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**

<table>
<thead>
<tr>
<th></th>
<th>median</th>
<th>90th</th>
<th>95th</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgresQL</td>
<td>1.00</td>
<td>2.08</td>
<td>6.10</td>
<td>207</td>
</tr>
<tr>
<td>DBMS A</td>
<td>1.01</td>
<td>1.33</td>
<td>1.98</td>
<td>43.4</td>
</tr>
<tr>
<td>DBMS B</td>
<td>1.00</td>
<td>6.03</td>
<td>30.2</td>
<td>104000</td>
</tr>
<tr>
<td>DBMS C</td>
<td>1.06</td>
<td>1677</td>
<td>5367</td>
<td>20471</td>
</tr>
<tr>
<td>HyPer</td>
<td>1.02</td>
<td>4.47</td>
<td>8.00</td>
<td>2084</td>
</tr>
</tbody>
</table>

**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)
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

[Graph showing comparisons between PostgreSQL estimates and true cardinalities]
Yet Another Difficulties

• SQL Queries are often issued from applications
• Optimized once using *prepare* statement, executed often
• The constants in the query are not known 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
Jayant Haritsa, ICDE’2019 tutorial

Different optimal plans for different C1, C2.
Query Plans
Figure 9: Cost distributions for 5 queries and different index configurations. The vertical green lines represent the cost of the optimal plan.
<table>
<thead>
<tr>
<th></th>
<th>PK indexes</th>
<th></th>
<th></th>
<th>PK + FK indexes</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>95%</td>
<td>max</td>
<td>median</td>
<td>95%</td>
<td>max</td>
</tr>
<tr>
<td>zig-zag</td>
<td>1.00</td>
<td>1.06</td>
<td>1.33</td>
<td>1.00</td>
<td>1.60</td>
<td>2.54</td>
</tr>
<tr>
<td>left-deep</td>
<td>1.00</td>
<td>1.14</td>
<td>1.63</td>
<td>1.06</td>
<td>2.49</td>
<td>4.50</td>
</tr>
<tr>
<td>right-deep</td>
<td>1.87</td>
<td>4.97</td>
<td>6.80</td>
<td>47.2</td>
<td>30931</td>
<td>738349</td>
</tr>
</tbody>
</table>

Table 2: Slowdown for restricted tree shapes in comparison to the optimal plan (true cardinalities)
<table>
<thead>
<tr>
<th></th>
<th>PK indexes</th>
<th>PK + FK indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PostgreSQL estimates</td>
<td>true cardinalities</td>
</tr>
<tr>
<td></td>
<td>median</td>
<td>95%</td>
</tr>
<tr>
<td>Dynamic Programming</td>
<td>1.03</td>
<td>1.85</td>
</tr>
<tr>
<td>Quickpick-1000</td>
<td>1.05</td>
<td>2.19</td>
</tr>
<tr>
<td>Greedy Operator Ordering</td>
<td>1.19</td>
<td>2.29</td>
</tr>
</tbody>
</table>

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
- \(|R|, |S| \leq N\)
- 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
• $|R|, |S| \leq N$
• 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

• $|R|, |S| \leq N$
• 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:

\[ |Q(D)| \leq N \]
Fix input statistics for D

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

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

No other info:

$|Q(D)| \leq N^2$

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

- S.Y has degree $\leq d$:
  
  $|Q(D)| \leq d \times N$
Fix input statistics for D

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

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

No other info:

$$|Q(D)| \leq N^2$$

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

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

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

No other info:
Upper Bounds

Fix input statistics for D

- $|R|, |S| \leq N$
- 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 $\leq d$: $|Q(D)| \leq d \times N$

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

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

Fix input statistics for $D$

- $|R|, |S| \leq N$
- 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 $\leq d$: $|Q(D)| \leq d \times N$

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

No other info: $|Q(D)| \leq N^{3/2}$
Simple Fact #1

$$Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m)$$

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

\[ Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m) \]

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:

\[ |Q| \leq |R_1|^{u_1} \times \cdots \times |R_m|^{u_m} \]
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 $e \in E$ is a set of two nodes.
Graphs and Hypergraphs

- An undirected graph $G = (V, E)$ where each edge $e \in E$ 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 *edge cover* 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

\[
\sum_i a_i^{1/2} b_i^{1/2} \leq (\sum_i a_i)^{1/2} (\sum_i b_i)^{1/2}
\]

\[a_i \geq 0, \text{ etc}\]
Inequalities

Cauchy-Schwartz

\[ \sum a_i^{1/2} b_i^{1/2} \leq (\sum a_i)^{1/2} (\sum b_i)^{1/2} \]

\[ a_i \geq 0, \text{ etc} \]

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

\[ \sum a_i^{u_1} b_i^{u_2} c_i^{u_3} \leq (\sum a_i)^{u_1} (\sum b_i)^{u_2} (\sum c_i)^{u_3} \]
Inequalities

Cauchy-Schwartz

$$\sum a_i^{1/2} b_i^{1/2} \leq (\sum a_i)^{1/2} (\sum b_i)^{1/2}$$

$$a_i \geq 0$$, etc

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

$$\sum a_i^{u_1} b_i^{u_2} c_i^{u_3} \leq (\sum a_i)^{u_1} (\sum b_i)^{u_2} (\sum c_i)^{u_3}$$

Friedgut 2004

$$\sum a_{ij}^{1/2} b_{jk}^{1/2} c_{ki}^{1/2} \leq (\sum a_{ij})^{1/2} (\sum b_{jk})^{1/2} (\sum c_{ki})^{1/2}$$
Let $G=(V,E)$ be a hypergraph, where:

\[ V = \{x_1, \ldots, x_k\}, \quad E = \{e_1, \ldots, e_m\} \]

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

\[
\sum_{x_1, \ldots, x_k} a_{\text{index}}^{1 \cdot e_1} \cdots a_{\text{index}}^{m \cdot e_m} \leq \left( \sum_{e_1} a_{\text{index}, e_1} \right)^{u_1 \cdot \text{index}} \cdots \left( \sum_{e_m} a_{\text{index}, e_m} \right)^{u_m \cdot \text{index}}
\]

Here, $a_{\text{index}, e_{xyz}}$ is a tensor; similarly $a_{2, \text{etc.}}$. Example: think of $a_{1,xy}, a_{2,yz}, a_{3,zx}$ as three matrices, like $a_{xy}, b_{yz}, c_{zx}$:
Proof

\[ V = \{x_1, \ldots, x_k\}, \quad E = \{e_1, \ldots, e_m\} \]

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

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
Proof

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

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

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

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

\[
\sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \sum_{x,y} a_{xy} (\sum_{z} b_{yz} c_{zx})^u_3
\]
Case 2: $k > 1$. First we illustrate on a special case:

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

Hölder. Why is $u_2 + u_3 \geq 1$?
Proof

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

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

$$\sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \sum_{x,y} a_{xy} (\sum_{z} b_{yz} c_{zx})^{u_3} \leq \sum_{x,y} a_{xy} (\sum_{z} b_{yz})^{u_2} (\sum_{z} c_{zx})^{u_3} = \sum_{x,y} a_{xy} B_{y}^{u_2} C_{x}^{u_3}$$

Hölder. Why is $u_2 + u_3 \geq 1$?

Notations:
$B_{y} = \sum_{z} b_{yz}$, $C_{z} = \sum_{z} c_{zx}$
Proof

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

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

\[
\sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \sum_{x,y} a_{xy} (\sum_{z} b_{yz} c_{zx})^{u_3}
\]

Hölder. Why is $u_2 + u_3 \geq 1$?

Notations:
$B_y = \sum_{z} b_{yz}$, $c_z = \sum_{z} c_{zx}$

Induction on $V'$, $E'$
Proof

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

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

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

Hölder. Why is \( u_2 + u_3 \geq 1? \)

Induction on \( V', E' \)

Substitute \( B_y, C_z \)

Notations:
\( B_y = \sum_z b_{yz}, \ C_z = \sum_z c_{zx} \)
Case 2: \( k > 1 \). The general proof:

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

Case 2: $k > 1$. The general proof:

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

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

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

Case 2: $k > 1$. The general proof:

$$V = \{x_1, \ldots, x_k\},$$

$$E = \{e_1, \ldots, e_m\}$$

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

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

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

Case 2: \( k > 1 \). The general proof:

\[
\sum_{x_1, \ldots, x_k} \prod_{j=1, m} a_j, e_j \sum_{u_j} = \\
\leq \sum_{x_1, \ldots, x_{k-1}} \prod_{j: x_k \notin e_j} a_j, e_j \left( \sum_{x_k} \prod_{j: x_k \in e_j} a_j, e_j \right)^u_j \\
= \sum_{x_1, \ldots, x_{k-1}} \prod_{j: x_k \notin e_j} a_j, e_j \prod_{j: x_k \in e_j} A_j, e_j ^u_j
\]
Proof

Case 2: $k > 1$. The general proof:

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

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

where $e_j' = e_j - \{x_k\}$

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

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

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 $V', E'$

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

Case 2: \( k > 1 \). The general proof:

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

\[
V' = \{x_1, \ldots, x_{k-1}\}, \\
E = \{e'_1, \ldots, e'_m\}
\]

where \( e'_j = e_j - \{x_k\} \)

\[
\begin{align*}
\sum_{x_1,\ldots,x_k} \prod_{j=1,m} a_{j,e_j}^u &= \\
&= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \not\in e_j} a_{j,e_j}^u \left( \sum_{x_k} \prod_{j:x_k \in e_j} a_{j,e_j}^u \right) \\
&\leq \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \not\in e_j} a_{j,e_j}^u \left( \prod_{j:x_k \in e_j} \left( \sum_{x_k} a_{j,e_j}^u \right)^u \right) \\
&= \sum_{x_1,\ldots,x_{k-1}} \prod_{j:x_k \not\in e_j} a_{j,e_j}^u \prod_{j:x_k \in e_j} A_{j,e_j}^u \\
&\leq \prod_{j:x_k \not\in e_j} \left( \sum_{e_j} a_{j,e_j}^u \right) \prod_{j:x_k \in e_j} \left( \sum_{e_j} A_{j,e_j}^u \right) \\
&= \prod_{j:x_k \not\in e_j} \left( \sum_{e_j} a_{j,e_j}^u \right) \prod_{j:x_k \in e_j} \left( \sum_{e_j} \sum_{x_k} a_{j,e_j}^u \right)
\end{align*}
\]
Proof

Case 2: $k > 1$. The general proof:

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

where $e_j' = e_j - \{x_k\}$

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

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

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

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

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

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

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

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

\[\sum_{e_j'} \sum_{x_k} a_{j,e_j} = \sum_{e_j} \]
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

\[ Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m) \]

**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

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

**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 \( n = \text{number of constants} \); for all \( x, y, z \in \{1, \ldots, n\} \) let:

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

\[ |Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} \]
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 \( n = \text{number of constants} \); for all \( x, y, z \in \{1, \ldots, n\} \) let:

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

\[
|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \\
= \sum_{x,y,z} a_{xy}^{1/2} b_{yz}^{1/2} c_{zx}^{1/2}
\]
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 \( n = \text{number of constants}; \) for all \( x, y, z \in \{1, \ldots, n\} \) let:

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

\[ |Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \]
\[ = \sum_{x,y,z} a_{xy}^{1/2} b_{yz}^{1/2} c_{zx}^{1/2} \leq (\sum_{xy} a_{xy})^{1/2} (\sum_{yz} b_{yz})^{1/2} (\sum_{zx} c_{zx})^{1/2} \]
Not so simple Fact #3

\[ Q(X_1, ..., X_k) = R_1(Vars_1) \bowtie \cdots \bowtie R_m(Vars_m) \]

**Theorem** [Atserias, Grohe, Marx]
Let \( u_1, u_2, ..., u_m \) 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 \( x, y, z \in \{1, ..., n\} \) let:

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

\[
|Q| = \sum_{x,y,z} a_{xy} b_{yz} c_{zx} = \\
= \sum_{x,y,z} a_{xy}^{1/2} b_{yz}^{1/2} c_{zx}^{1/2} \leq (\sum_{xy} a_{xy})^{1/2} (\sum_{yz} b_{yz})^{1/2} (\sum_{zx} c_{zx})^{1/2} \\
= |R|^{1/2} |S|^{1/2} |T|^{1/2}
\]
Not so simple Fact #3

\[ Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m) \]

**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 \( a_{j,x_1,x_2,\ldots} = 1 \) if \((x_1, x_2, \ldots) \in R_j\), 0 otherwise.
Not so simple Fact #3

\[ Q(X_1, \ldots, X_k) = R_1(Vars_1) \Join \cdots \Join R_m(Vars_m) \]

**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 \( a_{j,x_{j_1},x_{j_2},...} = 1 \) if \( (x_{j_1}, x_{j_2}, ...) \in R_j \), 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

\[ Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m) \]

**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.

\[ |Q| = \sum_{x_1, \ldots, x_k} a_{1, \text{Vars}_1} \cdots a_{m, \text{Vars}_m} \]
\[ = \sum_{x_1, \ldots, x_k} a_{1, \text{Vars}_1}^{u_1} \cdots a_{m, \text{Vars}_m}^{u_m} \] // because \( a_{j, \text{Vars}_j} = 0 \) or 1
\[ \leq (\sum_{\text{Vars}_1} a_{1, \text{Vars}_1})^{u_1} \cdots (\sum_{\text{Vars}_m} a_{m, \text{Vars}_m})^{u_m} \]
Not so simple Fact #3

\[ Q(X_1, \ldots, X_k) = R_1(\text{Vars}_1) \bowtie \cdots \bowtie R_m(\text{Vars}_m) \]

**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.

\[
|Q| = \sum_{x_1, \ldots, x_k} a_{1, \text{vars}_1} \cdots a_{m, \text{vars}_m} \\
= \sum_{x_1, \ldots, x_k} a_{\text{vars}_1}^{u_1} \cdots a_{\text{vars}_m}^{u_m} \quad \text{// because } a_{j, \text{vars}_j} = 0 \text{ or } 1 \\
\leq (\sum_{\text{vars}_1} a_{1, \text{vars}_1})^{u_1} \cdots (\sum_{\text{vars}_m} a_{m, \text{vars}_m})^{u_m} \\
= |R_1|^{u_1} \times \cdots \times |R_m|^{u_m} \]
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 $|R|, |S|, |T|, |K| \leq N$

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

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

Assume \(|R|, |S|, |T|, |K| \leq N\)

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

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

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

\[ R(X, Y) \bowtie S(Y, Z) \quad |R|, |S| \leq N \]

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

The **Query Extension** method:
- If \( Y \) is a key in some relation \( S \), then add all attributes of \( S \) relations containing \( Y \)
- Compute \( AGM(Q^{ext}) \)
Example

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

\[ Q \text{ is a key} \]
\[ Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z) \]

- \[ Q^\text{ext}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z), \]

\( Y \) is a key
Example

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

- \[ Q^{ext}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z), \]
- Edge cover: 1,0
- \[ AGM(Q^{ext}) = |R| \]
Example

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

- \[ Q^{\text{ext}}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z), \]
- Edge cover: 1,0
- \[ AGM(Q^{\text{ext}}) = |R| \]

\[ Q(X, Y, Z) = R(X, Y) \land S(Y, Z) \land T(Z, X) \]

Y is a key
Example

\[ Q(X, Y, Z) = R(X, Y) \Join S(Y, Z) \]

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

\[ Q(X, Y, Z) = R(X, Y) \land S(Y, Z) \land T(Z, X) \]

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

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

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

\[ Q(X, Y, Z) = R(X, Y) \land S(Y, Z) \land T(Z, X) \]

- \[ Q^{ext}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z) \land T(Z, X) \]
- Edge covers: 1,0,0 or 0,1,1
Example

\[ Q(X, Y, Z) = R(X, Y) \bowtie S(Y, Z) \]
• \( Q^{\text{ext}}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z), \)
• Edge cover: 1,0
• \( AGM(Q^{\text{ext}}) = |R| \)

\[ Q(X, Y, Z) = R(X, Y) \land S(Y, Z) \land T(Z, X) \]
• \( Q^{\text{ext}}(X, Y, Z) = R(X, Y, Z) \land S(Y, Z) \land T(Z, X) \)
• Edge covers: 1,0,0 or 0,1,1
• \( AGM(Q^{\text{exp}}) = \min(|R|, |S| \times |T|) \)
Equal Cardinalities

If \(|R_1|, |R_2|, \ldots, |R_m| \leq N\)

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

• \(\rho^* \overset{\text{def}}{=} \min_{\text{fractional edge cover}} (u_1 + \cdots + u_m)\)

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 $\text{AGM}(Q)$

• Proof is simple and instructive; we will show for special case $|R_1| = \cdots = |R_m| = N$

• In this case $\text{AGM}(Q) = N^{\rho^*}$
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 **fractional vertex packing** of a (hyper)graph is a set of non-negative numbers $\nu_x$, one for each node $x$, such that, for every edge $e$: $\sum_{x: x \in e} \nu_x \leq 1$

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

$\rho^* = 2$

\[ \begin{array}{ccc}
1 & 0 & 1 \\
1 & 1 & 1 \\
\end{array} \]

$\rho^* = 3/2$

\[ \begin{array}{ccc}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\end{array} \]

$\rho^* = 3$

\[ \begin{array}{cccc}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
\end{array} \]
The Bound is Tight

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

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

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

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

**Proof.**
For each variable $x_i$: $D_i \triangleq [N^{\nu x_i}] = \{1, 2, ..., N^{\nu x_i}\}$
The Bound is Tight

**Fact** For any fractional vertex packing \( \nu_x \), there exists a database instance such that 
\[
|R_1| \leq N, \ldots, |R_m| \leq N \text{ and } |Q| = N^{\sum_x \nu_x}
\]

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

**Proof.**
For each variable \( x_i \): \( D_i \overset{\text{def}}{=} [N^{\nu x_i}] = \{1,2,\ldots,N^{\nu x_i}\} \)
For each relation \( R_j \): 
\[
|R_j(x_{i_1},x_{i_2},\ldots)| \overset{\text{def}}{=} D_{i_1} \times D_{i_2} \times \ldots
\]
The Bound is Tight

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

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

Proof.
For each variable $x_i$: $D_i \overset{\text{def}}{=} \lfloor N^{\nu x_i} \rfloor = \{1, 2, \ldots, N^{\nu x_i}\}$
For each relation $R_j$: $|R_j(x_{i_1}, x_{i_2}, \ldots)| \overset{\text{def}}{=} D_{i_1} \times D_{i_2} \times \cdots$
(a) $|R_j| = N^{\nu_{i_1} + \nu_{i_2} + \cdots} \leq N$  (why?)
The Bound is Tight

**Fact** For any fractional vertex packing \( \nu_x \), there exists a database instance such that 
\(|R_1| \leq N, \ldots, |R_m| \leq N\) and 
\[|Q| = N^{\sum_x \nu_x}\]

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

**Proof.**

For each variable \( x_i \):
\[D_i \overset{\text{def}}{=} \lfloor N^{\nu x_i} \rfloor = \{1,2,\ldots,N^{\nu x_i}\}\]

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

(a) 
\[|R_j| = N^{\nu_{i_1} + \nu_{i_2} + \cdots} \leq N \quad \text{ (why?)}\]

(b) 
\[|Q| = N^{\sum_x \nu_x} \quad (\text{why?})\]
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: \( D_x = D_y = D_z \overset{\text{def}}{=} \left[ N^{1/2} \right] \)
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: $$D_x = D_y = D_z \overset{\text{def}}{=} \left[ N^{1/2} \right]$$

$$R(x, y) \overset{\text{def}}{=} D_x \times D_y$$, $$S(y, z) \overset{\text{def}}{=} D_y \times D_z$$, $$T(z, x) \overset{\text{def}}{=} D_z \times D_x$$
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: \(D_x = D_y = D_z \overset{\text{def}}{=} [N^{1/2}]\)

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

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| ≤ 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| ≤ \(N^3\)

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:

\[ D_x \stackrel{\text{def}}{=} [N], \quad D_y \stackrel{\text{def}}{=} [1], \quad D_z \stackrel{\text{def}}{=} [N], \quad D_u \stackrel{\text{def}}{=} [1], \quad D_v \stackrel{\text{def}}{=} [N], \quad D_w \stackrel{\text{def}}{=} [1] \]
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:
\( D_x \equiv [N], D_y \equiv [1], D_z \equiv [N], D_u \equiv [1], D_v \equiv [N], D_w \equiv [1] \)

\( R(x, y) \equiv [N] \times [1], S(y, z) \equiv [1] \times [N], T(z, u) \equiv [N] \times [1], \ldots \)
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:

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

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

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 $\tilde{O}(AGM(Q))$

$\tilde{O}(\cdots)$ means “times $\log(N)$”
Motivation

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

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

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

- When $R = S = T = \left(\left\lfloor \frac{N}{2} \right\rfloor \times [1]\right) \cup \left([1] \times \left\lceil \frac{N}{2} \right\rceil\right)$
- $|R| = |S| = |T| = N$ but Any query plan takes time $O(N^2)$, because of intermediate relations:
  $$|R \bowtie S| = |S \bowtie T| = |R \bowtie T| = \left\lfloor \frac{N^2}{4} \right\rfloor$$
- 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 \( R_{i_1}, R_{i_2}, \ldots \) be all relations containing \( x \)
compute \( D = \Pi_x(R_{i_1}) \cap \Pi_x(R_{i_2}) \cap \ldots \)
for every value \( v \in D \) do:
  compute \( Q_{x=v}, \)
where \( R_{i_1}, R_{i_2}, \ldots \) are restricted to \( x = v \)
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]

\[ A = \Pi_x(R(x, y)) \land \Pi_x(T(z, x)) \]
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]

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

for a in A do

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

\[ B = \Pi_y(R(a, y)) \cap \Pi_y(S(y, z)) \]
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]

\[ A = \Pi_x(R(x, y)) \land \Pi_x(T(z, x)) \]

**for a in A do**

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

\[ B = \Pi_y(R(a, y)) \land \Pi_y(S(y, z)) \]

**for b in B do**

/* compute \( Q(a, b, z) = R(a, b) \land S(b, z) \land T(z, a) \) */
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]

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

for \( a \) in \( A \) do

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

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

for \( b \) in \( B \) do

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

\[ C = \Pi_z(S(b, z)) \cap \Pi_z(T(z, a)) \]

for \( c \) in \( C \) do

output \((a, b, c)\)
Generic Join Example

\[ Q(x, y, z) = R(x, y) \land S(y, z) \land T(z, x), \]

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

for \( a \) in \( A \) do

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

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

for \( b \) in \( B \) do

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

\[ C = \Pi_z(S(b, z)) \cap \Pi_z(T(z, a)) \]

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 = \( \tilde{O}(\min(|R|, |S|)) \)
Let $x$ be any variable
Let $R_{i_1}, R_{i_2}, \ldots$ be all relations containing $x$
compute $D = \Pi_x(R_{i_1}) \cap \Pi_x(R_{i_2}) \cap \ldots$
for every value $\nu \in D$ do:
compute $Q_{x = \nu}$, where $R_{i_1}, R_{i_2}, \ldots$ are restricted to $x = \nu$
Assume all relations are pre-sorted
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 $u_1, u_2, \ldots$

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

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

Case 1: $k=1$ Then GJ is intersection

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

$Time(Q) = \sum_{v=1}^{n} Time(Q_{x=v}) = \tilde{O}\left(\left(\sum_{v=1}^{n} \prod_{j: x \in \text{vars}(R_j)} R_{j,x=v}^{u_j}\right) \prod_{j: x \notin \text{vars}(R_j)} R_j^{u_j}\right)$
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 $u_1, u_2 \ldots$

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

Case 1: $k=1$ Then GJ is intersection

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

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

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

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

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 $u_1, u_2 \ldots$
We prove: $Time(Q) = \tilde{O}(|R_1|^{u_1} \cdot |R_2|^{u_2} \ldots)$
Case 1: $k=1$ Then GJ is intersection
Case 2: $k>1$ Assume domain of $x$ is $|D| = n$

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

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

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

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

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
    ...

// 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
      ...

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 = \cap \text{domains for } x
For x in A do
  B = \cap \text{domains for } y
  For y in B do
    C = \cap \text{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

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Phone</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-1234</td>
<td>Seattle</td>
</tr>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-6543</td>
<td>Seattle</td>
</tr>
<tr>
<td>Joe</td>
<td>987-65-4321</td>
<td>908-555-2121</td>
<td>Westfield</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Phone</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-1234</td>
<td>Seattle</td>
</tr>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-6543</td>
<td>Seattle</td>
</tr>
<tr>
<td>Joe</td>
<td>987-65-4321</td>
<td>908-555-2121</td>
<td>Westfield</td>
</tr>
</tbody>
</table>
**Relation Decomposition**

Break the relation into two:

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Phone</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-1234</td>
<td>Seattle</td>
</tr>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>206-555-6543</td>
<td>Seattle</td>
</tr>
<tr>
<td>Joe</td>
<td>987-65-4321</td>
<td>908-555-2121</td>
<td>Westfield</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>123-45-6789</td>
<td>Seattle</td>
</tr>
<tr>
<td>Joe</td>
<td>987-65-4321</td>
<td>Westfield</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-45-6789</td>
<td>206-555-1234</td>
</tr>
<tr>
<td>123-45-6789</td>
<td>206-555-6543</td>
</tr>
<tr>
<td>987-65-4321</td>
<td>908-555-2121</td>
</tr>
</tbody>
</table>

The relation satisfies a join dependency!

\[ R(\text{Name},\text{SSN},\text{Phone},\text{City}) = R(\text{Name},\text{SSN},\text{City}) \bowtie R(\text{SSN},\text{Phone}) \]
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.** $R(X_1, ..., X_k)$ satisfies some JD iff

\[
R = R_1 \bowtie \cdots \bowtie R_k
\]

where $R_i = \Pi_{\{X_1, ..., X_k\} - \{X_i\}}(R)$

• **Solution:** compute $Q \triangleq R_1 \bowtie \cdots \bowtie R_k$

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