UCLA

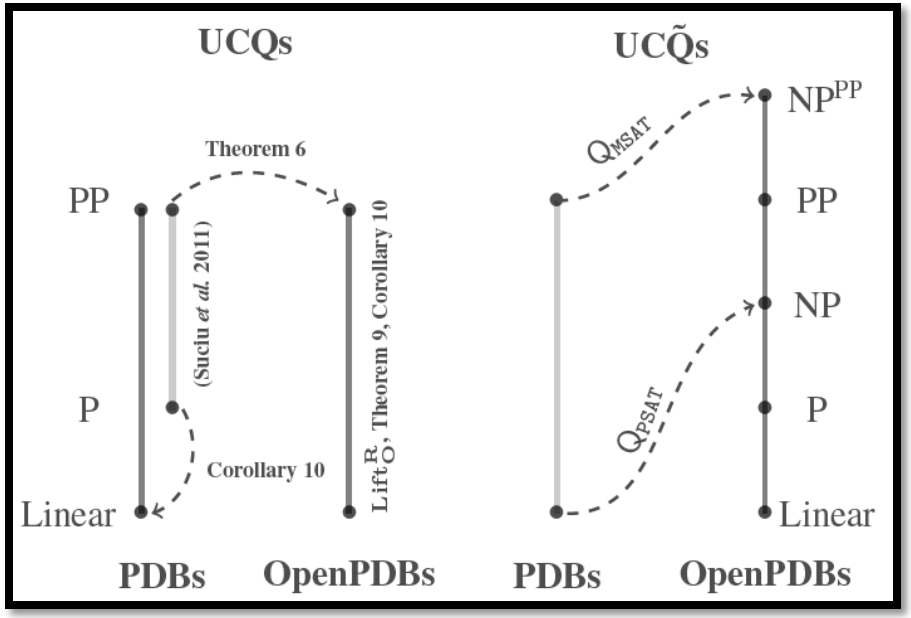
Open-World Probabilistic Databases

Guy Van den Broeck

On joint work with
Ismail Ilkan Ceylan, Adnan Darwiche

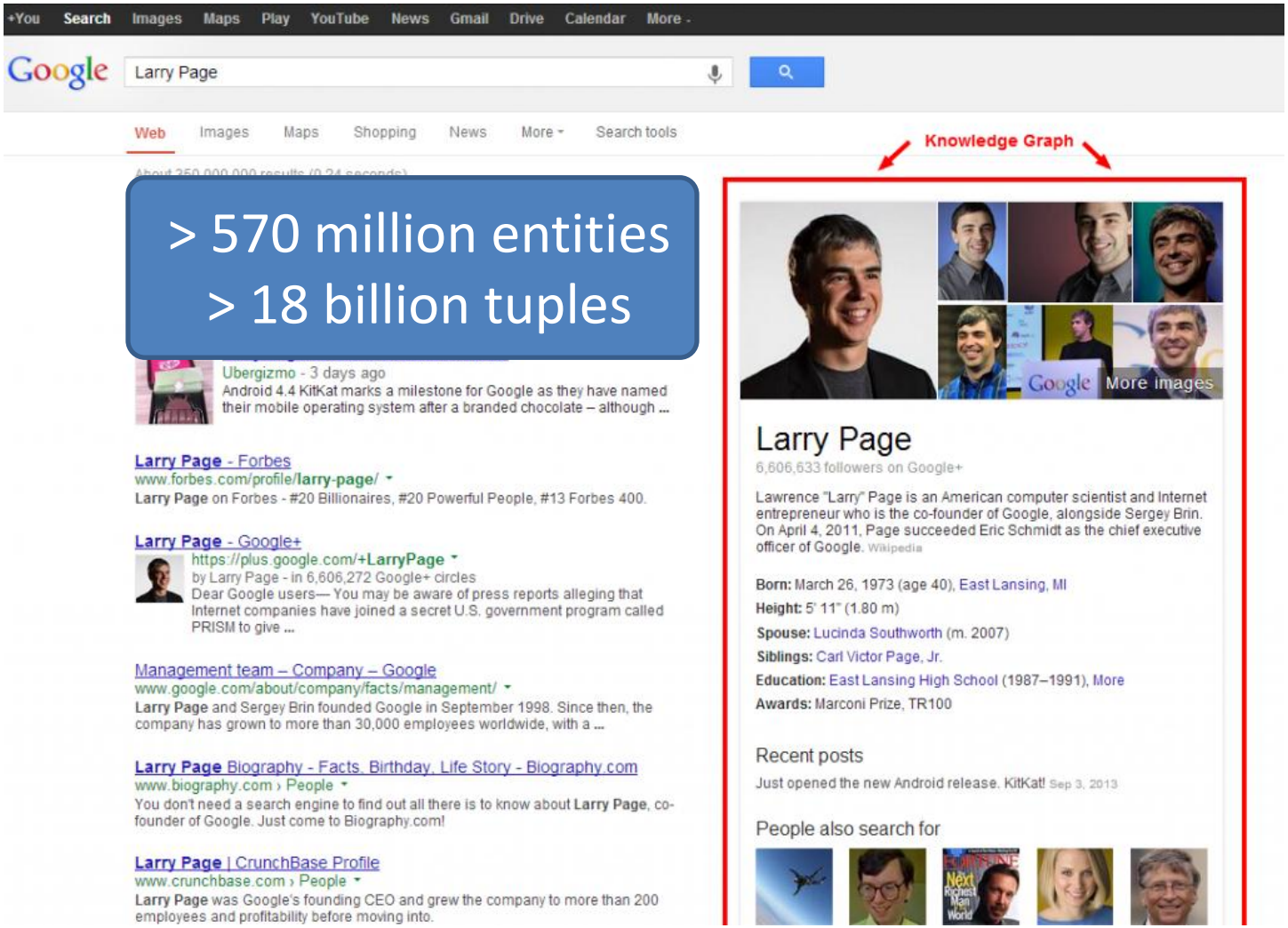
Feb 3, 2016, SML

Outline?

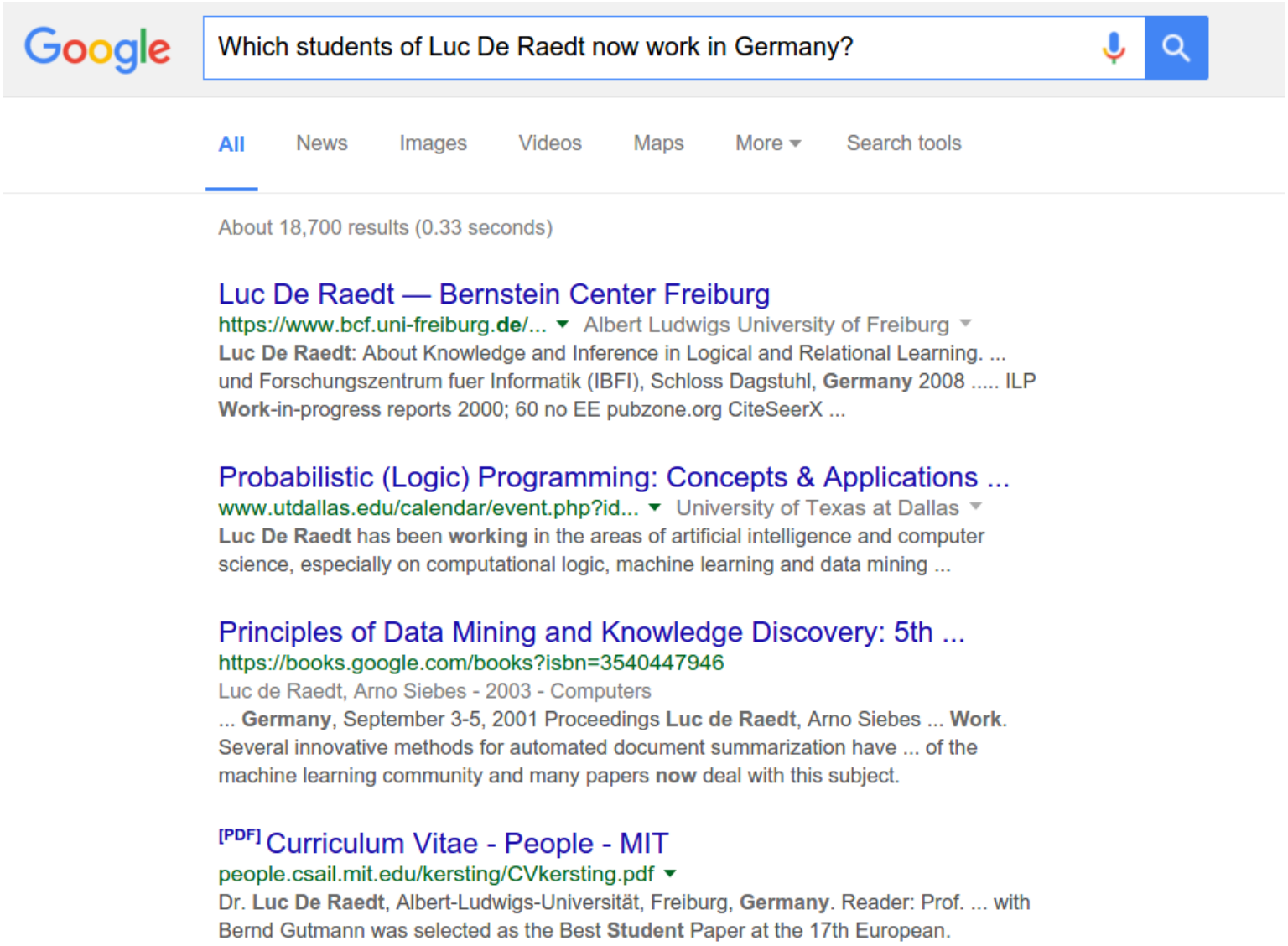

UCQs
UCQ̃s
Theorem 6
PP
P
Linear
(Suciu et al. 2011)
Corollary 10
Lift^R_O, Theorem 9, Corollary 10
NP^PP
PP
NP
P
Linear
Q_MSAT
Q_PSAT
PDBs
OpenPDBs
PDBs
OpenPDBs

or

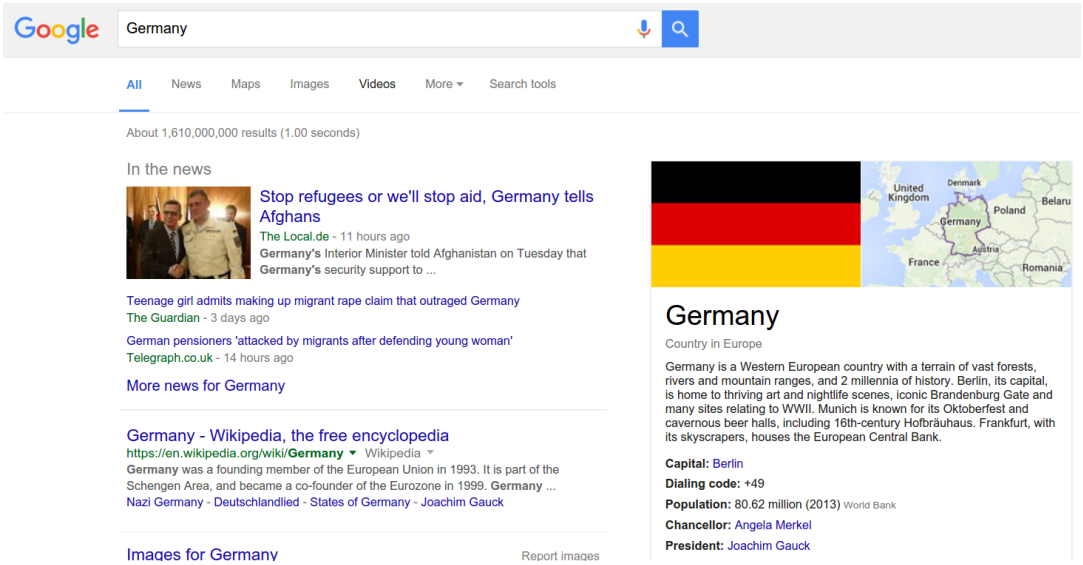
What we can do already…

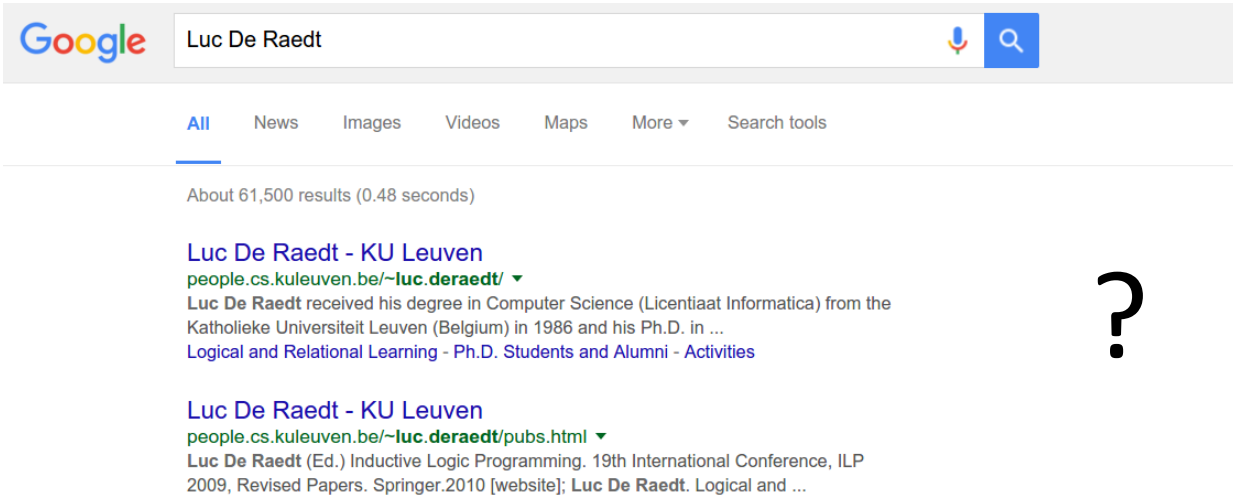


What I want to do…



Ingredients





?

Information Extraction

PhD Students Luc De Raedt

- Laura-Andrea Antanas(co-promotor Tinne Tuytelaars)
- Dries Van Daele (co-promotor Kathleen Marchal)
- Thanh Le Van (co-promotor Kathleen Marchal)
- Bogdan Moldovan
- Davide Nitti (co-promotor Tinne De Laet)
- José Antonio Oramas Mogrovejo (key supervisor Tinne Tuytelaars)
- Francesco Orsini (co-supervisor Paol Frasconi)
- Sergey Paramonov
- Joris Renkens
- Mathias Verbeke (with Bettina Berendt)
- Jonas Vlasselaer

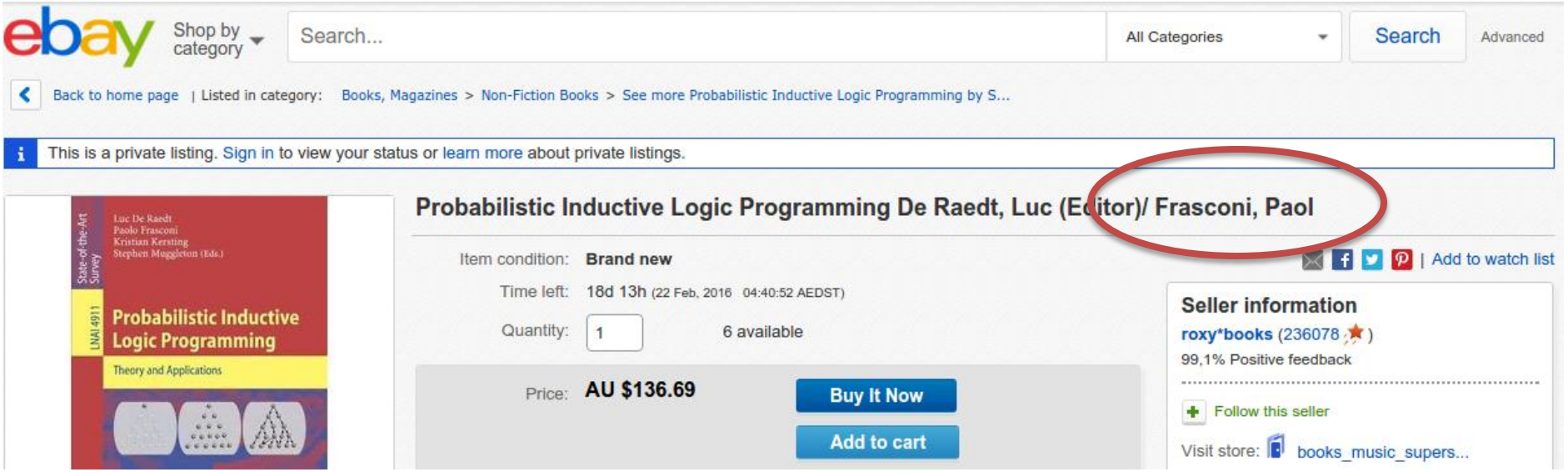
HasStudent

X	Y	P
Luc	Laura	0.7
Luc	Hendrik	0.6
Luc	Kathleen	0.3
Luc	Paol	0.3
Luc	Paolo	0.1

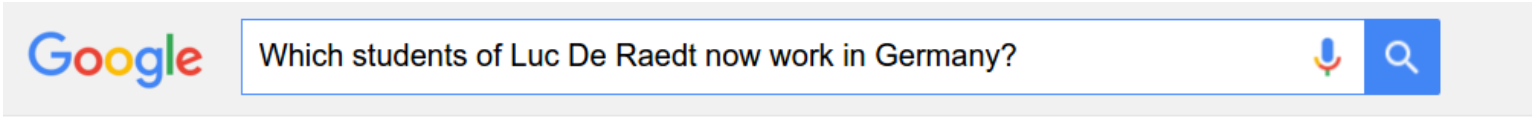
Alumni Luc De Raedt

1. Hendrik Blockeel, *Top-down induction of first order logical decision trees,* Ph.D. thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, december 1998, 202+xv pages. (Co-promotor Maurice Bruynooghe)
2. Luc Dehaspe, *Frequent pattern discovery in first-order logic,* Ph.D. thesis, Department of Computer

So noisy!


ebay
Shop by category
Search...
All Categories
Search
Advanced
Back to home page | Listed in category: Books, Magazines > Non-Fiction Books > See more Probabilistic Inductive Logic Programming by S...
This is a private listing. Sign in to view your status or learn more about private listings.
Luc De Raedt
Paolo Frasconi
Kristian Kersting
Stephen Muggleton (Eds.)
State-of-the-Art Survey
LNAI 4911
Probabilistic Inductive Logic Programming
Theory and Applications
Probabilistic Inductive Logic Programming De Raedt, Luc (Editor)/ Frasconi, Paol
Item condition: Brand new
Time left: 18d 13h (22 Feb, 2016 04:40:52 AEDST)
Quantity: 1
6 available
Price: AU $136.69
Buy It Now
Add to cart
Add to watch list
Seller information
roxy*books (236078)
99,1% Positive feedback
Follow this seller
Visit store: books_music_supers...

Desired Answer



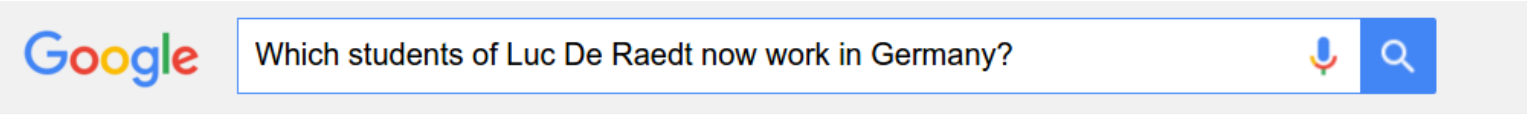
Kristian Kersting, Bjoern Bringmann, …

Ingo Thon, Niels Landwehr, …

Paol Frasconi, …

Justin Bieber, …

Observations



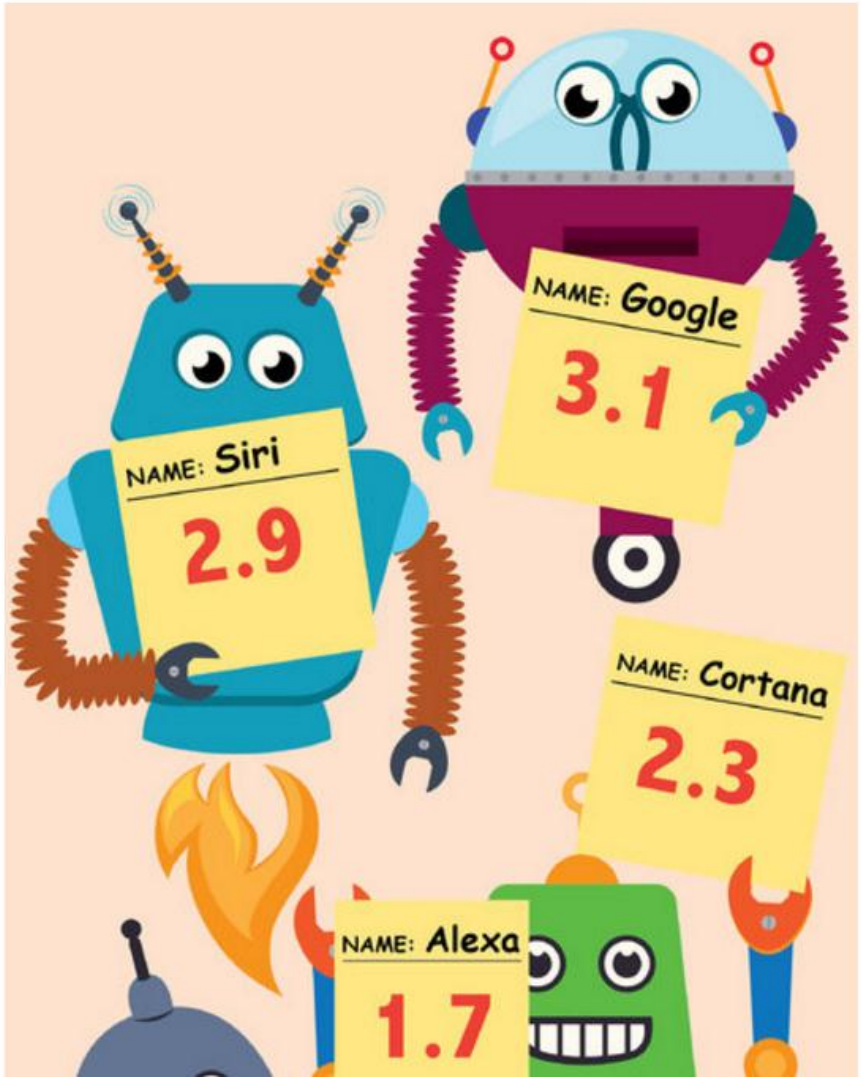
- Expose uncertainty
- Risk incorrect answers
- Cannot be labeled manually
- Join information extracted from many pages

Google, Microsoft, Amazon, Yahoo not ready?
How do we get there?

Siri, Alexa and Other Virtual Assistants Put to the Test

Tech Fix

By BRIAN X. CHEN JAN. 27, 2016



WHEN I asked Alexa earlier this week who was playing in the Super Bowl, she responded, somewhat monotonously, "Super Bowl 49's winner is New England Patriots."

"Come on, that's last year's Super Bowl," I said. "Even I can do better than that."

At the time, I was actually alone in my living room. I was talking to the virtual companion inside Amazon's wireless speaker, Echo, which was released last June. Known as Alexa, she has gained raves from Silicon Valley's tech-obsessed digerati and has become one of the newest members of the virtual assistants club.

All the so-called Frightful Five tech

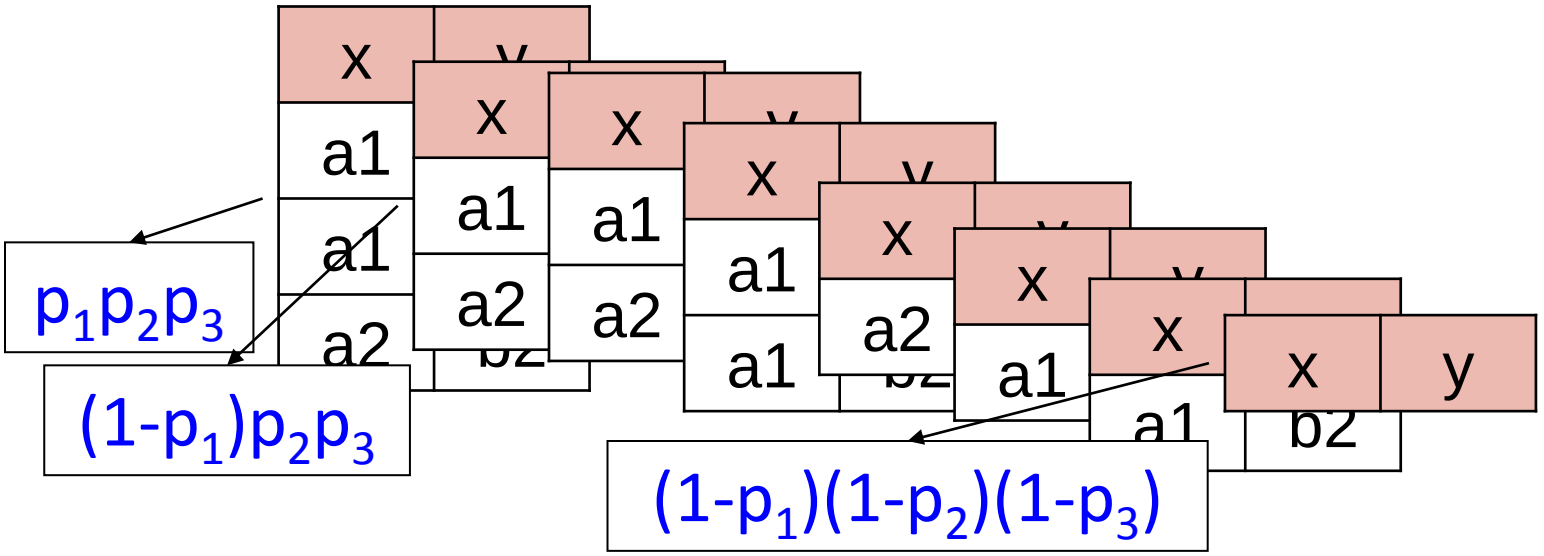
[NYTimes]

Probabilistic Databases

Probabilistic database D:

x	y	P
a1	b1	p_1
a1	b2	p_2
a2	b2	p_3

Possible worlds semantics:



Knowledge Base Completion

Given:

LivesIn

X	Y
Luc	Belgium
Guy	USA
Kristian	Germany

LocatedIn

X	Y
Siemens	Germany
Siemens	Belgium
UCLA	USA
TUDortmund	Germany
KU Leuven	Belgium

WorksFor

X	Y
Luc	KU Leuven
Guy	UCLA
Kristian	TUDortmund
Ingo	Siemens

Learn:

0.8::LivesIn(x,y) :- WorksFor(x,z) ∧ LocatedIn(z,x).

- Handle lots of noise, robust!
- Predict `LivesIn(Ingo,Germany)` with 80% prob.

How close are we?

- Do we have the technology available?
- NO! All of this stands on weak footing!
- Problems
  1. Broken learning loop
  2. Broken query semantics
  3. The curse of superlinearity
  4. How to measure success?

Problem 1: Broken Learning Loop

Bayesian view on learning:

- Prior belief:

  $\Pr(\texttt{HasStudent(Luc,Paol)}) = 0.01$

- Observe page

  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  $) = 0.2$

- Observe page

  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  ,  $) = 0.3$

Principled and sound reasoning!

Problem 1: Broken Learning Loop

Current view on Knowledge Base Completion:

- Prior belief:

  $\Pr(\texttt{HasStudent(Luc,Paol)}) = 0$

- Observe page

  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  $) = 0.2$

- Observe page

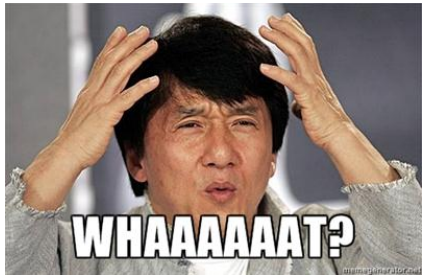
  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  ,  $) = 0.3$

Problem 1: Broken Learning Loop

Current view on Knowledge Base Completion:

- Prior belief:

  $\Pr(\text{HasStudent}(\text{Luc},\text{Paol})) = 0$



- Observe page

  $\Pr(\text{HasStudent}(\text{Luc},\text{Paol}) \mid$  $) = 0.2$

- Observe page

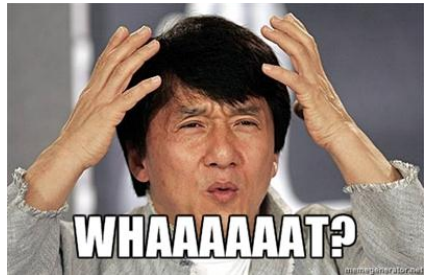
  $\Pr(\text{HasStudent}(\text{Luc},\text{Paol}) \mid$  ,  $) = 0.3$

Problem 1: Broken Learning Loop

Current view on Knowledge Base Completion:

- Prior belief:

  $\Pr(\texttt{HasStudent(Luc,Paol)}) = 0$



- Observe page

  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  $) = 0.2$

- Observe page

  $\Pr(\texttt{HasStudent(Luc,Paol)} \mid$  ,  $) = 0.3$

This is mathematical nonsense!

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

Q :- HasStudent(Luc,Ingo) ∧ WorksIn(Ingo,DE)

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)
Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,FR)

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE) ∧ Scientologist(z)

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- HasStudent(Luc,Ingo) ∧ WorksIn(Ingo,DE)

Q :- HasStudent(Luc,Kristian) ∧ ¬HasStudent(Luc,Kristian)

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- HasStudent(Luc,Ingo) ∧ WorksIn(Ingo,DE)

Q :- HasStudent(Luc,Kristian) ∧ WorksIn(Kristian,DE)

HasStudent

X	Y	P
Luc	Ingo	0.9
Luc	Kristian	0.6

Problem 2: Broken Query Semantics

Let's play a new drinking game: *higher or lower*.

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

Q :- ∃z HasStudent(Hendrik,z) ∧ WorksIn(z,DE)

HasStudent

X	Y	P
Luc	Ingo	0.9
Luc	Kristian	0.6
Hendrik	Nima	0.7

Problem 2: Broken Query Semantics

- Often probabilities will be **identical**

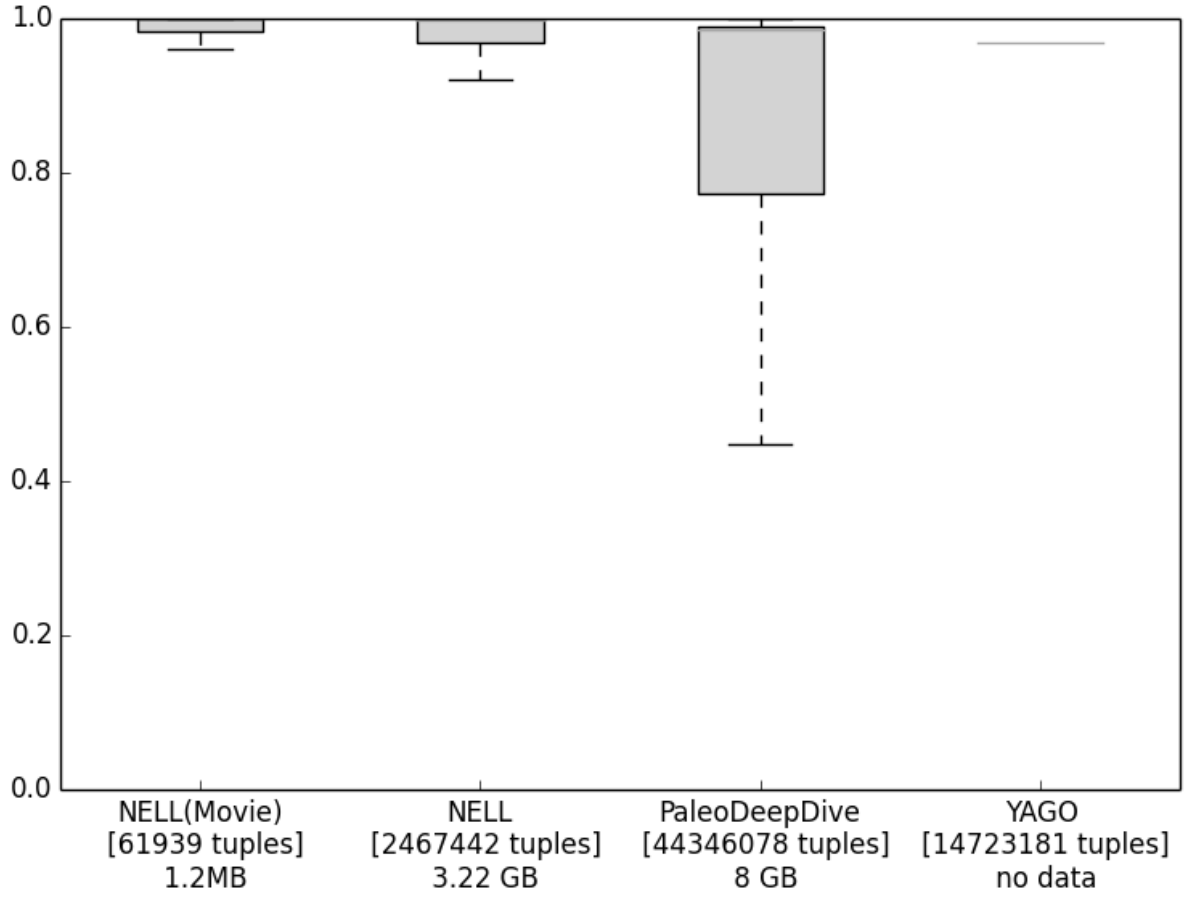
  Example: P(Q)=0 if WorksIn table empty

- Yet queries are clearly **different**..
  … IF you assume that tuples are missing!

- Not captured by existing query semantics ☹

Problem 3: Curse of Superlinearity

- Reality is worse!
- Tuples are intentionally missing!
- Every tuple has 99% pr.



Problem 3: Curse of Superlinearity

“This is all true, Guy,
but it’s just a temporary issue”

“No it’s not!”

Problem 3: Curse of Superlinearity

- A single table

Sibling

X	Y	P
…	…	…

- At the scale of facebook (billions of people)
- Real Bayesian belief about everyone

  I.e., all non-zero probabilities

⇒ 200 Exabytes of data

Problem 3: Curse of Superlinearity

All Google storage is a couple exabytes…

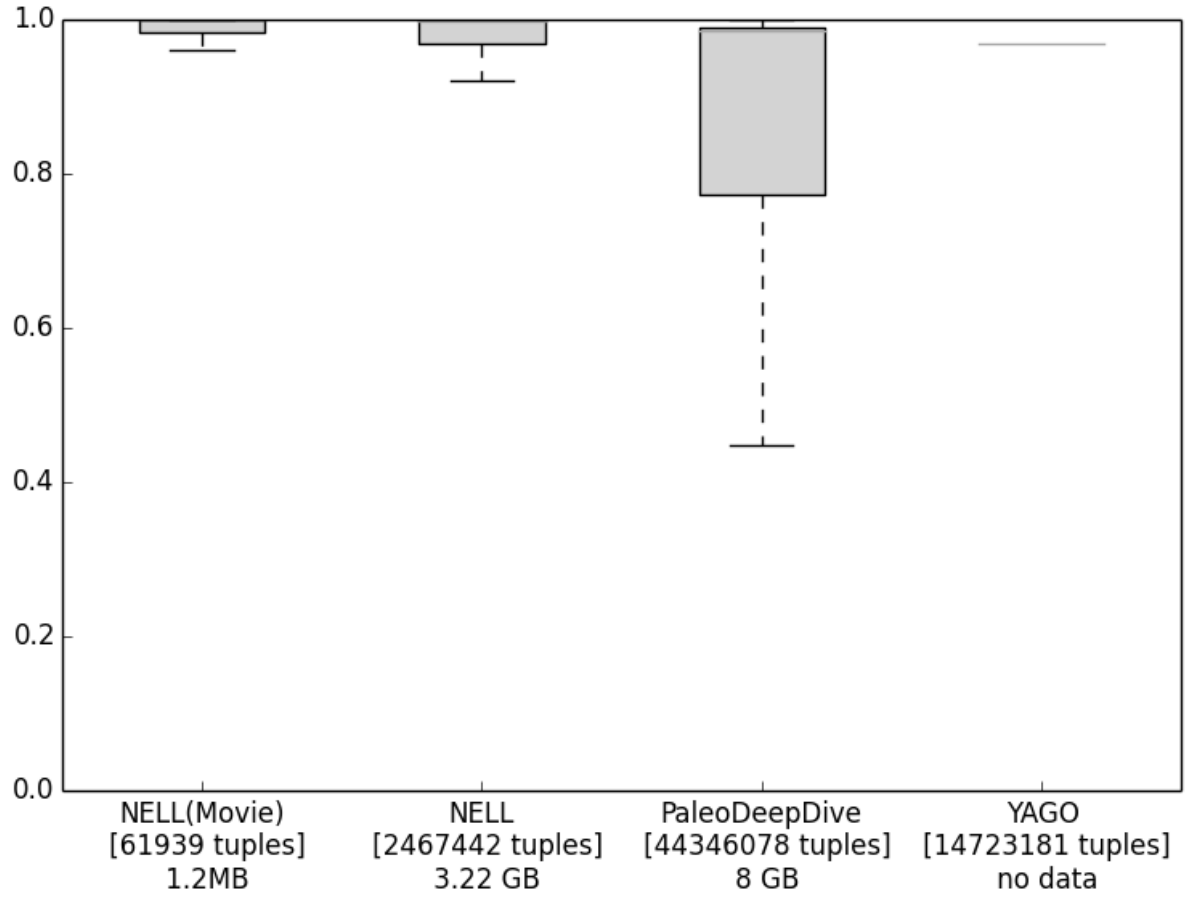
ing. In *Proc. of AAAI'15*. AAAI Press, 2015.

Randall Munroe. Google's datacenters on punch cards, 2015.

James D Park and Adnan Darwiche. Complexity Results and

Problem 3: Curse of Superlinearity

We should be here!



How to measure success?

Example: Knowledge base completion

WorksFor

X	Y	P
Luc	KU Leuven	0.7
Guy	UCLA	0.6
Kristian	TUDortmund	0.3
Ingo	Siemens	0.3

LocatedIn

X	Y	P
Siemens	Germany	0.7
Siemens	Belgium	0.5
UCLA	USA	0.8
TUDortmund	Germany	0.6
KU Leuven	Belgium	0.7

0.8::LivesIn(x,y) :- WorksFor(x,z) ∧ LocatedIn(z,x).

How to measure success?

Example: Knowledge base completion

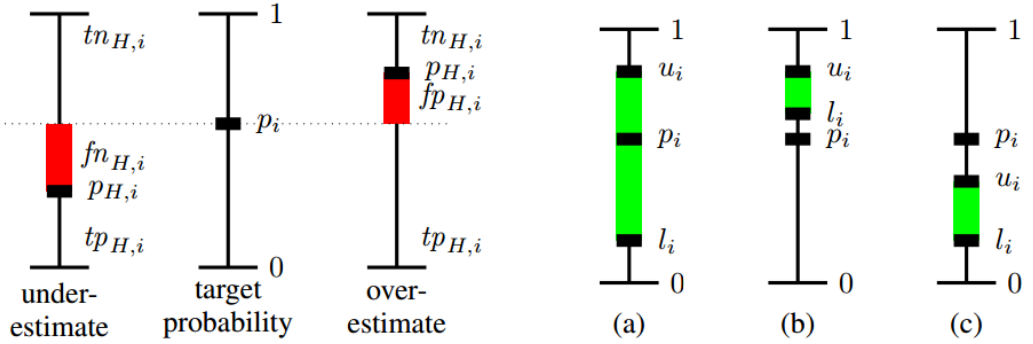
0.8::LivesIn(x,y) :- WorksFor(x,z) ∧ LocatedIn(z,x).

or

0.5::LivesIn(x,y) :- BornIn(x,y).

What is the likelihood, precision, accuracy, …?

ProbFOIL:



How to measure success?

Example: Knowledge base completion

If the query semantics are off,
how can these score be right?

Example: Relational pattern mining

[Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on World Wide Web]

Learners and miners are **led astray**… ☹

All of this to say…

… we need open-world semantics for knowledge bases.

Open Probabilistic Databases

- Intuition: What is missing from the database has low probability.
- Credal semantics:
  OpenPDB represents **set of distributions**.
- All closed-world databases extended with tuples <t,p> where $p < \lambda$.
- Query semantics: upper and lower bounds.

OpenPDB Example

HasStudent

X	Y	P
Luc	Ingo	0.9
Luc	Kristian	0.6

Q1 :- HasStudent(Luc,Ingo) ∧ WorksIn(Ingo,DE)
Q2 :- HasStudent(Luc,Kristian) ∧ WorksIn(Kristian,DE)

with λ=0.1

- Lower bound: Pr(Q1) = 0 Pr(Q2) = 0
- Upper bound: Pr(Q1) = 0.09 Pr(Q2) = 0.06

when

WorksIn

X	Y	P
Ingo	DE	0.1
Kristian	DE	0.1

OpenPDB Example

HasStudent

X	Y	P
Luc	Ingo	0.9
Luc	Kristian	0.6

Q :- HasStudent(Luc,Kristian) ∧ ¬HasStudent(Luc,Kristian)

with λ=0.1

- Lower bound: Pr(Q) = 0
- Upper bound: Pr(Q) = 0

<u>In general:</u>
Lower-higher relations observed in upper bound! ☺

Algorithm for UCQ

Q :- $\exists z$ HasStudent(Luc,z) $\wedge$ WorksIn(z,DE)

Q :- $\exists z$ HasStudent(Luc,z) $\wedge$ WorksIn(z,FR)

- Monotone sentence in logic
- More tuples is better
- More probability is better

$\Rightarrow$ Lower bound: Assume closed world

$\Rightarrow$ Upper bound: Add all tuples with prob λ

Is this a good algorithm?

- Polynomial time reduction to classic setting ☺
- Quadratic blowup of database ☹

  200 exabytes for Sibling!

Can we do open-world reasoning
with no overhead?

Probabilistic Database Inference

- $P(Q1 \wedge Q2) = P(Q1)P(Q2)$
  $P(Q1 \vee Q2) = 1 - (1 - P(Q1))(1 - P(Q2))$

Decomposable
$\wedge$/ $\vee$

- $P(\exists z\, Q) = 1 - \Pi_{a \in Domain} (1 - P(Q[a/z]))$
  $P(\forall z\, Q) = \Pi_{a \in Domain} P(Q[a/z])$

Decomposable
$\exists$/ $\forall$

- $P(Q1 \wedge Q2) = P(Q1) + P(Q2) - P(Q1 \vee Q2)$
  $P(Q1 \vee Q2) = P(Q1) + P(Q2) - P(Q1 \wedge Q2)$

Inclusion/
exclusion

Dalvi and Suciu's dichotomy theorem:

If rules succeed, prob. database query eval is in PTIME; else, PP-hard (in database size).

$$\text{Linear} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PP} \subseteq \text{P}^{\text{PP}} \subseteq \text{NP}^{\text{PP}} \subseteq \text{PSpace} \subseteq \text{ExpTime}$$

PTIME is not enough!

- We want linear-time!
- Theorem: Prob. database query eval is **LINEAR time** for all PTIME queries.
- Theorem: Open prob. database query eval is **LINEAR time** for all PTIME queries.

Algorithm 1 $\text{Lift}^{\text{R}}_{\text{O}}(\text{Q}, \mathcal{S}, \lambda, \mathcal{D})$, abbreviated by $\mathbf{L}(\text{Q}, \mathcal{S})$

Input: CNF Q , prob. tuples $\mathcal{S}$, threshold λ, and domain $\mathcal{D}$.
Output: The upper probability $\overline{\text{P}}_{(\mathcal{S},\lambda)}(\text{Q})$ over domain $\mathcal{D}$.

Step 0 *Base of Recursion*
  if Q is a single ground atom t **then**
    if $\langle t : p \rangle \in \mathcal{S}$ **then return** p **else return** λ
Step 1 *Rewriting of Query*
  Convert Q to unions of CNFs: $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \ldots \vee \text{Q}_m$
Step 2 *Decomposable Disjunction*
  if $m > 1$ and $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    $q_1 \leftarrow \mathbf{L}(\text{Q}_1, \mathcal{S}|_{Q_1})$ and $q_2 \leftarrow \mathbf{L}(\text{Q}_2, \mathcal{S}|_{Q_2})$
    return $1 - (1 - q_1) \cdot (1 - q_2)$
Step 3 *Inclusion-Exclusion*
  if $m > 1$ but Q_{UCNF} has no independent Q_i **then**
    return $\sum_{s \subseteq m} (-1)^{|s|} \cdot \mathbf{L}(\wedge_{i \in s} \text{Q}_i, \mathcal{S}|_{\wedge_{i \in s} \text{Q}_i})$
Step 4 *Decomposable Conjunction*
  if $\text{Q} = \text{Q}_1 \wedge \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    return $\mathbf{L}(\text{Q}_1, \mathcal{S}) \cdot \mathbf{L}(\text{Q}_2, \mathcal{S})$
Step 5 *Decomposable Universal Quantifier*
  if Q has a *separator variable* x **then**
    let T be all constants as x-*argument* in $\mathcal{S}$
    $q_c \leftarrow \prod_{t \in T} \mathbf{L}(\text{Q}[x/t], \mathcal{S}|_{x=t})$
    $q_o \leftarrow \mathbf{L}(\text{Q}[x/t], \varnothing)$ for some $t \in \mathcal{D} \setminus T$
    return $q_c \cdot q_o^{|\mathcal{D} \setminus T|}$
Step 6 *Fail*

Algorithm 1 $\text{Lift}^{\text{R}}_{\text{O}}(\text{Q}, \mathcal{S}, \lambda, \mathcal{D})$, abbreviated by $\mathbf{L}(\text{Q}, \mathcal{S})$

Input: CNF Q , prob. tuples $\mathcal{S}$, threshold λ, and domain $\mathcal{D}$.
Output: The upper probability $\overline{\text{P}}_{(\mathcal{S},\lambda)}(\text{Q})$ over domain $\mathcal{D}$.

Step 0 *Base of Recursion*
  if Q is a single ground atom t **then**
    if $\langle t : p \rangle \in \mathcal{S}$ **then return** p **else return** λ
Step 1 *Rewriting of Query*
  Convert Q to unions of CNFs: $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \ldots \vee \text{Q}_m$
Step 2 *Decomposable Disjunction*
  if $m > 1$ and $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    $q_1 \leftarrow \mathbf{L}(\text{Q}_1, \mathcal{S}|_{Q_1})$ and $q_2 \leftarrow \mathbf{L}(\text{Q}_2, \mathcal{S}|_{Q_2})$
    return $1 - (1 - q_1) \cdot (1 - q_2)$
Step 3 *Inclusion-Exclusion*
  if $m > 1$ but Q_{UCNF} has no independent Q_i **then**
    return $\sum_{s \subseteq m} (-1)^{|s|} \cdot \mathbf{L}(\wedge_{i \in s} \text{Q}_i, \mathcal{S}|_{\wedge_{i \in s} \text{Q}_i})$
Step 4 *Decomposable Conjunction*
  if $\text{Q} = \text{Q}_1 \wedge \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    return $\mathbf{L}(\text{Q}_1, \mathcal{S}) \cdot \mathbf{L}(\text{Q}_2, \mathcal{S})$
Step 5 *Decomposable Universal Quantifier*
  if Q has a *separator variable* x **then**
    let T be all constants as x-*argument* in $\mathcal{S}$
    $q_c \leftarrow \prod_{t \in T} \mathbf{L}(\text{Q}[x/t], \mathcal{S}|_{x=t})$
    $q_o \leftarrow \mathbf{L}(\text{Q}[x/t], \varnothing)$ for some $t \in \mathcal{D} \setminus T$
    return $q_c \cdot q_o^{|\mathcal{D} \setminus T|}$
Step 6 *Fail*

Existing Rules (see before)

Algorithm 1 $\text{Lift}^{\text{R}}_{\text{O}}(\text{Q}, \mathcal{S}, \lambda, \mathcal{D})$, abbreviated by $\text{L}(\text{Q}, \mathcal{S})$

Input: CNF Q , prob. tuples $\mathcal{S}$, threshold λ, and domain $\mathcal{D}$.
Output: The upper probability $\overline{\text{P}}_{(\mathcal{S},\lambda)}(\text{Q})$ over domain $\mathcal{D}$.

Step 0 *Base of Recursion*
  if Q is a single ground atom t **then**
    if $\langle t : p \rangle \in \mathcal{S}$ **then return** p **else return** λ
Step 1 *Rewriting of Query*
  Convert Q to unions of CNFs: $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \ldots \vee \text{Q}_m$
Step 2 *Decomposable Disjunction*
  if $m > 1$ and $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    $q_1 \leftarrow \text{L}(\text{Q}_1, \mathcal{S}|_{Q_1})$ and $q_2 \leftarrow \text{L}(\text{Q}_2, \mathcal{S}|_{Q_2})$
    return $1 - (1 - q_1) \cdot (1 - q_2)$
Step 3 *Inclusion-Exclusion*
  if $m > 1$ but Q_{UCNF} has no independent Q_i **then**
    return $\sum_{s \subseteq m} (-1)^{|s|} \cdot \text{L}(\wedge_{i \in s} \text{Q}_i, \mathcal{S}|_{\wedge_{i \in s} \text{Q}_i})$
Step 4 *Decomposable Conjunction*
  if $\text{Q} = \text{Q}_1 \wedge \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    return $\text{L}(\text{Q}_1, \mathcal{S}) \cdot \text{L}(\text{Q}_2, \mathcal{S})$
Step 5 *Decomposable Universal Quantifier*
  if Q has a *separator variable* x **then**
    let T be all constants as *x-argument* in $\mathcal{S}$
    $q_c \leftarrow \prod_{t \in T} \text{L}(\text{Q}[x/t], \mathcal{S}|_{x=t})$
    $q_o \leftarrow \text{L}(\text{Q}[x/t], \varnothing)$ for some $t \in \mathcal{D} \setminus T$
    return $q_c \cdot q_o^{|\mathcal{D} \setminus T|}$
Step 6 *Fail*

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

HasStudent(L,I) ∧ WorksIn(I,DE)
HasStudent(L,K) ∧ WorksIn(K,DE)
HasStudent(L,A) ∧ WorksIn(I,DE)

Recurse and multiply probs

Algorithm 1 $\text{Lift}^{\text{R}}_{\text{O}}(\text{Q}, \mathcal{S}, \lambda, \mathcal{D})$, abbreviated by $\mathbf{L}(\text{Q}, \mathcal{S})$

Input: CNF Q , prob. tuples $\mathcal{S}$, threshold λ, and domain $\mathcal{D}$.
Output: The upper probability $\overline{\text{P}}_{(\mathcal{S},\lambda)}(\text{Q})$ over domain $\mathcal{D}$.

Step 0 *Base of Recursion*
  if Q is a single ground atom t **then**
    if $\langle t : p \rangle \in \mathcal{S}$ **then return** p **else return** λ
Step 1 *Rewriting of Query*
  Convert Q to unions of CNFs: $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \ldots \vee \text{Q}_m$
Step 2 *Decomposable Disjunction*
  if $m > 1$ and $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    $q_1 \leftarrow \mathbf{L}(\text{Q}_1, \mathcal{S}|_{Q_1})$ and $q_2 \leftarrow \mathbf{L}(\text{Q}_2, \mathcal{S}|_{Q_2})$
    return $1 - (1 - q_1) \cdot (1 - q_2)$
Step 3 *Inclusion-Exclusion*
  if $m > 1$ but Q_{UCNF} has no independent Q_i **then**
    return $\sum_{s \subseteq m} (-1)^{|s|} \cdot \mathbf{L}(\wedge_{i \in s} \text{Q}_i, \mathcal{S}|_{\wedge_{i \in s} \text{Q}_i})$
Step 4 *Decomposable Conjunction*
  if $\text{Q} = \text{Q}_1 \wedge \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    return $\mathbf{L}(\text{Q}_1, \mathcal{S}) \cdot \mathbf{L}(\text{Q}_2, \mathcal{S})$
Step 5 *Decomposable Universal Quantifier*
  if Q has a *separator variable* x **then**
    let T be all constants as x-*argument* in $\mathcal{S}$
    $q_c \leftarrow \prod_{t \in T} \mathbf{L}(\text{Q}[x/t], \mathcal{S}|_{x=t})$
    $q_o \leftarrow \mathbf{L}(\text{Q}[x/t], \varnothing)$ for some $t \in \mathcal{D} \setminus T$
    return $q_c \cdot q_o^{|\mathcal{D} \setminus T|}$
Step 6 *Fail*

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

HasStudent(L,I) ∧ WorksIn(I,DE)
HasStudent(L,K) ∧ WorksIn(K,DE)
HasStudent(L,A) ∧ WorksIn(I,DE)

Recurse and ‘multiply’ probs
Multiply by $\mathbf{q_o}$: open world correction

Algorithm 1 $\text{Lift}^{\text{R}}_{\text{O}}(\text{Q}, \mathcal{S}, \lambda, \mathcal{D})$, abbreviated by $\text{L}(\text{Q}, \mathcal{S})$

Input: CNF Q , prob. tuples $\mathcal{S}$, threshold λ, and domain $\mathcal{D}$.
Output: The upper probability $\overline{\text{P}}_{(\mathcal{S},\lambda)}(\text{Q})$ over domain $\mathcal{D}$.

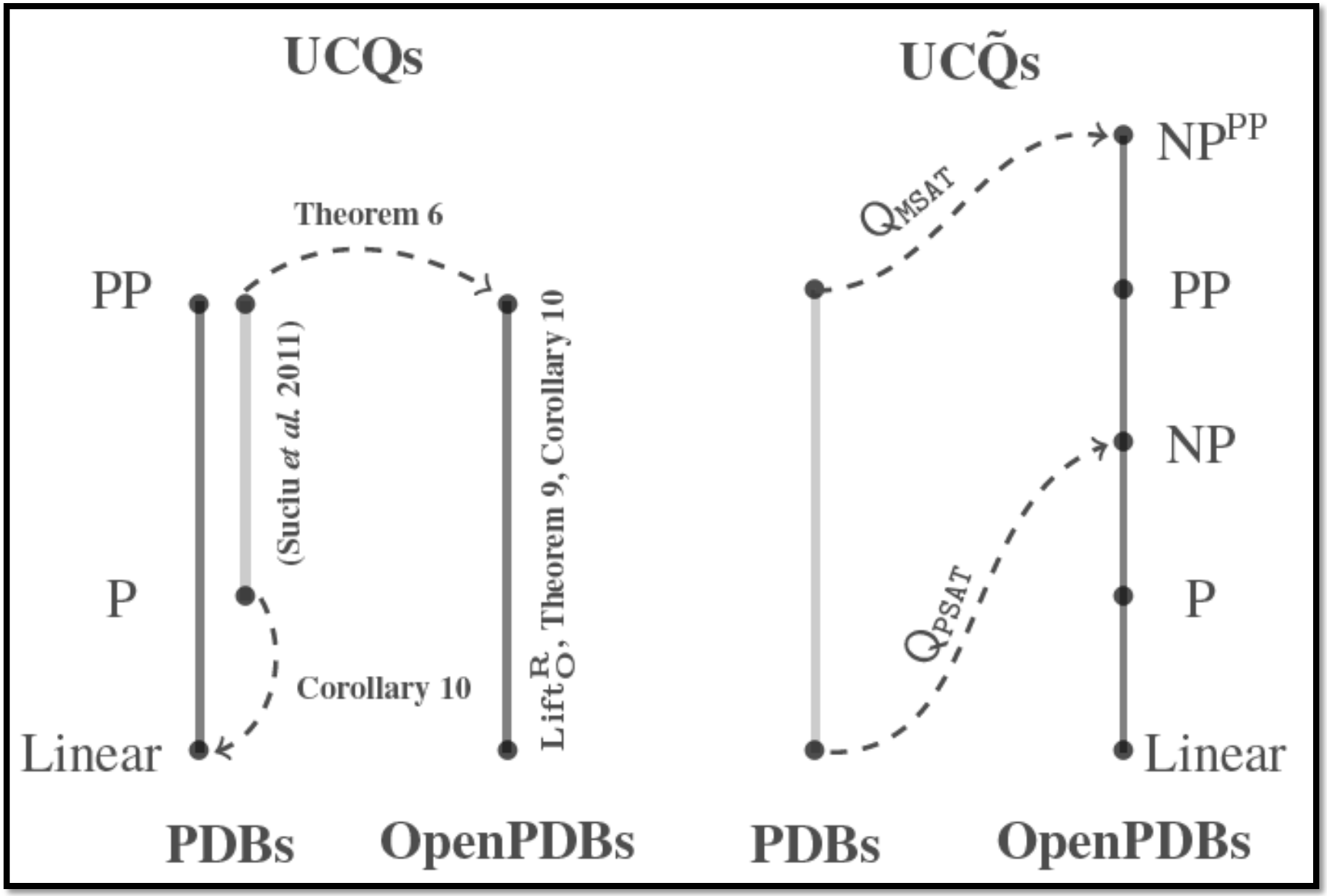
Step 0 *Base of Recursion*
  if Q is a single ground atom t **then**
    if $\langle t : p \rangle \in \mathcal{S}$ **then return** p **else return** λ
Step 1 *Rewriting of Query*
  Convert Q to unions of CNFs: $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \ldots \vee \text{Q}_m$
Step 2 *Decomposable Disjunction*
  if $m > 1$ and $\text{Q}_{\text{UCNF}} = \text{Q}_1 \vee \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    $q_1 \leftarrow \text{L}(\text{Q}_1, \mathcal{S}|_{Q_1})$ and $q_2 \leftarrow \text{L}(\text{Q}_2, \mathcal{S}|_{Q_2})$
    return $1 - (1 - q_1) \cdot (1 - q_2)$
Step 3 *Inclusion-Exclusion*
  if $m > 1$ but Q_{UCNF} has no independent Q_i **then**
    return $\sum_{s \subseteq m} (-1)^{|s|} \cdot \text{L}(\wedge_{i \in s} \text{Q}_i, \mathcal{S}|_{\wedge_{i \in s} \text{Q}_i})$
Step 4 *Decomposable Conjunction*
  if $\text{Q} = \text{Q}_1 \wedge \text{Q}_2$ where $\text{Q}_1 \perp \text{Q}_2$ **then**
    return $\text{L}(\text{Q}_1, \mathcal{S}) \cdot \text{L}(\text{Q}_2, \mathcal{S})$
Step 5 *Decomposable Universal Quantifier*
  if Q has a *separator variable* x **then**
    let T be all constants as x-*argument* in $\mathcal{S}$
    $q_c \leftarrow \prod_{t \in T} \text{L}(\text{Q}[x/t], \mathcal{S}|_{x=t})$
    $q_o \leftarrow \text{L}(\text{Q}[x/t], \varnothing)$ for some $t \in \mathcal{D} \setminus T$
    return $q_c \cdot q_o^{|\mathcal{D} \setminus T|}$
Step 6 *Fail*

q_o is lifted inference!
WFOMC/FOVE/…

Q :- ∃z HasStudent(Luc,z) ∧ WorksIn(z,DE)

HasStudent(L,I) ∧ WorksIn(I,DE)
HasStudent(L,K) ∧ WorksIn(K,DE)
HasStudent(L,A) ∧ WorksIn(I,DE)

Recurse and ‘multiply’ probs
Multiply by q_o: open world correction

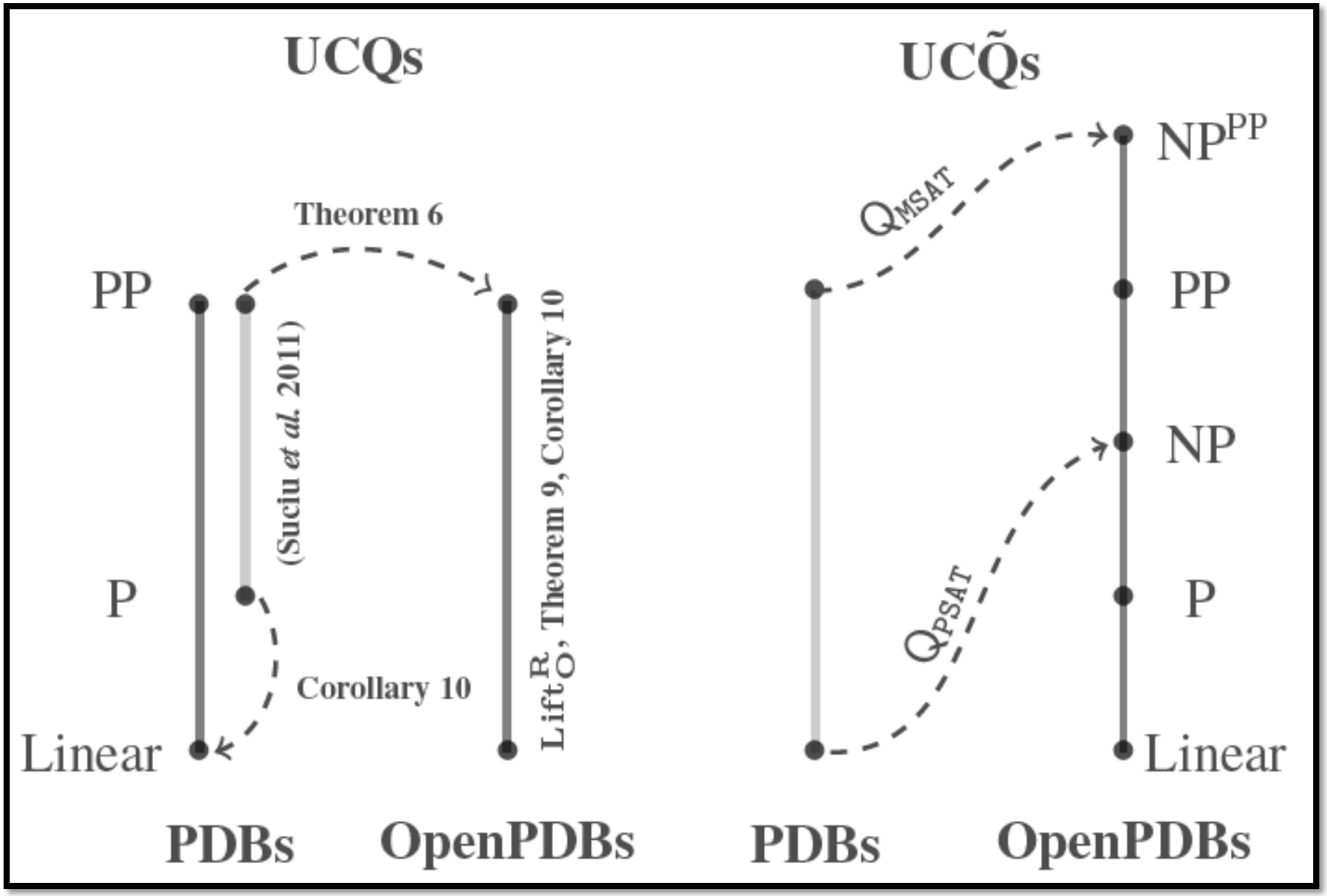


$$\text{Linear} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PP} \subseteq \text{P}^{\text{PP}} \subseteq \text{NP}^{\text{PP}} \subseteq \text{PSpace} \subseteq \text{ExpTime}$$

UCQ with negation

- Theorem:
  Linear time queries on closed-world databases can become **NP-complete** on OpenPDBs

- Theorem:
  PP queries on closed-world databases can become **NP^{PP}-complete** on OpenPDBs



$$\text{Linear} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PP} \subseteq \text{P}^{\text{PP}} \subseteq \text{NP}^{\text{PP}} \subseteq \text{PSpace} \subseteq \text{ExpTime}$$

Conclusions

- Open-world semantics makes a lot of sense
- Matches how these systems are employed
- Open-world reasoning is FREE for UCQs
- Beyond UCQs, can pay a hefty price
- Future work
  - More refined models of the open world
    E.g., (types, MLNs, additional statistics)
  - Efficient algorithms for hard case

References

Ismail Ilkan Ceylan, Adnan Darwiche, Guy Van den Broeck. Open-World Probabilistic Databases, In Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2016.

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. Foundations of databases, volume 8. Addison-Wesley Reading.

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. The Description Logic Handbook: Theory, Implementation, and Applica- tions. Cambridge University Press, 2nd edition.

Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction for the web. In Proc. of IJCAI'07, volume 7, 2670–2676.

Beame, P.; Van den Broeck, G.; Suciu, D.; and Gribkoff, E. 2015. Symmetric Weighted First-Order Model Counting. In Proc. of PODS'15, 313–328. ACM Press.

Bienvenu, M.; Cate, B. T.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, csp, and mmsnp. ACM Trans. Database Syst. 39(4):33:1–33:44.

Bishop, C. M. 2006. Pattern recognition and machine learn- ing. Springer.

Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In AAAI'11.

Ceylan, İ. İ., and Peñaloza, R. 2015. Probabilistic Query Answering in the Bayesian Description Logic BEL. In Proc. of SUM'15, volume 9310 of LNAI, 21–35. Springer.

Cozman, F. G. 2000. Credal networks. AIJ 120(2):199–233. Dalvi, N., and Suciu, D. 2012. The dichotomy of proba- bilistic inference for unions of conjunctive queries. JACM 59(6):1–87.

De Campos, C. P., and Cozman, F. G. 2005. The inferential complexity of bayesian and credal networks. In Proc. of IJCAI'05, AAAI Press, 1313–1318.

de Campos, C. P., and Cozman, F. G. 2007. Inference in credal networks through integer programming. In Proc. of SIPTA.

De Raedt, L.; Dries, A.; Thon, I.; Van den Broeck, G.; and Verbeke, M. 2015. Inducing probabilistic relational rules from probabilistic examples. In Proc. of IJCAI'15.

Dong, X. L.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In Proc. of ACM SIGKDD'14, KDD'14, 601–610. ACM.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In Proceedings of EMNLP, 1535–1545. Ass. for Computational Linguistics.

Fink, R., and Olteanu, D. 2014. A dichotomy for non- repeating queries with negation in probabilistic databases. In Proc. of PODS, 144–155. ACM.

Fink, R., and Olteanu, D. 2015. Dichotomies for Queries with Negation in Probabilistic Databases. ACM Transac- tions on Database Systems (TODS).

Galárraga, L. A.; Teflioudi, C.; Hose, K.; and Suchanek, F. 2013. Amie: association rule mining under incom- plete evidence in ontological knowledge bases. In Proc. of WWW'2013, 413–422.

Gill, J. 1977. Computational complexity of probabilistic

turing machines. SIAM Journal on Computing 6(4):675–

695.

Gottlob, G.; Lukasiewicz, T.; Martinez, M. V.; and Simari, G. I. 2013. Query answering under Probabilistic Uncertainty in Datalog +/- Ontologies. Ann. Math. AI 69(1):37–72.

Gribkoff, E.; Suciu, D.; and Van den Broeck, G. 2014. Lifted probabilistic inference: A guide for the database researcher. Bulletin of the Technical Committee on Data Engineering 37(3):6–17.

Gribkoff, E.; Van den Broeck, G.; and Suciu, D. 2014. Un- derstanding the Complexity of Lifted Inference and Asym- metric Weighted Model Counting. In Proc. of UAI'14, 280–289. AUAI Press.

References

Halpern, J. Y. 2003. Reasoning about uncertainty. MIT Press.

Hinrichs, T., and Genesereth, M. 2006. Herbrand logic. Technical Report LG-2006-02, Stanford University.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. Yago2: A spatially and temporally enhanced knowl- edge base from wikipedia. In Proc. of IJCAI'2013, 3161–3165. AAAI Press.

Jung, J. C., and Lutz, C. 2012. Ontology-Based Access to Probabilistic Data with OWL QL. In Proc. of ISWC'12, volume 7649 of LNCS, 182–197. Springer Verlag.

Kersting, K. 2012. Lifted probabilistic inference. In Proc. of ECAI'12, 33–38. IOS Press.

Levi, I. 1980. The Enterprise of Knowledge. MIT Press.

Libkin, L. 2014. Certain answers as objects and knowledge. In Proc. of KR'14. AAAI Press.

Littman, M. L.; Majercik, S. M.; and Pitassi, T. 2001. Stochastic Boolean Satisability. J. of Automated Reasoning 27(3):251–296.

Lukasiewicz, T. 2000. Credal networks under maximum entropy. In Proc. of UAI'00, 363–370.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.;and Kolobov, A. 2007. Blog: Probabilistic models with un- known objects. Statistical relational learning 373.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Dis- tant supervision for relation extraction without labeled data. In Proc. of ACL-IJCNLP, 1003–1011.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Bet- teridge, J.; Carlson, A.; Dalvi, B.; and Gardner, M. 2015. Never-Ending Learning. In Proc. of AAAI'15. AAAI Press.

Munroe, R. 2015. Google's datacenters on punch cards.

Park, J. D., and Darwiche, A. 2004. Complexity Results and Approximation Strategies for MAP Explanations. JAIR 21(1):101–133.

Patel-Schneider, P. F., and Horrocks, I. 2006. Position paper: a comparison of two modelling paradigms in the semantic web. In Proc. of WWW'06, 3–12. ACM.

Poole, D. 2003. First-order probabilistic inference. In Proc. IJCAI'03, volume 3, 985–991.

Reiter, R. 1978. On closed world data bases. Logic and Data Bases 55–76.

Reiter, R. 1980. A logic for default reasoning. Artificial intelligence 13(1):81–132.

Shin, J.; Wu, S.; Wang, F.; De Sa, C.; Zhang, C.; and Ré, C. 2015. Incremental knowledge base construction using deepdive. Proc. of VLDB 8(11):1310–1321.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In Proc. of NIPS'13, 926–934.

Suciu, D.; Olteanu, D.; Ré, C.; and Koch, C. 2011. Proba- bilistic Databases.

Sutton, C., and McCallum, A. 2011. An introduction to conditional random fields. Machine Learning 4(4):267–373.

Tseitin, G. S. 1983. On the complexity of derivation in propositional calculus. In Automation of reasoning. Springer. 466–483.

Valiant, L. G. 1979. The complexity of computing the per- manent. Theor. Comput. Sci. 8:189–201.

Van den Broeck, G. 2013. Lifted Inference and Learning in Statistical Relational Models. Ph.D. Dissertation, KU Leu- ven.

Wang, W. Y.; Mazaitis, K.; and Cohen, W. W. 2013. Pro- gramming with personalized pagerank: a locally groundable first-order probabilistic logic. In Proc. of CIKM, 2129–2138. ACM.

Wu, W.; Li, H.; Wang, H.; and Zhu, K. Q. 2012. Probase: A probabilistic taxonomy for text understanding. In Proc. of SIGMOD, 481–492. ACM.