Logic and Databases

Phokion G. Kolaitis

UC Santa Cruz & IBM Research - Almaden

Lecture 3
Aspects of Conjunctive Query Evaluation

Today, we will carry out a fine-grained examination of conjunctive query evaluation, which includes:

• The universal instance problem
• A closer look at the query complexity of conjunctive query evaluation.
• Islands of tractability for the combined complexity of conjunctive query evaluation.
• A brief look at the parameterized complexity of conjunctive query evaluation.
• Tight estimates on the size of natural joins.
The Universal Instance Problem

- If $S$ is a relation, then $\text{atr}(S)$ is the list of attributes of $S$

The Universal Instance Problem: Given relations $R_1, R_2, \ldots, R_m$, is there a relation $R$ such that for every $i \leq m$,

$$R_i = \pi_{\text{atr}(R_i)}(R)?$$

- Such an $R$ is called a universal instance for $R_1, R_2, \ldots, R_m$
- In this case, we say that $R_1, R_2, \ldots, R_m$ are join-consistent.

Lemma 1: If $R$ is a universal instance for $R_1, R_2, \ldots, R_m$, then

$$R \subseteq R_1 \bowtie R_2 \bowtie \ldots \bowtie R_m$$

Lemma 2: The following statements are equivalent:

1. A universal instance for $R_1, R_2, \ldots, R_m$ exists.
2. $R_1 \bowtie R_2 \bowtie \ldots \bowtie R_m$ is a universal instance for $R_1, R_2, \ldots, R_m$
The Universal Instance Problem

Example: $R(A,B), S(B,C), T(A,C)$

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>S</th>
<th>B</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Clearly, $R \bowtie S \bowtie T = \emptyset$
  Hence (by Lemma 1 or by Lemma 2), no universal instance for $R, S, T$ exists.
- Note that $R, S, T$ have the same projections on their common attributes, namely, \{0,1\}.
- In fact, every pair from $R, S, T$ has a universal instance
Complexity of the Universal Instance Problem

Theorem (Honeyman, Ladner, Yannakakis – 1980)
The Universal Instance Problem is NP-complete.

Proof of NP-hardness: Reduction from 3-Colorability
Given a graph G=(V,E): for each edge e = (u,v) of E, introduce a binary relation $R_e$ with attributes u, v and populate with all valid 3-colorings for (u,v), i.e.,

$$R_e = \{ (r,b), (b,r), (r,g), (g,r), (b,g), (g,b) \}.$$

Fact: G is 3-colorable if and only if the relations $R_e$, $e \in E$, have a universal instance.

In fact, the 3-colorings of G are the members of the natural join

$$\bigdsum_{e \in E} R_e$$

of all the relations $R_e$, $e \in E$. 
# The Complexity of Database Query Languages

<table>
<thead>
<tr>
<th></th>
<th>Relational Calculus</th>
<th>Conjunctive Queries</th>
<th>Unions of Conjunctive Queries</th>
<th>Datalog Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query Eval: Combined / Query Complexity</strong></td>
<td>PSPACE-complete</td>
<td>NP-complete</td>
<td>NP-complete</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td><strong>Query Eval.: Data Complexity</strong></td>
<td>In LOGSPACE (hence, in P)</td>
<td>In LOGSPACE (hence, in P)</td>
<td>In LOGSPACE (hence, in P)</td>
<td>P-complete</td>
</tr>
<tr>
<td><strong>Query Equivalence</strong></td>
<td>Undecidable</td>
<td>NP-complete</td>
<td>NP-complete</td>
<td>Undecidable</td>
</tr>
<tr>
<td><strong>Query Containment</strong></td>
<td>Undecidable</td>
<td>NP-complete</td>
<td>NP-complete</td>
<td>Undecidable</td>
</tr>
</tbody>
</table>
Conjunctive Query Evaluation: Summary

- **Data Complexity of CQ-evaluation** is in LOGSPACE (fixed conjunctive query $q$; the input is a database $D$).

- **Combined Complexity of CQ-evaluation** is NP-complete (fixed database $D$; the input is a conjunctive query $q$).

- **Query Complexity of CQ-evaluation** is NP-complete (the input is a conjunctive query $q$ and a database $D$).
A Closer Look at the Query Complexity of CQs

Definition:
For every database D, let $P_D(CQ)$ be the following decision problem:
Given a Boolean CQ $q$, does $D \models q$?

Theorem:
The query complexity of CQ-evaluation is NP-complete, i.e.,
• $P_D(CQ)$ is in NP, for every database D (but can be in P; e.g. $D = K_2$)
• $P_D(CQ)$ is NP-complete, for some databases D (e.g., $D = K_3$)

Feder-Vardi Dichotomy Conjecture (1993):
For every database D, one of the following holds:
• $P_D(CQ)$ is in P.
• $P_D(CQ)$ is NP-complete.
Moreover, this dichotomy is effective.
Ladner’s Theorem and Complexity Dichotomies

• **Ladner’s Theorem (1975)**
  If $P \neq NP$, then there are decision problems $T$ such that
  – $T$ is in $NP$.
  – $T$ is **not** in $P$.
  – $T$ is **not** $NP$-complete.

• **Feder-Vardi Dichotomy Conjecture (1993) - restated:**
  There is no database $D$ for which $P_D(CQ)$ is a Ladner-type decision problem.
Feder-Vardi Dichotomy Conjecture

• The Feder-Vardi Dichotomy Conjecture was originally formulated in the context of Constraint Satisfaction (viewed as the Homomorphism Problem).

• The Feder-Vardi Dichotomy Conjecture has been confirmed for several special cases, including
  – For databases $D$ such that $|\text{adom}(D)| = 2$
    Schaefer – 1978 (Generalized Satisfiability Problems)
  – For databases $D$ such that $|\text{adom}(D)| = 3$
    Bulatov – 2002
  – For undirected graphs $D$
    Hell and Nešetřil – 1990

• The Feder-Vardi Dichotomy Conjecture remains open to date.
Islands of Tractability of CQ Evaluation

Major Research Program:
Identify tractable cases of the combined complexity of conjunctive query evaluation.

Note:
Over the years, this program has been pursued by two different research communities:
- The Database Theory community.
- The Constraint Satisfaction community.

Explanation:
Constraint Satisfaction Problem $\equiv$ (Feder-Vardi, 1993)
Homomorphism Problem $\equiv$ (Chandra-Merlin, 1977)
Conjunctive Query Evaluation
An Early Large Island of Tractability

- In 1981, Mihalis Yannakakis discovered a large and useful tractable case of the Conjunctive Query Evaluation Problem.

Specifically,

- Yannakakis showed that the Query Evaluation Problem is tractable for **Acyclic Conjunctive Queries**.
Acyclic Conjunctive Queries

Definition: A conjunctive query $Q$ is **acyclic** if it has a **join tree**.

Definition: Let $Q$ be a conjunctive query of the form

$$Q(x): \exists y \ (R_1(z_1) \land R_2(z_2) \land \ldots \land R_m(z_m)).$$

A **join tree** for $Q$ is a tree $T$ such that

- The nodes of $T$ are the atoms $R_i(z_i), 1 \leq i \leq m,$ of $Q$.
- For every variable $w$ occurring in $Q$, the set of the nodes of $T$ that contain $w$ forms a subtree of $T$;
  - in other words, if a variable $w$ occurs in two different atoms $R_j(z_j)$ and $R_k(z_k)$ of $Q$, then it occurs in each atom on the unique path of $T$ joining $R_j(z_j)$ and $R_k(z_k)$.  

Acyclic Conjunctive Queries

- Path of length 4 is acyclic
  \[ P_4(x_1, x_4) : \exists x_2 x_3 (E(x_1, x_2) \land E(x_2, x_3) \land E(x_3, x_4)) \]
  - Join tree is a path

- Cycle of length 4 is cyclic
  \[ C_4(\cdot) : \exists x_1 x_2 x_3 x_4 (E(x_1, x_2) \land E(x_2, x_3) \land E(x_3, x_4) \land E(x_4, x_1)) \]

- The following query Q is acyclic
  \[ Q(\cdot) : \exists x y z u v w (A(x, y, z) \land B(y, v) \land C(y, z, v) \land D(z, u, v) \land F(u, v, w)) \]
Acyclic Conjunctive Queries

\[ Q( ) : \exists x \ y \ z \ u \ v \ w \]
\[ (A(x,y,z) \land B(y,v) \land C(y,z,v) \land D(z,u,v) \land F(u,v,w)) \]
Acyclic Conjunctive Queries

Q( ) : \exists x y z u v w

(A(x,y,z) \land B(y,v) \land C(y,z,v) \land D(z,u,v) \land F(u,v,w))

Join Tree for Q
**Acyclic Conjunctive Queries**

**Theorem** (Yannakakis – 1981)
The *Acyclic Conjunctive Query Evaluation Problem* is tractable. More precisely, there is an algorithm for this problem having the following properties:

- If $Q$ is a Boolean acyclic conjunctive query, then the algorithm runs in time $O(|Q||D|)$.

- If $Q$ is a $k$-ary acyclic conjunctive query, $k \geq 1$, then the algorithm runs in time $O(|Q||D||Q(D)|)$, i.e., it runs in input/output polynomial time (which is the “right” notion of tractability in this case).
Yannakakis’ Algorithm

Dynamic Programming Algorithm

Input: Boolean acyclic conjunctive query $Q$, database $D$

1. Construct a join tree $T$ of $Q$
2. Populate the nodes of $T$ with the matching relations of $D$.
3. Traverse the tree $T$ bottom up:
   - For each node $R_k(z_k)$, compute the semi-joins of the (current) relation in the node $R_k(z_k)$ with the (current) relations in the children of the node $R_k(z_k)$.
4. Examine the resulting relation $R$ at the root of $T$
   - If $R$ is non-empty, then output $Q(D) = 1$ ($D$ satisfies $Q$).
   - If $R$ is empty, then output $Q(D) = 0$ ($D$ does not satisfy $Q$).
Yannakakis’ Algorithm

\[ Q(\ ) : \exists x y z u v w \]
\[ (A(x,y,z) \land B(y,v) \land C(y,z,v) \land D(z,u,v) \land F(u,v,w)) \]

- **D(z,u,v)**
- **C(y,z,v)**
- **F(u,v,w)**
- **A(x,y,z)**
- **B(y,v)**

\[ C(y,z,v) \text{ semi-join } A(x,y,z) \]
\[ = \]
\[ \text{all triples } (y,z,v) \text{ in } C \text{ that “match” a triple } (x,y,z) \text{ in } A \]
More on Yannakakis’ Algorithm

• The join tree makes it possible to avoid exponential explosion in intermediate computations.

• The algorithm can be extended to non-Boolean conjunctive queries using two more traversals of the join tree.

• There are efficient algorithms for detecting acyclicity and computing a join tree.
  – Tarjan and Yannakakis – 1984
    Linear-time algorithm for detecting acyclicity and computing a join tree.
    Detecting acyclicity is in SL (hence, by Reingold’s Theorem detecting acyclicity is in L).
Subsequent Developments

Yannakakis’ algorithm became the catalyst for numerous subsequent investigations in different directions, including:

- **Direction 1:** Identify the exact complexity of **Boolean Acyclic Conjunctive Query Evaluation**.
  - Yannakakis’ algorithm is sequential (e.g., if the join tree is a path of length $n$, then $n-1$ semi-joins in sequence are needed).
  - Is Boolean Acyclic Conjunctive Query Evaluation P-complete? Is it in some parallel complexity class?

- **Direction 2:** Identify other tractable cases of **Conjunctive Query Evaluation**.
Complexity of Acyclic Conjunctive Query Evaluation

Theorem (Dalhaus – 1990)
Boolean Acyclic Conjunctive Query Evaluation is in NC^2.

Theorem (Gottlob, Leone, Scarcello - 1998)
Boolean Acyclic Conjunctive Query Evaluation is LOGCFL-complete, where LOGCFL is the class of all problems having a logspace-reduction to some context-free language.

Fact:
- NL \subseteq LOGCFL \subseteq AC^1 \subseteq NC^2 \subseteq P
- LOGCFL is closed under complements (Borodin et al. - 1989)
Tractable Conjunctive Query Evaluation

• Extensive pursuit of tractable cases of conjunctive query evaluation during the past three decades.

• Two different branches of investigation
  – The relational database schema $\mathbf{S}$ is fixed in advance; in this case, the input conjunctive query is over $\mathbf{S}$.
  – Both the relational database schema and the query are part of the input.

• Note that in Yannakakis’ algorithm both the relational database schema and the query are part of the input.
Enter Tree Decompositions and Treewidth

Definition: \( S \) a fixed relational database schema, \( D \) a database over \( S \).

- A **tree decomposition** of \( D \) is a tree \( T \) such that
  - Every node of \( T \) is labeled by a set of values from \( D \).
  - For every relation \( R \) of \( D \) and every tuple \( (d_1, \ldots, d_m) \in R \), there is a node of \( T \) whose label contains \( \{d_1, \ldots, d_m\} \).
  - For every value \( d \) in \( \text{adom}(D) \), the set \( X \) of nodes of \( T \) whose labels include \( d \) forms a subtree of \( T \).

- \( \text{width}(T) = \max(\text{cardinality of a label of } T) - 1 \)

- **Treewidth**: \( \text{tw}(D) = \min \{ \text{width}(T): T \text{ tree decomposition of } D \} \)
Conjunctive Queries and Treewidth

Definition: \( S \) a fixed relational database schema, \( Q \) a Boolean conjunctive query over \( S \).
- \( \text{tw}(Q) = \text{tw}(Q^D) \), where \( Q^D \) is the canonical database of \( Q \).

- \( TW(k,S) = \) All Boolean conjunctive queries \( Q \) over \( S \) with \( \text{tw}(Q) \leq k \).

Note: Fix a relational database schema \( S \).
- If \( Q \) is a Boolean acyclic conjunctive query over \( S \), then \( \text{tw}(Q) \leq \max \{ \text{arity}(R) : R \text{ is a relation symbol of } S \} - 1 \).

- The converse is not true. For every \( n \geq 3 \), the query \( C_n = \) “is there a cycle of length \( n \)” is cyclic, yet \( \text{tw}(C_n) = 2 \).
Conjunctive Queries and Treewidth

Theorem (Dechter & Pearl – 1989, Freuder 1990)

- For every relational database schema $S$ and every $k \geq 1$, the query evaluation problem for $TW(k, S)$ is tractable.
- In words, there is a polynomial-time algorithm for the following problem: given a database $D$ and a Boolean conjunctive query $Q$ over $S$ of treewidth at most $k$, does $D \models Q$?

Note:
This result was obtained in the quest for islands of tractability of the Constraint Satisfaction Problem.
Beyond Bounded Treewidth

**Question:** Are there islands of tractability for conjunctive query evaluation larger than bounded treewidth?

**Definition:** Two queries $Q$ and $Q'$ are equivalent, denoted $Q \equiv Q'$, if $Q(D) = Q'(D)$, for every database $D$.

**Fact:** Let $Q$ and $Q'$ be Boolean conjunctive queries. Then $Q \equiv Q'$ if and only if $D^Q$ and $D^{Q'}$ are homomorphically equivalent, i.e., there are homomorphisms $h: D^Q \rightarrow D^{Q'}$ and $h': D^{Q'} \rightarrow D^Q$.

**Note:** This follows from the Chandra-Merlin Theorem.
Beyond Bounded Treewidth

**Definition:** \( S \) a fixed relational schema, \( Q \) a Boolean conjunctive query over \( S \).
- \( HTW(k,S) = \) All Boolean conjunctive queries \( Q \) over \( S \) such that \( Q \equiv Q' \), for some \( Q' \in TW(k,S) \).

**Fact:** \( Q \in HTW(k,S) \) if and only if \( \text{core}(Q) \in TW(k,S) \), where \( \text{core}(Q) \) is the minimization of \( Q \), i.e., the smallest subquery of \( Q \) that is equivalent to \( Q \).

**Note:** \( TW(k,S) \) is properly contained in \( HTW(k,S) \)

**Reason:**
The \( k \times k \) grid has treewidth \( k \), but it is 2-colorable, hence it is homomorphically equivalent to \( K_2 \), which has treewidth 1.
Beyond Bounded Treewidth

Theorem (Dalmau, K …, Vardi – 2002)

- For every relational schema $S$ and every $k \geq 1$, the evaluation problem for $HTW(k,S)$ is tractable.
- In words, there is a polynomial-time algorithm for the following problem: given a database $D$ and a Boolean conjunctive query $Q$ that is equivalent to some conjunctive query of treewidth at most $k$, does $D \models Q$?
- In fact, this problem is in Least Fixpoint Logic.

Algorithm:

- Determine the winner in a certain pebble game, known as the existential $k$-pebble game.
- No tree decomposition is used (actually, computing tree decompositions is hard).
A Logical Characterization of Treewidth

Definition: $S$ a relational database schema, $k$ positive integer.
$L^k$ is the collection of all first-order formulas with $k$ variables,
containing all atoms of $S$, and closed under $\land$ and $\exists$.

Theorem (Dalmau, K ..., Vardi – 2002)
$S$ a relational database schema, $Q$ a Boolean conjunctive query over $S$.
Then the following statements are equivalent:
- $Q \in HTW(k,S)$
- $\text{core}(Q) \in TW(k,S)$
- $Q$ is equivalent to some $L^{k+1}$-sentence.

Example: The query $C_n$ : “is there a cycle of length $n$?”
can be expressed in $L^3$. For instance, $C_5$ is equivalent to
$\exists x (\exists y (E(x,y) \land \exists z (E(y,z) \land \exists y (E(z,y) \land \exists z (E(y,z) \land E(z,x)))))$
The Largest Islands of Tractability

Question: Are there islands of tractability larger than $\text{HTW}(k, S)$?

Answer: “No”, modulo a complexity-theoretic hypothesis.

Theorem (Grohe – 2007)
Assume that $\text{FPT} \neq W[1]$.
Let $S$ be a relational database schema and $C$ a recursively enumerable collection of Boolean conjunctive queries over $S$ such that the query evaluation problem for $C$ is tractable.
Then there is a positive integer $k$ such that $C \subseteq \text{HTW}(k, S)$.

Proof: Uses the Excluded Grid Theorem by Robertson & Seymour.
Fixed vs. Variable Relational Schemas

• The preceding results assume that we have a fixed relational database schema $S$, and the conjunctive queries are over $S$.

• As mentioned earlier, in Yannakakis’ algorithm both the relational schema and the query are part of the input.

• When the relational schema is part of the input, then acyclic queries may have (cores of) unbounded treewidth.
  – $Q_n(\ )$: $\exists x_1 \ldots \exists x_n R_n(x_1,\ldots,x_n)$

• Thus, the preceding results do not subsume Yannakakis’ work in the case in which the relational schema is part of the input.
Variable Relational Schemas

- Extensive pursuit of tractable cases of conjunctive query evaluation when the relational schema is part of the input.
  - Several extensions of treewidth have been explored.
  - Hypertree decomposition notions have been studied.

- Chekuri & Rajaraman – 1997: query-width

- Gottlob, Leone, Scarcello – 2000: hypertree-width:
  - Acyclicity amounts to hypertree-width = 1.
  - Tractable conjunctive query evaluation for conjunctive queries of bounded hypertree-width.
Parameterized Complexity

**Theorem** (Papadimitriou & Yannakakis – 1997)

For both fixed and variable relational database schemas, and with the query size as the parameter:

- The parameterized complexity of conjunctive query evaluation is \( \text{W}[1]\)-complete.
- The parameterized complexity of relational calculus query evaluation is \( \text{W}[t]\)-hard, for all \( t \).

**Note:** Several subsequent investigations of the parameterized complexity of query evaluation by

- Downey, Fellows and Taylor
- Flum, Frick and Grohe
- …
Estimates on the Size of Natural Joins

- **Definition:** A full conjunctive query is a quantifier-free conjunctive query.
- **Important Special Case:** Natural Joins

- **Examples:**
  - $Q(x,y,z,w) : - R(x,y), S(z,w)$
  - $Q(x,y,z) :- R(x,y), S(y,z)$
  - $Q(x,y,z,w) :- R(x,y), S(y,z), T(z,w)$
  - $Q(x,y,z) :- R(x,y), S(y,z), T(z,x)$

- **Question:** Given a database $D$ in which $|P(D)| \leq N$, for each relation $P$, how big can $|Q(D)|$ be?
Estimates on the Size of Natural Joins

• **Question:** Given a database D in which $|P(D)| \leq N$, for each relation P, how big can $|Q(D)|$ be?

• **Examples:**
  - $Q(x,y,z,w) : - R(x,y), S(z,w)$ \hspace{1cm} $|Q(D)| \leq N^2$
  - $Q(x,y,z) : - R(x,y), S(y,z)$ \hspace{1cm} $|Q(D)| \leq N^2$
  - $Q(x,y,z,w) : - R(x,y), S(y,z), T(z,w)$ \hspace{1cm} $|Q(D)| \leq N^2$
  - $Q(x,y,z) : - R(x,y), S(y,z), T(z,x)$ \hspace{1cm} $|Q(D)| \leq N^2$

**Non-Obvious Fact:**

- $Q(x,y,z) : - R(x,y), S(y,z), T(z,x)$ \hspace{1cm} $|Q(D)| \leq N^{3/2}$

Where does the $N^{3/2}$ bound come from?
Estimates on the Size of Natural Joins

Atserias, Grohe, Marx (2011):
Established \textit{tight} estimates on the size of natural joins.
Results extend to the size of full conjunctive queries.

Interesting mix of ingredients:
• Fractional edge covers of hypergraphs
• Linear Programming (duality theory)
• Entropy (Shearer’s Lemma)
Edge Covers and Fractional Edge Covers

- **Hypergraph**: $G=(V,E)$ such that if $e \in E$, then $e \subseteq V$
- **Edge Cover of $G=(V,E)$**: Set $C \subseteq E$ such that for every $v \in V$, there is $e \in C$ with $v \in e$.
- **Edge Cover Number** $\rho(G)$: minimum cardinality of edge covers of $G$.
- **Edge Cover Number as a 0-1 Linear Programming Problem**
  - Variable $x_e$, taking values in $\{0, 1\}$, for each $e \in E$
  - $\min \left( \sum_{e \in E} x_e \right)$ subject to
    $$\sum_{v \in e} x_e \geq 1, \text{ for each } v \in V.$$
- **Fractional Cover Number** $\rho^*(H)$: optimal value of the LP relaxation
- **Natural Join** $Q(x)$: - $R_1(x_1), R_2(x_2), \ldots, R_m(x_m)$ as a hypergraph
  - $V =$ set of variables
  - Edge $e_R$ consisting of the variables of $R$, for each atom $R$ of $Q$.
- **Edge cover number** $\rho(Q)$ and **fractional edge cover number** $\rho^*(Q)$
Edge Covers and Fractional Edge Covers

Example: \( Q(x,y,z) :\) R(x,y), S(y,z), T(z,x)
- Linear Program for Edge Cover and Fractional Edge Cover
  \[
  \min (x_R + x_S + x_T)
  \]
  subject to
  \[
  \begin{align*}
  x_R + x_S & \geq 1 \\
  x_S + x_T & \geq 1 \\
  x_R + x_T & \geq 1 \\
  \end{align*}
  \]
- Edge Cover Number: \( \rho(Q) = 2 \)
- Fractional Edge Cover Number: \( \rho^*(Q) = 3/2 \)
Tight Estimates on the Size of Natural Joins

Theorem (Atserias, Grohe, Marx – 2011)
Natural Join $Q(x) : - R_1(x_1), R_2(x_2), \ldots, R_m(x_m)$
If $D$ is a database such that $|R_i(D)| \leq N$, for all $i \leq m$, then
$$|Q(D)| \leq N^{\rho^*(Q)}.$$ Moreover, this upper bound is tight.

Proof Ingredients:
- Upper Bound: Entropy and Shearer’s Lemma
- Lower Bound: Linear Programming Duality Theory
Crash Course on Entropy

• Random Variable $X$ taking values $a_1, \ldots, a_m$
  
  Entropy $H[X] = \Sigma_i \Pr(X = a_i) \log (1/\Pr(X = a_i))$

• Basic Facts:
  – If $X$ is uniform on a space of size $n$, then $H[X] = \log n$
  – If the support of $X$ has cardinality $n$, then $H[X] \leq \log n$
    (Reason: $\log x$ is a concave function)

• Shearer’s Lemma: Let $X = (X_j, j \in J)$ be a r.v. and let $A_1, \ldots, A_m$
  be subsets of $J$ such that each $j$ appears in at least $k$ of them.
  Then $k \cdot H[X] \leq H[X_{A_1}] + \ldots + H[X_{A_m}]$
Tight Bounds on the Size of Natural Joins

Theorem (Atserias, Grohe, Marx – 2011) – Upper Bound by Example

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

If \( D \) is such that \(|R(D)| \leq N\), \(|S(D)| \leq N\), \(|T(D)| \leq N\), then \(|Q(D)| \leq N^{3/2}\).

Proof: Let \( X_{x,y,z} \) be the uniform distribution on \( Q(D) \).

Consider the projections \( X_{x,y}, X_{y,z}, X_{z,x} \).

- **Shearer’s Lemma** applies with \( k = 2 \) and implies that
  \[
  2 \cdot H[X_{x,y,z}] \leq H[X_{x,y}] + H[X_{y,z}] + H[X_{z,x}]
  \]

  \[
  H[X_{x,y,z}] = \log(|Q(D)|) \quad \text{(}X_{x,y,z} \text{ is uniform)}
  \]

  \[
  H[X_{x,y}] \leq \log(|R(D)|) \leq \log(N) \quad \text{(support is contained in } R(D) \text{)}
  \]

  \[
  H[X_{y,z}] \leq \log(|S(D)|) \leq \log(N) \quad \text{(support is contained in } S(D) \text{)}
  \]

  \[
  H[X_{z,x}] \leq \log(|T(D)|) \leq \log(N) \quad \text{(support is contained in } T(D) \text{)}
  \]

Thus, \( 2 \cdot \log(|Q(D)|) \leq 3 \cdot \log(N) \), hence \(|Q(D)| \leq N^{3/2}\).
Natural Joins and Fractional Edge Covers

Theorem (Atserias, Grohe, Marx – 2011)
Let $Q$ be a class of natural join queries. The following statements are equivalent:
1. Queries in $Q$ have answers bounded by a polynomial in $|D|$.
2. Queries in $Q$ can be evaluated in time bounded by a polynomial in $|Q|$ and $|D|$.
3. There is a fixed bound on the fractional edge cover number of queries in $Q$.

Note: Only 2. $\Rightarrow$ 1. is obvious

Corollary: The Universal Instance Problem is solvable in polynomial time on inputs of bounded fractional edge cover.