Semantic acyclicity

General goal: evaluate queries over (relational or graph) databases

NP-complete problem $\Rightarrow$ find restrictions on the queries
Semantic acyclicity

**General goal:** evaluate queries over (relational or graph) databases

**NP-complete problem** \( \Rightarrow \) find **restrictions** on the queries

![Diagram: query Q to associated graph]

**Fact:** evaluation is **polynomial** for **acyclic** queries
Semantic acyclicity

General goal: evaluate queries over (relational or graph) databases

NP-complete problem \( \Rightarrow \) find restrictions on the queries

Fact: evaluation is polynomial for acyclic queries
Semantic acyclicity

General goal: evaluate queries over (relational or graph) databases

NP-complete problem \(\Rightarrow\) find restrictions on the queries

Fact: evaluation is polynomial for acyclic queries
Semantic acyclicity

**General goal:** evaluate queries over (relational or graph) databases

**NP-complete problem** $\Rightarrow$ find **restrictions** on the queries

**Fact:** evaluation is **polynomial** for **acyclic** queries
Semantic acyclicity

General goal: evaluate queries over (relational or graph) databases

NP-complete problem $\Rightarrow$ find restrictions on the queries

Fact: evaluation is polynomial for acyclic queries

a query is semantically acyclic if it is equivalent to an acyclic query
Semantic acyclicity

- A query is **semantically acyclic** if it is equivalent to an acyclic query.

- Semantic acyclicity is well-understood for **relational databases**.

<table>
<thead>
<tr>
<th>Relational DB</th>
<th>Non-acyclic</th>
<th>Sem. acyclic</th>
<th>Acyclic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NP-complete</td>
<td>tractable (Chen, Dalmau)</td>
<td>linear (Yannakakis)</td>
</tr>
</tbody>
</table>
Semantic acyclicity

- A query is **semantically acyclic** if it is equivalent to an acyclic query.

- Semantic acyclicity is well-understood for **relational databases**.

<table>
<thead>
<tr>
<th>Relational DB</th>
<th>Non-acyclic</th>
<th>Sem. acyclic</th>
<th>Acyclic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NP-complete</td>
<td>tractable (Chen, Dalmau)</td>
<td>linear (Yannakakis)</td>
</tr>
</tbody>
</table>

- **Goal**: explore semantic acyclicity for **graph databases**.
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over $\Sigma$
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over $\Sigma$

\[
\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))
\]

where the $r_i$s are regular expressions over $\Sigma$

$r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over \( \Sigma \)
- **CRPQ**: \( \exists y_1, \ldots, y_n \left( r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k) \right) \)
  where the \( r_i \)'s are regular expressions over \( \Sigma \)
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over $\Sigma$
- **CRPQ**: $\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))$
  where the $r_i$s are regular expressions over $\Sigma$
- $r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over Σ
- **CRPQ**: $\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))$
  where the $r_i$s are regular expressions over Σ
- $r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$

$\exists v, w (\text{carry}^*(v, w) \land \text{fly}(v, w) \land \text{rainbow}(w, w))$

selects the animals who can reach the rainbow
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over $\Sigma$
- **CRPQ**: $\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))$
  where the $r_i$’s are regular expressions over $\Sigma$
- $r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$

$\exists w, \overset{\circ}{w} (\text{carry}^*(w, \overset{\circ}{w}) \land \text{fly}(w, \overset{\circ}{w}) \land \text{rainbow}(\overset{\circ}{w}, w))$

selects the animals who can reach the rainbow
Moving to the setting of graph databases

- **graph databases**: directed graphs with labeled edges over $\Sigma$
- **CRPQ**: $\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))$
  where the $r_i$s are regular expressions over $\Sigma$
- $r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$

$\exists w, \text{no} \ (\text{carry}^*(w, \text{no}) \land \text{fly}(\text{no}, w) \land \text{rainbow}(\text{no}, w))$
selects the animals who can reach the rainbow
Moving to the setting of graph databases

- **Graph databases**: directed graphs with labeled edges over $\Sigma$

- **CRPQ**: $\exists y_1, \ldots, y_n (r_1(u_1, v_1) \land \cdots \land r_k(u_k, v_k))$
  where the $r_i$'s are regular expressions over $\Sigma$

- $r(u, v) \equiv$ there is a path satisfying $r$ from $u$ to $v$

$$\exists v, w \ (\text{carry}^*(v, w) \land \text{fly}(v, w) \land \text{rainbow}(w, w))$$

selects the animals who can reach the rainbow.
Acyclicity on graph databases

- evaluation of CRPQs is **NP-complete** ($|G|^{O(|Q|)}$)
Acyclicity on graph databases

- Evaluation of CRPQs is **NP-complete** \((|G|^{O(|Q|)})\)

- A CRPQ is **acyclic** if its associated graph does not contain a cycle of length \(> 2\)

\[
\exists v, w [\text{carried}^*(v, u) \land \text{fly}(v, w) \land \text{rainbow}(w, w)]
\]
Acyclicity on graph databases

- evaluation of CRPQs is NP-complete ($|G|^{O(|Q|)}$)

- a CRPQ is acyclic if its associated graph does not contain a cycle of length $> 2$

\[ \exists v, w \ [\text{carried}^*(v, u) \land \text{fly}(v, w) \land \text{rainbow}(w, w)] \]

- evaluation of acyclic CRPQ $Q$ over a graph $G$: $O(|G|^2|Q|^2)$
Acylicity on graph databases

- Evaluation of CRPQs is \( \text{NP-complete} \ (|G|^O(|Q|)) \)

- A CRPQ is **acyclic** if its associated graph does not contain a cycle of length > 2

  \[ \exists v, w \ [ \text{carried}^*(v, u) \land \text{fly}(v, w) \land \text{rainbow}(w, w) ] \]

- Evaluation of acyclic CRPQ \( Q \) over a graph \( G \): \( O(|G|^2|Q|^2) \)

- A UCRPQ is **semantically acyclic** if it is equivalent to a union of acyclic CRPQs
Main result

There is a \textit{2EXPSPACE} algorithm that on input $Q$

- checks whether $Q$ is semantically acyclic
- if so, outputs an equivalent acyclic UCRPQ $Q'$ of exponential size

\[ Q \xrightarrow{2 \textit{ExpSpace}} Q' \]

\[ |Q'| \leq 2^{\text{poly}(|Q|)} \]
Main result

There is a $2\text{EXPSPACE}$ algorithm that on input $Q$

- checks whether $Q$ is semantically acyclic
- if so, outputs an equivalent acyclic UCRPQ $Q'$ of exponential size

\[ Q \xrightarrow{2\text{EXPSPACE}} Q' \]

Remark. It cannot be dramatically improved

- checking whether a UCRPQ is semantically acyclic is EXSPACE-hard
- the exponential size of $Q'$ is optimal
Main result

There is a \(2\text{EXPSPACE}\) algorithm that on input \(Q\)

- checks whether \(Q\) is semantically acyclic
- if so, outputs an equivalent acyclic UCRPQ \(Q'\) of \text{exponential} size

\[
\begin{array}{c}
Q \xrightarrow{2\text{EXPSPACE}} Q' \\
\text{Acyclic} \quad |Q'| \leq 2^{\text{poly}(|Q|)}
\end{array}
\]

Remark. It cannot be dramatically improved

- checking whether a UCRPQ is semantically acyclic is \(\text{EXSPACE-hard}\)
- the \text{exponential} size of \(Q'\) is \text{optimal}

Consequence. evaluation of semantically acyclic UCRPQs is \text{fixed parameter tractable}:

\[
O(|G|^2 f(|Q|)) \quad \text{(as opposed to the general case } |G|^{O(|Q|)})
\]
Evaluation of semantically acyclic UCRPQs is fixed parameter tractable

Recall. If $Q$ semantically acyclic, we can compute in 2EXPSPACE an equivalent acyclic UCRPQ $Q'$ of size exponential in $Q$.

\[ |G|^{o(|Q|)} \]
Evaluation of semantically acyclic UCRPQs is fixed parameter tractable

Recall. If $Q$ semantically acyclic, we can compute in $2\text{EXPSPACE}$ an equivalent acyclic UCRPQ $Q'$ of size exponential in $Q$.
Evaluation of semantically acyclic UCRPQs is fixed parameter tractable

Recall. If \( Q \) semantically acyclic, we can compute in \( 2\text{EXPSPACE} \) an equivalent acyclic UCRPQ \( Q' \) of size exponential in \( Q \)
Evaluation of semantically acyclic UCRPQs is fixed parameter tractable

Recall. If $Q$ semantically acyclic, we can compute in $2\text{EXPSPACE}$ an equivalent acyclic UCRPQ $Q'$ of size exponential in $Q$
Evaluation of semantically acyclic UCRPQs is fixed parameter tractable

Recall. If $Q$ semantically acyclic, we can compute in $2\text{EXPSPACE}$ an equivalent acyclic UCRPQ $Q'$ of size exponential in $Q$. 

Evaluation of semantically acyclic UCRPQs is fixed parameter tractable
Conclusion and open questions

- checking for semantic acyclicity is in 2EXPSPACE and EXPSPACE-hard
- evaluation of semantically acyclic queries is fixed-parameter tractable
Conclusion and open questions

- checking for semantic acyclicity is in 2EXPSPACE and EXPSPACE-hard
- evaluation of semantically acyclic queries is fixed-parameter tractable

- extension to wider classes (e.g. bounded treewidth)
- exact complexity of evaluation of semantically acyclic UCRPQs