Exam Re-Take

Uncertain Data Management
Université Paris-Saclay, M2 Data&Knowledge

June 6th, 2016

This is the re-take of the final exam for the Uncertain Data Management class. The grade in this exam will replace your grade of the first session of the final exam, and will become your final grade for the class. The exam consists of four independent exercises.

You are allowed up to two A4 sheets of personal notes (i.e., four page sides), printed or written by hand, with font size of 10 points at most. If you use such personal notes, you must hand them in along with your answers. You may not use any other written material.

Write your name clearly on the top right of every sheet used for your exam answers. Number every page. It is highly recommended to answer the exercises on separate sheets.

The exam is strictly personal: any communication or influence between students, or use of outside help, is prohibited. Any violation of the rules may result in a grade of 0 and/or disciplinary action.

Exercise 1: Unions of TID (3 points)

Consider the three TID instances R, S, and T defined as follows:

R

attr1	**attr2**	
a	b	0.5

S

attr1	**attr2**	
c	d	0.8

T

attr1	**attr2**	
a	b	0.2

Question 1 (0.5 point). Let U_1 be the probabilistic instance defined by the query $R \cup S$ (in relational algebra). Write a representation of U_1 as a TID instance.

Answer. We can represent U_1 as the following TID instance:

U_1

attr1	**attr2**	
a	b	0.5
c	d	0.8

Question 2 (0.5 point). Define likewise U_2 as $R \cup T$. Write a representation of U_2 as a TID instance.

Answer. We can represent U_2 as the following TID instance:

U_1

attr1	**attr2**	
a	b	0.6

In this instance, 0.6 is computed as $1 - (1 - 0.5) \times (1 - 0.8)$.

Question 3 (2 points). Show that the union $R_1 \cup R_2$ of two arbitrary TID instances R_1 and R_2 can always be represented as a TID instance U. Describe how U is constructed from R_1 and R_2.

Answer. Let R_1 and R_2 be arbitrary TID instances. The possible tuples of U are clearly those of R_1 and those of R_2. For the tuples that are only in R_1, their probability of occurring in U is clearly their probability in R_1, and they are independent from any other possible tuple of U. The same claim holds for the tuples that are only in R_2. For the tuples that are both in R_1 and R_2, they are independent from any other tuple of U, and their probability of occurring in U is their probability of occurring in either R_1 and R_2, these two events being independent.

Hence, the TID representation of U consists of the tuples that are only in R_1 with the same probabilities as in R_1, the tuples that are only in R_2 with the same probabilities as in R_2, and the tuples that are both in R_1 and R_2, each tuple t having probability $1 - (1 - p_1) \times (1 - p_2)$ where p_1 and p_2 are respectively the probabilities of t in R_1 and in R_2.

Exercise 2: Unions of BID (7 points)

Consider the three BID instances R, S, and T defined as follows:

R

attr1	**attr2**	
a	b	0.3
a	c	0.4

S

attr1	**attr2**	
d	e	0.1
d	f	0.1

T

attr1	**attr2**	
a	g	0.1

Question 1 (0.5 point). Define U_1 as $R \cup S$. Write a representation of U_1 as a BID instance. How many blocks does U_1 contain?

Answer. We can represent U_1 as the following BID instance:

U_1

attr1	**attr2**	
a	b	0.3
a	c	0.4
d	e	0.1
d	f	0.1

U_1 contains two blocks.

Question 2 (1.5 points). Let U_2 be $R \cup T$. Prove that U_2 cannot be represented as a BID instance with key **attr1**.

Answer. Consider the possible world of R where the tuple $t = (a, b)$ is retained, and the possible world of T where the tuple $t' = (a, g)$ is retained. As these possible worlds each have probability > 0, there is a possible world of U_2 containing the two tuples t and t'. Now, assuming by way of contradiction that we can represent U_2 as a BID, U_2 contains both t and t'. These two tuples are different but match on the key attribute **attr1**. But then they should be mutually exclusive, so there is no possible world of U_2 where both t and t' occur. We have reached a contradiction, which concludes the proof.

Question 3 (1 point). Can U_2 be represented as a BID instance over **attr1**, **attr2** but with some other choice of key attributes? If yes, write such a representation; if not, prove that there is no such representation.

Answer. U_2 cannot be represented as a BID instance, no matter what the key is. We will proceed by case disjunction on the possible keys.

Observe first that, as R has a possible world containing $t_1 = (a, b)$ and a possible world containing $t_2 = (a, c)$, U_2 should contain t_1 and t_2. However, as they are mutually exclusive in R, there should be no possible world of U_2 where both t_1 and t_2 occur. This means that the key for a BID representation of U_2 cannot be **attr2** and cannot be **attr1**, **attr2**, as in both cases t_1 and t_2 would be in different blocks and there would then be a possible world where both occur.

Observe second that, as R has a possible world containing t_1 and T has a possible world containing (a, g), U_2 has a possible world containing both t_1 and t_2. This excludes the case of an empty key, as all tuples of U_2 would then be mutually exclusive, so U_2 would have no possible world with more than one tuple. As we eliminated the case of the key **attr1** in the previous question, no possibilities remain, which concludes.

Question 4 (1 point). Give two *different* BID instances R' and S' over the schema **attr1**, **attr2** and with key **attr1** such that R' and S' both contain some common tuple t with some probability $0 < p < 1$, and yet $R' \cup S'$ can be represented as a BID instance. Specifically, write a suitable choice of BID R' and S' and write the representation of their union $R' \cup S'$ as a BID.

Answer. Consider the following BID instances:

R'

attr1	**attr2**	
a	b	0.5

S'

attr1	**attr2**	
a	b	0.5
c	d	0.5

R' and S' have one tuple in common with the same probability. Yet, we can represent $R' \cup S'$ as the following BID instance:

$R' \cup S'$

attr1	**attr2**	
a	b	0.75
c	d	0.5

Question 5 (1 point). Let R_1 and R_2 be two arbitrary BID instances over the schema **attr1**, **attr2** with key **attr1**. Give a characterization of when $R_1 \cup R_2$ cannot be represented as a BID instance over this schema with this key. In other words, write a necessary and sufficient condition on R_1 and R_2 that holds if and only if $R_1 \cup R_2$ cannot be represented as a BID instance with key **attr1**. You are not required to prove that your proposed condition is correct. (Hint: use the examples of questions 1, 2, and 4 to verify that your condition correctly classifies them.)

Answer. $R_1 \cup R_2$ cannot be represented as a BID instance if and only if the following condition holds on R_1 and R_2: there is some value of **attr1** for which both R_1 and R_2 have a block, and these two blocks are not both singleton blocks over the same tuple (disregarding the probability). In other words, we fail if there are two blocks on a common key value and one of them is not a singleton, or they are both singletons but contain different tuples.

The following is not required in the answer and is just here as an explanation. To see that the negation of this condition is sufficient to ensure that the result can be represented as a BID instance, considering R_1 and R_2 that violate the condition, the blocks of R_1 for key values with no block in R_2 can be kept as-is, and likewise for the blocks in R_2 having no counterpart in R_1. For common key values, as the two blocks must be singletons of the same tuple, we can create a singleton block for them in the result, with a probability computed as that of the independent disjunction.

To see why the negation of this condition is necessary for the result to be representable as a BID instance, assuming that there are two counterexample blocks, we see as in Question 3 that there is a possible world of the union with two different tuples for that block, which is not possible if the result is to be represented as a BID instance.

Question 6 (1 point) Given a BID instance W over **attr1**, **attr2** with key **attr1**, we denote by $\overline{W}$ the result of keeping the same tuples with the same probabilities, but changing the key attribute to be **attr2** instead of **attr1**. For instance, remembering S from the beginning of the exercise, we define $\overline{S}$ as:

$\overline{S}$

attr1	**attr2**	
d	e	0.1
d	f	0.1

However, this operation is not always well-defined: i.e., it is not always possible to interpret its result as a BID instance. To illustrate this, give an example of a BID instance W over **attr1**, **attr2** with key **attr1** such that $\overline{W}$ cannot be interpreted as a BID instance, and quickly explain why.

Answer. Consider the following BID instance:

W

attr1	**attr2**	
a	c	1
b	c	1

$\overline{W}$ would be as follows:

$\overline{W}$

attr1	**attr2**	
a	c	1
b	c	1

This is not a valid BID instance because the probabilities in the only block of the instance sum up to 2 which is > 1.

Question 7 (1 point). Remember the BID instance R defined at the beginning of the exercise. Design a BID instance W' over **attr1**, **attr2** with key **attr1** such that $R \cup W'$ can be represented as a BID instance, $\overline{W'}$ is well-defined (i.e., it is a BID instance), but $\overline{R} \cup \overline{W'}$ cannot be represented as a BID instance. Use Question 5 to justify that your choice of W' satisfies these conditions.

Answer. Consider the following BID instance:

W'

attr1	**attr2**	
b	b	1

It is clear that $\overline{W'}$ is well-defined. We can represent $R \cup W'$ as a BID instance, because, using the criterion of Question 5, there is no value of the key for which R and W' both have a block. However, for the value b, $\overline{R}$ and $\overline{W'}$ both have a block, and the two blocks are not singletons of the same tuple, so, using the same criterion, we cannot represent $\overline{R} \cup \overline{W'}$ as a BID instance.

Exercise 3: Probabilistic Query Processing (6 points)

Consider the following TID instances:

A

s	**m**	
a	b	p_1
a	c	p_2
b	c	p_3
b	d	p_4
b	e	p_5
c	d	p_6
c	e	p_7
e	d	p_8

B

m	**t**	
b	c	r_1
c	b	r_2
c	d	r_3
c	e	r_4

Question 1 (1 point). Consider the following query in relational calculus:

$$Q(s,t) := \exists x, y \; A(s,x) \wedge B(x,y) \wedge A(y,t).$$

Write an SQL query that evaluates $Q(a,e)$.

Answer. A possible SQL query is:

```
SELECT COUNT(1) FROM A a1, B b, A a2
    WHERE a1.s='a' AND a1.m=b.m AND b.t=a2.s AND a2.m='e';
```

Question 2 (2 points). We assign probabilistic events to the tuples in A and B as follows:

A		
s	**m**	
a	b	X_1
a	c	X_2
b	c	X_3
b	d	X_4
b	e	X_5
c	d	X_6
c	e	X_7
e	d	X_8

B		
m	**t**	
b	c	Y_1
c	b	Y_2
c	d	Y_3
c	e	Y_4

Consider again the query $Q(a, e)$. Write the lineage of the query, i.e., a Boolean formula in disjunctive normal form over the probabilistic events which evaluates to true iff the query is true when the corresponding facts are kept. Use this to write the probability $\Pr[Q(a, e)]$ that the query is true, as a function of the probability values $p_1, \ldots, p_8, r_1, \ldots, r_4$.

Answer. The lineage of $Q(a, e)$ is:

$$(X_1 \wedge Y_1 \wedge X_7) \vee (X_2 \wedge Y_2 \wedge X_5).$$

This is a read-once formula, and thus we can use independent-AND and independent-OR to get the probability formula:

$$\Pr[Q(a, e)] = 1 - (1 - p_1 r_1 p_7)(1 - p_2 r_2 p_5).$$

Question 3 (2 points). Consider the query $Q' := \pi_{\mathbf{s}}(\sigma_{\mathbf{s}=\text{'a'} \vee \mathbf{s}=\text{'b'}}(A \bowtie B))$ in relational algebra, and the following plan for Q': $\pi_{\mathbf{s}}\big(\sigma_{\mathbf{s}=\text{'a'} \vee \mathbf{s}=\text{'b'}}(A) \bowtie \pi_{\mathbf{m}}(B)\big)$. Compute Q' following this plan, detailing the steps on the instance represented above. Is it true that the probabilities obtained by evaluating the plan match the correct probabilities for the result of the query? Can the resulting relation be represented as a TID instance?

Answer.

The two sides of the join are the following:

$\sigma_{\mathbf{s}=\text{'a'} \vee \mathbf{s}=\text{'b'}}(A)$		
s	**m**	
a	b	p_1
a	c	p_2
b	c	p_3
b	d	p_4
b	e	p_5

$\pi_{\mathbf{m}}(B)$	
m	
b	r_1
c	$1 - (1 - r_2)(1 - r_3)(1 - r_4)$

The result of the join is:

$\sigma_{\mathbf{s}=\text{'a'} \vee \mathbf{s}=\text{'b'}}(A) \bowtie \pi_{\mathbf{m}}(B)$		
s	**m**	
a	b	$p_1 r_1$
a	c	$p_2(1 - (1 - r_2)(1 - r_3)(1 - r_4))$
b	c	$p_3(1 - (1 - r_2)(1 - r_3)(1 - r_4))$

The result of the final projection is:

Q'

s	
a	$1-(1-p_1r_1)(1-p_2(1-(1-r_2)(1-r_3)(1-r_4)))$
b	$p_3(1-(1-r_2)(1-r_3)(1-r_4))$

The computed probabilities are correct. The result cannot be represented as a TID instance, since the presence of the a-tuple and that of the b-tuple in the result are not independent probabilistic events.

Question 4 (1 point). Consider the same plan as above in the general case, i.e., on any TID instance of the tables A and B. Is the plan safe in general? Briefly explain why or why not. What does this imply about the safeness of the query?

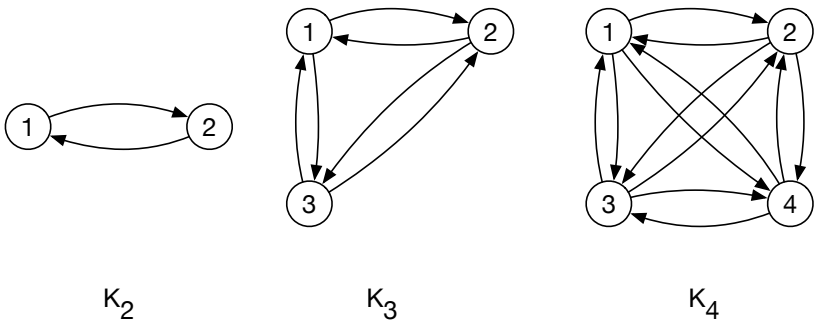
Answer. In general, the plan is safe for Q' and any two TID relations having the schema $A(s,t)$, $B(s,t)$. Going in order through the plan:

1. the selection on A will keep the probabilities correct and results in a TID relation,
2. the projection on B will keep the probabilities correct (it is a projection on a TID instance) and results in a TID relation,
3. the join step maintains the correctness of the probabilities (it is a join between two independent relations) but *does not* necessarily result in a TID instance,
4. the final projection on $A.s$ contains correct probabilities: even if the join is no longer a TID instance, the tuples having a given value of $A.s$ have independent probabilities.

The query in general is safe, since it has as a safe plan, namely, the one described above.

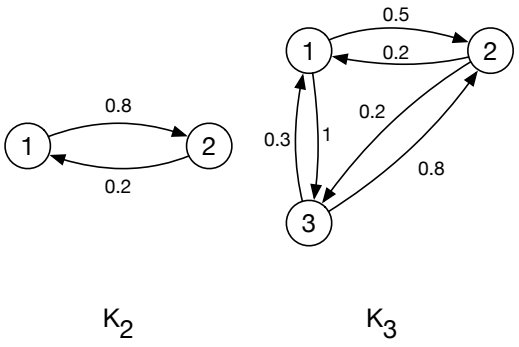
Exercise 4: Reachability Queries (4 points)

In this exercise, we will consider reachability queries on probabilistic complete graphs. A *complete graph* of n nodes, denoted K_n, is a directed graph where there exists an edge from each node to each node of the graph (excluding self-loops, i.e., there is never an edge from one node to itself). We represent the graphs K_2, K_3, and K_4 below:



A *probabilistic complete graph* is a complete graph where each edge has a non-zero probability of existence. The *reachability query* $\mathsf{Reach}(s,t)$ on a probabilistic complete graph asks, given a source node s and a target node t with $s \neq t$, what is the probability that node t is reachable via a directed path from s.

Question 1 (2 points). Consider the following probabilistic complete graphs, along with their attached probabilities:



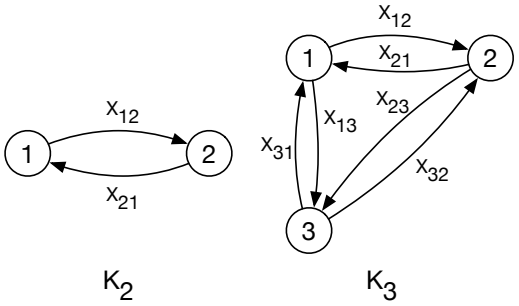
Compute the probability for $\mathsf{Reach}(1, 2)$ in the graph K_2 above. Do the same in the graph K_3 above.

Answer. In K_2, the probability $\mathsf{Reach}(1, 2)$ is the probability of the edge $1 \to 2$. Hence, $\mathsf{Reach}(1, 2) = 0.8$.

In K_3, the probability $\mathsf{Reach}(1, 2)$ is derived from the probability of the edge $1 \to 2$ and the path going from the edges $1 \to 3$, $3 \to 2$:

$$\mathsf{Reach}(1, 2) = 1 - (1 - 0.5)(1 - 1 \times 0.8) = 1 - 0.1 = 0.9.$$

Question 2 (2 points). Consider the general case of the K_2 and K_3 graphs, where each edge is annotated with an independent probabilistic event:



Remember that a *lineage* of a query is a Boolean formula over the uncertain events of the graph which evaluates to true iff the query is true when the corresponding edges are kept; and that the lineage is *read-once* if it can be written in a form where each variable occurs at most once.

Show that, for any $n \in \{2, 3\}$, for any vertices $s \neq t$ in K_n, any lineage for $\mathsf{Reach}(s, t)$ on K_n is read-once.

Answer. Take K_2. The only simple path between two nodes in this graph is the edge between them, so that a lineage consists of the single occurrence of that variable. Hence, the lineage of $\mathsf{Reach}(1, 2)$ is X_{12} and the lineage of $\mathsf{Reach}(2, 1)$ is X_{21}, which are both read-once formulas.

Consider now K_3. The lineage of any reachability query between two nodes $s \neq t$ is the conjunction of the only two simple paths from s to t: the direct edge $s \to t$ and the two-hop path going through the third node u (with $u \neq s$ and $u \neq t$), namely, $s \to u$, $u \to t$. Hence, the lineage can be written as the following lineage formula:

$$X_{st} \vee (X_{su} \wedge X_{ut}).$$