Task 1: Minimal Total Joining Network of Tuples (MTJNTs)

Given are the following database, its database graph, the query $K$ and the size control parameter $T_{\text{max}}$ (maximum number of tuples in MTJNT) as follows.

$K = \{\text{Michelle, XML}\}$

$T_{\text{max}} = 5$

Find all MTJNTs for query $K$.

Solution:

contains $(a_3, \text{“Michelle”})$
contains $(p_1, \text{“Michelle”})$
contains $(p_2, \text{“XML”})$
contains $(p_3, \text{“XML”})$
Task 2: Candidate Networks (CN)

Given are the following CNs:

Write SQL query expressions to generate \(C_1, \ldots, C_4\).

You can also try your queries online here: https://sqliteonline.com. First, load the database file linked on the lecture's website (https://www2.kbs.uni-hannover.de/399.html). When you execute your query online, use “a LIKE %b%” instead of “contains(a,b)”.

Solution (Online queries)\(^1\):

<table>
<thead>
<tr>
<th></th>
<th>SQL Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_1)</td>
<td><code>SELECT * FROM Paper as P1, Cite as C, Paper as P2 WHERE P1.Title LIKE &quot;%Michelle%&quot; AND NOT P1.Title LIKE &quot;%XML%&quot; AND P1.TID = C.PID2 AND C.PID1 = P2.TID AND P2.Title LIKE &quot;%XML%&quot; AND P2.Title NOT LIKE &quot;%Michelle%&quot;</code></td>
</tr>
<tr>
<td>(C_2)</td>
<td><code>SELECT * FROM Paper as P1, Write as W1, Author as A, Write as W2, Paper as P2 WHERE P1.Title LIKE &quot;%Michelle%&quot; AND NOT P1.Title LIKE &quot;%XML%&quot; AND P1.TID = W1.PID AND W1.AID = A.TID AND A.TID = W2.AID AND P2.TID = W2.PID AND P2.Title LIKE &quot;%XML%&quot; AND P2.Title NOT LIKE &quot;%Michelle%&quot;</code></td>
</tr>
<tr>
<td>(C_3)</td>
<td><code>SELECT * FROM Author as A, Write as W, Paper as P WHERE A.Name LIKE &quot;%Michelle%&quot; AND NOT A.Name LIKE &quot;%XML%&quot; AND A.TID = W.AID AND W.PID = P.TID AND P.Title LIKE &quot;%XML%&quot; AND P.Title NOT LIKE &quot;%Michelle%&quot;</code></td>
</tr>
<tr>
<td>(C_4)</td>
<td><code>SELECT * FROM Author as A, Write as W, Paper as P1, Cite as C, Paper as P2 WHERE A.Name LIKE &quot;%Michelle%&quot; AND NOT A.Name LIKE &quot;%XML%&quot; AND A.TID = W.AID AND W.PID = P1.TID AND P1.Title NOT LIKE &quot;%XML%&quot; AND P1.Title NOT LIKE &quot;%Michelle%&quot; AND P1.TID = C.PID2 AND C.PID1 = P2.TID AND P2.TID = P2.TID AND P2.Title LIKE &quot;%XML%&quot; AND P2.Title NOT LIKE &quot;%Michelle%&quot;</code></td>
</tr>
</tbody>
</table>

\(^1\) In a suited setting, you would use the "CONTAINS" predicate, e.g. \(P1.CONTAINS(P1.Title, "Michelle")\).
Task 3: CN generation algorithm

Given are the CN generation algorithm, a schema graph and the query keyword relations as follows.

CN generation algorithm (BFS-based):

Algorithm 1 Discover-CNGen \((Q, \text{Tmax}, G_S)\)

Input: an \(l\)-keyword query \(Q = \{k_1, k_2, \ldots, k_l\}\), the size control parameter \(\text{Tmax}\),
the schema graph \(G_S\).

Output: the set of CNs \(\mathcal{C} = \{C_1, C_2, \ldots\}\).

1. \(Q \leftarrow \emptyset; \mathcal{C} \leftarrow \emptyset\)
2. for all \(R_i \in V(G_S), K' \subseteq Q\) do
3. \(Q.enqueue(R_i[K'])\)
4. while \(Q \neq \emptyset\) do
5. \(T \leftarrow Q.dequeue()\)
6. if \(T\) is minimal and total and \(T\) does not satisfy Rule-1 then
7. \(\mathcal{C} \leftarrow \mathcal{C} \cup \{T\}; \text{ continue}\)
8. if the size of \(T < \text{Tmax}\) then
9. for all \(R_i \in T\) do
10. for all \((R_i, R_j) \in E(G_S)\) or \((R_j, R_i) \in E(G_S)\) do
11. \(T' \leftarrow T \cup (R_i, R_j)\)
12. if \(T'\) does not satisfy Rule-2 or Rule-3 then
13. \(Q.enqueue(T')\)
14. return \(\mathcal{C}\).

Rule 1: There are duplicates in \(C\).
Rule 2: The CN is not minimal.
Rule 3: The CN contains cycles.

Schema graph:

Keyword relations:
\(A\{\text{Michelle}\}, P\{\text{XML}\}, P\{\text{Michelle}\}\)

Write down the essential steps of the algorithm until the first valid (i.e. total and minimal) CN is generated.
Solution:

\[ Q = \langle A\{\text{Michelle} \}, P \{\text{XML} \}, P \{\text{Michelle} \} \rangle \]

\[ T = A\{\text{Michelle} \} \]

\[ T' = A\{\text{Michelle} \} \Join^{ADV} W \{\} \]

\[ Q = \langle P \{\text{XML} \}, P \{\text{Michelle} \}, A\{\text{Michelle} \} \Join^{ADV} W \{\} \rangle \]

\[ T = P \{\text{XML} \} \]

\[ T' = P \{\text{XML} \} \Join^{PDC1} C \{\} \]

\[ Q = \langle P \{\text{Michelle} \}, A\{\text{Michelle} \} \Join^{ADV} W \{\}, P \{\text{XML} \} \Join^{PDC1} C \{\} \rangle \]

\[ T' = P \{\text{XML} \} \Join^{PDC2} C \{\} \]

\[ Q = \langle P \{\text{Michelle} \}, A\{\text{Michelle} \} \Join^{ADV} W \{\}, P \{\text{XML} \} \Join^{PDC1} C \{\}, P \{\text{XML} \} \Join^{PDC2} C \{\} \rangle \]

\[ T = C \{\} \]

\[ T' = C \{\} \Join^{ADV} W \{\} \Join^{PDC} P \{\text{XML} \} \]

\[ T = A\{\text{Michelle} \} \Join^{ADV} W \{\} \Join^{PDC} P \{\text{XML} \} \]

\[ T \text{ is minimal and total and does not satisfy Rule 1} \]

\[ C = \{A\{\text{Michelle} \} \Join^{ADV} W \{\} \Join^{PDC} P \{\text{XML} \} \} \]