Tractable Query Answering Under Probabilistic Constraints

Antoine Amarilli[1], Pierre Bourhis[2], Pierre Senellart[1,3]

[1]Télécom ParisTech

[2]CNRS-LIFL

[3]National University of Singapore

September 4th, 2014

Tractable Query Evaluation On Probabilistic Instances

Antoine Amarilli[1], Pierre Bourhis[2], Pierre Senellart[1,3]

[1]Télécom ParisTech

[2]CNRS-LIFL

[3]National University of Singapore

September 4th, 2014

Instances and queries

- Given a relational instance (= set of facts, hypergraph)
  $I = \{R(a, b), R(b, c), S(c)\}$
- Given a conjunctive query (CQ) (existentially quantified)
  $q : \exists xy\, R(x, y) \wedge S(y)$

Instances and queries

- Given a relational instance (= set of facts, hypergraph)
  $I = \{R(a, b), R(b, c), S(c)\}$
- Given a conjunctive query (CQ) (existentially quantified)
  $q : \exists xy\, R(x, y) \wedge S(y)$

$\rightarrow$ Query evaluation (model checking) of q on I

Instances and queries

- Given a relational instance (= set of facts, hypergraph)
  $I = \{R(a, b), R(b, c), S(c)\}$
- Given a conjunctive query (CQ) (existentially quantified)
  $q : \exists xy\, R(x, y) \wedge S(y)$

$\rightarrow$ Query evaluation (model checking) of q on I

Instances and queries

- Given a relational instance (= set of facts, hypergraph)
  $I = \{R(a,b), R(b,c), S(c)\}$
- Given a conjunctive query (CQ) (existentially quantified)
  $q : \exists xy\, R(x,y) \wedge S(y)$

$\rightarrow$ Query evaluation (model checking) of q on I

Instances and queries

- Given a relational instance (= set of facts, hypergraph)
  $I = \{R(a, b), R(b, c), S(c)\}$
- Given a conjunctive query (CQ) (existentially quantified)
  $q : \exists xy\, R(x, y) \wedge S(y)$

$\rightarrow$ Query evaluation (model checking) of q on I

$\rightarrow$ Data complexity: q is fixed

Uncertain and probabilistic instances

- Set of uncertain events

e_{flight} CDG $\rightarrow$ VIE flight AF1756 takes place
e_{bus} Vienna $\rightarrow$ Bratislava buses are running

Uncertain and probabilistic instances

- Set of uncertain events

e_{flight}	CDG $\rightarrow$ VIE flight AF1756 takes place
e_{bus}	Vienna $\rightarrow$ Bratislava buses are running

- Annotate instance facts with formulae on the events

IsIn(AA, Paris)	$\neg e_{\text{flight}}$
IsIn(AA, Vienna)	$e_{\text{flight}} \land \neg e_{\text{bus}}$
IsIn(AA, Bratislava)	$e_{\text{flight}} \land e_{\text{bus}}$

Uncertain and probabilistic instances

- Set of uncertain events

e_{flight} CDG $\rightarrow$ VIE flight AF1756 takes place
e_{bus} Vienna $\rightarrow$ Bratislava buses are running

- Annotate instance facts with formulae on the events

IsIn(AA, Paris)	$\neg e_{\text{flight}}$
IsIn(AA, Vienna)	$e_{\text{flight}} \wedge \neg e_{\text{bus}}$
IsIn(AA, Bratislava)	$e_{\text{flight}} \wedge e_{\text{bus}}$

$\rightarrow$ Semantics: a set of instances (possible worlds).

Uncertain and probabilistic instances

- Set of uncertain events

e_{flight} CDG $\rightarrow$ VIE flight AF1756 takes place
e_{bus} Vienna $\rightarrow$ Bratislava buses are running

- Annotate instance facts with formulae on the events

IsIn(AA, Paris)	$\neg e_{\text{flight}}$
IsIn(AA, Vienna)	$e_{\text{flight}} \wedge \neg e_{\text{bus}}$
IsIn(AA, Bratislava)	$e_{\text{flight}} \wedge e_{\text{bus}}$

$\rightarrow$ Semantics: a set of instances (possible worlds).

- Add a probability distribution on each event
  - each event has probability $0 < p < 1$ of being true
  - all events are assumed to be independent

Uncertain and probabilistic instances

- Set of uncertain events

  e_{flight} CDG $\to$ VIE flight AF1756 takes place

  e_{bus} Vienna $\to$ Bratislava buses are running

- Annotate instance facts with formulae on the events

IsIn(AA, Paris)	$\neg e_{\text{flight}}$
IsIn(AA, Vienna)	$e_{\text{flight}} \land \neg e_{\text{bus}}$
IsIn(AA, Bratislava)	$e_{\text{flight}} \land e_{\text{bus}}$

$\to$ Semantics: a set of instances (possible worlds).

- Add a probability distribution on each event
  - each event has probability $0 < p < 1$ of being true
  - all events are assumed to be independent

$\to$ Semantics: a probability distribution on instances.

Uncertain and probabilistic instances

- Set of uncertain events

$$\begin{array}{rl} e_{\text{flight}} & \text{CDG} \to \text{VIE flight AF1756 takes place} \\ e_{\text{bus}} & \text{Vienna} \to \text{Bratislava buses are running} \end{array}$$

- Annotate instance facts with formulae on the events

IsIn(AA, Paris)	$\neg e_{\text{flight}}$
IsIn(AA, Vienna)	$e_{\text{flight}} \wedge \neg e_{\text{bus}}$
IsIn(AA, Bratislava)	$e_{\text{flight}} \wedge e_{\text{bus}}$

$\rightarrow$ Semantics: a set of instances (possible worlds).

- Add a probability distribution on each event
  - each event has probability $0 < p < 1$ of being true
  - all events are assumed to be independent

$\rightarrow$ Semantics: a probability distribution on instances.

$\rightarrow$ Query evaluation: determine the probability of q on $\widehat{I}$.

Hardness and tractability

- With arbitrary annotations
  - → Query evaluation is #P-hard even with a single fact (Immediate reduction from #SAT)
- With simple annotations (one unique event per tuple)
  - → Query evaluation is #P-hard on arbitrary instances (Use the instance to do the reduction)

Hardness and tractability

- With arbitrary annotations
  - → Query evaluation is #P-hard even with a single fact (Immediate reduction from #SAT)
- With simple annotations (one unique event per tuple)
  - → Query evaluation is #P-hard on arbitrary instances (Use the instance to do the reduction)
- Existing work:
  - → Fix a simple annotation scheme
  - → Show dichotomy between #P-hard and PTIME queries

Hardness and tractability

- With arbitrary annotations
  - → Query evaluation is #P-hard even with a single fact (Immediate reduction from #SAT)
- With simple annotations (one unique event per tuple)
  - → Query evaluation is #P-hard on arbitrary instances (Use the instance to do the reduction)
- Existing work:
  - → Fix a simple annotation scheme
  - → Show dichotomy between #P-hard and PTIME queries
- Our approach:
  - → Find a restriction on the instance and annotations
  - → Show that many queries are tractable in this case

Bounded treewidth

An idea from instances without probabilities...

- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...

Bounded treewidth

An idea from instances without probabilities...

- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...

instance I

$R(a, b)$ $R(b, c)$ $S(c)$

Bounded treewidth

An idea from instances without probabilities...

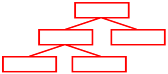
- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...

instance I $\longrightarrow$ tree encoding T_I

$R(a, b)$ $R(b, c)$ $S(c)$

tree decomposition

$O(|I|)$ for fixed width



Bounded treewidth

An idea from instances without probabilities...

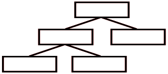
- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...

instance I $\longrightarrow$ tree encoding T_I

$R(a, b)$ $R(b, c)$ $S(c)$

tree decomposition

$O(|I|)$ for fixed width



$\exists xy\ R(x, y) \wedge S(y)$

query q

Bounded treewidth

An idea from instances without probabilities...

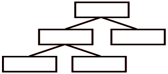
- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...

instance I ⟶ tree encoding T_I

$R(a, b)$ $R(b, c)$ $S(c)$

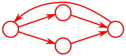
tree decomposition

$O(|I|)$ for fixed width



rewriting

$O(1)$ data complexity



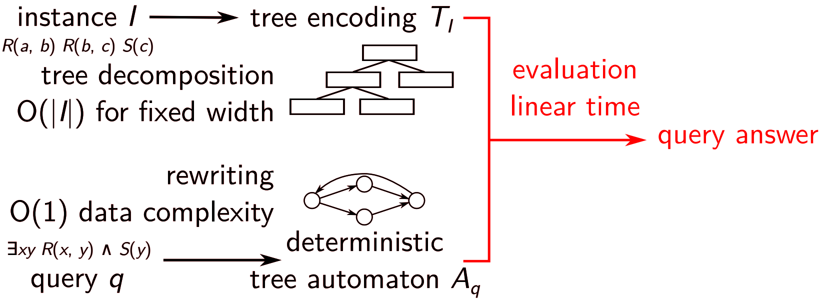
$\exists xy\, R(x, y) \land S(y)$

query q ⟶ deterministic tree automaton A_q

Bounded treewidth

An idea from instances without probabilities...

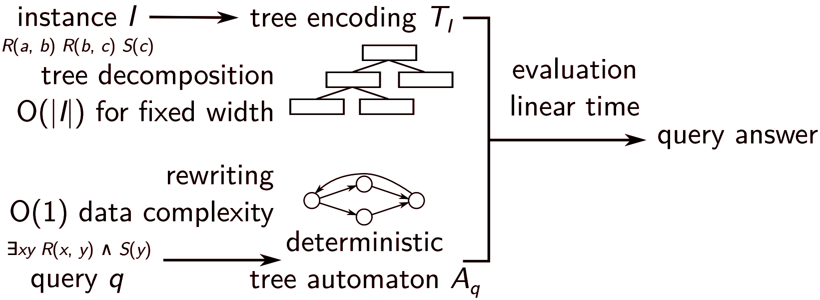
- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...



Bounded treewidth

An idea from instances without probabilities...

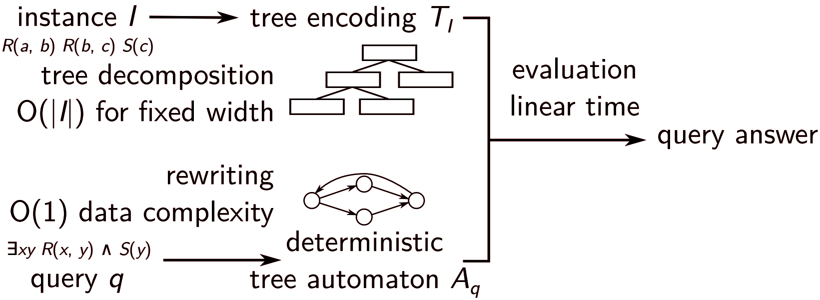
- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...



Bounded treewidth

An idea from instances without probabilities...

- If an instance has low treewidth then it is almost a tree
- Assume that the instance treewidth is constant...



→ Linear time data complexity

Tractable inference

An idea from probabilities without instances...

- Represent a propositional formula F as a Boolean circuit
- Assume the circuit has constant treewidth

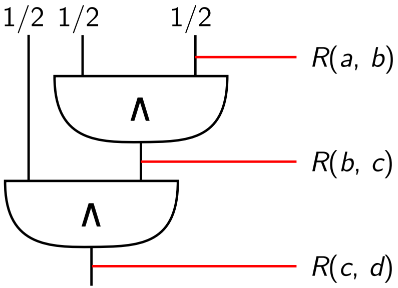
$\rightarrow$ Probability of F can be computed in linear time
(using junction tree algorithm for Bayesian networks)
(assuming constant-time arithmetic operations)

cc-tables

- Boolean circuit for the annotations

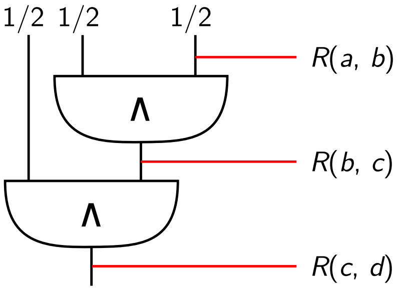
cc-tables

- Boolean circuit for the annotations



cc-tables

- Boolean circuit for the annotations

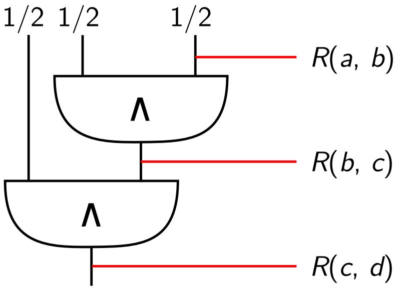


- Circuit must have low treewidth
- Instance must have low treewidth

→ Need simultaneous decomposition

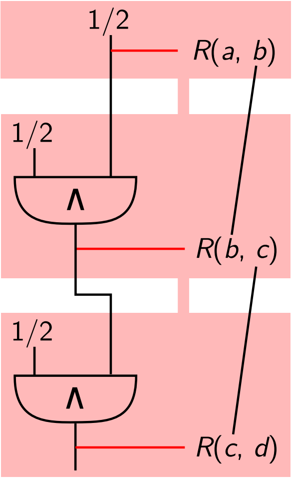
cc-tables

- Boolean circuit for the annotations



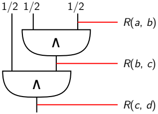
- Circuit must have low treewidth
- Instance must have low treewidth

$\rightarrow$ Need simultaneous decomposition

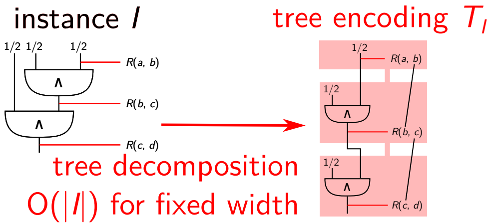


Main result

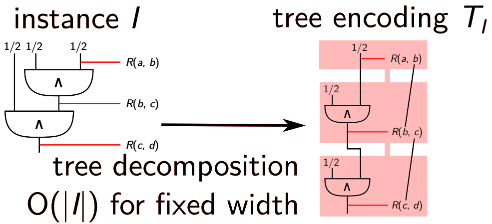
instance I



Main result


instance I
1/2 1/2 1/2
$R(a, b)$
$R(b, c)$
$R(c, d)$
tree encoding T_I
1/2
$R(a, b)$
1/2
$R(b, c)$
1/2
$R(c, d)$
tree decomposition
$O(|I|)$ for fixed width

Main result

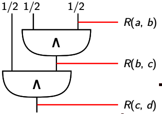


$\exists xy\ R(x, y) \wedge S(y)$

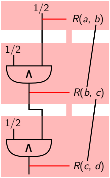
query q

Main result

instance I



tree encoding T_I



tree decomposition

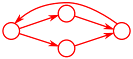
$O(|I|)$ for fixed width

rewriting

$O(1)$ data complexity

$\exists xy\, R(x, y) \wedge S(y)$

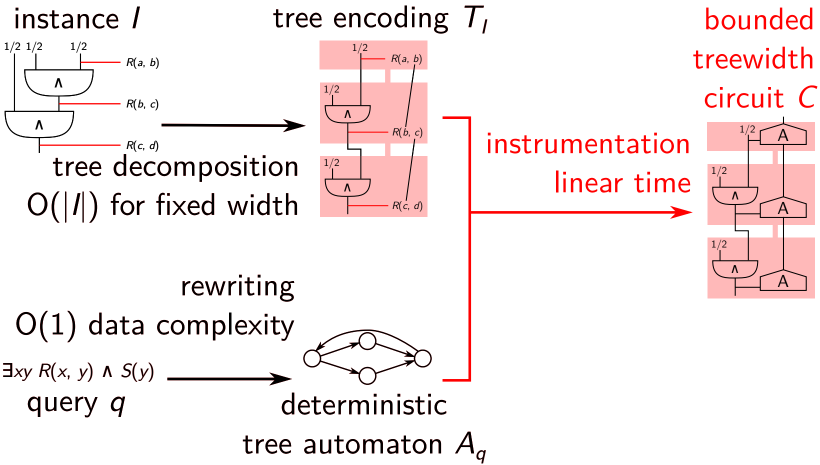
query q



deterministic

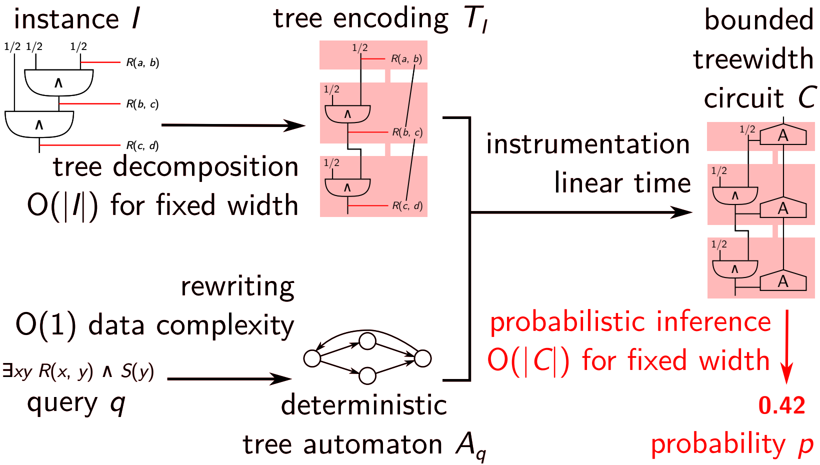
tree automaton A_q

Main result

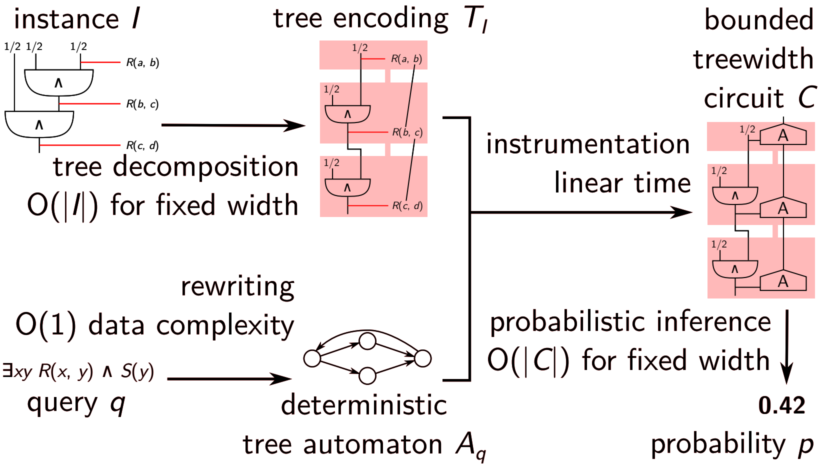


instance I
1/2 1/2 1/2
R(a, b)
R(b, c)
R(c, d)
tree decomposition
O(|I|) for fixed width
tree encoding T_I
bounded
treewidth
circuit C
instrumentation
linear time
rewriting
O(1) data complexity
$\exists xy\ R(x, y) \land S(y)$
query q
deterministic
tree automaton A_q

Main result


instance I
1/2 1/2 1/2
R(a, b)
R(b, c)
R(c, d)
tree decomposition
O(|I|) for fixed width
tree encoding T_I
bounded
treewidth
circuit C
instrumentation
linear time
rewriting
O(1) data complexity
∃xy R(x, y) ∧ S(y)
query q
deterministic
tree automaton A_q
probabilistic inference
O(|C|) for fixed width
0.42
probability p

Main result


instance I
1/2 1/2 1/2
$R(a, b)$
$R(b, c)$
$R(c, d)$
tree decomposition
$O(|I|)$ for fixed width
tree encoding T_I
bounded
treewidth
circuit C
instrumentation
linear time
rewriting
$O(1)$ data complexity
$\exists xy\ R(x, y) \wedge S(y)$
query q
deterministic
tree automaton A_q
probabilistic inference
$O(|C|)$ for fixed width
0.42
probability p

Consequences

- For queries representable as deterministic automata ...
  - $\rightarrow$ CQs
  - $\rightarrow$ Monadic second-order
  - $\rightarrow$ Guarded second-order

Consequences

- For queries representable as deterministic automata ...
  - → CQs
  - → Monadic second-order
  - → Guarded second-order
- ... on various probabilistic models ...
  - → Tuple-independent tables
  - → Block-independent disjoint tables
  - → pc-tables (presented before)
  - → Probabilistic XML

Consequences

- For queries representable as deterministic automata ...
  - $\rightarrow$ CQs
  - $\rightarrow$ Monadic second-order
  - $\rightarrow$ Guarded second-order
- ... on various probabilistic models ...
  - $\rightarrow$ Tuple-independent tables
  - $\rightarrow$ Block-independent disjoint tables
  - $\rightarrow$ pc-tables (presented before)
  - $\rightarrow$ Probabilistic XML
- ... assuming bounded treewidth (for reasonable definitions) ...

Consequences

- For queries representable as deterministic automata ...
  - → CQs
  - → Monadic second-order
  - → Guarded second-order
- ... on various probabilistic models ...
  - → Tuple-independent tables
  - → Block-independent disjoint tables
  - → pc-tables (presented before)
  - → Probabilistic XML
- ... assuming bounded treewidth (for reasonable definitions) ...

→ ... probability of fixed q can be computed in $O(|\widehat{I}|)$!

Conclusion

- We can combine the following techniques:
  - Computing tree decompositions
  - Encoding problems to automata on tree encodings of instances
  - Evaluating probabilities on bounded-treewidth circuits

Conclusion

- We can combine the following techniques:
  - Computing tree decompositions
  - Encoding problems to automata on tree encodings of instances
  - Evaluating probabilities on bounded-treewidth circuits
- Applications:
  - Tractable probabilistic query evaluation in practice?
  - Reasoning under uncertain rules
    (hence the bait-and-switch on the title...)

Conclusion

- We can combine the following techniques:
  - Computing tree decompositions
  - Encoding problems to automata on tree encodings of instances
  - Evaluating probabilities on bounded-treewidth circuits
- Applications:
  - Tractable probabilistic query evaluation in practice?
  - Reasoning under uncertain rules
    (hence the bait-and-switch on the title...)
- Questions:
  - Other semirings than Boolean AND/OR?
  - Other tasks than probabilistic inference?

Conclusion

- We can combine the following techniques:
  - Computing tree decompositions
  - Encoding problems to automata on tree encodings of instances
  - Evaluating probabilities on bounded-treewidth circuits
- Applications:
  - Tractable probabilistic query evaluation in practice?
  - Reasoning under uncertain rules
    (hence the bait-and-switch on the title...)
- Questions:
  - Other semirings than Boolean AND/OR?
  - Other tasks than probabilistic inference?

5 votes 0 answers 48 views

What are bounded-treewidth circuits good for?

circuit-complexity pr.probability treewidth arithmetic-circuits

http://cstheory.stackexchange.com/q/25624 modified aug 28 at 13:05 a3nm 1,432

Conclusion

- We can combine the following techniques:
  - Computing tree decompositions
  - Encoding problems to automata on tree encodings of instances
  - Evaluating probabilities on bounded-treewidth circuits
- Applications:
  - Tractable probabilistic query evaluation in practice?
  - Reasoning under uncertain rules
    (hence the bait-and-switch on the title...)
- Questions:
  - Other semirings than Boolean AND/OR?
  - Other tasks than probabilistic inference?

5 votes 0 answers 48 views

What are bounded-treewidth circuits good for?

circuit-complexity pr.probability treewidth arithmetic-circuits

http://cstheory.stackexchange.com/q/25624

modified aug 28 at 13:05 a3nm 1,432

Thanks for your attention!