CSE544
Data Management

Lectures 9-10

Advanced Query Processing

Discuss the paper

- Why do they use the IMDB database instead of TPC-H?

- Do cardinality estimators typically under- or over-estimate?

- From cardinality to cost: how critical is that?

[How good are they]

Single Table Estimation

	median	90th	95th	max
PostgreSQL	1.00	2.08	6.10	207
DBMS A	1.01	1.33	1.98	43.4
DBMS B	1.00	6.03	30.2	104000
DBMS C	1.06	1677	5367	20471
HyPer	1.02	4.47	8.00	2084

Table 1: Q-errors for base table selections

Discuss histograms v.s. samples

Single Table Estimation

- 1d Histograms: accurate for selection on a single equality or range predicate; poor for multiple predicates; useless for LIKE
- Samples: great for correlations, or predicates like LIKE; poor for low selectivity predicates: estimate is 0, then use ”magic constants”

Joins (0 to 6)

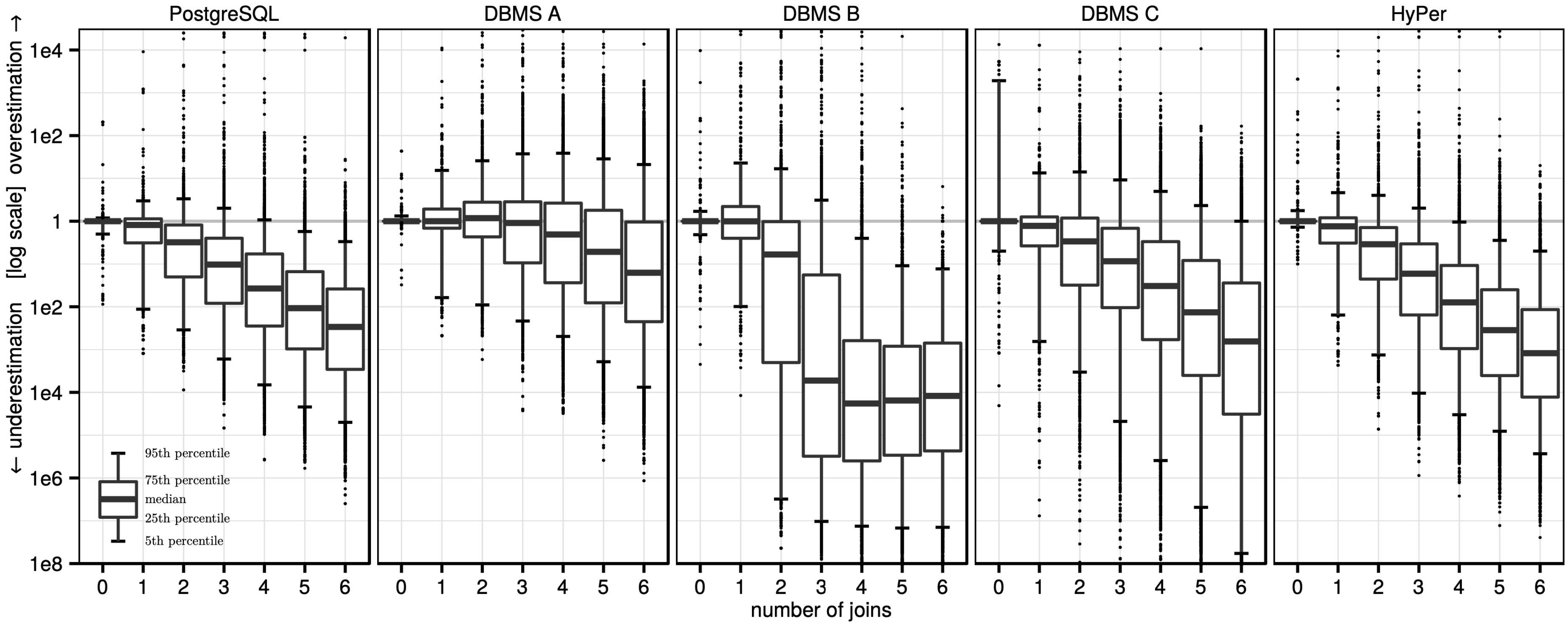
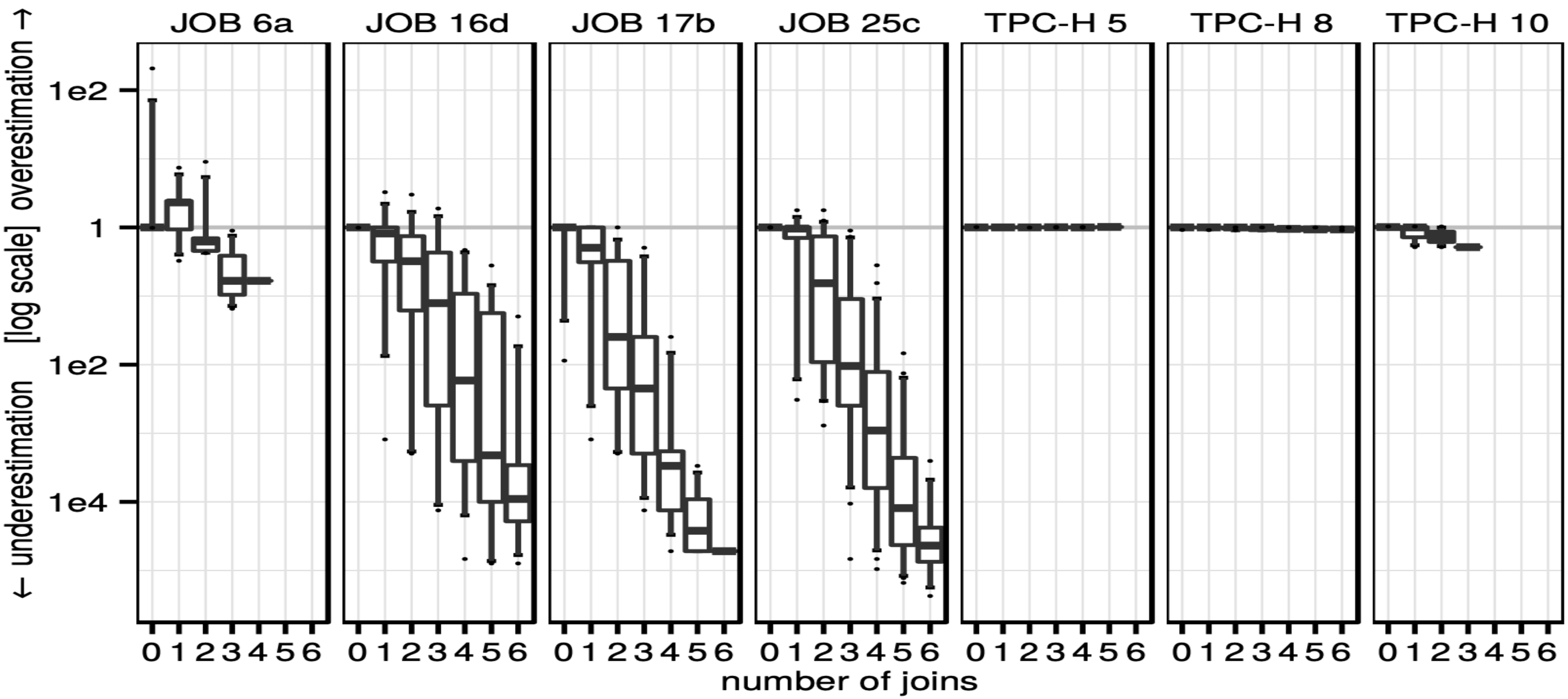


Figure 3: Quality of cardinality estimates for multi-join queries in comparison with the true cardinalities. Each boxplot summarizes the error distribution of all subexpressions with a particular size (over all queries in the workload)

TPC-H v.s. Real Data (IMDB)



[How good are they]

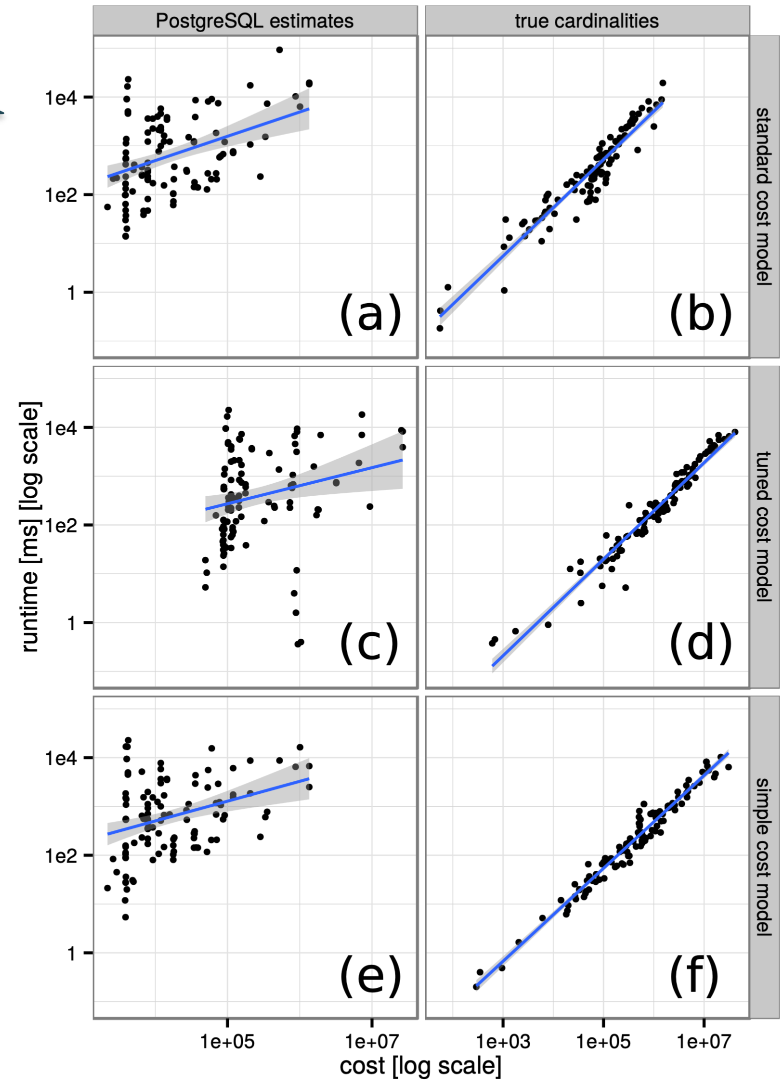
Cardinalities to Cost

- Cardinality estimation creates largest errors
- Complex or simple cost models don't differ much

Postgres cost

No I/O, keep only CPU

Their own simple formula

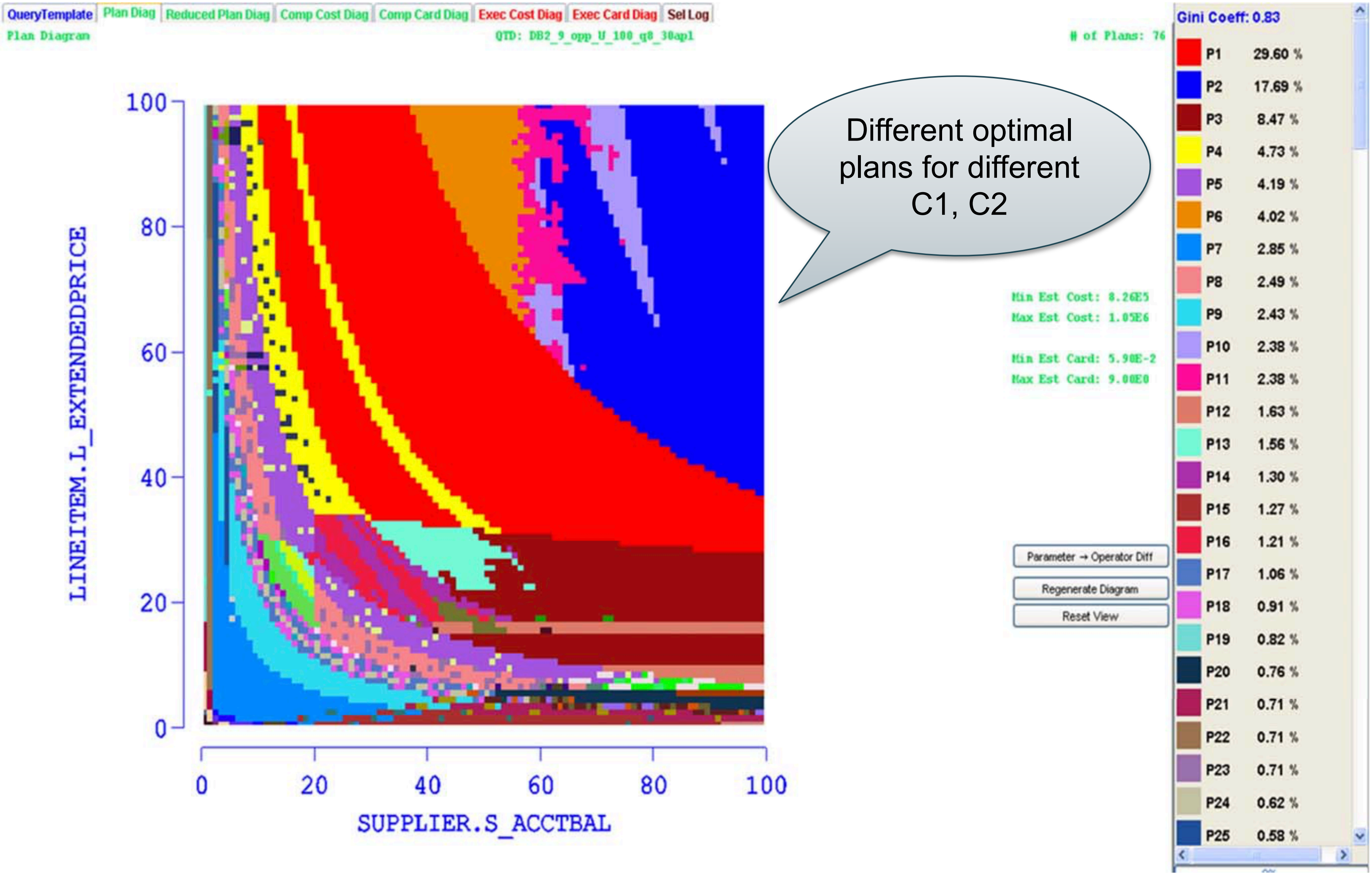


Yet Another Difficulties

- SQL Queries are often issued from applications
- Optimized once using *prepare* statement, executed often
- The constants in the query are not know until execution time: optimized plan may be suboptimal

```
select
  o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(volume)
from
 (select YEAR(o_orderdate) as o_year,
              l_extendedprice * (1 - l_discount) as volume,
                    n2.n_name as nation
  from part, supplier, lineitem, orders,
      customer, nation n1, nation n2, region
  where p_partkey = l_partkey and s_suppkey = l_suppkey
    and l_orderkey = o_orderkey and o_custkey = c_custkey
    and c_nationkey = n1.n_nationkey
    and n1.n_regionkey = r_regionkey
    and r_name = 'AMERICA'
    and s_nationkey = n2.n_nationkey
    and o_orderdate between '1995-01-01'
    and '1996-12-31'
    and p_type = 'ECONOMY ANODIZED STEEL'
   and s_acctbal ≤ C1 and l_extendedprice ≤ C2 ) as all_nations
group by o_year order by o_year
```

Optimize without knowing C1, C2


QueryTemplate
Plan Diag
Reduced Plan Diag
Comp Cost Diag
Comp Card Diag
Exec Cost Diag
Exec Card Diag
Sel Log
Plan Diagram
QTD: DB2_9_opp_U_100_q8_30apl
# of Plans: 76
Gini Coeff: 0.83
P1 29.60 %
P2 17.69 %
P3 8.47 %
P4 4.73 %
P5 4.19 %
P6 4.02 %
P7 2.85 %
P8 2.49 %
P9 2.43 %
P10 2.38 %
P11 2.38 %
P12 1.63 %
P13 1.56 %
P14 1.30 %
P15 1.27 %
P16 1.21 %
P17 1.06 %
P18 0.91 %
P19 0.82 %
P20 0.76 %
P21 0.71 %
P22 0.71 %
P23 0.71 %
P24 0.62 %
P25 0.58 %
Different optimal plans for different C1, C2
Min Est Cost: 8.26E5
Max Est Cost: 1.05E6
Min Est Card: 5.90E-2
Max Est Card: 9.00E0
Parameter → Operator Diff
Regenerate Diagram
Reset View
LINEITEM.L_EXTENDEDPRICE
SUPPLIER.S_ACCTBAL
0
20
40
60
80
100

Query Plans

[How good are they]

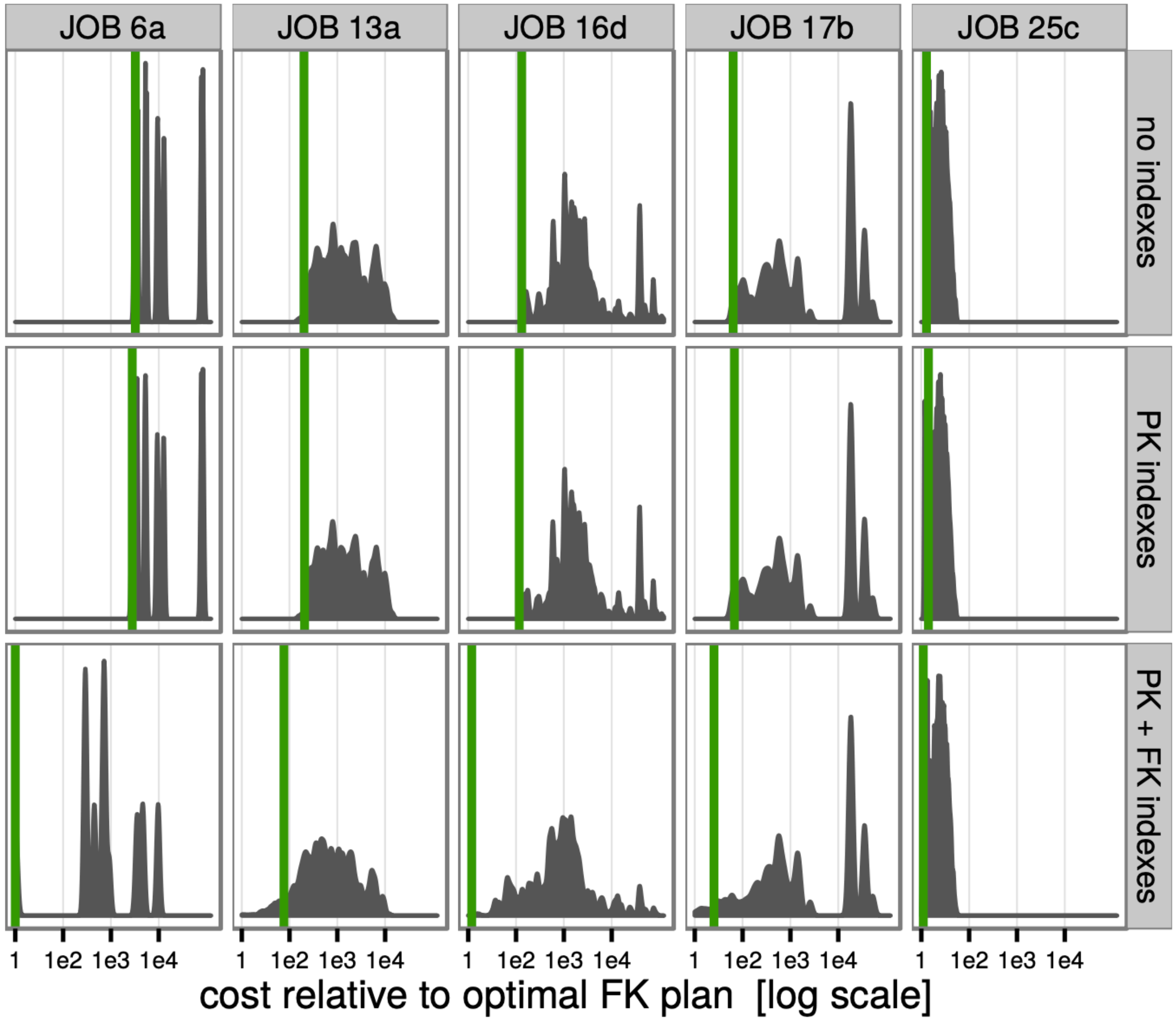


Figure 9: Cost distributions for 5 queries and different index configurations. The vertical green lines represent the cost of the optimal plan

[How good are they]

	PK indexes			PK + FK indexes		
	median	95%	max	median	95%	max
zig-zag	1.00	1.06	1.33	1.00	1.60	2.54
left-deep	1.00	1.14	1.63	1.06	2.49	4.50
right-deep	1.87	4.97	6.80	47.2	30931	738349

Table 2: Slowdown for restricted tree shapes in comparison to the optimal plan (true cardinalities)

[How good are they]

	PK indexes						PK + FK indexes					
	PostgreSQL estimates			true cardinalities			PostgreSQL estimates			true cardinalities		
	median	95%	max	median	95%	max	median	95%	max	median	95%	max
Dynamic Programming	1.03	1.85	4.79	1.00	1.00	1.00	1.66	169	186367	1.00	1.00	1.00
Quickpick-1000	1.05	2.19	7.29	1.00	1.07	1.14	2.52	365	186367	1.02	4.72	32.3
Greedy Operator Ordering	1.19	2.29	2.36	1.19	1.64	1.97	2.35	169	186367	1.20	5.77	21.0

Table 3: Comparison of exhaustive dynamic programming with the Quickpick-1000 (best of 1000 random plans) and the Greedy Operator Ordering heuristics. All costs are normalized by the optimal plan of that index configuration

Advanced Query Processing

State of the art

- A lot based on heuristics

Advanced techniques

- Find principled, provable techniques

Outline

- AGM bound: today

- Next week:
  – Worst-case optimal algorithm
  – Acyclic queries, Yannakakis algorithm
  – Tree decomposition of cyclic queries

Upper Bounds

Fix input statistics for D

- $|R|, |S| \leq N$
- How large are the answers to these queries?

$$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$$

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key:

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key: $|Q(D)| \leq N$

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key: $|Q(D)| \leq N$
- S.Y has degree ≤ d: $|Q(D)| \leq d \times N$

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key: $|Q(D)| \leq N$
- S.Y has degree ≤ d: $|Q(D)| \leq d \times N$

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

No other info:

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key: $|Q(D)| \leq N$
- S.Y has degree ≤ d: $|Q(D)| \leq d \times N$

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

No other info: $|Q(D)| \leq N^{3/2}$

Upper Bounds

Fix input statistics for D

- |R|, |S| ≤ N
- How large are the answers to these queries?

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

No other info: $|Q(D)| \leq N^2$

- S.Y is a key: $|Q(D)| \leq N$
- S.Y has degree ≤ d: $|Q(D)| \leq d \times N$

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

No other info: $|Q(D)| \leq N^{3/2}$

WOW!

Simple Fact #1

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Then:

$$|Q| \leq |R_1| \times \cdots \times |R_m|$$

Simple Fact #2

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Suppose $R_{i_1}, R_{i_2}, \ldots, R_{i_\ell}$ contain all variables (attributes) $X_1, \ldots, X_k$. Then:

$$|Q| \leq |R_{i_1}| \times \cdots \times |R_{i_\ell}|$$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover,* meaning: for each variables X_i: $\sum_{j:R_j \text{ contains } X_i} u_j \geq 1$. Then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover,* meaning: for each variables X_i: $\sum_{j: R_j \text{ contains } X_i} u_j \geq 1$. Then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Example: $Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

$|Q| \leq |R|^{1/2} |S|^{1/2} |T|^{1/2} = N^{3/2}$

Discussion

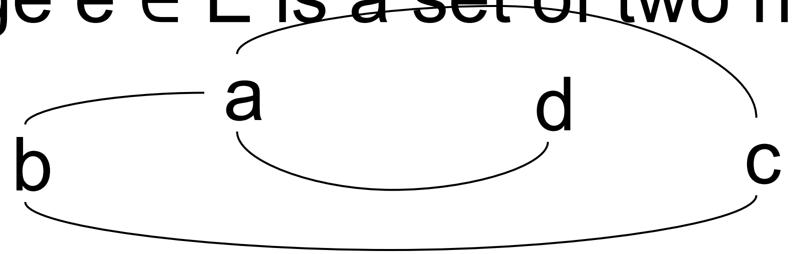
- The “simple fact #3” is called the AGM bound, after Atserias,Grohe,Marx
- We will prove this bound next
- First: a detour in graph theory (fractional edge covers) and inequalities
- Next time: an algorithm with a matching runtime, derived from the proof of the AGM bound

Quick Review

- Graphs, hypergraphs
- Edge cover
- Fractional edge cover

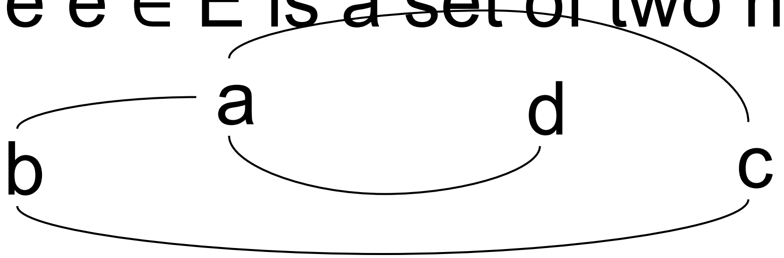
Graphs and Hypergraphs

- An undirected graph G = (V, E) where each edge e ∈ E is a set of two nodes

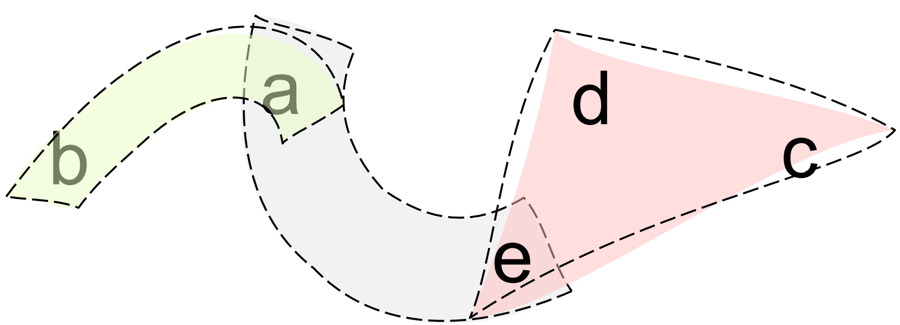


Graphs and Hypergraphs

- An undirected graph G = (V, E) where each edge e ∈ E is a set of two nodes

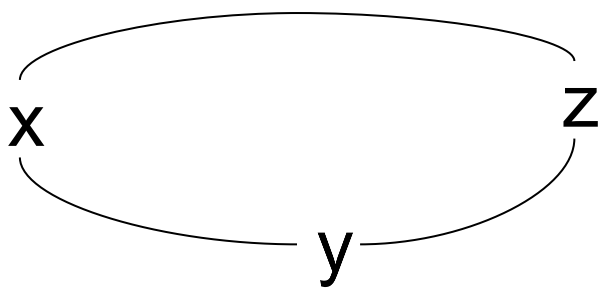


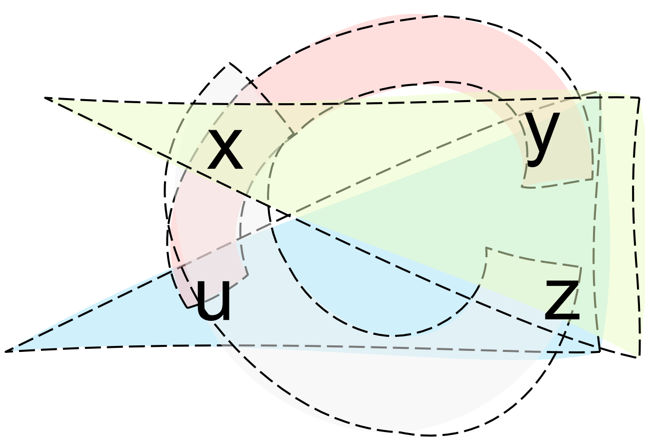
- A hypergraph is G = (V, E) where each edge is some set (of 1 or 2 or >2 nodes)



Edge Cover

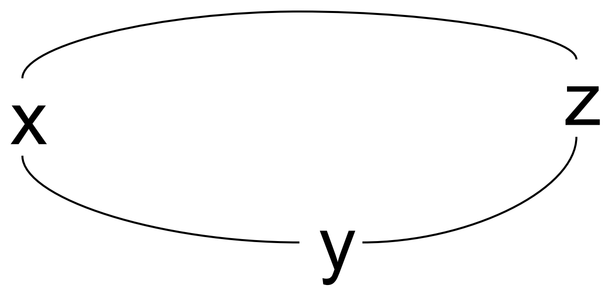
- An *<u>edge cover</u>* of a (hyper)graph is a subset of edges that contain all the vertices

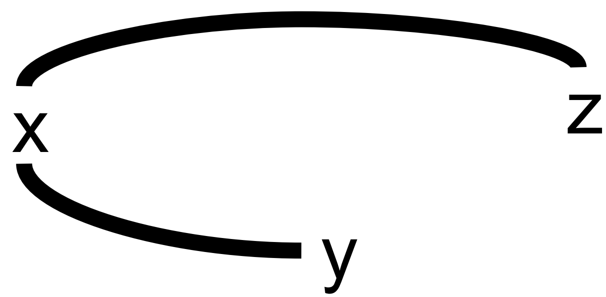


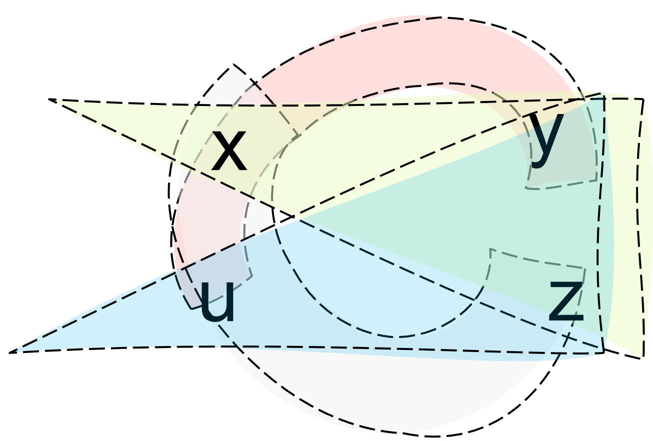


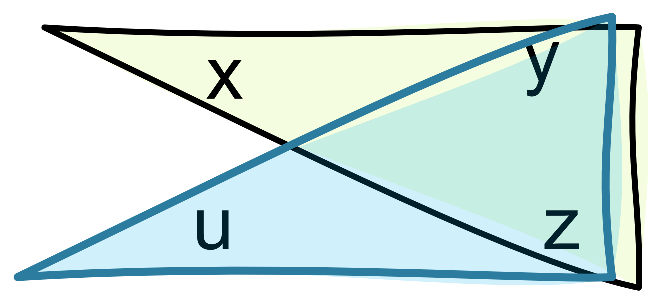
Edge Cover

- An *edge cover* of a (hyper)graph is a subset of edges that contain all the vertices





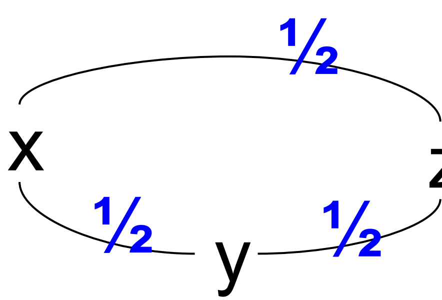


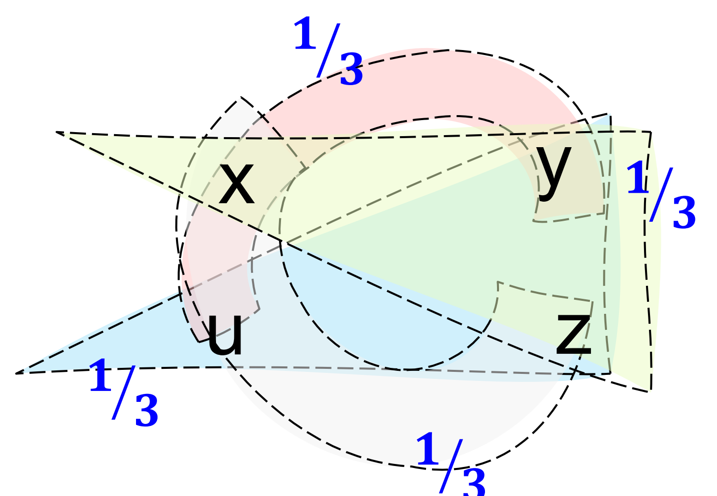


Fractional Edge Cover

- A ***fractional edge cover*** of a (hyper)graph are numbers $u_e \geq 0$, one for each edge e, such that, for every vertex x:

$$\sum_{e:x \in e} u_e \geq 1$$





Inequalities

Cauchy-Schwartz

$$\sum_i a_i^{1/2} b_i^{1/2} \leq \left(\sum_i a_i\right)^{1/2} \left(\sum_i b_i\right)^{1/2}$$

$a_i \geq 0$, etc

Inequalities

Cauchy-Schwartz

$$\sum_i a_i^{1/2} b_i^{1/2} \leq \left(\sum_i a_i\right)^{1/2} \left(\sum_i b_i\right)^{1/2}$$

$a_i \geq 0$, etc

Generalized Hölder. If $u_1 + u_2 + u_3 \geq 1$ then:

$$\sum_i a_i^{u_1} b_i^{u_2} c_i^{u_3} \leq \left(\sum_i a_i\right)^{u_1} \left(\sum_i b_i\right)^{u_2} \left(\sum_i c_i\right)^{u_3}$$

Inequalities

Cauchy-Schwartz

$$\sum_i a_i^{1/2} b_i^{1/2} \leq \left(\sum_i a_i\right)^{1/2} \left(\sum_i b_i\right)^{1/2}$$

$a_i \geq 0$, etc

Generalized Hölder. If $u_1 + u_2 + u_3 \geq 1$ then:

$$\sum_i a_i^{u_1} b_i^{u_2} c_i^{u_3} \leq \left(\sum_i a_i\right)^{u_1} \left(\sum_i b_i\right)^{u_2} \left(\sum_i c_i\right)^{u_3}$$

Friedgut 2004

$$\sum_{i,j,k} a_{ij}^{1/2} b_{jk}^{1/2} c_{ki}^{1/2} \leq \left(\sum_{i,j} a_{ij}\right)^{1/2} \left(\sum_{j,k} b_{jk}\right)^{1/2} \left(\sum_{k,i} c_{ki}\right)^{1/2}$$

Friedgut's Inequality (2004)

Let G=(V,E) be a hypergraph, where:

$$V = \{x_1, \ldots, x_k\}, \qquad E = \{e_1, \ldots, e_m\}$$

Let $u_1, u_2, \ldots, u_m$ be a fractional edge cover. Then:

$$\sum_{x_1,\ldots,x_k} a_{1,e_1}^{u_1} \cdots a_{m,e_m}^{u_m} \leq \left(\sum_{e_1} a_{1,e_1}\right)^{u_1} \cdots \left(\sum_{e_m} a_{m,e_m}\right)^{u_m}$$

Here, $a_{1,xyz\ldots}$ is a tensor; similarly $a_{2,\ldots}$ etc.

Example: think of $a_{1,xy}, a_{2,yz}, a_{3,zx}$ as three matrices, like a_{xy}, b_{yz}, c_{zx}:

Proof

$V = \{x_1, \dots, x_k\}, \qquad E = \{e_1, \dots, e_m\}$

$$\sum_{x_1,\dots,x_k} a_{1,e_1}^{u_1} \cdots a_{m,e_m}^{u_m} \leq \left(\sum_{e_1} a_{1,e_1}\right)^{u_1} \cdots \left(\sum_{e_m} a_{m,e_m}\right)^{u_m}$$

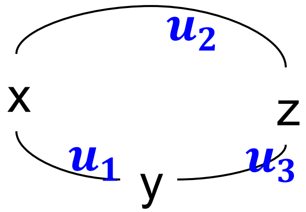
Proof: by induction on the number of nodes k

Case 1: k=1

Then $e_1 = e_2 = \cdots = e_m = \{x_1\}$.

The inequality is generalized Hölder's inequality

$V = \{x, y, z\}, E = \{xy, yz, zx\}$

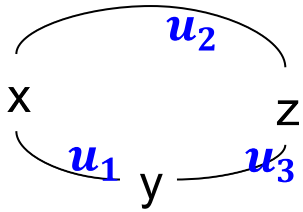


Proof

Case 2: k > 1. First we illustrate on a special case:

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$V = \{x, y, z\}, E = \{xy, yz, zx\}$$



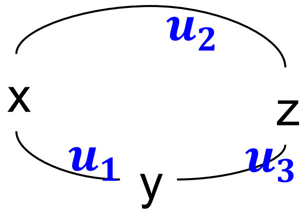
Proof

Case 2: k > 1. First we illustrate on a special case:

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} = \sum_{x,y} a_{xy}^{u_1} \left(\sum_{z} b_{yz}^{u_2} c_{zx}^{u_3}\right)$$

$V = \{x, y, z\}, E = \{xy, yz, zx\}$



Proof

Case 2: k > 1. First we illustrate on a special case:

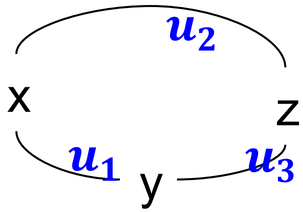
$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \le \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} = \sum_{x,y} a_{xy}^{u_1} \left(\sum_{z} b_{yz}^{u_2} c_{zx}^{u_3}\right)$$

$$\le \sum_{x,y} a_{xy}^{u_1} \left(\sum_{z} b_{yz}\right)^{u_2} \left(\sum_{z} c_{zx}\right)^{u_3}$$

Hölder. Why is $u_2 + u_3 \geq 1$?

$V = \{x, y, z\}, E = \{xy, yz, zx\}$



Proof

Case 2: k > 1. First we illustrate on a special case:

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} = \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}^{u_2} c_{zx}^{u_3}\right)$$

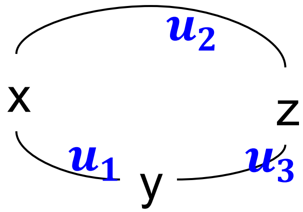
$$\leq \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}\right)^{u_2} \left(\sum_z c_{zx}\right)^{u_3}$$

$$= \sum_{x,y} a_{xy}^{u_1} B_y^{u_2} C_x^{u_3}$$

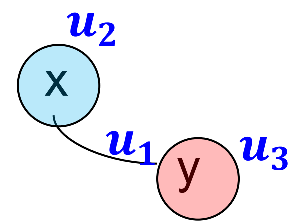
Hölder. Why is $u_2 + u_3 \geq 1$?

Notations:
$B_y = \sum_z b_{yz}$, $C_z = \sum_z c_{zx}$

$V = \{x, y, z\}, E = \{xy, yz, zx\}$ $\quad$ $V' = \{x, y\}, E' = \{xy, y, x\}$



Proof



Case 2: k > 1. First we illustrate on a special case:

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} = \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}^{u_2} c_{zx}^{u_3}\right)$$

$$\leq \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}\right)^{u_2} \left(\sum_z c_{zx}\right)^{u_3}$$

$$= \sum_{x,y} a_{xy}^{u_1} B_y^{u_2} C_x^{u_3}$$

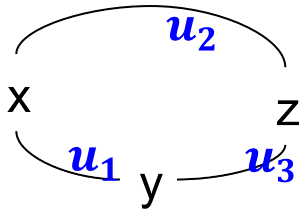
$$\leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_y B_y\right)^{u_2} \left(\sum_x C_x\right)^{u_3}$$

Hölder. Why is $u_2 + u_3 \geq 1$?

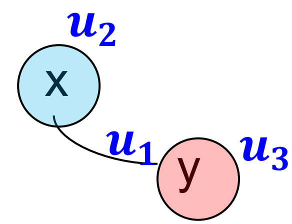
Notations:
$B_y = \sum_z b_{yz}$, $C_z = \sum_z c_{zx}$

Induction on V', E'

$V = \{x, y, z\}, E = \{xy, yz, zx\}$ $\quad V' = \{x, y\}, E' = \{xy, y, x\}$



Proof



Case 2: k > 1. First we illustrate on a special case:

$$\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} \leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}$$

$$\begin{aligned}
\sum_{x,y,z} a_{xy}^{u_1} b_{yz}^{u_2} c_{zx}^{u_3} &= \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}^{u_2} c_{zx}^{u_3}\right) \\
&\leq \sum_{x,y} a_{xy}^{u_1} \left(\sum_z b_{yz}\right)^{u_2} \left(\sum_z c_{zx}\right)^{u_3} \\
&= \sum_{x,y} a_{xy}^{u_1} B_y^{u_2} C_x^{u_3} \\
&\leq \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_y B_y\right)^{u_2} \left(\sum_x C_x\right)^{u_3} \\
&= \left(\sum_{x,y} a_{xy}\right)^{u_1} \left(\sum_{y,z} b_{yz}\right)^{u_2} \left(\sum_{z,x} c_{zx}\right)^{u_3}
\end{aligned}$$

Hölder. Why is $u_2 + u_3 \geq 1$?

Notations: $B_y = \sum_z b_{yz}$, $C_z = \sum_z c_{zx}$

Induction on V', E'

Substitute B_y , C_z

$V = \{x_1, \ldots, x_k\}$,
$E = \{e_1, \ldots, e_m\}$

Proof

Case 2: k > 1. The general proof:

$$\sum_{x_1,\ldots,x_k} \prod_{j=1,m} {a_{j,e_j}}^{u_j} =$$

$V = \{x_1, \ldots, x_k\}$,
$E = \{e_1, \ldots, e_m\}$

Proof

Case 2: k > 1. The general proof:

$$\sum_{x_1,\ldots,x_k} \prod_{j=1,m} {a_{j,e_j}}^{u_j} =$$

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j} \right)$$

$V = \{x_1, \dots, x_k\}$,
$E = \{e_1, \dots, e_m\}$

Proof

Case 2: k > 1. The general proof:

$$\sum_{x_1,\dots,x_k} \prod_{j=1,m} {a_{j,e_j}}^{u_j} =$$

$$= \sum_{x_1,\dots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j} \right)$$

Hölder

$$\leq \sum_{x_1,\dots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\prod_{j:x_k \in e_j} \left(\sum_{x_k} a_{j,e_j} \right)^{u_j} \right)$$

$V = \{x_1, \dots, x_k\}$,
$E = \{e_1, \dots, e_m\}$

Proof

Case 2: k > 1. The general proof:

$$\sum_{x_1,\dots,x_k} \prod_{j=1,m} {a_{j,e_j}}^{u_j} =$$

$$= \sum_{x_1,\dots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j} \right)$$

Hölder

$$\leq \sum_{x_1,\dots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\prod_{j:x_k \in e_j} \left(\sum_{x_k} a_{j,e_j} \right)^{u_j} \right)$$

Notation

$$= \sum_{x_1,\dots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \prod_{j:x_k \in e_j} A_{j,e_j'}^{u_j}$$

$V = \{x_1, \ldots, x_k\}$,
$E = \{e_1, \ldots, e_m\}$

Proof

$V' = \{x_1, \ldots, x_{k-1}\}$,
$E = \{e_1', \ldots, e_m'\}$
where $e_j' = e_j - \{x_k\}$

Case 2: k > 1. The general proof:

$$\sum_{x_1,\ldots,x_k} \prod_{j=1,m} {a_{j,e_j}}^{u_j} =$$

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j} \right)$$

$$\leq \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\prod_{j:x_k \in e_j} \left(\sum_{x_k} a_{j,e_j} \right)^{u_j} \right)$$

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \prod_{j:x_k \in e_j} A_{j,e_j'}^{u_j}$$

$$\leq \prod_{j:x_k \notin e_j} \left(\sum_{e_j} a_{j,e_j} \right)^{u_j} \prod_{j:x_k \in e_j} \left(\sum_{e_j'} A_{j,e_j} \right)^{u_j}$$

Hölder

Notation

Induction on V',E'

$V = \{x_1, \ldots, x_k\}$,
$E = \{e_1, \ldots, e_m\}$

$V' = \{x_1, \ldots, x_{k-1}\}$,
$E = \{e_1{}', \ldots, e_m{}'\}$
where $e_j' = e_j - \{x_k\}$

Proof

Case 2: k > 1. The general proof:

$$\sum_{x_1,\ldots,x_k} \prod_{j=1,m} a_{j,e_j}{}^{u_j} =$$

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j} \right)$$

Hölder

$$\leq \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\prod_{j:x_k \in e_j} \left(\sum_{x_k} a_{j,e_j} \right)^{u_j} \right)$$

Notation

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \prod_{j:x_k \in e_j} A_{j,e_j'}^{u_j}$$

Induction on V',E'

$$\leq \prod_{j:x_k \notin e_j} \left(\sum_{e_j} a_{j,e_j} \right)^{u_j} \prod_{j:x_k \in e_j} \left(\sum_{e_j'} A_{j,e_j} \right)^{u_j}$$

$$= \prod_{j:x_k \notin e_j} \left(\sum_{e_j} a_{j,e_j} \right)^{u_j} \prod_{j:x_k \in e_j} \left(\sum_{e_j'} \sum_{x_k} a_{j,e_j} \right)^{u_j}$$

$V = \{x_1, \ldots, x_k\},$
$E = \{e_1, \ldots, e_m\}$

Proof

$V' = \{x_1, \ldots, x_{k-1}\},$
$E = \{e_1', \ldots, e_m'\}$
where $e_j' = e_j - \{x_k\}$

Case 2: k > 1. The general proof:

$$\sum_{x_1,\ldots,x_k} \prod_{j=1,m} a_{j,e_j}{}^{u_j} =$$

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^{u_j}\right)$$

Hölder

$$\leq \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \left(\prod_{j:x_k \in e_j} \left(\sum_{x_k} a_{j,e_j}\right)^{u_j}\right)$$

Notation

$$= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \notin e_j} a_{j,e_j}^{u_j} \prod_{j:x_k \in e_j} A_{j,e_j'}^{u_j}$$

Induction on V',E'

$$\leq \prod_{j:x_k \notin e_j} \left(\sum_{e_j} a_{j,e_j}\right)^{u_j} \prod_{j:x_k \in e_j} \left(\sum_{e_j'} A_{j,e_j}\right)^{u_j}$$

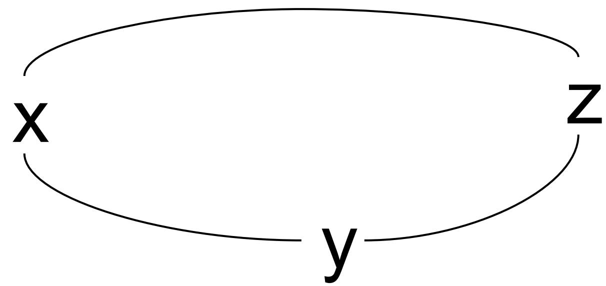
$$= \prod_{j:x_k \notin e_j} \left(\sum_{e_j} a_{j,e_j}\right)^{u_j} \prod_{j:x_k \in e_j} \left(\sum_{e_j'} \sum_{x_k} a_{j,e_j}\right)^{u_j}$$

$$= \prod_{j=1,m} \left(\sum_{e_j} a_{j,e_j}\right)^{u_j}$$

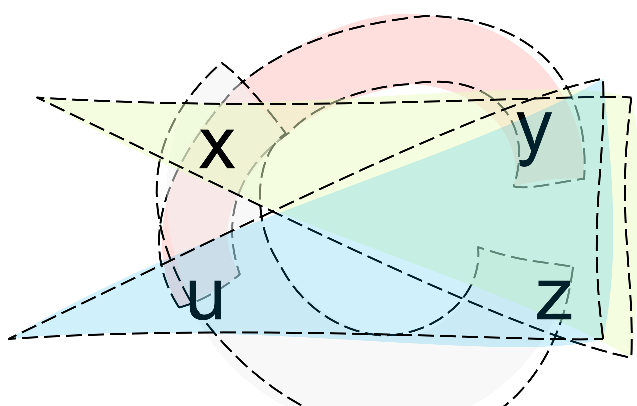
$\sum_{e_j'} \sum_{x\_k} \quad = \sum_{e_j}$

Conjunctive Queries are Hypergraphs

$Q(x, y, z) = R(x, y) \bowtie S(y, z) \bowtie T(z, x)$



$Q(x, y, z) = A(x, y, z) \bowtie B(x, y, u) \bowtie C(x, z, u) \bowtie D(y, z, u)$



Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Special case $R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

Fix instance R, S, T, let n = number of constants; for all $x, y, z \in \{1, \ldots, n\}$ let:

Not so simple Fact #3

$$Q(X_1, \dots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]
Let $u_1, u_2, \dots, u_m$ be *fractional edge cover*, then:

$$|Q| \le |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Special case $R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$
Fix instance R, S, T, let n = number of constants; for all $x, y, z \in \{1, \dots, n\}$ let:

$$a_{xy} = \begin{cases} 1, & (x,y) \in R \\ 0, & \text{otherwise} \end{cases} \qquad b_{yz} = \begin{cases} 1, & (y,z) \in S \\ 0, & \text{otherwise} \end{cases} \qquad c_{zx} = \begin{cases} 1, & (z,x) \in T \\ 0, & \text{otherwise} \end{cases}$$

$$|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx}$$

Not so simple Fact #3

$$Q(X_1, \dots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \dots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Special case $R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

Fix instance R, S, T, let n = number of constants; for all $x, y, z \in \{1, \dots, n\}$ let:

$$a_{xy} = \begin{cases} 1, & (x,y) \in R \\ 0, & \text{otherwise} \end{cases} \qquad b_{yz} = \begin{cases} 1, & (y,z) \in S \\ 0, & \text{otherwise} \end{cases} \qquad c_{zx} = \begin{cases} 1, & (z,x) \in T \\ 0, & \text{otherwise} \end{cases}$$

$$|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} =$$
$$= \sum_{x,y,z} a_{xy}^{1/2}\, b_{yz}^{1/2} c_{zx}^{1/2}$$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Special case $R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

Fix instance R, S, T, let n = number of constants; for all $x, y, z \in \{1, \ldots, n\}$ let:

$$a_{xy} = \begin{cases} 1, & (x,y) \in R \\ 0, & \text{otherwise} \end{cases} \qquad b_{yz} = \begin{cases} 1, & (y,z) \in S \\ 0, & \text{otherwise} \end{cases} \qquad c_{zx} = \begin{cases} 1, & (z,x) \in T \\ 0, & \text{otherwise} \end{cases}$$

$$|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} =$$

$$= \sum_{x,y,z} a_{xy}^{1/2} b_{yz}^{1/2} c_{zx}^{1/2} \leq \left(\sum_{xy} a_{xy}\right)^{1/2} \left(\sum_{yz} b_{yz}\right)^{1/2} \left(\sum_{zx} c_{zx}\right)^{1/2}$$

Not so simple Fact #3

$$Q(X_1, \dots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \dots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Special case $R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

Fix instance R, S, T, let n = number of constants; for all $x, y, z \in \{1, \dots, n\}$ let:

$$a_{xy} = \begin{cases} 1, & (x,y) \in R \\ 0, & \text{otherwise} \end{cases} \qquad b_{yz} = \begin{cases} 1, & (y,z) \in S \\ 0, & \text{otherwise} \end{cases} \qquad c_{zx} = \begin{cases} 1, & (z,x) \in T \\ 0, & \text{otherwise} \end{cases}$$

$$|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} =$$

$$= \sum_{x,y,z} a_{xy}^{1/2} \, b_{yz}^{1/2} c_{zx}^{1/2} \leq \left(\sum_{xy} a_{xy}\right)^{1/2} \left(\sum_{yz} b_{yz}\right)^{1/2} \left(\sum_{zx} c_{zx}\right)^{1/2}$$

$$= |R|^{1/2} |S|^{1/2} \, |T|^{1/2}$$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Let $a_{j,x_{j_1},x_{j_2},\ldots} = 1$ if $(x_{j_1}, x_{j_2}, \ldots) \in R_j$, 0 otherwise.

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Let $a_{j,x_{j_1},x_{j_2},\ldots} = 1$ if $(x_{j_1}, x_{j_2}, \ldots) \in R_j$, 0 otherwise.

$|Q| = \sum_{x_1,\ldots,x_k} a_{1,vars_1} \cdots a_{m,vars_m}$

Not so simple Fact #3

$$Q(X_1, \ldots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \ldots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Let $a_{j,x_{j_1},x_{j_2},\ldots} = 1$ if $\left(x_{j_1}, x_{j_2}, \ldots\right) \in R_j$, 0 otherwise.

$|Q| = \sum_{x_1,\ldots,x_k} a_{1,vars_1} \cdots a_{m,vars_m}$

$= \sum_{x_1,\ldots,x_k} a_{1,vars_1}^{u_1} \cdots a_{m,vars_m}^{u_m}$ // because $a_{j,vars_j} = 0$ or 1

Not so simple Fact #3

$$Q(X_1, \dots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \dots, u_m$ be *fractional edge cover*, then:

$$|Q| \le |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Let $a_{j,x_{j_1},x_{j_2},\dots} = 1$ if $(x_{j_1}, x_{j_2}, \dots) \in R_j$, 0 otherwise.

$$|Q| = \sum_{x_1,\dots,x_k} a_{1,vars_1} \cdots a_{m,vars_m}$$

$$= \sum_{x_1,\dots,x_k} a_{1,vars_1}^{u_1} \cdots a_{m,vars_m}^{u_m}$$ // because $a_{j,vars_j} = 0$ or 1

$$\le \left(\sum_{vars_1} a_{1,vars_1}\right)^{u_1} \cdots \left(\sum_{vars_m} a_{m,vars_m}\right)^{u_m}$$

Not so simple Fact #3

$$Q(X_1, \dots, X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m)$$

Theorem [Atserias,Grohe,Marx]

Let $u_1, u_2, \dots, u_m$ be *fractional edge cover*, then:

$$|Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$$

Proof. Let $a_{j,x_{j_1},x_{j_2},\dots} = 1$ if $\left(x_{j_1}, x_{j_2}, \dots\right) \in R_j$, 0 otherwise.

$|Q| = \sum_{x_1,\dots,x_k} a_{1,vars_1} \cdots a_{m,vars_m}$

$= \sum_{x_1,\dots,x_k} a_{1,vars_1}^{u_1} \cdots a_{m,vars_m}^{u_m}$ // because $a_{j,vars_j} = 0$ or 1

$\leq \left(\sum_{vars_1} a_{1,vars_1}\right)^{u_1} \cdots \left(\sum_{vars_m} a_{m,vars_m}\right)^{u_m}$

$= |R_1|^{u_1} \times \cdots \times |R_m|^{u_m}$

Announcements

This week:

- Big HW3 is due on Friday!
- No paper review, no project task

Will read your project proposals soon

No class next Monday: Presidents Day

Review: Upper Bound

Set semantics!

Assume $|R|, |S|, |T|, |K| \leq N$

$$Q(x, y, z, u) = R(x, y, z) \bowtie S(x, y, u) \bowtie T(x, z, u) \bowtie K(y, z, u)$$

Review: Upper Bound

Set semantics!

Assume $|R|, |S|, |T|, |K| \leq N$

$$Q(x, y, z, u) = R(x, y, z) \bowtie S(x, y, u) \bowtie T(x, z, u) \bowtie K(y, z, u)$$

Fact #1: $|Q| \leq |R| \cdot |S| \cdot |T| \cdot |K| \leq N^4$

Review: Upper Bound

Set semantics!

Assume $|R|, |S|, |T|, |K| \leq N$

$$Q(x, y, z, u) = R(x, y, z) \bowtie S(x, y, u) \bowtie T(x, z, u) \bowtie K(y, z, u)$$

Fact #1: $|Q| \leq |R| \cdot |S| \cdot |T| \cdot |K| \leq N^4$

Fact #2: $|Q| \leq |R| \cdot |S| \leq N^2$

Review: Upper Bound

Set semantics!

Assume $|R|, |S|, |T|, |K| \leq N$

$$Q(x,y,z,u) = R(x,y,z) \bowtie S(x,y,u) \bowtie T(x,z,u) \bowtie K(y,z,u)$$

Fact #1: $|Q| \leq |R| \cdot |S| \cdot |T| \cdot |K| \leq N^4$

Fact #2: $|Q| \leq |R| \cdot |S| \leq N^2$

better: $|Q| \leq \min(|R| \cdot |S|, |R| \cdot |T|, \ldots, |T| \cdot |K|)$

Review: Upper Bound

Set semantics!

Assume $|R|, |S|, |T|, |K| \leq N$

$$Q(x,y,z,u) = R(x,y,z) \bowtie S(x,y,u) \bowtie T(x,z,u) \bowtie K(y,z,u)$$

Fact #1: $|Q| \leq |R| \cdot |S| \cdot |T| \cdot |K| \leq N^4$

Fact #2: $|Q| \leq |R| \cdot |S| \leq N^2$

better: $|Q| \leq \min(|R| \cdot |S|, |R| \cdot |T|, \ldots, |T| \cdot |K|)$

Fact #3: $|Q| \leq |R|^{1/3} \cdot |S|^{1/3} \cdot |T|^{1/3} \cdot |K|^{1/3} \leq N^{4/3}$

AGM Bound

Review: Friedgut's Inequality

Interesting special case

$$\sum_{i,j,k} a_{ij}^{1/2} b_{jk}^{1/2} c_{ki}^{1/2} \leq \left(\sum_{i,j} a_{ij}\right)^{1/2} \left(\sum_{j,k} b_{jk}\right)^{1/2} \left(\sum_{k,i} c_{ki}\right)^{1/2}$$

Hypergraph $V = \{x_1, \dots, x_k\},\ \ E = \{e_1, \dots, e_m\}$

Fractional edge cover: $u_1, u_2, \dots, u_m$

$$\sum_{x_1,\dots,x_k} a_{1,e_1}^{u_1} \cdots a_{m,e_m}^{u_m} \leq \left(\sum_{e_1} a_{1,e_1}\right)^{u_1} \cdots \left(\sum_{e_m} a_{m,e_m}\right)^{u_m}$$

Extension: Keys

$R(X, Y) \bowtie S(Y, Z)$ $|R|, |S| \leq N$

- No other info: $|Q(D)| \leq N^2$
- S.Y is a key: $|Q(D)| \leq N$

The <u>*Query Extension*</u> method:

- If Y is a key in some relation S, then add all attributes of S relations containing Y
- Compute $AGM(Q^{ext})$

Example

Y is a key

$$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$$

Example

Y is a key

$Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z)$

- $Q^{ext}(X, Y, Z) = R(X, Y, Z) \wedge S(Y, Z),$

Example

Y is a key

$Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z)$

- $Q^{ext}(X, Y, Z) = R(X, Y, Z) \wedge S(Y, Z)$,
- Edge cover: 1,0
- $AGM(Q^{ext}) = |R|$

Example

Y is a key

$Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z)$

- $Q^{ext}(X, Y, Z) = R(X, Y, Z) \wedge S(Y, Z)$,
- Edge cover: 1,0
- $AGM(Q^{ext}) = |R|$

$Q(X, Y, Z) = R(X, Y) \wedge S(Y, Z) \wedge T(Z, X)$

Example

Y is a key

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z),$
- Edge cover: 1,0
- $AGM(Q^{ext}) = |R|$

$Q(X,Y,Z) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,X)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z) \wedge T(Z,X)$

Example

Y is a key

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z)$,
- Edge cover: 1,0
- $AGM(Q^{ext}) = |R|$

$Q(X,Y,Z) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,X)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z) \wedge T(Z,X)$
- Edge covers: 1,0,0 or 0,1,1

Example

Y is a key

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z)$,
- Edge cover: 1,0
- $AGM(Q^{ext}) = |R|$

$Q(X,Y,Z) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,X)$

- $Q^{ext}(X,Y,Z) = R(X,Y,Z) \wedge S(Y,Z) \wedge T(Z,X)$
- Edge covers: 1,0,0 or 0,1,1
- $AGM(Q^{exp}) = \min(|R|, |S| \times |T|)$

Equal Cardinalities

If $|R_1|, |R_2|, \cdots |R_m| \leq N$

then: $|Q| \leq |R_1|^{u_1} \cdots |R_m|^{u_m} \leq N^{u_1+u_2+\cdots+u_m}$

- $\rho^* \stackrel{\text{def}}{=} \min_{fract\ edge\ cover} (u_1 + \cdots + u_m)$

Simplified AGM bound: $|Q| \leq N^{\rho^*}$

Tightness

- There exists instances $R_1, R_2, \ldots$ such that the size of the query's output is AGM(Q)
- Proof is simple and instructive; we will show for special case $|R_1| = \cdots = |R_m| = N$
- In this case $AGM(Q) = N^{\rho^*}$

Fractional Vertex Packing

- A *fractional vertex packing* of a (hyper)graph is a set of non-negative numbers v_x, one for each node x, such that, for every edge e: $\sum_{x:x \in e} v_x \leq 1$

Fractional Vertex Packing

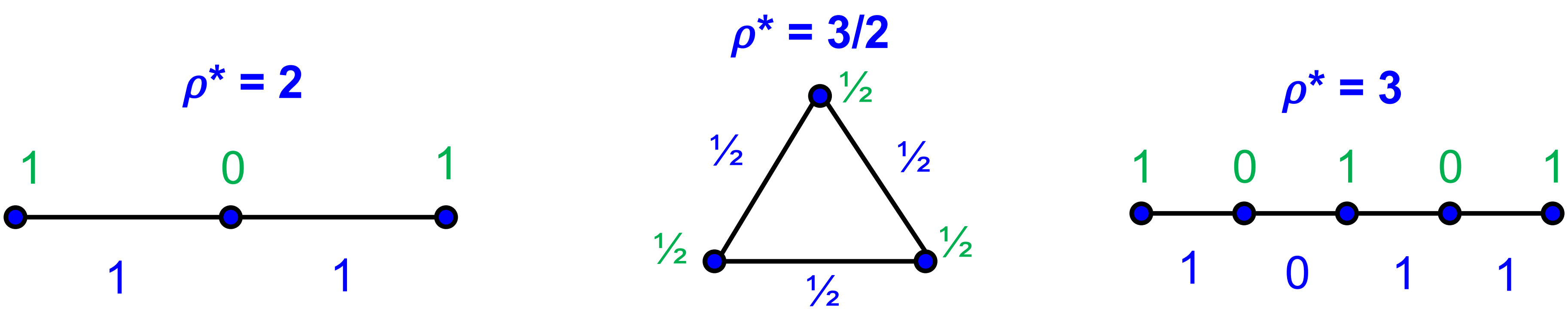
- A *fractional vertex packing* of a (hyper)graph is a set of non-negative numbers v_x, one for each node x, such that, for every edge e: $\sum_{x:x \in e} v_x \leq 1$

Theorem $\max \sum_x v_x = \rho^* = \min \sum_e u_e$

Fractional Vertex Packing

- A <u>*fractional vertex packing*</u> of a (hyper)graph is a set of non-negative numbers v_x, one for each node x, such that, for every edge e: $\sum_{x: x \in e} v_x \leq 1$

Theorem $\max \sum_x v_x = \rho^* = \min \sum_e u_e$



The Bound is Tight

Fact For any fractional vertex packing v_x, there exists a database instance such that $|R_1| \leq N, \ldots, |R_m| \leq N$ and $|Q| = N^{\sum_x v_x}$

In particular, there exists an instance s.t. $|Q| = N^{\rho^*}$

The Bound is Tight

Fact For any fractional vertex packing v_x, there exists a database instance such that $|R_1| \leq N$, …, $|R_m| \leq N$ and $|Q| = N^{\Sigma_x v_x}$

In particular, there exists an instance s.t. $|Q| = N^{\rho^*}$

Proof.

For each variable x_i: $D_i \stackrel{\text{def}}{=} [N^{v_{x_i}}] = \{1,2, \ldots, N^{v_{x_i}}\}$

The Bound is Tight

Fact For any fractional vertex packing v_x, there exists a database instance such that $|R_1| \leq N$, …, $|R_m| \leq N$ and $|Q| = N^{\sum_x v_x}$

In particular, there exists an instance s.t. $|Q| = N^{\rho^*}$

Proof.

For each variable x_i: $D_i \stackrel{\text{def}}{=} [N^{v_{x_i}}] = \{1,2, \ldots, N^{v_{x_i}}\}$

For each relation R_j: $|R_j(x_{i_1}, x_{i_2}, \ldots)| \stackrel{\text{def}}{=} D_{i_1} \times D_{i_2} \times \cdots$

The Bound is Tight

Fact For any fractional vertex packing v_x, there exists a database instance such that $|R_1| \leq N$, …, $|R_m| \leq N$ and $|Q| = N^{\sum_x v_x}$

In particular, there exists an instance s.t. $|Q| = N^{\rho^*}$

Proof.

For each variable x_i: $D_i \stackrel{\text{def}}{=} [N^{v_{x_i}}] = \{1,2,\ldots,N^{v_{x_i}}\}$

For each relation R_j: $|R_j(x_{i_1}, x_{i_2}, \ldots)| \stackrel{\text{def}}{=} D_{i_1} \times D_{i_2} \times \cdots$

(a) $|R_j| = N^{v_{i_1}+v_{i_2}+\cdots} \leq N$ (why?)

The Bound is Tight

Fact For any fractional vertex packing v_x, there exists a database instance such that $|R_1| \leq N, \ldots, |R_m| \leq N$ and $|Q| = N^{\Sigma_x v_x}$

In particular, there exists an instance s.t. $|Q| = N^{\rho^*}$

Proof.

For each variable x_i: $D_i \stackrel{\text{def}}{=} [N^{v_{x_i}}] = \{1,2, \ldots, N^{v_{x_i}}\}$

For each relation R_j: $|R_j(x_{i_1}, x_{i_2}, \ldots)| \stackrel{\text{def}}{=} D_{i_1} \times D_{i_2} \times \cdots$

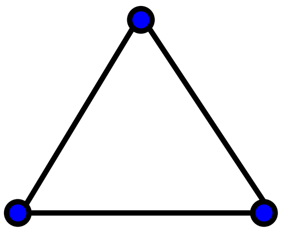
(a) $|R_j| = N^{v_{i_1}+v_{i_2}+\cdots} \leq N$ (why?)

(b) $|Q| = N^{\Sigma_x v_x}$ (why?)

Example 1

$|R|, |S|, |T| \leq N$

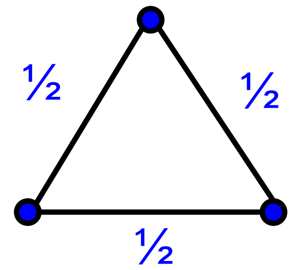
$Q(x, y, z) = R(x, y) \bowtie S(y, z) \bowtie T(z, x)$



Example 1

$|R|, |S|, |T| \leq N$

$Q(x, y, z) = R(x, y) \bowtie S(y, z) \bowtie T(z, x)$



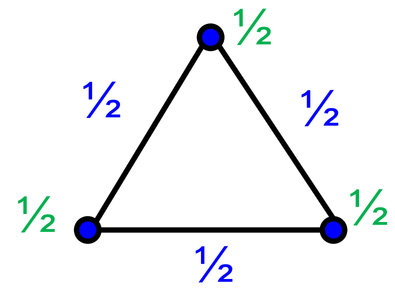
We know $|Q| \leq N^{3/2}$

Find an instance where $|Q| = N^{3/2}$

Example 1

$|R|, |S|, |T| \leq N$

$Q(x, y, z) = R(x, y) \bowtie S(y, z) \bowtie T(z, x)$



We know $|Q| \leq N^{3/2}$

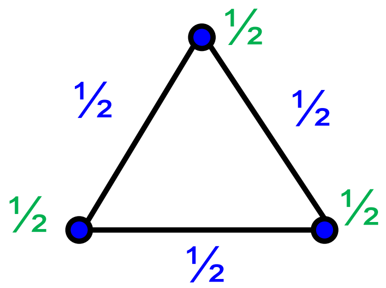
Find an instance where $|Q| = N^{3/2}$

Answer: $D_x = D_y = D_z \stackrel{\text{def}}{=} [N^{1/2}]$

Example 1

$|R|, |S|, |T| \leq N$

$Q(x,y,z) = R(x,y) \bowtie S(y,z) \bowtie T(z,x)$



We know $|Q| \leq N^{3/2}$

Find an instance where $|Q| = N^{3/2}$

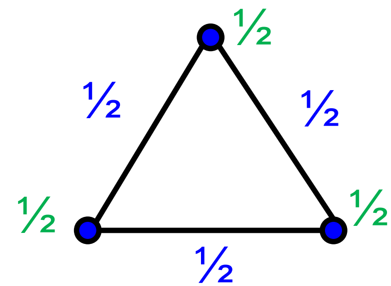
Answer: $D_x = D_y = D_z \stackrel{\text{def}}{=} [N^{1/2}]$

$R(x,y) \stackrel{\text{def}}{=} D_x \times D_y$, $\quad S(y,z) \stackrel{\text{def}}{=} D_y \times D_z$, $\quad T(z,x) \stackrel{\text{def}}{=} D_z \times D_x$

Example 1

$|R|, |S|, |T| \leq N$

$Q(x, y, z) = R(x, y) \bowtie S(y, z) \bowtie T(z, x)$



We know $|Q| \leq N^{3/2}$

Find an instance where $|Q| = N^{3/2}$

Answer: $D_x = D_y = D_z \stackrel{\text{def}}{=} [N^{1/2}]$

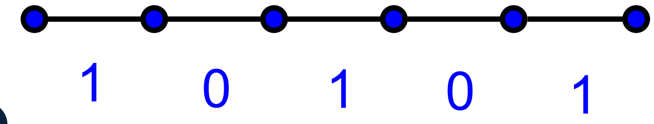
$R(x, y) \stackrel{\text{def}}{=} D_x \times D_y$, $S(y, z) \stackrel{\text{def}}{=} D_y \times D_z$, $T(z, x) \stackrel{\text{def}}{=} D_z \times D_x$

Then: $Q(x, y, z) = D_x \times D_y \times D_z$

Example 2

$|R|, |S|, |T|, |K|, |L| \leq N$

$$Q(x, y, z, u, v, w) = R(x, y) \bowtie S(y, z) \bowtie T(z, u) \bowtie K(u, v) \bowtie L(v, w)$$



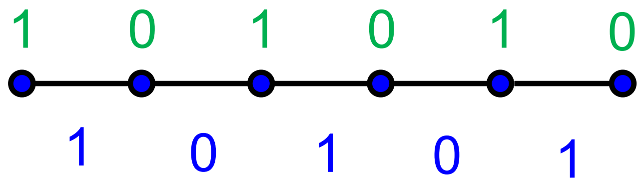
Example 2

$|R|, |S|, |T|, |K|, |L| \leq N$

$Q(x, y, z, u, v, w) = R(x, y) \bowtie S(y, z) \bowtie T(z, u) \bowtie K(u, v) \bowtie L(v, w)$

$|Q| \leq N^3$

Find an instance where $|Q| = N^3$



Example 2

$|R|, |S|, |T|, |K|, |L| \leq N$

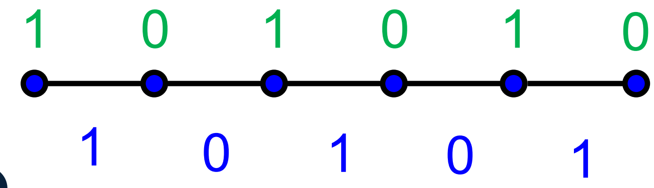
$Q(x, y, z, u, v, w) = R(x, y) \bowtie S(y, z) \bowtie T(z, u) \bowtie K(u, v) \bowtie L(v, w)$

$|Q| \leq N^3$

Find an instance where $|Q| = N^3$

Answer:

$D_x \stackrel{\text{def}}{=} [N],\ D_y \stackrel{\text{def}}{=} [1], D_z \stackrel{\text{def}}{=} [N],\ D_u \stackrel{\text{def}}{=} [1], D_v \stackrel{\text{def}}{=} [N],\ D_w \stackrel{\text{def}}{=} [1]$



Example 2

$|R|, |S|, |T|, |K|, |L| \leq N$

$$Q(x, y, z, u, v, w) = R(x, y) \bowtie S(y, z) \bowtie T(z, u) \bowtie K(u, v) \bowtie L(v, w)$$

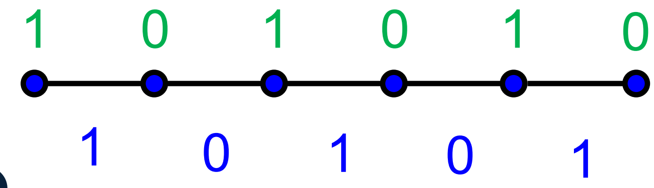
$|Q| \leq N^3$

Find an instance where $|Q| = N^3$

Answer:

$D_x \stackrel{\text{def}}{=} [N],\ D_y \stackrel{\text{def}}{=} [1],\ D_z \stackrel{\text{def}}{=} [N],\ D_u \stackrel{\text{def}}{=} [1],\ D_v \stackrel{\text{def}}{=} [N],\ D_w \stackrel{\text{def}}{=} [1]$

$R(x, y) \stackrel{\text{def}}{=} [N] \times [1],\ S(y, z) \stackrel{\text{def}}{=} [1] \times [N],\ T(z, u) \stackrel{\text{def}}{=} [N] \times [1], \ldots$



Example 2

$|R|, |S|, |T|, |K|, |L| \leq N$

$Q(x, y, z, u, v, w) = R(x, y) \bowtie S(y, z) \bowtie T(z, u) \bowtie K(u, v) \bowtie L(v, w)$

$|Q| \leq N^3$

Find an instance where $|Q| = N^3$

Answer:

$D_x \stackrel{\text{def}}{=} [N],\ D_y \stackrel{\text{def}}{=} [1], D_z \stackrel{\text{def}}{=} [N],\ D_u \stackrel{\text{def}}{=} [1], D_v \stackrel{\text{def}}{=} [N],\ D_w \stackrel{\text{def}}{=} [1]$

$R(x, y) \stackrel{\text{def}}{=} [N]\times[1], S(y, z) \stackrel{\text{def}}{=} [1]\times[N], T(z, u) \stackrel{\text{def}}{=} [N]\times[1]$, ...

Then: $Q(x, y, z) = [N]\times[1]\times[N]\times[1]\times[N]\times[1]$

Outline

- AGM bound
- Worst-case optimal join algorithm
- Acyclic queries, Yannakakis algorithm
- Tree decomposition of cyclic queries

Next

Later…
(next week)

Motivation

Multijoin query: $Q = R_1 \bowtie R_2 \bowtie \cdots$

Goal: compute in time $\tilde{O}(AGM(Q))$

$\tilde{O}(\ldots)$ means "times $\log(N)$"

Motivation

Multijoin query: $Q = R_1 \bowtie R_2 \bowtie \cdots$

Goal: compute in time $\tilde{O}(AGM(Q))$

$\tilde{O}(\dots)$ means "times $\log(N)$"

Why non-trivial: $Q = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

- When $R = S = T = \left(\left[\frac{N}{2}\right] \times [1]\right) \cup \left([1] \times \left[\frac{N}{2}\right]\right)$
- $|R| = |S| = |T| = N$ but Any query plan takes time $O(N^2)$, because of intermediate relations:

$$|R \bowtie S| = |S \bowtie T| = |R \bowtie T| = \left[\frac{N^2}{4}\right]$$

- Yet $|Q| = 1$

History

- Worst-Case-Optimal-Join Algorithm (WCOJ)
- First by Ngo, Porat, Re, Rudra in 2012
  - "NPRR algorithm"
  - Very complicated
- Veldhuizen'2014:
  - "Leapfrog-Tree-Join" (LFTJ)
  - Had been implemented by Logicblox much earlier
- Ngo, Re, Rudra 2013:
  - Simplified further; "Generic Join" (GJ)
- Today: WCOJ or LFTJ or GJ mean same thing

Generic Join Algorithm

Let x be any variable
Let R_{i_1}, R_{i_2}, ... be all relations containing x
compute $D = \Pi_x(R_{i_1}) \cap \Pi_x(R_{i_2}) \cap \cdots$
for every value $v \in D$ **do**:
    compute $Q_{x=v}$,
    where R_{i_1}, R_{i_2}, ... are restricted to $x = v$

Generic Join Example

$Q(x, y, z) \; = \; R(x, y) \wedge S(y, z) \wedge T(z, x),$

Generic Join Example

$Q(x, y, z) \; = \; R(x, y) \wedge S(y, z) \wedge T(z, x),$

$A \; = \; \Pi_x(R(x, y)) \; \cap \; \Pi_x(T(z, x))$

Generic Join Example

$Q(x, y, z) = R(x, y) \wedge S(y, z) \wedge T(z, x)$,

$A = \Pi_x(R(x, y)) \cap \Pi_x(T(z, x))$

for a in A **do**

/* compute $Q(a, y, z) = R(a, y) \wedge S(y, z) \wedge T(z, a)$ */

$B = \Pi_y(R(a, y)) \cap \Pi_y(S(y, z))$

Generic Join Example

$Q(x, y, z) = R(x, y) \wedge S(y, z) \wedge T(z, x)$,

$A = \Pi_x(R(x, y)) \cap \Pi_x(T(z, x))$

for a in A **do**

/* compute $Q(a, y, z) = R(a, y) \wedge S(y, z) \wedge T(z, a)$ */

$B = \Pi_y(R(a, y)) \cap \Pi_y(S(y, z))$

for b in B **do**

/* compute $Q(a, b, z) = R(a, b) \wedge S(b, z) \wedge T(z, a)$ */

Generic Join Example

$Q(x, y, z) = R(x, y) \wedge S(y, z) \wedge T(z, x)$,

$A = \Pi_x(R(x, y)) \cap \Pi_x(T(z, x))$
for a in A **do**
    /* compute $Q(a, y, z) = R(a, y) \wedge S(y, z) \wedge T(z, a)$ */
    $B = \Pi_y(R(a, y)) \cap \Pi_y(S(y, z))$
    for b in B **do**
    /* compute $Q(a, b, z) = R(a, b) \wedge S(b, z) \wedge T(z, a)$ */
        $C = \Pi_z(S(b, z)) \cap \Pi_z(T(z, a))$
        for c in C **do**
            output (a,b,c)

Generic Join Example

$Q(x, y, z) = R(x, y) \wedge S(y, z) \wedge T(z, x)$,

Runs in time $\tilde{O}\big(AGM(Q)\big) = \tilde{O}(N^{3/2})$

$A = \Pi_x(R(x, y)) \cap \Pi_x(T(z, x))$
for a in A **do**
  /* compute $Q(a, y, z) = R(a, y) \wedge S(y, z) \wedge T(z, a)$ */
  $B = \Pi_y(R(a, y)) \cap \Pi_y(S(y, z))$
  for b in B **do**
  /* compute $Q(a, b, z) = R(a, b) \wedge S(b, z) \wedge T(z, a)$ */
    $C = \Pi_z(S(b, z)) \cap \Pi_z(T(z, a))$
    for c in C **do**
      output (a,b,c)

Generic Join: Intersection

Intersection is the main building block of G.J.

- $Q(X) = R(X) \bowtie S(X)$
- What is $AGM(Q)$?

Generic Join: Intersection

Intersection is the main building block of G.J.

- $Q(X) = R(X) \bowtie S(X)$
- What is $AGM(Q)$?
  - Edge covers of Q: 1,0 and 0,1;
  - $AGM(Q) = \min(|R|, |S|)$

Generic Join: Intersection

Intersection is the main building block of G.J.

- $Q(X) = R(X) \bowtie S(X)$
- What is $AGM(Q)$?
  - Edge covers of Q: 1,0 and 0,1;
  - $AGM(Q) = \min(|R|, |S|)$
- Assume R, S are already sorted
  - Merge-join – what is runtime?

Generic Join: Intersection

Intersection is the main building block of G.J.

- $Q(X) = R(X) \bowtie S(X)$
- What is $AGM(Q)$?
  - Edge covers of Q: 1,0 and 0,1;
  - $AGM(Q) = \min(|R|, |S|)$
- Assume R, S are already sorted
  - Merge-join – what is runtime? runtime $= O(|R| + |S|)$
  - Improved merge-join: runtime $= \tilde{O}(\min(|R|, |S|))$

Generic Join Algorithm

Assume all relations are pre-sorted

Let x be any variable

Let $R_{i_1}, R_{i_2}, \ldots$ be all relations containing x

compute $D = \Pi_x(R_{i_1}) \cap \Pi_x(R_{i_2}) \cap \cdots$

for every value $v \in D$ **do**:

compute $Q_{x=v}$,

where $R_{i_1}, R_{i_2}, \ldots$ are restricted to $x = v$

needs to be done in time $\tilde{O}(min_j \Pi_x(R_j))$

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Case 1: k=1 Then GJ is intersection

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Case 1: k=1 Then GJ is intersection

Case 2: k>1 Assume domain of x is $|D| = n$

$$Time(Q) = \sum_{v=1,n} Time(Q_{x=v})$$

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Case 1: k=1 Then GJ is intersection

Case 2: k>1 Assume domain of x is $|D| = n$

By induction

$$Time(Q) = \sum_{v=1,n} Time(Q_{x=v}) =$$

$$= \tilde{O}\left(\left(\sum_{v=1,n} \prod_{j:x \in vars(R_j)} \left|R_{j,x=v}\right|^{u_j}\right) \prod_{j:x \notin vars(R_j)} \left|R_j\right|^{u_j}\right)$$

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Case 1: k=1 Then GJ is intersection

Case 2: k>1 Assume domain of x is $|D| = n$

$$Time(Q) = \sum_{v=1,n} Time(Q_{x=v}) =$$

$$= \tilde{O}\left(\left(\sum_{v=1,n} \prod_{j:x \in vars(R_j)} \left|R_{j,x=v}\right|^{u_j}\right) \prod_{j:x \notin vars(R_j)} \left|R_j\right|^{u_j}\right)$$

$$\leq \tilde{O}\left(\prod_{j:x \in vars(R_j)} \left(\sum_{v=1,n} \left|R_{j,x=v}\right|\right)^{u_j} \prod_{j:x \notin vars(R_j)} \left|R_j\right|^{u_j}\right)$$

By induction

Friedgut

Analysis of Generic Join

Theorem. Assume all relations are pre-sorted. Then runtime of GJ is $\tilde{O}(AGM(Q))$

Proof: Fix any edge cover u_1, u_2 ...

We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \cdots)$

Case 1: k=1 Then GJ is intersection

Case 2: k>1 Assume domain of x is $|D| = n$

$$Time(Q) = \sum_{v=1,n} Time(Q_{x=v}) =$$

$$= \tilde{O}\left(\left(\sum_{v=1,n} \prod_{j:x \in vars(R_j)} \left|R_{j,x=v}\right|^{u_j}\right) \prod_{j:x \notin vars(R_j)} |R_j|^{u_j}\right)$$

$$\leq \tilde{O}\left(\prod_{j:x \in vars(R_j)} \left(\sum_{v=1,n} \left|R_{j,x=v}\right|\right)^{u_j} \prod_{j:x \notin vars(R_j)} |R_j|^{u_j}\right)$$

$$= \tilde{O}\left(\prod_j |R_j|^{u_j}\right)$$

By induction

Friedgut

Discussion

- All relations need to be presorted, or indexed
- Runtime is guaranteed to be worst-case optimal, *no matter* what variable order we choose
- In practice, the variable order does matter, but how exactly is poorly understood to date

Comparison to Naïve Nested Loop

Naïve nested loop:

```
// tuple at a time:
For t1 in R1 do
  for t2 in R2 do
    …
```

Comparison to Naïve Nested Loop

Naïve nested loop:

```
// tuple at a time:
For t1 in R1 do
  for t2 in R2 do
    …

// value at a time:
For x in Domain do
  For y in Domain do
    For z in Domain do
      ...
```

Comparison to Naïve Nested Loop

Naïve nested loop:

```
// tuple at a time:
For t1 in R1 do
  for t2 in R2 do
    …

// value at a time:
For x in Domain do
  For y in Domain do
    For z in Domain do
      ...
```

Generic-join

```
A = ∩ domains for x
For x in A do
   B = ∩ domains for y
   For y in B do
      C = ∩ domains for z
      For z in C do
         ...
```

An Application

- Fix a relational instance $R(X_1, \ldots, X_k)$
- Let $V_1 \cup \cdots \cup V_\ell$ be a partition of the variables. Then:

$$R \subseteq \Pi_{V_1}(R) \bowtie \cdots \bowtie \Pi_{V_\ell}(R)$$

- A *join dependency* is a partition where this is an equality
- Application to schema design

Relational Schema Design

Name	SSN	Phone	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield

One person may have multiple phones, but lives in only one city

Primary key is thus (SSN, PhoneNumber)

What is the problem with this schema?

Anomalies

Name	SSN	Phone	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield

- Redundancy = repeat data for Fred
- Update anomalies = what if Fred moves to "Bellevue"?
- Deletion anomalies = what if Joe deletes his phone number?

Relation Decomposition

Break the relation into two:

Name	SSN	Phone	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield

Name	SSN	City
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

SSN	Phone
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121

The relation satisfies a join dependency!

$$R(\text{Name,SSN,Phone,City}) = R(\text{Name,SSN,City}) \bowtie R(\text{SSN,Phone})$$

Problem

- Given the instance $R(X_1, \ldots, X_k)$
- Check if there exists a JD:

$$R = \Pi_{V_1}(R) \bowtie \cdots \bowtie \Pi_{V_\ell}(R)$$

- Notice: we don't ask to find it, only *check if one exists*

Solution

- **Fact.** $R(X_1, \ldots, X_k)$ satisfies some JD iff
$$R = R_1 \bowtie \cdots \bowtie R_k$$
where $R_i = \Pi_{\{X_1,\ldots,X_k\}-\{X_i\}}(R)$

- **Solution**: compute $Q \stackrel{\text{def}}{=} R_1 \bowtie \cdots \bowtie R_k$ and check if $|Q| = |R|$

- Runtime: $AGM(|Q|) = N^{\frac{k}{k-1}}$

Final Takeaways

- Useful beyond 544:
  - fractional edge cover/ vertex packing;
  - inequalities
- The AGM bound
  - Simple intuition based on “covers”
  - Useful recipe to compute a “bad” instance based on “packings”
- Generic Join:
  - Best choice for cyclic queries