On the Connections between Relational and XML Probabilistic Data Models

Antoine Amarilli [1] Pierre Senellart [2]

[1]École normale supérieure, Paris, France
& University of Oxford, Oxford, United Kingdom

[2]Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI, Paris, France

Table of contents

Probabilistic data

A probabilistic instance is a finite collection of possible states of the data (the possible worlds) with associated probability.

$p = 0.16$

document	topic
#42	bncod
#42	oxford

$p = 0.04$

document	topic
#42	oxford

$p = 0.64$

document	topic
#42	bncod

$p = 0.16$

document	topic

Efficient representation

We want to represent the possible worlds in a concise manner:

document	**topic**	p
#42	bncod	0.8
#42	oxford	0.2

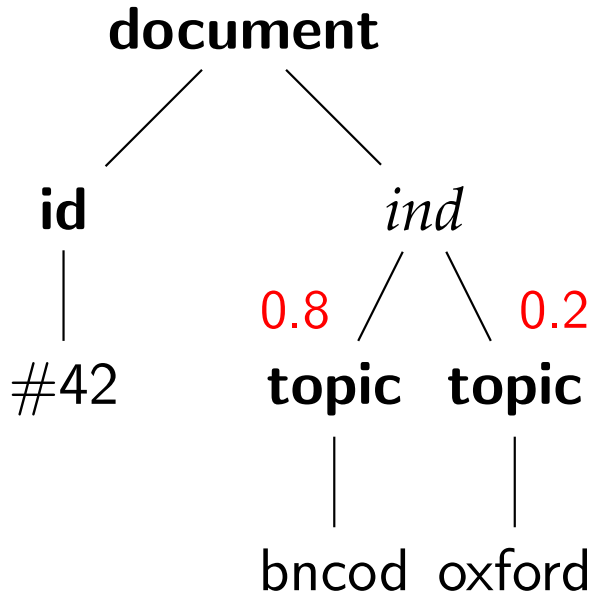
We want to evaluate queries efficiently on all possible worlds:

document	p
#42	0.8

$q(x)$: SELECT **document** WHERE **topic** = 'bncod'

Relational data and XML

Probabilistic relational and XML data models have been developed in isolation.



document	topic	p
#42	bncod	0.8
#42	oxford	0.2

We will show how query complexity results in both models can be connected through encodings from one model to the other.

Relational tuple-independent model

Each tuple is annotated with a probability score independently from other tuples.

document	**topic**	p
#42	bncod	0.8
#42	oxford	0.2

This is not a very expressive model! For instance:

conference	**loc**	**iso**	p
bncod	oxford	gb	0.8
bncod	london	gb	0.2

Relational block-independent-disjoint model

Use a key to divide the relation attributes. For one value of the key (a block), choose at most one of the matching rows, independently between blocks.

name	**city**	**iso**	p
bncod	oxford	gb	0.8
bncod	london	gb	0.2
icalp	riga	lv	0.9
icalp	riga	lt	0.1

Complexity results

We will be interested in conjunctive queries (CQs), unions of conjunctive queries (UCQs), and the relational algebra. We always study data complexity.

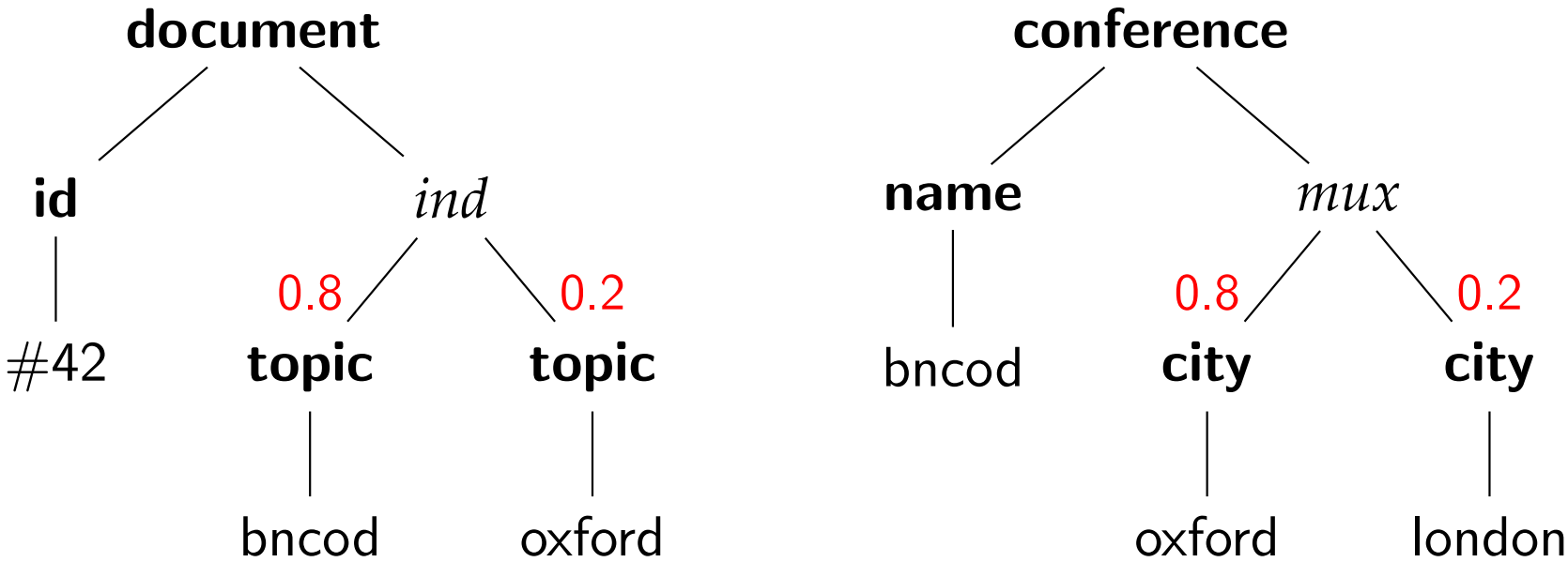
- Relational algebra evaluation over BIDs is in $\mathrm{FP}^{\#\mathrm{P}}$ [DS07b].
- CQ evaluation over tuple-independent databases is $\mathrm{FP}^{\#\mathrm{P}}$-hard.

Finer results exist:

- UCQs over tuple-independent databases : dichotomy between $\mathrm{FP}^{\#\mathrm{P}}$-hard queries and FP queries [DSS10] with a doubly exponential time test (exact complexity open).
- CQs without self-joins over BID: similar dichotomy, polynomial-time test [DS07b].
- CQ without self-joins over tuple-independent: $\mathrm{FP}^{\#\mathrm{P}}$-hard iff not hierarchical. [DS07a]. Being non-hierarchical is a sufficient condition for $\mathrm{FP}^{\#\mathrm{P}}$-hardness of any relational calculus query.

PrXML$^{\{ind,mux\}}$ model

Documents with additional *ind* nodes to keep or remove children independently, and *mux* nodes to choose at most one child.



Complexity results

- Tree-pattern queries (with joins) (TPQ(J)s): tree patterns labeled with constants and variables (possibly multiple occurrences), with child and descendent edges.

Results:

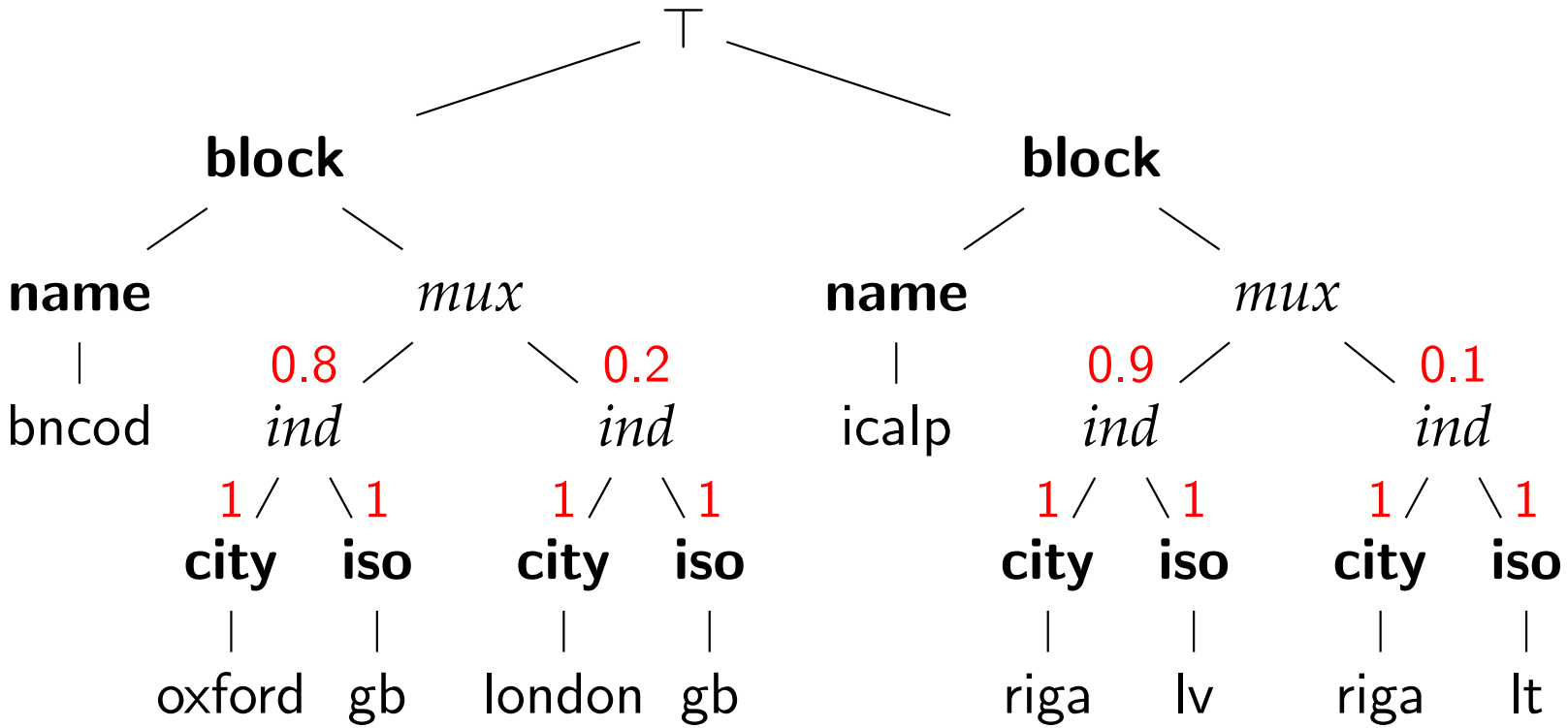
- TPQJ evaluation over $\text{PrXML}^{\{\text{ind,mux}\}}$ is in $\text{FP}^{\#\text{P}}$ [KNS11].
- TPQJ evaluation over $\text{PrXML}^{\{\text{ind,mux}\}}$ is $\text{FP}^{\#\text{P}}$-hard.

More precisely:

- TPQs over $\text{PrXML}^{\{\text{ind,mux}\}}$ are linear time (actually also true for MSO [CKS09]).
- TPQJs with a single join over $\text{PrXML}^{\{\text{ind,mux}\}}$ are $\text{FP}^{\#\text{P}}$-hard if not equivalent to a join-free query [KNS11], with a Σ_2^{P}-complete test.

Relational to XML

We can encode any BID table (and hence any tuple-independent table) to PrXML$^{\{\text{mux,ind}\}}$ in linear time:



Likewise, we can encode CQs to TPQJs in linear time.

Complexity consequences

Some results from the relational or XML world can thus be reproven from the corresponding result in the other world:

- TPQJ evaluation over $\text{PrXML}^{\{\text{ind,mux}\}}$ is in $\text{FP}^{\#\text{P}}$
  $\Rightarrow$ CQ evaluation over BIDs is in $\text{FP}^{\#\text{P}}$
- CQ evaluation over tuple-independent databases is $\text{FP}^{\#\text{P}}$-hard
  $\Rightarrow$ TPQJ evaluation over $\text{PrXML}^{\{\text{ind}\}}$ is $\text{FP}^{\#\text{P}}$-hard;
  $\Rightarrow$ We can identify examples of hard TPQJ query classes (e.g., the image of classes of non-hierarchical CQs)
- MSO evaluation over $\text{PrXML}^{\{\text{ind,mux}\}}$ is linear time
  $\Rightarrow$ Read-once relational algebra on BID is linear time

We cannot encode $\text{PrXML}^{\{\text{ind,mux}\}}$ to BIDs: the result of a query over $\text{PrXML}^{\{\text{ind,mux}\}}$ can be represented in $\text{PrXML}^{\{\text{ind,mux}\}}$ whereas BIDs are not a strong representation system for CQs.

Relational *pc*-tables

Each tuple is annotated with a Boolean condition over variables drawn independently with an associated probability.

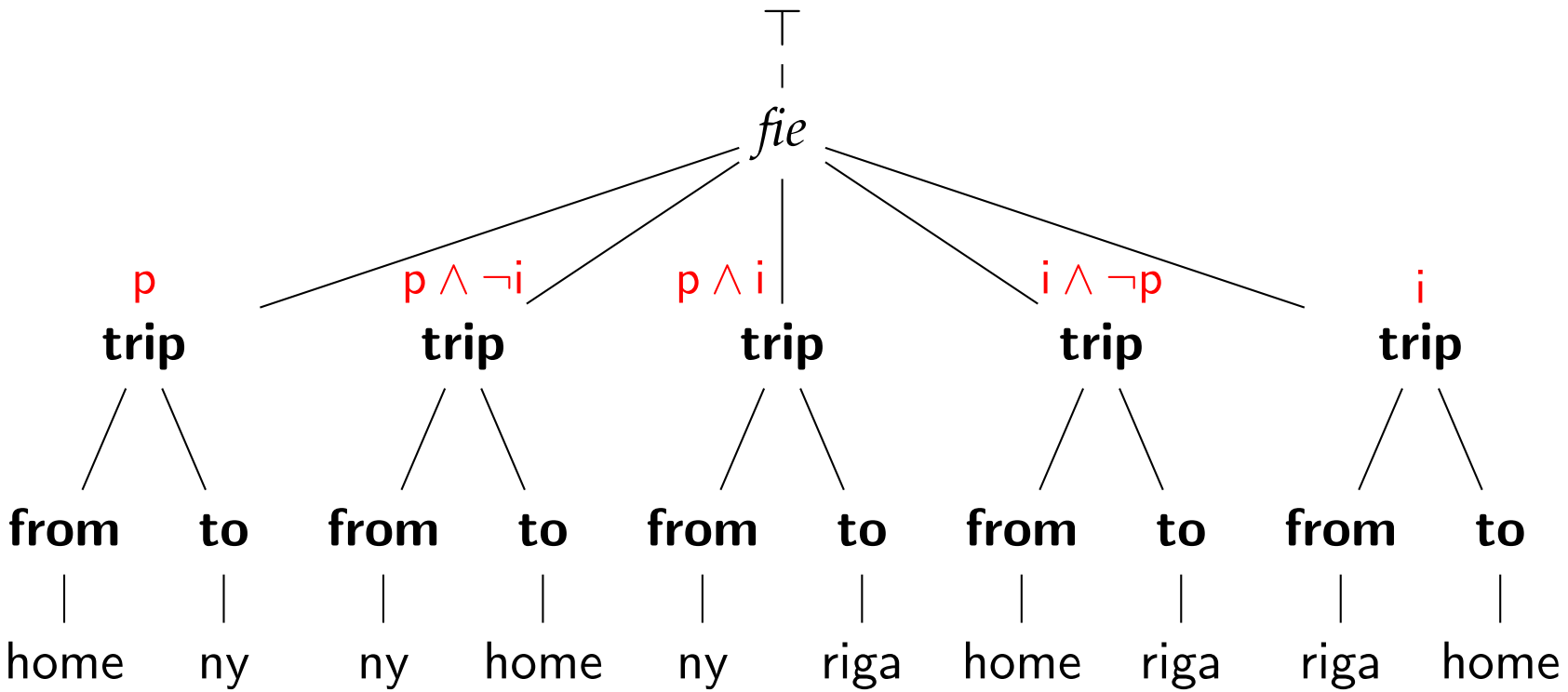
from	**to**	C
home	ny	pods
ny	home	pods $\wedge \neg$icalp
ny	riga	icalp $\wedge$ pods
home	riga	icalp $\wedge \neg$pods
riga	home	icalp

$p(\text{pods}) = 0.2, \quad p(\text{icalp}) = 0.2$

This can represent any finite distribution. However, evaluating a single-atom CQ over a *pc*-table with only conjunctions is already $\text{FP}^{\#\text{P}}$-hard.

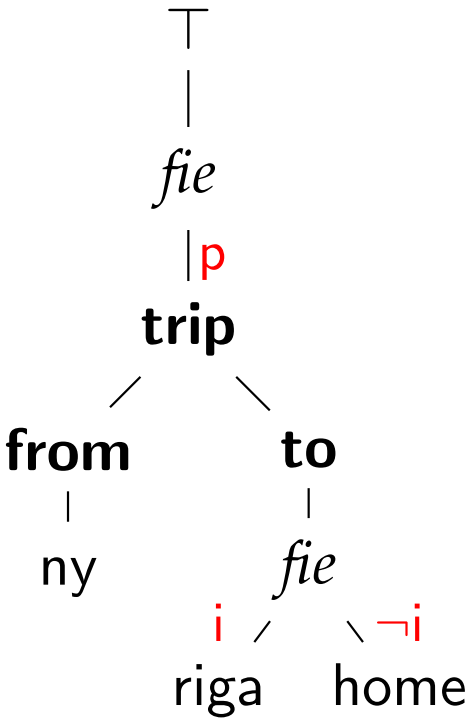
PrXML$^{\{\text{fie}\}}$ model

Boolean conditions on children can be expressed with *fie* nodes like in *pc*-tables.



Encodings

As before, we can encode *pc*-tables in PrXML$^{\{\text{fie}\}}$.
In the other direction, we materialize the descendant relation to handle descendant edges in TPQJs and the construction is cubic time.



node	desc	C
⊤	**trip**	p
⊤	**from**	p
⊤	ny	p
⊤	**to**	p
⊤	riga	p ∧ i
⊤	home	p ∧ ¬i
trip	**from**	t
trip	ny	t
trip	**to**	t
trip	riga	i
trip	home	¬i
from	ny	t
to	riga	i
to	home	¬i

node	child	C
⊤	**trip**	p
trip	**from**	t
trip	**to**	t
to	riga	icalp
to	home	¬icalp

Complexity consequences

- TPQJ evaluation over $\text{PrXML}^{\{\text{fie}\}}$ is in $\text{FP}^{\#\text{P}}$
  - $\Rightarrow$ CQ evaluation over *pc*-tables is in $\text{FP}^{\#\text{P}}$
- CQ evaluation over *pc*-tables is in $\text{FP}^{\#\text{P}}$
  - $\Rightarrow$ TPQJ evaluation over $\text{PrXML}^{\{\text{fie}\}}$ is in $\text{FP}^{\#\text{P}}$
- Certain TPQJ queries are $\text{FP}^{\#\text{P}}$-hard over $\text{PrXML}^{\{\text{fie}\}}$
  - $\Rightarrow$ Examples of $\text{FP}^{\#\text{P}}$-hard relational queries

Summary and open problems

- The query evaluation problems on relational and XML probabilistic data models have been studied in isolation.
- Simple encodings shows that the broad results ($\mathrm{FP}^{\#\mathrm{P}}$ membership and hardness) can be transferred from one setting to the other.
- There is no straightforward correspondence for finer results (e.g., dichotomies).
- The connection between the two settings could be used to suggest tractable classes of queries and instances that are the preimage of a known tractable class in the other setting.

Summary and open problems

- The query evaluation problems on relational and XML probabilistic data models have been studied in isolation.
- Simple encodings shows that the broad results ($FP^{\#P}$ membership and hardness) can be transferred from one setting to the other.
- There is no straightforward correspondence for finer results (e.g., dichotomies).
- The connection between the two settings could be used to suggest tractable classes of queries and instances that are the preimage of a known tractable class in the other setting.

Thanks for your attention!

References

- Sara Cohen, Benny Kimelfeld, and Yehoshua Sagiv, *Running tree automata on probabilistic XML*, PODS, 2009.
- Nilesh N. Dalvi and Dan Suciu, *Efficient query evaluation on probabilistic databases*, VLDB Journal **16** (2007), no. 4.
- _____, *Management of probabilistic data: foundations and challenges*, PODS, 2007.
- Nilesh N. Dalvi, Karl Schnaitter, and Dan Suciu, *Computing query probability with incidence algebras*, PODS, 2010.
- Evgeny Kharlamov, Werner Nutt, and Pierre Senellart, *Value joins are expensive over (probabilistic) XML*, Proc. LID, March 2011.