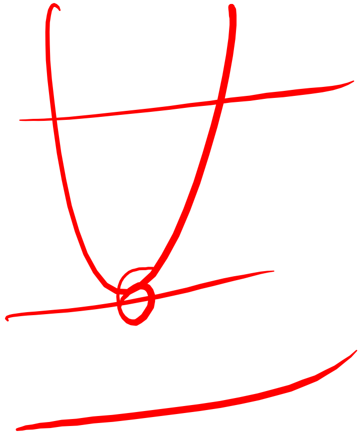
Unions of Conjunctive Queries

Theorem (Ioannidis & Ramakrishnan – 1995):
Under bag semantics, the containment problem for unions of conjunctive queries is **undecidable**.

Hint of Proof:
Reduction from Hilbert's 10^{th} Problem.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

$4x_1^7 - 18x_2^5 + 2 = 0 \quad x \in \mathbb{Z}$



Hilbert's 10th Problem

- Hilbert's 10th Problem – 1900
  (10th in Hilbert's list of 23 problems)

  Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.

  In effect, Hilbert's 10th Problem is:
  Find an algorithm for the following problem:
  Given a polynomial $P(x_1,\ldots,x_n)$ with integer coefficients, does it have an all-integer solution?

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Hilbert's 10^{th} Problem

- Hilbert's 10^{th} Problem – 1900

  (10^{th} in Hilbert's list of 23 problems)

  Find an algorithm for the following problem:

  Given a polynomial $P(x_1,\ldots,x_n)$ with integer coefficients, does it have an all-integer solution?

- Y. Matiyasevich – 1971

  (building on M. Davis, H. Putnam, and J. Robinson)

  – Hilbert's 10^{th} Problem is **undecidable**, hence **no** such algorithm exists.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Hilbert's 10^{th} Problem

- Fact: The following variant of Hilbert's 10^{th} Problem is undecidable:
  - Given two polynomials $p_1(x_1,\ldots x_n)$ and $p_2(x_1,\ldots x_n)$ with positive integer coefficients and no constant terms, is it true that $p_1 \leq p_2$?

    In other words, is it true that $p_1(a_1,\ldots,a_n) \leq p_2(a_1,\ldots a_n)$, for all positive integers $a_1,\ldots,a_n$?

- Thus, there is no algorithm for deciding questions like:
  - Is $3x_1^4x_2x_3 + 2x_2x_3 \leq x_1^6 + 5x_2x_3$?

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Unions of Conjunctive Queries

Theorem (Ioannidis & Ramakrishnan – 1995):
Under bag semantics, the containment problem for unions of conjunctive queries is **undecidable**.

Hint of Proof:

- Reduction from the previous variant of Hilbert's 10th Problem:
  - Use joins of unary relations to encode monomials (products of variables).
  - Use unions to encode sums of monomials.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Unions of Conjunctive Queries

Example: Consider the polynomial $3x_1{}^4x_2x_3 + 2x_2x_3$

- The monomial $x_1{}^4x_2x_3$ is encoded by the conjunctive query $P_1(w),P_1(w),P_1(w), P_1(w), P_2(w),P_3(w)$.
- The monomial x_2x_3 is encoded by the conjunctive query $P_2(w),P_3(w)$.
- The polynomial $3x_1{}^4x_2x_3 + 2x_2x_3$ is encoded by the union having:
  - three copies of $P_1(w),P_1(w),P_1(w), P_1(w), P_2(w),P_3(w)$ and
  - two copies of $P_2(w),P_3(w)$.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Complexity of Query Containment

Class of Queries	Complexity – Set Semantics	Complexity – Bag Semantics
Conjunctive queries	NP-complete CM – 1977	
Unions of conj. queries	NP-complete SY - 1980	Undecidable IR - 1995
Conj. queries with $\neq$, $\leq$, $\geq$	Π_2^p-complete vdM - 1992	
First-order (SQL) queries	Undecidable Trakhtenbrot - 1949	Undecidable

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Conjunctive Queries with ≠

Theorem (Jayram, K …, Vee – 2006):
Under bag semantics, the containment problem for conjunctive queries with ≠ is **undecidable**.

In fact, this problem is **undecidable** even if

- the queries use only a single relation of arity 2;
- the number of inequalities in the queries is at most some fixed (albeit huge) constant.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Complexity of Query Containment

Class of Queries	**Complexity – Set Semantics**	**Complexity – Bag Semantics**
Conjunctive queries	NP-complete CM – 1977	**Open**
Unions of conj. queries	NP-complete SY - 1980	Undecidable IR - 1995
Conj. queries with $\neq$, $\leq$, $\geq$	Π_2^p-complete vdM - 1992	Undecidable JKV - 2006
First-order (SQL) queries	Undecidable Trakhtenbrot - 1949	Undecidable

26

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01

Subsequent Developments

- Some progress has been made towards identifying special classes of conjunctive queries for which the containment problem under bag semantics is decidable.
  - Afrati, Damigos, Gergatsoulis – 2010
    - Projection-free conjunctive queries.
  - Kopparty and Rossman – 2011
    - A large class of boolean conjunctive queries on graphs.

Source: Phokion Kolaitis: https://simons.berkeley.edu/talks/phokion-kolaitis-2016-09-01