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_{max} (maximum number of tuples in MTJNT) as follows.

TID	Name
a_1	Charlie Carpenter
a_2	Michael Richardson
a_3	Michelle

(a) Author

TID	Title
p_1	Contributions of Michelle
p_2	Keyword Search in XML
p_3	Pattern Matching in XML
p_4	Algorithms for TopK Query

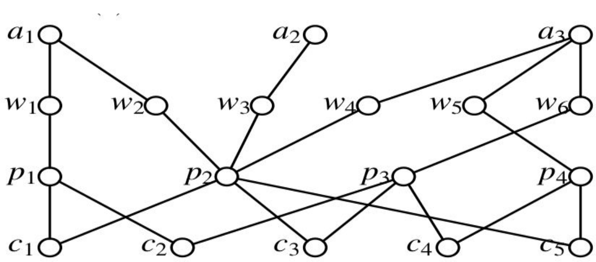
(b) Paper

TID	AID	PID
w_1	a_1	p_1
w_2	a_1	p_2
w_3	a_2	p_2
w_4	a_3	p_2
w_5	a_3	p_4
w_6	a_3	p_3

(c) Write

TID	PID1	PID2
c_1	p_2	p_1
c_2	p_3	p_1
c_3	p_2	p_3
c_4	p_3	p_4
c_5	p_2	p_4

(d) Cite



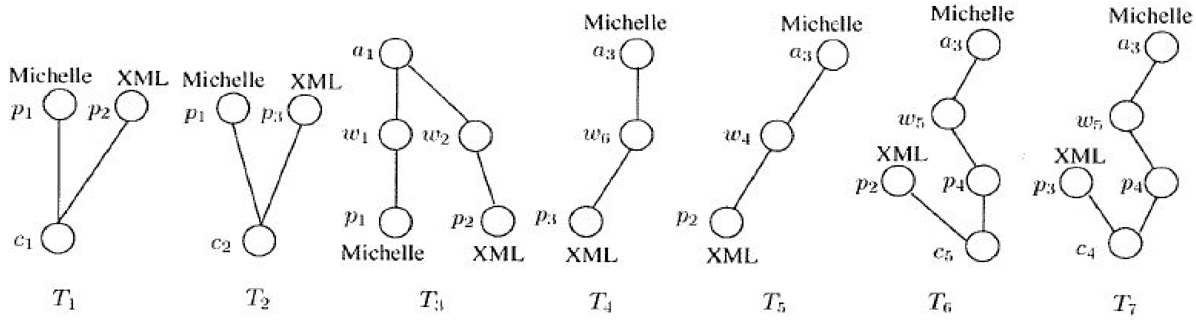
(e) Tuple Connections

$K = \{Michelle, XML\}$
$T_{max} = 5$

Find all MTJNTs for query K.

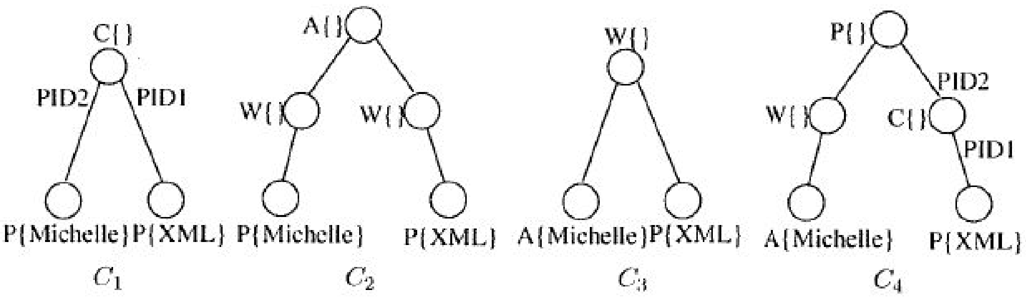
Solution:

contains (a_3, "Michelle")
contains (p_1, "Michelle")
contains (p_2, "XML")
contains (p_3, "XML")



Task 2: Candidate Networks (CN)

Given are the following CNs:



Write SQL query expressions to generate $C1, ..., C4$.

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]:

$C1$	$C3$
SELECT * FROM Paper as P1, Cite as C, Paper as P2 WHERE P1.Title LIKE "%Michelle%" AND NOT P1.Title LIKE "%XML%" AND P1.TID = C.PID2 AND C.PID1 = P2.TID AND P2.Title LIKE "%XML%" AND P2.Title NOT LIKE "%Michelle%"	SELECT * FROM Author as A, Write as W, Paper as P WHERE A.Name LIKE "%Michelle%" AND NOT A.Name LIKE "%XML%" AND A.TID = W.AID AND W.PID = P.TID AND P.Title LIKE "%XML%" AND P.Title NOT LIKE "%Michelle%"
$C2$	**$C4$**
SELECT * FROM Paper as P1, Write as W1, Author as A, Write as W2, Paper as P2 WHERE P1.Title LIKE "%Michelle%" AND NOT P1.Title LIKE "%XML%" AND P1.TID = W1.PID AND W1.AID = A.TID AND A.TID = W2.AID AND P2.TID = W2.PID AND P2.Title LIKE "%XML%" AND P2.Title NOT LIKE "%Michelle%"	SELECT * FROM Author as A, Write as W, Paper as P1, Cite as C, Paper as P2 WHERE A.Name LIKE "%Michelle%" AND NOT A.Name LIKE "%XML%" AND A.TID = W.AID AND W.PID = P1.TID AND P1.Title NOT LIKE "%XML%" AND P1.Title NOT LIKE "%Michelle%" AND P1.TID = C.PID2 AND C.PID1 = P2.TID AND P2.Title LIKE "%XML%" AND P2.Title NOT LIKE "%Michelle%"

[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):

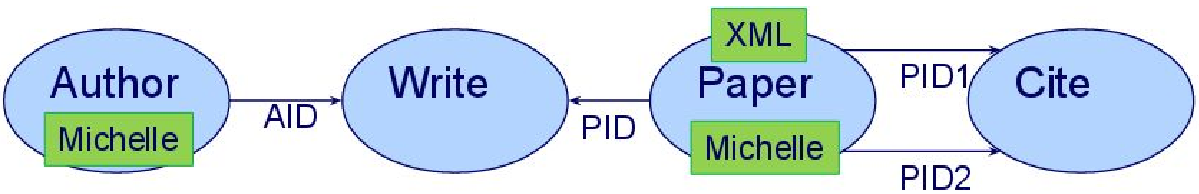
Algorithm 1 Discover-CNGen (Q, Tmax, G_S)

Input: an l-keyword query $Q = \{k_1, k_2, \cdots, k_l\}$, the size control parameter Tmax, the schema graph G_S.
Output: the set of CNs $\mathcal{C} = \{C_1, C_2, \cdots\}$.

$\mathcal{Q} \leftarrow \emptyset; \mathcal{C} \leftarrow \emptyset$
for all $R_i \in V(G_S)$, $K' \subseteq Q$ **do**
  $\mathcal{Q}.enqueue(R_i\{K'\})$
while $\mathcal{Q} \neq \emptyset$ **do**
  $T \leftarrow \mathcal{Q}.dequeue()$
  if T is minimal and total **and** T does not satisfy Rule-1 **then**
    $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{T\}$; **continue**
  if the size of T < Tmax **then**
    for all $R_i \in T$ **do**
      for all $(R_i, R_j) \in E(G_S)$ **or** $(R_j, R_i) \in E(G_S)$ **do**
        $T' \leftarrow T \bigcup (R_i, R_j)$
        if T' does not satisfy Rule-2 or Rule-3 **then**
          $\mathcal{Q}.enqueue(T')$
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:
$A\{Michelle\}$, $P\{XML\}$, $P\{Michelle\}$

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

<u>Solution:</u>

$Q =< A\{Michelle\},\ P\{XML\},\ P\{Michelle\} >$

$T = A\{Michelle\}$

$T' = A\{Michelle\} \bowtie^{AID} W\{\}$

$Q =< P\{XML\},\ P\{Michelle\}, A\{Michelle\} \bowtie^{AID} W\{\} >$

$T = P\{XML\}$

$T' = P\{XML\} \bowtie^{PID1} C\{\}$

$Q =< P\{Michelle\}, A\{Michelle\} \bowtie^{AID} W\{\}, P\{XML\} \bowtie^{PID1} C\{\} >$

$T' = P\{XML\} \bowtie^{PID2} C\{\}$

$Q =< P\{Michelle\}, A\{Michelle\} \bowtie^{AID} W\{\}, P\{XML\} \bowtie^{PID1} C\{\},$
$P\{XML\} \bowtie^{PID2} C\{\} >$

$T = P\{Michelle\}$

$T' = P\{Michelle\} \bowtie^{PID1} C\{\}$

$Q =< A\{Michelle\} \bowtie^{AID} W\{\}, P\{XML\} \bowtie^{PID1} C\{\},$
$P\{XML\} \bowtie^{PID2} C\{\}, P\{Michelle\} \bowtie^{PID1} C\{\} >$

$T' = P\{Michelle\} \bowtie^{PID2} C\{\}$

$Q =< A\{Michelle\} \bowtie^{AID} W\{\}, P\{XML\} \bowtie^{PID1} C\{\},$
$P\{XML\} \bowtie^{PID2} C\{\}, P\{Michelle\} \bowtie^{PID1} C\{\}, P\{Michelle\} \bowtie^{PID1} C\{\} >$

$T = A\{M\} \bowtie^{AID} W\{\}$

$T' = A\{Michelle\} \bowtie^{AID} W\{\} \bowtie^{PID} P\{XML\}$

...

...

$T = A\{Michelle\} \bowtie^{AID} W\{\} \bowtie^{PID} P\{XML\}$

T is minimal and total and does not satisfy Rule 1

$C = \{A\{Michelle\} \bowtie^{AID} W\{\} \bowtie^{PID} P\{XML\}\}$

...