Boundedness of Regular Path Queries in Data Integration Systems

Gösta Grahne, Alex Thomo
Regular Path Queries

Useful for expressing desired paths to follow in graph DB’s.

E.g.
I want to go from Victoria to Munich taking Air Canada or Lufthansa or United.

Query:
(Air Canada+Lufthansa+United)*

Answer:
{ (Victoria,Vancouver), (Victoria,Frankfurt), (Victoria,Munich), ...
}
Data Sources

Suppose I have a not available the previous DB. What I have is “data sources” (views)

\[ V: \text{(Air Canada+Lufthansa)} \]

Extension:
{(Victoria,Vancouver), (Victoria,Frankfurt), (Victoria,Munich), (Victoria,Hanover), ...}

LAV (local-as-view) data integration

**Global Schema:**
\[ \Delta = \text{Air Canada, Lufthansa, United, BA, AA, Alaska,...} \]

**Local Schema:**
\[ \Omega = \{v, ...\} \]

*User posses queries on the global schema*
Query Answering

Q: (Air Canada+Lufthansa+United)*

V: (Air Canada+Lufthansa)*

Two approaches for answering queries:

• Compute the **certain answer** (very expensive w.r.t to the data)

• Compute **view-based rewriting** and answer it on the view-graph (polynomial w.r.t. to data)
  
  Will go with this here.

**View-Based Rewriting**

[Calvanese, DeGiacomo, Lenzerini, Vardi PODS 1999]

Q' = v*:  
All words on Ω whose substitution is contained in Q.
Unnecessary Recursion

\[ Q' = v^* \]

But why not just:

\[ Q'' = v \]

Surely: \( Q' \neq Q'' \)

...as languages on \( \Omega \).

However, they are equivalent should we “substitute” \( v \) by \( V \), and have languages on \( \Delta \).

Hence, we should rather talk about \( \Omega/\Delta \) equivalence.
Unnecessary Recursion – Another Example

\[ Q = R \ast R^k \]

\[ V = R^+ \]

\[ Q' = (v^k)^+ \quad \text{Recall, it's all words on } \Omega \text{ whose substitution is contained in } Q \]

but...

\[ Q'' = v^k \quad \text{which is clearly better.} \]
Possible Databases and Valid View-Graphs

- $\text{poss}(\mathcal{V})$: Set of all databases from which a given view-graph $\mathcal{V}$ might have been generated.

- Valid $\mathcal{V}$: when $\text{Poss}(\mathcal{V})$ not empty.

- Under exact view assumption, not all view graphs are valid.
  - E.g., consider $\mathcal{V} = R^*$ and $\mathcal{V}$

\[
\begin{array}{cccc}
  a & v & b & v & c \\
\end{array}
\]

$\text{poss}(\mathcal{V}) = \emptyset$. because $\mathcal{V}$ “misses” a v-edge from $a$ to $c$. 
Characterization Theorem

**Theorem.** Let $Q_1$ and $Q_2$ be queries on $\Omega$. Under exact view assumption,

\[ Q_1 \equiv_{\Omega/\Delta} Q_2 \quad \text{iff} \quad \text{for each valid view graph } V \]

\[ \text{ans}(Q_1, V) = \text{ans}(Q_2, V). \]

**Corollary.** Minimize as much as possible a query on $\Omega$ (i.e. a view-based rewriting) without losing query-power as long as $\Omega/\Delta$-equivalence is preserved.

...and $\Omega/\Delta$-equivalence is algebraically weaker than $\Omega$-equivalence.
Sound Views

• Previous theorem doesn’t hold for sound views.
• E.g., consider $V = R^*$, which is $\Omega/\Delta$-equivalent with $V^*$, and

\[ V \]

\[ \begin{array}{ccc}
  a & \rightarrow & v \\
  \downarrow & & \downarrow \\
  b & \rightarrow & v \\
  \downarrow & & \downarrow \\
  c & \rightarrow & \\
\end{array} \]

For $V$, we have that $ans(v^*, V) \neq ans(v, V)$.

• Clearly, the answer of $V$ will be equal to the answer of $V^*$ on each database on $\Delta$,

  …but because the view is assumed to be sound we cannot enforce $V$ to have an additional $v$-edge from $a$ to $c$.  

Two Notions of Boundedness

• $Q_k$ set of all $\Omega$-words in $Q$, of length not more than $k$.

Definition

1. $Q$ is $k$-bounded iff $Q_k \equiv_{\Omega/\Delta} Q$.
2. $Q$ is finitely bounded iff $\exists k \in \mathbb{N}$, such that $Q$ is $k$-bounded.
Theorems

• $k$-boundedness is PSPACE-complete w.r.t. the size of the query.

• Finite boundedness can be decided in EXPTIME w.r.t. the size of the query.
Limitedness Problem in Distance Automata

• Let $A$ be an $\varepsilon$-free weighted automaton (known as distance automata.)
  
  – $d_A(p,w,q) = \inf\{\text{weight}(\pi) : \pi \text{ is a path spelling } w, \text{ from } p \text{ to } q \text{ in } A\}$
  
  – $d(A) = \sup\{d_A(s,w,f) : s \text{ start state, } f \text{ final state}\}$
  
  – $A$ is limited in distance iff $d(A) < \infty$

• Limitedness Problem [Hashiguchi 82]:
  
  *Is a given distance automaton $A$ limited in distance?*
Reduction (I)

View definition

View-based

Rewriting

Weighted transducer
Reduction (I)

Drop output and obtain a weighted automaton.

Do epsilon removal.
Characterization

• Our characterization:

\( Q \) is bounded iff \( A_Q^V \) is limited in distance.
References

• Gösta Grahne, Alex Thomo. Boundedness of Regular Path Queries in Data Integration Systems. IDEAS 2007: 85-92
• Gösta Grahne, Alex Thomo: New Rewritings and Optimizations for Regular Path Queries. ICDT 2003: 242-258
• Gösta Grahne, Alex Thomo: Query containment and rewriting using views for regular path queries under constraints. PODS 2003: 111-122
• Gösta Grahne, Alex Thomo: Algebraic Rewritings for Optimizing Regular Path Queries. ICDT 2001: 301-315
• Gösta Grahne, Alex Thomo: Approximate Reasoning in Semistructured Data. KRDB 2001
• Gösta Grahne, Alex Thomo: An Optimization Technique for Answering Regular Path Queries. WebDB (Selected Papers) 2000: 215-225