Combined Complexity of Probabilistic Query Evaluation

Mikaël Monet

October 18th, 2019

Millennium Institute for Foundational Research on Data, Chile

Context

- At Télécom ParisTech, supervised by **Pierre Senellart** and **Antoine Amarilli**
- (Now doing a postdoc with Pablo Barceló at IMFD, Chile)

Context

- At Télécom ParisTech, supervised by **Pierre Senellart** and **Antoine Amarilli**
- (Now doing a postdoc with Pablo Barceló at IMFD, Chile)

Context of my thesis:

- Real life data can be **uncertain**
  - → imperfect sensor precision, error-prone automatic information extraction processes, data integration from multiple sources, missing information, etc.
- Querying this data without taking this uncertainty into account can lead to incorrect answers

Context

- At Télécom ParisTech, supervised by **Pierre Senellart** and **Antoine Amarilli**
- (Now doing a postdoc with Pablo Barceló at IMFD, Chile)

Context of my thesis:

- Real life data can be **uncertain**
  - → imperfect sensor precision, error-prone automatic information extraction processes, data integration from multiple sources, missing information, etc.
- Querying this data without taking this uncertainty into account can lead to incorrect answers

→ Probabilistic databases: a framework to model data uncertainty and reason about it

Tuple-independent databases (TID)

- A tuple-independent database (TID) (D, π) consists of:
  - A relational database D
  - A probability valuation π mapping each fact of D to $[0, 1]$

Tuple-independent databases (TID)

- A tuple-independent database (TID) (D, π) consists of:
  - A relational database D
  - A probability valuation π mapping each fact of D to $[0, 1]$

- Semantics of a TID (D, π): a probability distribution on $D' \subseteq D$:
  - Each fact $F \in D$ is either present or absent with probability $\pi(F)$
  - Assume independence across facts

  $\rightarrow$ For $D' \subseteq D$, $\Pr(D') = (\prod_{F \in D'} \pi(F)) \times (\prod_{F \in D \setminus D'} (1 - \pi(F)))$

Tuple-independent databases (TID)

- A tuple-independent database (TID) (D, π) consists of:
  - A relational database D
  - A probability valuation π mapping each fact of D to $[0, 1]$

- Semantics of a TID (D, π): a probability distribution on $D' \subseteq D$:
  - Each fact $F \in D$ is either present or absent with probability $\pi(F)$
  - Assume independence across facts

  $\rightarrow$ For $D' \subseteq D$, $\Pr(D') = (\prod_{F \in D'} \pi(F)) \times (\prod_{F \in D \setminus D'} (1 - \pi(F)))$

$\rightarrow$ **Succinctly** represent probabilistic data

Example: TID

D $=$

R	
a	b
a	c

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$.5 \times .2$	
R	
a	b
a	c

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$.5 \times .2$	
R	
a	b
a	c

$.5 \times (1 - .2)$	
R	
a	b

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$
R	R	R
a b	a b	
a c		a c

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$q = \exists x\, y\ R(x, y)$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	

$\Pr(D \models q) =$

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$q = \exists x\, y\ R(x, y)$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	
✔			

$\Pr(D \models q) =$

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$q = \exists x\, y\ R(x, y)$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	

$\Pr(D \models q) = .5 \times .2$

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$q = \exists x\, y\ R(x, y)$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	
✔	✔		

$\Pr(D \models q) = .5 \times .2$

Example: TID

$$(D, \pi) = \quad \begin{array}{|ccc|} \hline \multicolumn{3}{|c|}{\mathbf{R}} \\ \hline a & b & .5 \\ a & c & .2 \\ \hline \end{array} \qquad q = \exists x\, y\ R(x, y)$$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	**R**	**R**	**R**
a b	a b		
a c		a c	
✔	✔		

$$\Pr(D \models q) = .5 \times .2 + .5 \times (1 - .2)$$

Example: TID

$(D, \pi) =$

R		
a	b	.5
a	c	.2

$q = \exists x\, y\ R(x, y)$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	R	R	R
a b	a b		
a c		a c	
✔	✔	✔	

$\Pr(D \models q) = .5 \times .2 + .5 \times (1 - .2)$

Example: TID

$$(D, \pi) = \quad \begin{array}{|ccc|} \hline \multicolumn{3}{|c|}{\mathbf{R}} \\ \hline a & b & .5 \\ a & c & .2 \\ \hline \end{array} \qquad q = \exists x\, y\ R(x, y)$$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	**R**	**R**	**R**
a b	a b		
a c		a c	
✔	✔	✔	

$$\Pr(D \models q) = .5 \times .2 + .5 \times (1 - .2) + (1 - .5) \times .2$$

Example: TID

$$(D, \pi) = \quad \begin{array}{|ccc|} \hline \multicolumn{3}{|c|}{\mathbf{R}} \\ \hline a & b & .5 \\ a & c & .2 \\ \hline \end{array} \qquad q = \exists x\, y\ R(x, y)$$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	**R**	**R**	**R**
a b	a b		
a c		a c	
✔	✔	✔	✘

$$\Pr(D \models q) = .5 \times .2 + .5 \times (1 - .2) + (1 - .5) \times .2$$

Example: TID

$$(D, \pi) = \quad \begin{array}{|ccc|} \hline \multicolumn{3}{|c|}{\mathbf{R}} \\ \hline a & b & .5 \\ a & c & .2 \\ \hline \end{array} \qquad q = \exists x\, y\ R(x, y)$$

$.5 \times .2$	$.5 \times (1 - .2)$	$(1 - .5) \times .2$	$(1 - .5) \times (1 - .2)$
R	**R**	**R**	**R**
a b	a b		
a c		a c	
✔	✔	✔	✘

$$\Pr(D \models q) = .5 \times .2 + .5 \times (1 - .2) + (1 - .5) \times .2$$

$$= 1 - [(1 - .5) \times (1 - .2)]$$

Probabilistic query evaluation (PQE)

Let us fix:

- Class $\mathcal{D}$ of relational databases
- Class $\mathcal{Q}$ of Boolean queries

Probabilistic query evaluation (PQE)

Let us fix:

- Class $\mathcal{D}$ of relational databases
- Class $\mathcal{Q}$ of Boolean queries

Probabilistic query evaluation (PQE) problem for $\mathcal{Q}$ and $\mathcal{D}$:

- Given a query $q \in \mathcal{Q}$
- Given a database $D \in \mathcal{D}$ and a probability valuation π
- Compute the probability that (D, π) satisfies q

Probabilistic query evaluation (PQE)

Let us fix:

- Class $\mathcal{D}$ of relational databases
- Class $\mathcal{Q}$ of Boolean queries

Probabilistic query evaluation (PQE) problem for $\mathcal{Q}$ and $\mathcal{D}$:

- Given a query $q \in \mathcal{Q}$
- Given a database $D \in \mathcal{D}$ and a probability valuation π
- Compute the probability that (D, π) satisfies q

$\rightarrow \Pr((D, \pi) \models q) = \sum_{D' \subseteq D,\ D' \models q} \Pr(D')$

Complexity of probabilistic query evaluation (PQE)

Question: what is the (data, combined) complexity of PQE depending on the class $\mathcal{D}$ of databases and class $\mathcal{Q}$ of queries?

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data
  - $\mathcal{D}_k$ = all databases whose treewidth is bounded by $k \in \mathbb{N}$

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data
  - $\mathcal{D}_k$ = all databases whose treewidth is bounded by $k \in \mathbb{N}$
  - $\mathcal{Q} = \{q\}$, for q a MSO query

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data
  - $\mathcal{D}_k$ = all databases whose treewidth is bounded by $k \in \mathbb{N}$
  - $\mathcal{Q} = \{q\}$, for q a MSO query

  $\rightarrow$ PQE has linear data complexity [Amarilli et al., 2015]

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data
  - $\mathcal{D}_k$ = all databases whose treewidth is bounded by $k \in \mathbb{N}$
  - $\mathcal{Q} = \{q\}$, for q a MSO query

  $\rightarrow$ PQE has linear data complexity [Amarilli et al., 2015]

  $\rightarrow$ (Conversely, there is an FO query for which PQE is **#P-hard** on any unbounded-treewidth database class $\mathcal{D}$ (under some assumptions))

Data complexity results: related work

- Data dichotomy result on queries [Dalvi and Suciu, 2012]
  - $\mathcal{D}_{\text{all}}$ = all possible databases
  - $\mathcal{Q} = \{q\}$, for q a UCQ ($\approx$ SQL with **select**, **where**, **union**)

  $\rightarrow$ q is 'safe' $\implies$ PQE is PTIME

  $\rightarrow$ q is not 'safe' $\implies$ PQE is **#P-hard**

- Data dichotomy result on data
  - $\mathcal{D}_k$ = all databases whose treewidth is bounded by $k \in \mathbb{N}$
  - $\mathcal{Q} = \{q\}$, for q a MSO query

  $\rightarrow$ PQE has linear data complexity [Amarilli et al., 2015]

  $\rightarrow$ (Conversely, there is an FO query for which PQE is **#P-hard** on any unbounded-treewidth database class $\mathcal{D}$ (under some assumptions))

What about combined complexity?

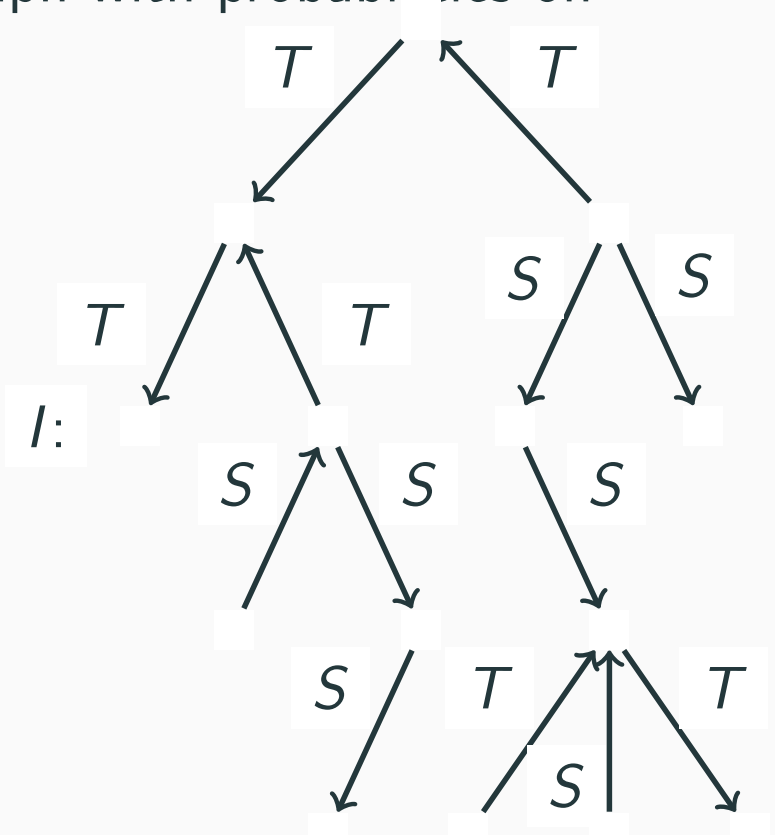
My thesis in 3 slides

1) Combined complexity of probabilistic graph homomorphism

- **With:** Antoine Amarilli, Pierre Senellart.
- **What:** Compute the probability that a (query) graph has a homomorphism to another (data) graph with probabilities on the edges

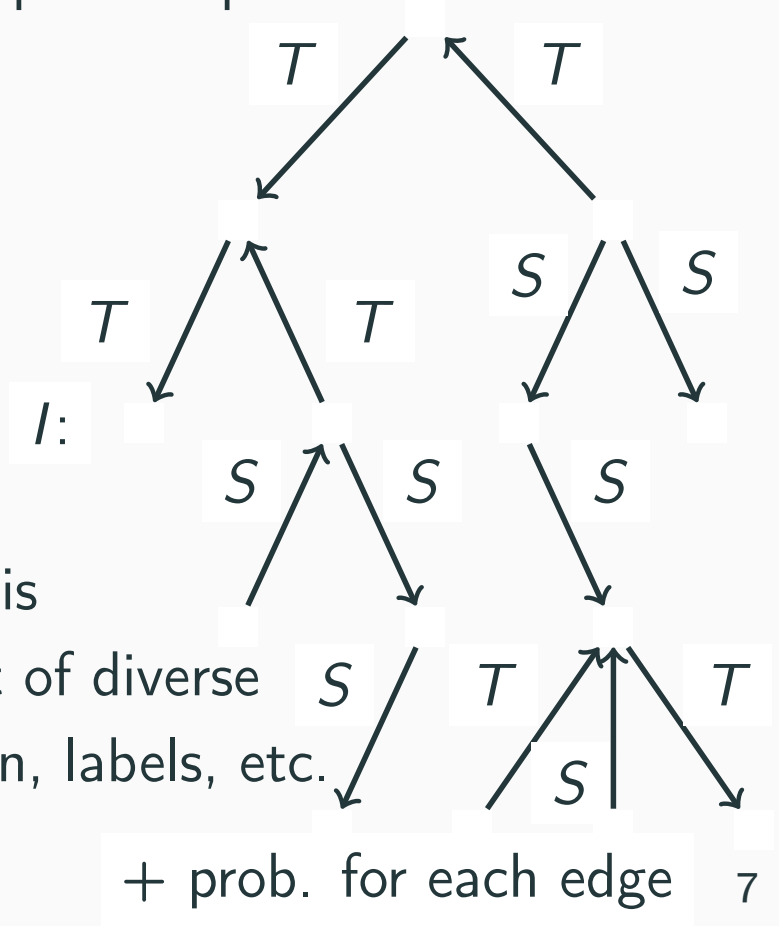
1) Combined complexity of probabilistic graph homomorphism

- **With:** Antoine Amarilli, Pierre Senellart.
- **What:** Compute the probability that a (query) graph has a homomorphism to another (data) graph with probabilities on the edges



1) Combined complexity of probabilistic graph homomorphism

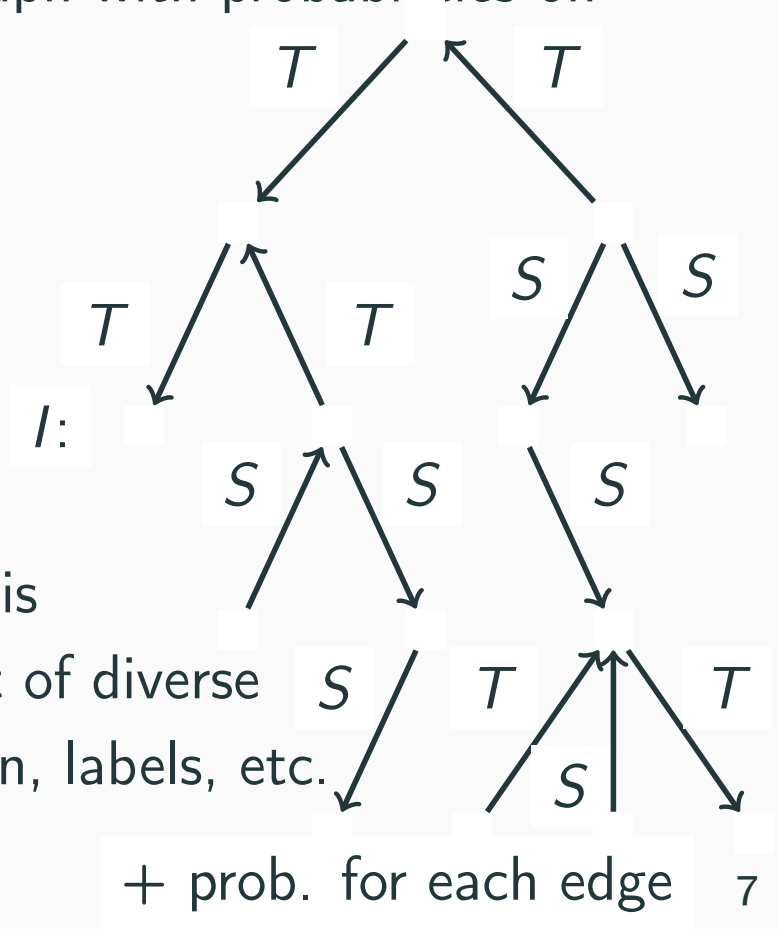
- **With:** Antoine Amarilli, Pierre Senellart.
- **What:** Compute the probability that a (query) graph has a homomorphism to another (data) graph with probabilities on the edges



→ Study the combined complexity of this problem for trees, analyse the impact of diverse features: branching, global orientation, labels, etc.

1) Combined complexity of probabilistic graph homomorphism

- **With:** Antoine Amarilli, Pierre Senellart.
- **What:** Compute the probability that a (query) graph has a homomorphism to another (data) graph with probabilities on the edges



→ Study the combined complexity of this problem for trees, analyse the impact of diverse features: branching, global orientation, labels, etc.

→ Dichotomies into FP or #P-hard

2) Combined complexity of querying bounded treewidth data

- **With:** Antoine Amarilli, Pierre Bourhis, Pierre Senellart.
- (treewidth: a measure telling how far a database is from being "acyclic")

2) Combined complexity of querying bounded treewidth data

- **With:** Antoine Amarilli, Pierre Bourhis, Pierre Senellart.
- (treewidth: a measure telling how far a database is from being "acyclic")
- **What:** We introduce a fragment of Datalog with negations (called CFG-Datalog) such that, when parameterized by the body size k_Q of the Datalog query Q and by the treewidth k_D of the database D, then:
  - $\rightarrow$ We can compute in time $f(k_D, k_Q) \times |Q| \times |D|$ (*Fixed Parameter Tractability*) a *cyclic Boolean circuit* (called cycluit) representing the provenance of Q on D

2) Combined complexity of querying bounded treewidth data

- **With:** Antoine Amarilli, Pierre Bourhis, Pierre Senellart.
- (treewidth: a measure telling how far a database is from being "acyclic")
- **What:** We introduce a fragment of Datalog with negations (called CFG-Datalog) such that, when parameterized by the body size k_Q of the Datalog query Q and by the treewidth k_D of the database D, then:
  - → We can compute in time $f(k_D, k_Q) \times |Q| \times |D|$ (*Fixed Parameter Tractability*) a *cyclic Boolean circuit* (called cycluit) representing the provenance of Q on D
  - → If D is a probabilistic tuple-independent database (TID), then we can compute in time $f(k_D, k_Q) \times 2^{2^{\mathrm{poly}(|Q|)}} \times |D|$ the probability that D satisfies Q

3) Connecting knowledge compilation classes and width parameter

- **With:** Antoine Amarilli, Florent Capelli, Pierre Senellart.
- **What:** We connect the (tree-, path-) width of Boolean circuits (or CNF formulas, or DNF) with formalisms used in knowledge compilation for representing Boolean functions, such as *Ordered Binary Decision Diagrams* (OBDDs), *Deterministic Decomposable Negation Normal Forms* (d-DNNFs), etc.

3) Connecting knowledge compilation classes and width parameter

- **With:** Antoine Amarilli, Florent Capelli, Pierre Senellart.
- **What:** We connect the (tree-, path-) width of Boolean circuits (or CNF formulas, or DNF) with formalisms used in knowledge compilation for representing Boolean functions, such as *Ordered Binary Decision Diagrams* (OBDDs), *Deterministic Decomposable Negation Normal Forms* (d-DNNFs), etc.
  - → Some upper bounds, e.g.: given a Boolean circuit C of treewidth k, we can compute in time $2^{ck} \times |C|$ an equivalent (structured) d-DNNF
  - → Some lower bounds, e.g.: the size of any structured d-DNNF for a monotone DNF formula F must be exponential in the treewidth of F

Thanks for your attention!

Bibliography I


Amarilli, A., Bourhis, P., and Senellart, P. (2015).
Provenance circuits for trees and treelike instances.
In *Proceedings, Part II, of the 42Nd International Colloquium on Automata, Languages, and Programming - Volume 9135*, ICALP 2015, pages 56–68, Berlin, Heidelberg. Springer-Verlag.


Dalvi, N. N. and Suciu, D. (2012).
The dichotomy of probabilistic inference for unions of conjunctive queries.
Journal of the ACM, 59(6):30.