Non-Standard-Datenbanken

Probabilistische Datenbanken

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Presentations have been adapted from

Lifted Probabilistic Inference in Relational Models
Guy Van den Broeck, KU Leuven
Dan Suciu, U. of Washington

A TUTORIAL ON PROBABILISTIC DATABASES
Dan Suciu, U. of Washington

Top-K Query Evaluation on Probabilistic Data
Christopher Ré, Nilesh Dalvi and Dan Suciu

Open-World Probabilistic Database
Ismail Ilkan Ceylan, Adnan Darwiche and Guy Van den Broeck

Weighted First-Order Model Counting (WFOMC)

Modell = Erfüllende Belegung einer aussagenlogischen Formel Δ

$\Delta = \forall d\ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday **Tuesday**}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 3 * 4 * 6 = 72
T	F	T	T	No	0
F	T	T	T	Yes	2 * 3 * 4 * 6 = 144
F	F	T	T	Yes	2 * 5 * 4 * 6 = 240
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 3 * 1 * 6 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 3 * 1 * 6 = 36
F	F	F	T	Yes	2 * 5 * 1 * 6 = 60
T	T	F	F	Yes	1 * 3 * 1 * 2 = 6
T	F	F	F	No	0
F	T	F	F	Yes	2 * 3 * 1 * 2 = 12
F	F	F	F	Yes	2 * 5 * 1 * 2 = 20
				+	+
				#SAT = 9	WFOMC = 608

Gogate, V., & Domingos, P., Probabilistic Theorem Proving. Proc. UAI, **2012**.

Van den Broeck, G., Taghipour, N., Meert, W., Davis, J., & De Raedt, L., Lifted probabilistic inference by first-order knowledge compilation. In Proc.IJCAI-11, pp. 2178-2185, **2011**.

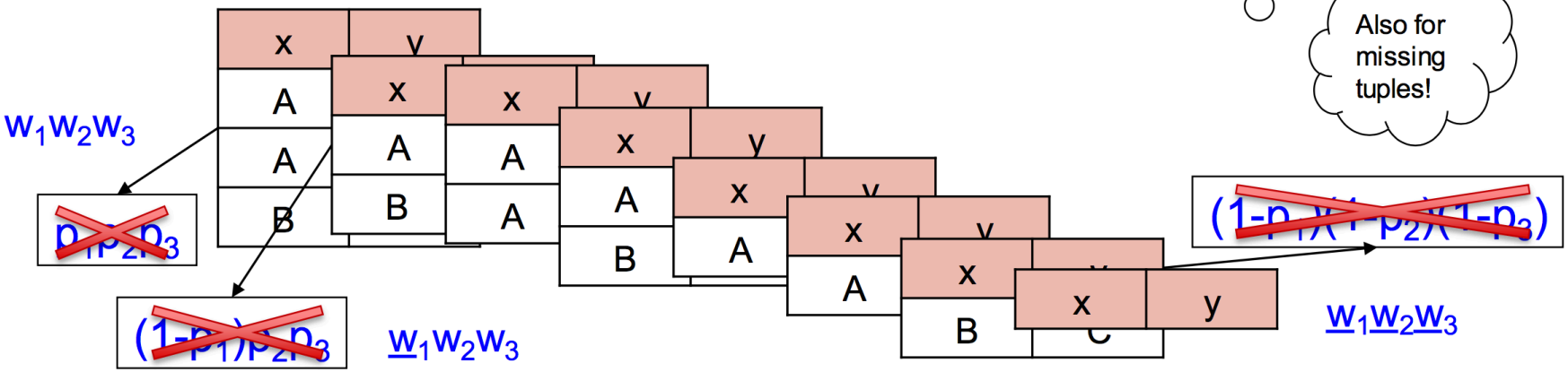
Von Wahrscheinlichkeiten zu Gewichten

Friend

x	y	P
A	B	p_1
A	C	p_2
B	C	p_3

→

x	y	w(Friend(x,y))	w(¬Friend(x,y))
A	B	$w_1 = p_1$	$\underline{w_1} = 1-p_1$
A	C	$w_2 = p_2$	$\underline{w_2} = 1-p_2$
B	C	$w_3 = p_3$	$\underline{w_3} = 1-p_3$
A	A	$w_4 = 0$	$\underline{w_4} = 1$
A	C	$w_5 = 0$	$\underline{w_5} = 1$
	…	…	



Intensionale Anfrageevaluation

- ProbDB D + Anfrage Q
  → Herkunftsformel (*lineage expression)* F
  - Boolesche Variablen w_1, $\underline{w}_1$, ...
    korrespondierend zu Tupeln t1, t2, ...
  - Die Herkunftsformel F sagt, wann Q wahr ist
- Berechne P(F) mit DPLL-artigem System

N. Fuhr; T. Rölleke, A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. ACM Transactions on Information Systems 14(1), **1997**.

Diskussion

- Einfache Idee: Ersetze p, 1-p durch w, w
  - Gewichte nicht-notwendigerweise Wahrscheinlichkeiten
- Anfragebeantwortung durch WFOMC
  - Für Wahrscheinlichkeitsraum:
    Dividiere Weltgewicht durch Z = Summe aller Weltgewichte

- Warum Gewichte statt Wahrscheinlichkeiten?
- Verschiedene Formalismen zur Darstellung von Wahrscheinlichkeitsverteilungen

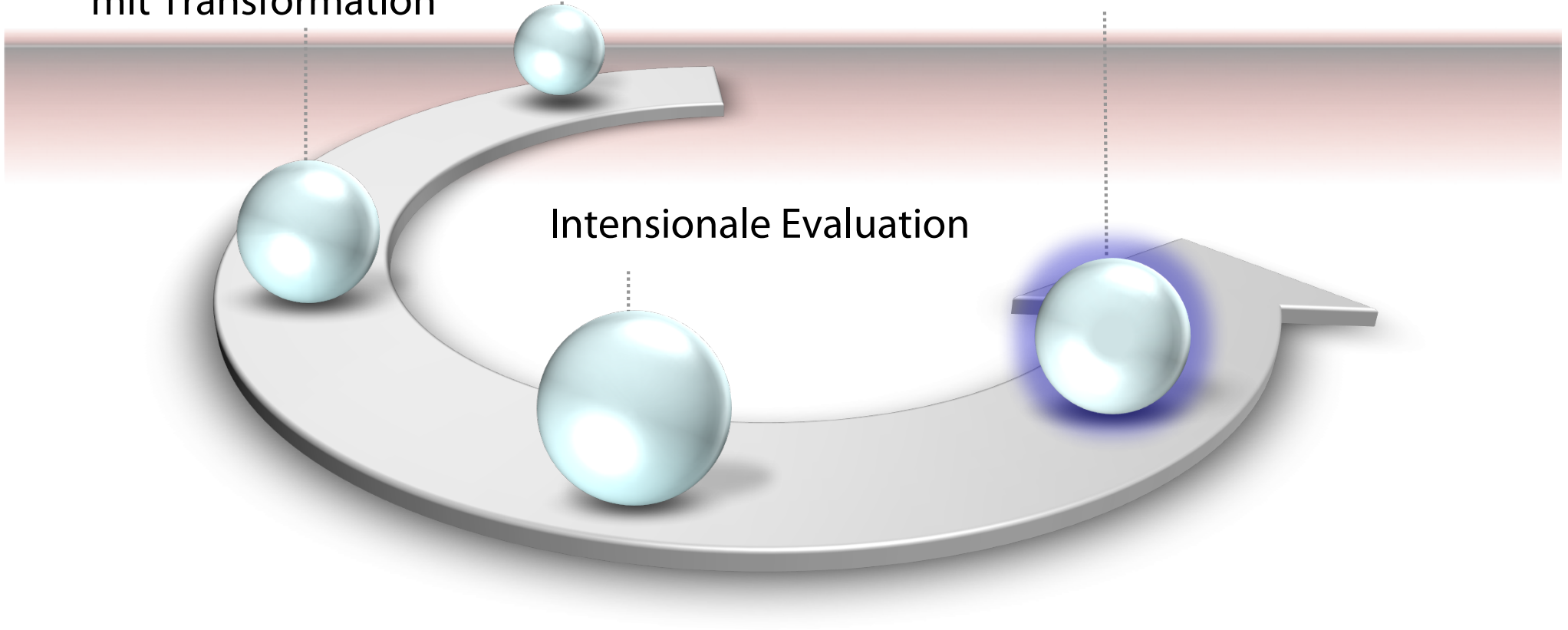
Non-Standard-Datenbanken

Probabilistische Datenbanken

Extensionale Anfragebeantwortung mit Transformation

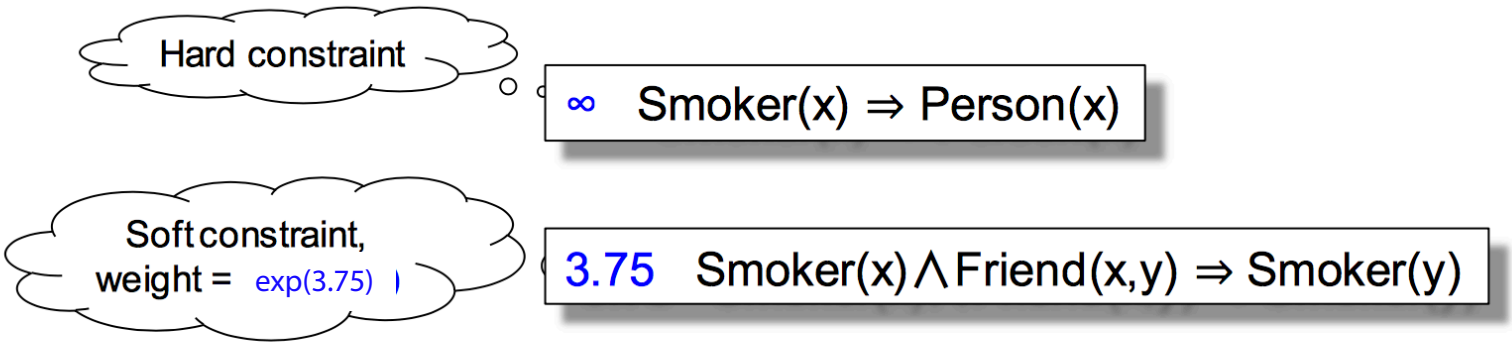
Probabilistisches Datenmodell

Anfragebeantwortung bzgl. probabilistischer Constraints



Markov-Logik

Gewichtete Formeln zur Modellierung von Einschränkungen



An MLN is a set of constraints (w, Γ(**x**)), where w=weight, Γ(**x**)=FO formula

Weight of a world = product of exp(w), for all MLN rules (w, Γ(**x**))
and grounding Γ(**a**) that hold in that world

Probability of a world = Weight / Z
Z = sum of weights of all worlds (no longer a simple expression!)

Richardson, M., & Domingos, P. Markov logic networks. Machine learning, 62(1-2), 107-136. **2006**.

Warum exp?

- Log-linear-Modelle
- Sei D eine Menge von Konstanten …
- … und $\omega \in \{0,1\}^m$ eine Welt mit m Atomen bzgl. D
- $weight(\omega) = \Pi_{\{ (w, \Gamma(\mathbf{x})) \in MLN \mid \exists \mathbf{a} \in D^n : \omega \vDash \Gamma(\mathbf{a}) \}} \; exp(w)$
- $ln(weight(\omega)) = \Sigma_{\{(w, \Gamma(\mathbf{x})) \in MLN \mid \exists \mathbf{a} \in D^n : \omega \vDash \Gamma(\mathbf{a})\}} \; w$
  - Summe ermöglicht komponentenweise Optimierung beim Lernen der Gewichte bei gegebenen Präferenzen (Gewichte) über Welten
- $Z = \Sigma_{\omega \in \{0,1\}^m} \; ln(weight(\omega))$
- $P(\omega) = ln(weight(\omega)) \; / \; Z$

Einschub

- Gegeben:
  - Zustände, Formeln, Ereignisse,…: $s_1, s_2, \ldots, s_n$
  - Dichte $p(s) = p_s$
- Maximum-Entropie-Prinzip:
  - Ohne weitere Information, wähle Dichte p_s, so dass Entropie maximiert wird

$$-\sum_s p_s(s)\ log\ p_s(s) = \quad -p_s \log p_s$$

  - in Bezug auf Einschränkungen

$$\sum_s p_s f_i(s) = D_i, \quad \forall i$$

Einschub

- Betrachte Lagrange-Funktional zur Bestimmung von p_s

$$L = -p_s \log p_s - \sum_i \lambda_i (\sum_s p_s f_i(s) - D_i) - \mu(\sum_s p_s - 1)$$

- Partielle Ableitungen von L in Bezug auf p_s →

  Nullstellenbestimmung ergibt (Boltzmann-Gibbs-Dichte):

$$p_s(s) = \frac{\exp\left(-\sum_i \lambda_i f_i(s)\right)}{Z}$$

  wobei Z ein geeigneter Normalisierungsfaktor ist

Anfragebeantwortungsproblem

Gegeben

MLN:

$$0.7\ \text{Actor}(a) \Rightarrow \neg\text{Director}(a)$$
$$1.2\ \text{Director}(a) \Rightarrow \neg\text{WorkedFor}(a,b)$$
$$1.4\ \text{InMovie}(m,a) \wedge \text{WorkedFor}(a,b) \Rightarrow \text{InMovie}(m,b)$$

Datenbanktabellen (wenn fehlend w=1)

Actor:

Name	w
Brando	2.9
Cruise	3.8
Coppola	1.1

WorkedFor:

Actor	Director	w
Brando	Coppola	2.5
Coppola	Brando	0.2
Cruise	Coppola	1.7

Berechne

$$P(\text{InMovie}(\text{GodFather}, \text{Brando})) = ??$$

Z-Berechnung

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty,\Gamma(\mathbf{x}))\in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN

b) Add new probabilistic relation $F_i(\mathbf{x})$

c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

For all constants **A**, relations F_i,

set $w(F_i(\mathbf{A})) = \exp(w_i)$, $w(\neg F_i(\mathbf{A})) = 1$

Theorem: $Z = WFOMC(\Delta)$

Van den Broeck, G., Meert, W., & Darwiche, A.,. Skolemization for weighted first-order model counting. In Proc. KR-13, **2013**.

Jha, A., & Suciu, D.,.Probabilistic databases with MarkoViews. Proceedings of the VLDB Endowment, 5(11), 1160-1171, **2012**.

Beispiel

1. Formula Δ

∞	Smoker(x) ⇒ Person(x)
3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

$$\Delta = \forall x\,(\text{Smoker}(x) \Rightarrow \text{Person}(x)) \wedge \forall x \forall y\,(F(x,y) \Leftrightarrow [\text{Smoker}(x) \wedge \text{Friend}(x,y) \Rightarrow \text{Smoker}(y)])$$

2. Weight function w(.)

F

x	y	w(F(x,y))	w(¬F(x,y))
A	A	exp(3.75)	1
A	B	exp(3.75)	1
A	C	exp(3.75)	1
B	A	exp(3.75)	1
	. . .	. . .	

Note: if no tables given for Smoker, Person, etc, (i.e. no evidence) then set their w = $\underline{w}$ = 1

$$Z = WFOMC(\Delta)$$

Weighted First-Order Model Counting

Modell = Erfüllende Belegung einer aussagenlogischen Formel Δ

$\Delta = \forall d\ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday **Tuesday**}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

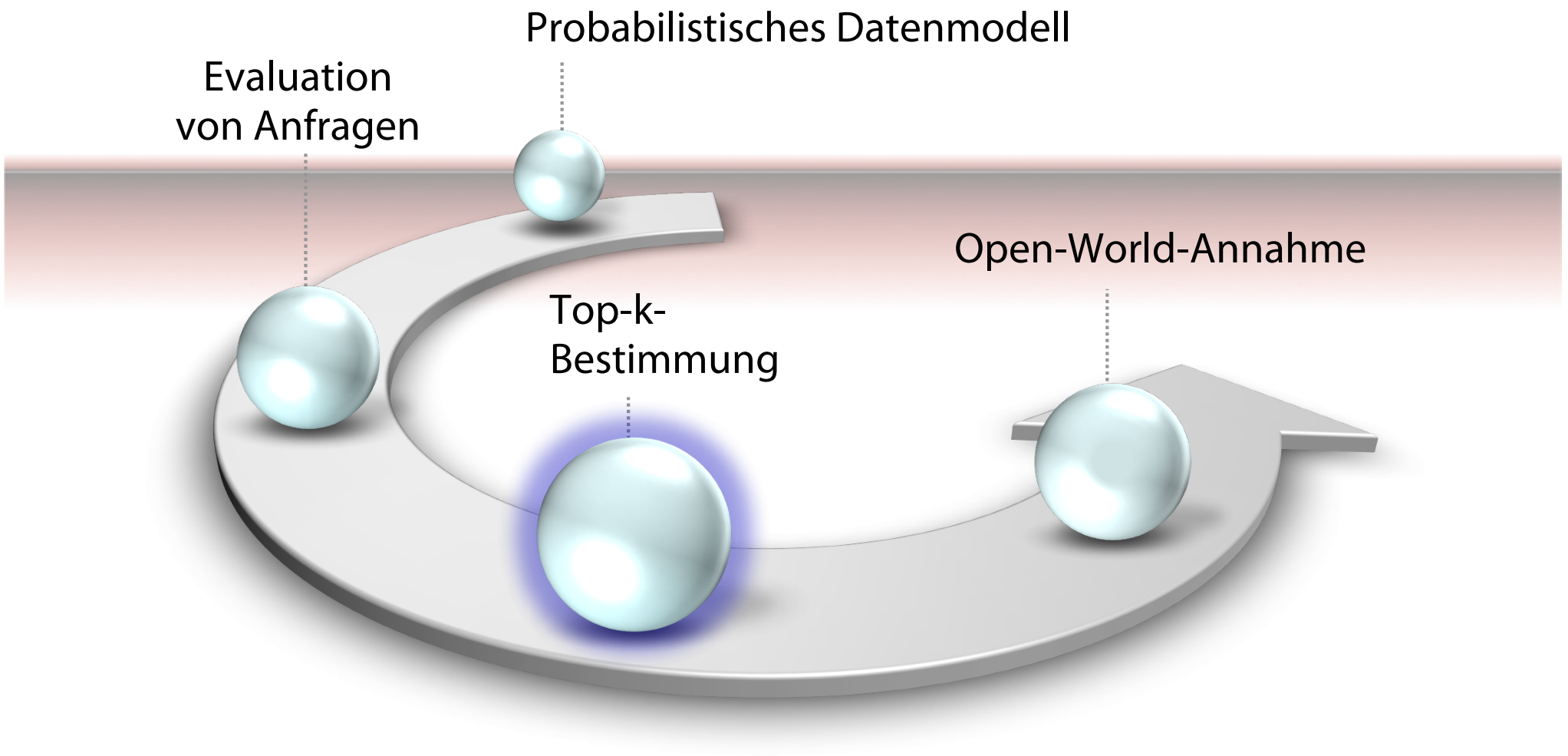
Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 3 * 4 * 6 = 72
T	F	T	T	No	0
F	T	T	T	Yes	2 * 3 * 4 * 6 = 144
F	F	T	T	Yes	2 * 5 * 4 * 6 = 240
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 3 * 1 * 6 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 3 * 1 * 6 = 36
F	F	F	T	Yes	2 * 5 * 1 * 6 = 60
T	T	F	F	Yes	1 * 3 * 1 * 2 = 6
T	F	F	F	No	0
F	T	F	F	Yes	2 * 3 * 1 * 2 = 12
F	F	F	F	Yes	2 * 5 * 1 * 2 = 20
				+	+
				#SAT = 9	WFOMC = 608

Gogate, V., & Domingos, P. Probabilistic theorem proving. Proc. UAI, **2012**.

Van den Broeck, G., Taghipour, N., Meert, W., Davis, J., & De Raedt, L., Lifted probabilistic inference by first-order knowledge compilation. In Proc.IJCAI-11, pp. 2178-2185, **2011**.

Non-Standard-Datenbanken

Probabilistische Datenbanken



Top-K Query Evaluation on Probabilistic Data

Christopher Ré, Nilesh Dalvi and Dan Suciu

Google-Patent ☹

Publication number	US7814113 B2
Publication type	Grant
Application number	US 11/935,230
Publication date	12 Oct 2010
Filing date	5 Nov 2007
Priority date ⓘ	7 Nov 2006
Fee status ⓘ	Paid
Also published as	US20080109428
Inventors	Dan Suciu, Christopher Re
Original Assignee	University Of Washington Through Its Center For Commercialization
Export Citation	BiBTeX, EndNote, RefMan

Patent Citations (7), Non-Patent Citations (3), Referenced by (9), Classifications (6), Legal Events (4)

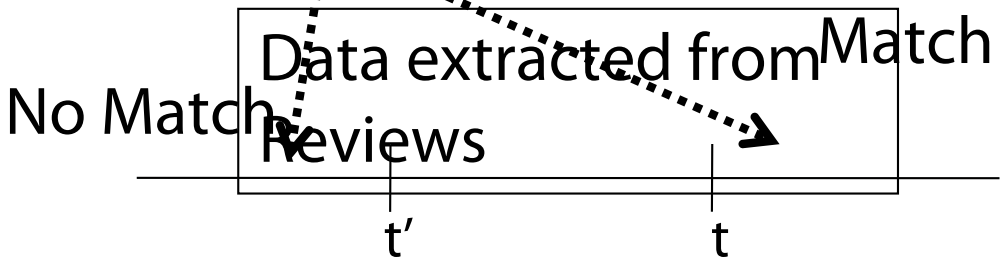
External Links: USPTO, USPTO Assignment, Espacenet

Unschärfe ist überall…

RID	Title
r124	12 Monkeys
r155	Twelve Monkeys
r175	2 Monkey
r194	Monk

MID	Title
m232	12 Monkeys
m143	Monkey Love

Fellegi-Sunter-Ansatz:
Score für jedes (RID,MID)



Clean IMDB Data

Our Approach:
Output: (RID,MID) pairs
Convert scores to probabilities

Fellegi, Ivan; Sunter, Alan. "A Theory for Record Linkage" Journal of the American Statistical Association. 64 (328): pp. 1183–1210, **1969**.

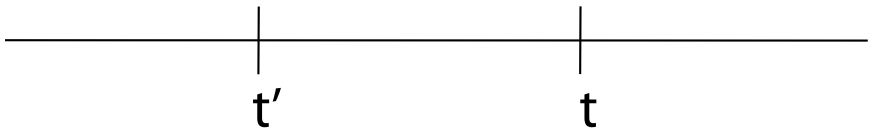
Unschärfe ist überall…

RID	Title
r124	12 Monkeys
r155	Twelve Monkeys
r175	**2 Monkey**
r194	Monk

MID	Title
m232	**12 Monkeys**
m143	**Monkey Love**

Fellegi-Sunter-Ansatz:
Score für jedes (RID,MID)

No Match Match



RID	MID	Prob
r175	m232	0.8
r175	m143	0.2

Fellegi, Ivan; Sunter, Alan. "A Theory for Record Linkage" Journal of the American Statistical Association. 64 (328): pp. 1183–1210, **1969**.

Anfragebeantwortung mit Herkunftsformel

- Intensionale Anfrage-beantwortung [FR97]
- Assoziiere mit jedem Tupel ein Ereignis
- Wahrscheinlichkeit, dass Ereignis eintritt / erfüllt ist = Anfragewert

RID	MID	Prob	
r175	m232	0.8	e_1
r175	m143	0.2	e_2

Anfrageverarbeitung generiert Ereignisausdruck

$\times((t_1,[e_1]),(t_2,[e_2]))$	$=((t_1,t_2),[e_1 \wedge e_2])$
$\sigma_c((t,[e]))$	$=\begin{cases}(t,[e]) & c(t) = \text{true} \\ \bot & c(t) = \text{false}\end{cases}$
$\Pi_A((t_1,[e_1]),\ldots,(t_n,[e_n]))$	$=(t_1[A],[e_1 \vee \cdots \vee e_n])$

Projektion sorgt für DNF

N. Fuhr; T. Rölleke, A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. ACM Transactions on Information Systems 14(1), **1997**.

Problemdefinition

Gegeben: $G = \{t_1, \ldots, t_n\}$ eine Menge von n Objekten mit unbekannten Wahrscheinlichkeiten $p_1, \ldots, p_n$ und eine Zahl $k \leq n$.

Ziel: Finde Menge von k Objekten mit höchsten Wahrscheinlichkeiten, genannt Top-k-Teilmenge von G

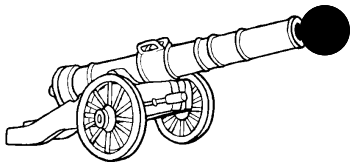
Lösungsidee: Verwende parallele Monte-Carlo-Simulationen, eine für jede Kandidatenantwort, und approximiere Wahrscheinlichkeiten nur soweit, wie es nötig ist, um die k besten Antworten zu finden

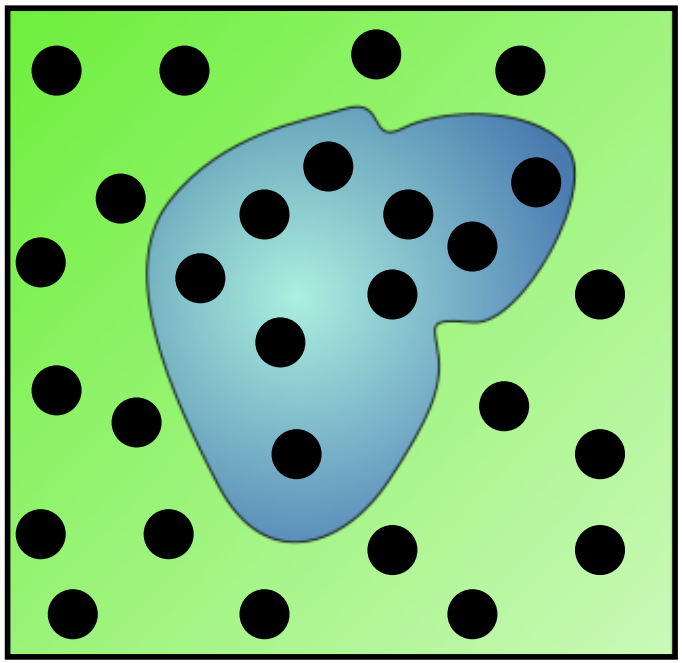
Monte-Carlo-Simulation: Einführung

- Gegeben: Gesamtfläche eines Landstücks (Quadrat)
- Wie kann die Fläche des darauf liegenden Sees bestimmt werden?

$\text{Fläche}_{\text{Land}} = 1000\ m^2$ (Quadrat)
X = Anzahl Kanonenschüsse
N = Einschläge auf grünem Land

$$\frac{\text{Fläche}_{\text{Land}}}{\text{Fläche}_{\text{See}}} = \frac{X}{X-N} \Rightarrow \text{Fläche}_{\text{See}} = \frac{(X-N)\times \text{Fläche}_{\text{Land}}}{X}$$





$$\text{Fläche}_{\text{See}} = 1000$$

$$\text{Fläche}_{\text{See}} = 500$$

$$\text{Fläche}_{\text{See}} = 333.\overline{3}$$

...

$$\text{Fläche}_{\text{See}} = 375$$

Monte Carlo Simulation: (ε, δ)-Approximation

Function MS-Naiv(G)
 Wähle N mal zufällig eine mögliche Welt.
 Berechne Wahrheitswert von DNF-Formel G.
 Wahrscheinlichkeit p=P(G) approximiert durch Frequenz $\tilde{p}$, mit der G wahr wird

Function MS-Karp-Luby(G)
 Fix an order on the disjuncts: $\{t_1, t_2, \ldots, t_m\} = G$
 $C := 0$
 repeat
  Choose a random disjunct $t_i \in G$
  Choose a random truth assignment s.t. $t_i = true$
  if forall $j < i$ *it holds that* $t_j = false$ then $C := C + 1$
 until *N times*
 return C/N

$\tilde{p}$:= Karp-Luby(G) $\quad \varepsilon = \sqrt{4m \log(2/\delta)/N} \quad a^N = \tilde{p} - \varepsilon \quad b^N = \tilde{p} + \varepsilon$

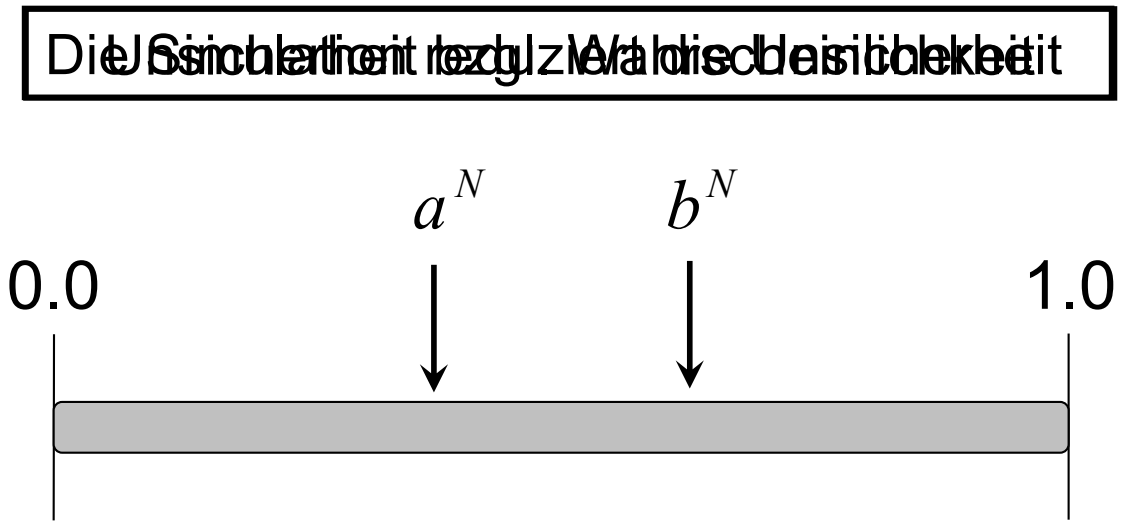
Richard M. Karp, Michael Luby, Monte-Carlo Algorithms for Enumeration and Reliability Problems. FOCS: 56-64, **1983**.

Paul Dagum, Richard M. Karp, Michael Luby, Sheldon M. Ross: An Optimal Algorithm for Monte Carlo Estimation. SIAM J. Comput. 29(5): 1484-1496, **2000**.

Karp-Luby

Nach N Simulationen garantiert:

$$\mathbf{P}(p \in [a^N, b^N]) \quad > \quad 1 - \delta$$ Konfidenzintervall



Anfragebeantwortung

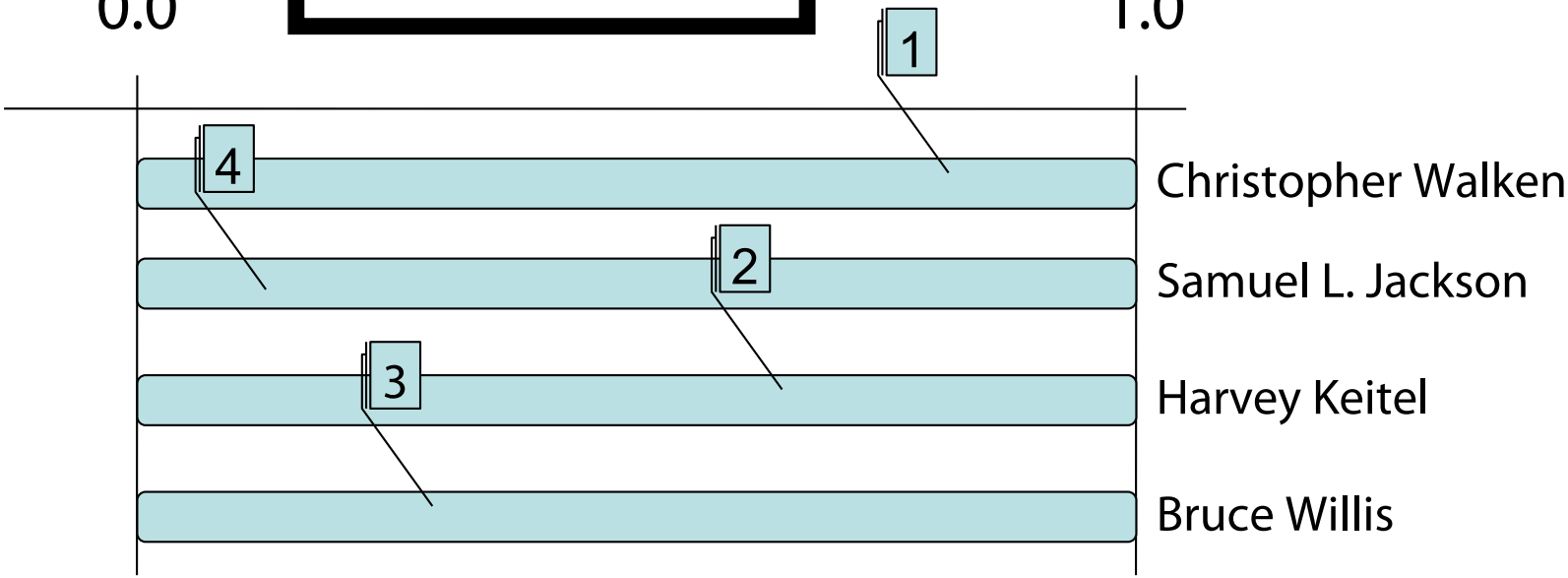
Anfrage: Finde Top-k Regisseure von guten Filmen (Score ≥ 4)
NB: Ranking nach P-Werten

Verfahren: Simuliere für jeden Kandidaten mit Karp-Luby bis ε klein

MID	Director
m232	Christopher Walken
…	…

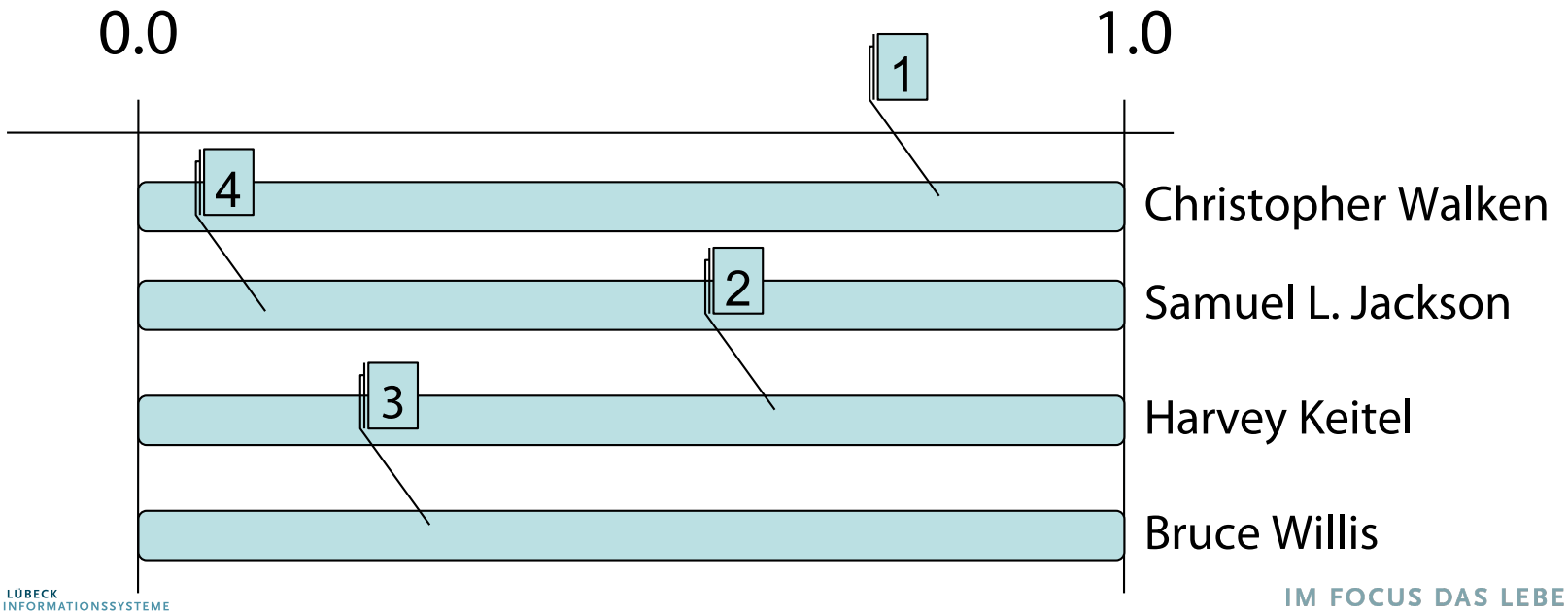
RID	Score
r157	4
…	…

Geht das besser?



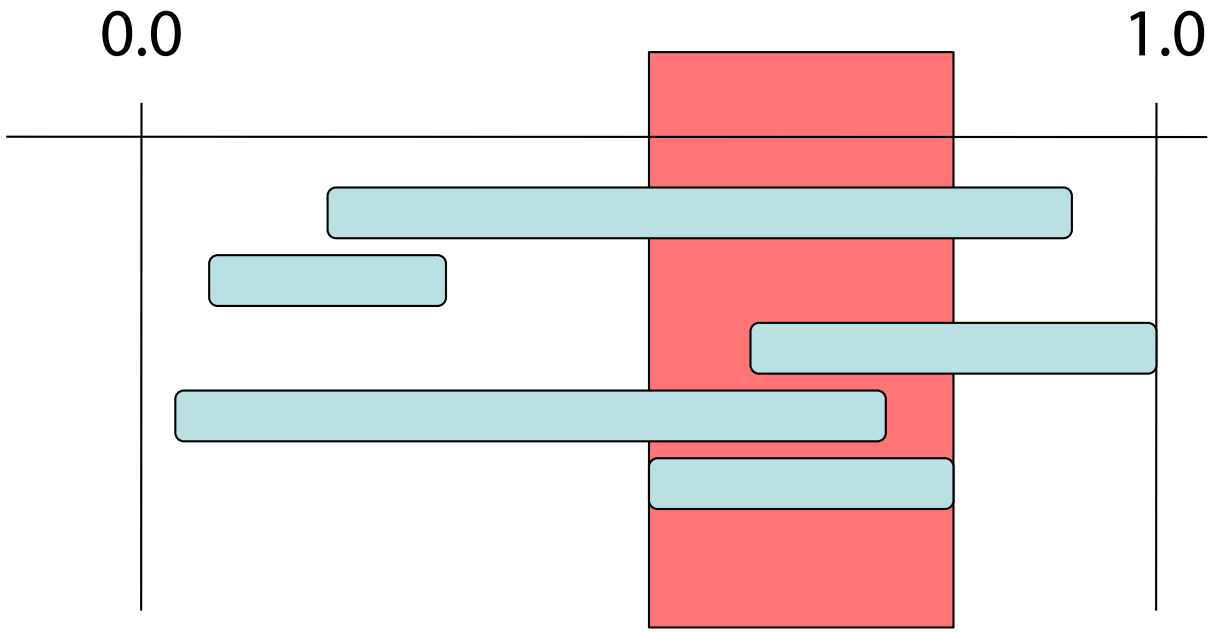
Besseres Verfahren: Multisimulation

- Trenne Top-K mit wenigen Simulationen
  - Betrachte Intervalle im Top-k-Bereich
  - Am Ende: Intervalle verschränkt
- Vergleich mit imaginärem Verfahren OPT
  - "Weiß" welche Intervall betrachtet werden müssen



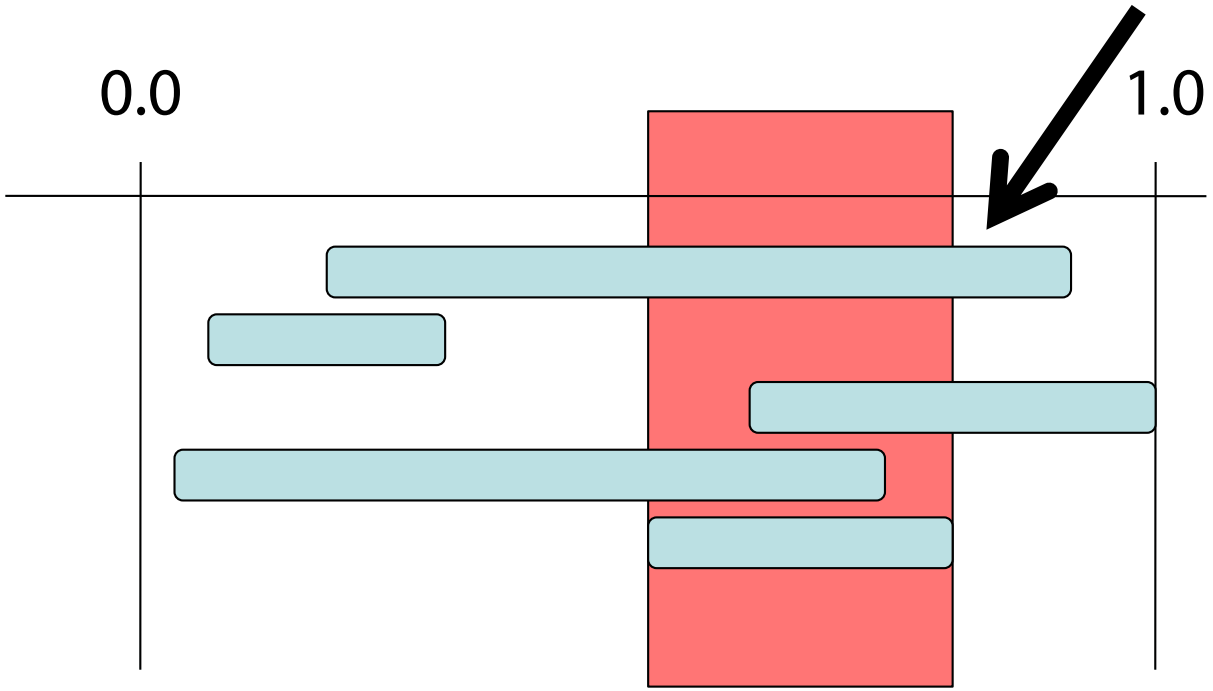
Die kritische Region

- Die kritische Region ist das Intervall
  - (k-höchste Min, k+1-höchste Max)
  - Für k = 2



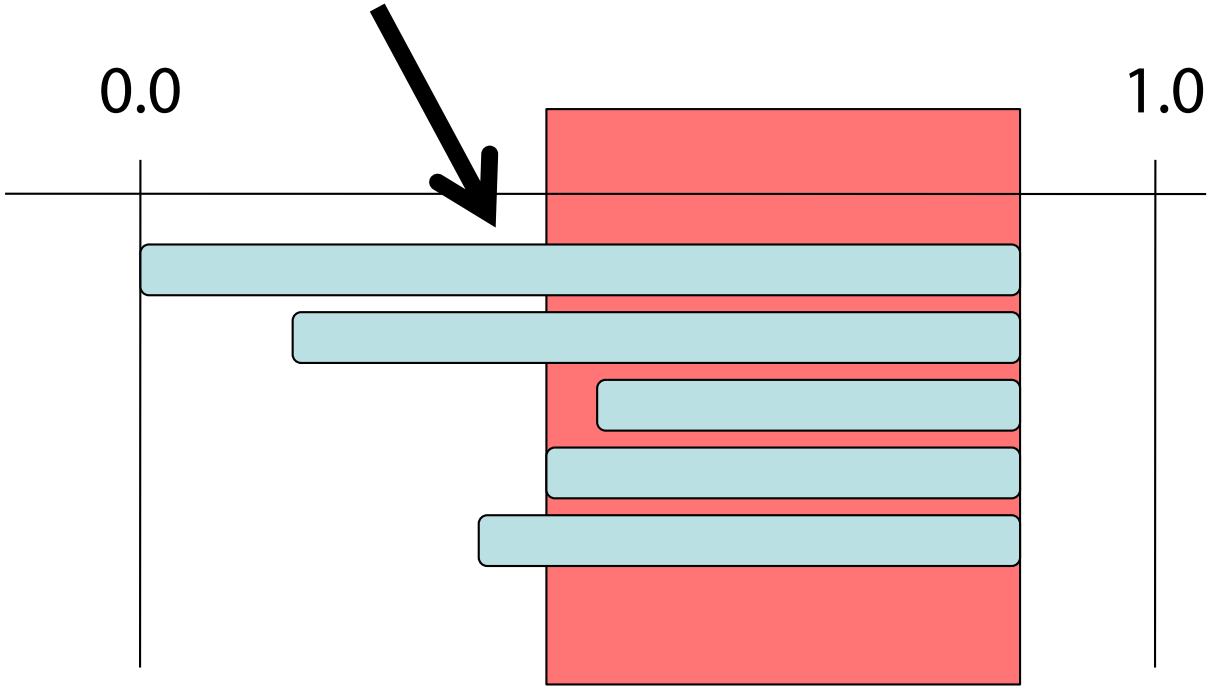
Drei einfache Regeln: Regel 1

- Wähle “Double Crosser”
  - *OPT* muss dieses Intervall auch wählen



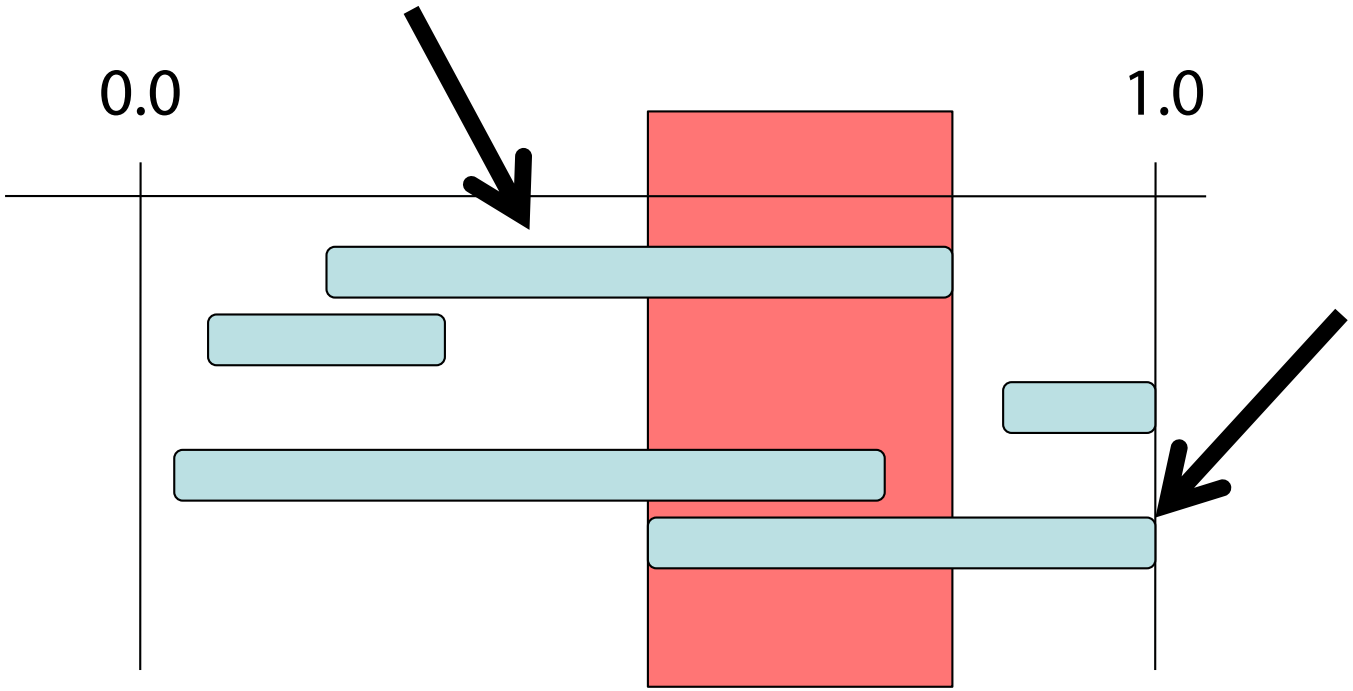
Drei einfache Regeln: Regel 2

- Nur noch "Lower/Upper Crosser"? dann wähle maximale Intervall
  - *OPT* muss das auch machen



Drei einfache Regeln: Regel 3

- Wähle Upper- und Lower-Crosser
  - *OPT* könnte nur ein Intervall wählen



Multisimulation (MS)

function MS_TopK($\{G_1, \ldots, G_n\}$, k):

  $[a_1, b_1] := \ldots := [a_n, b_n] = [0, 1]$

repeat

  $(c, d) := (\text{topk}(a_1, \ldots, a_n), \text{topk+1}(b_1, \ldots, b_n))$

  $T := \{G_i \mid d \leq a_i\}$

  $B := \{G_i \mid b_i \leq c\}$

  Case 1: choose a double crosser to simulate(T, B)

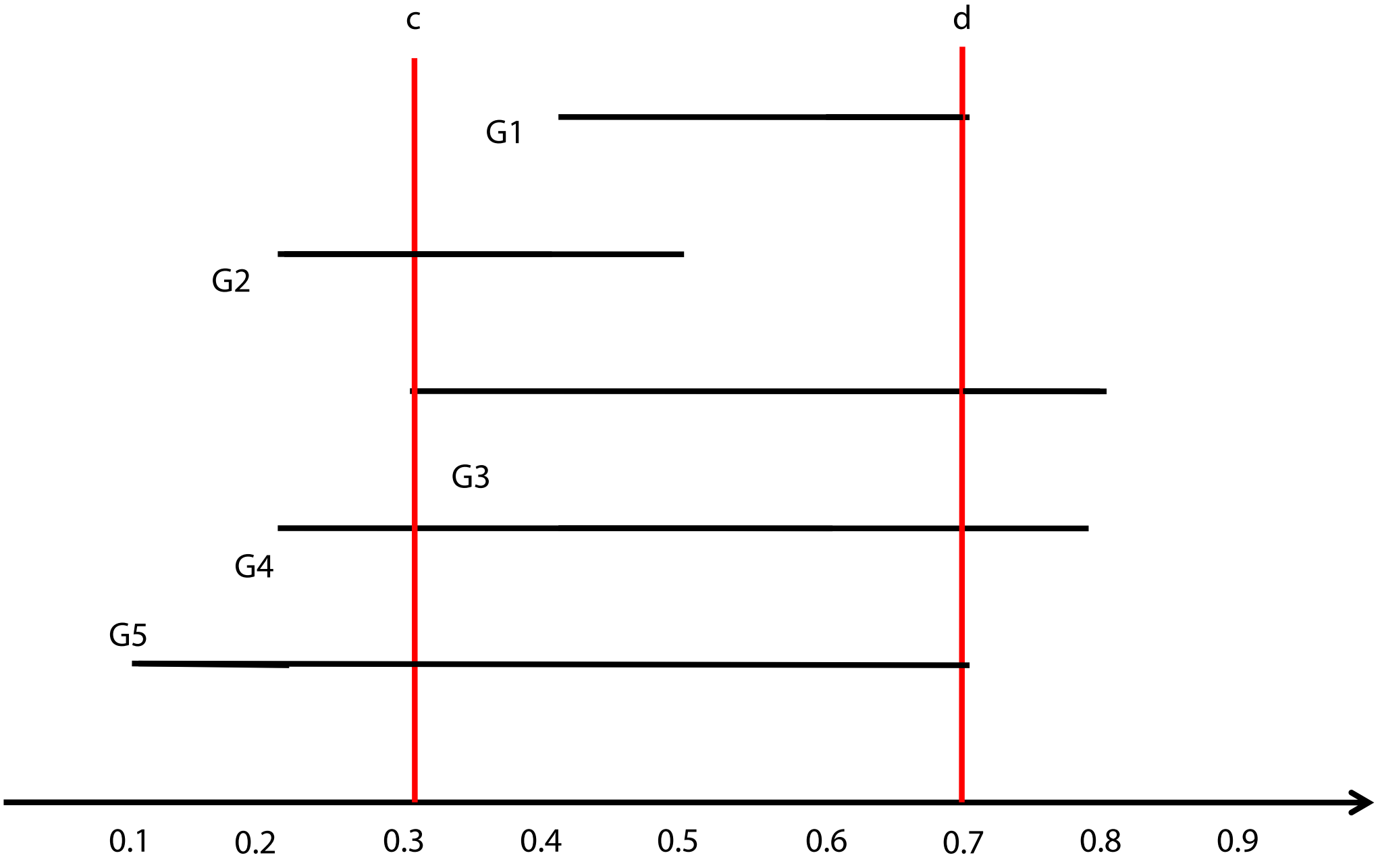
  Case 2: choose upper and lower crosser to simulate(T, B)

  Case 3: choose a maximal crosser to simulate(T, B)

until $c > d$

return T

Example : Let us select Top 2

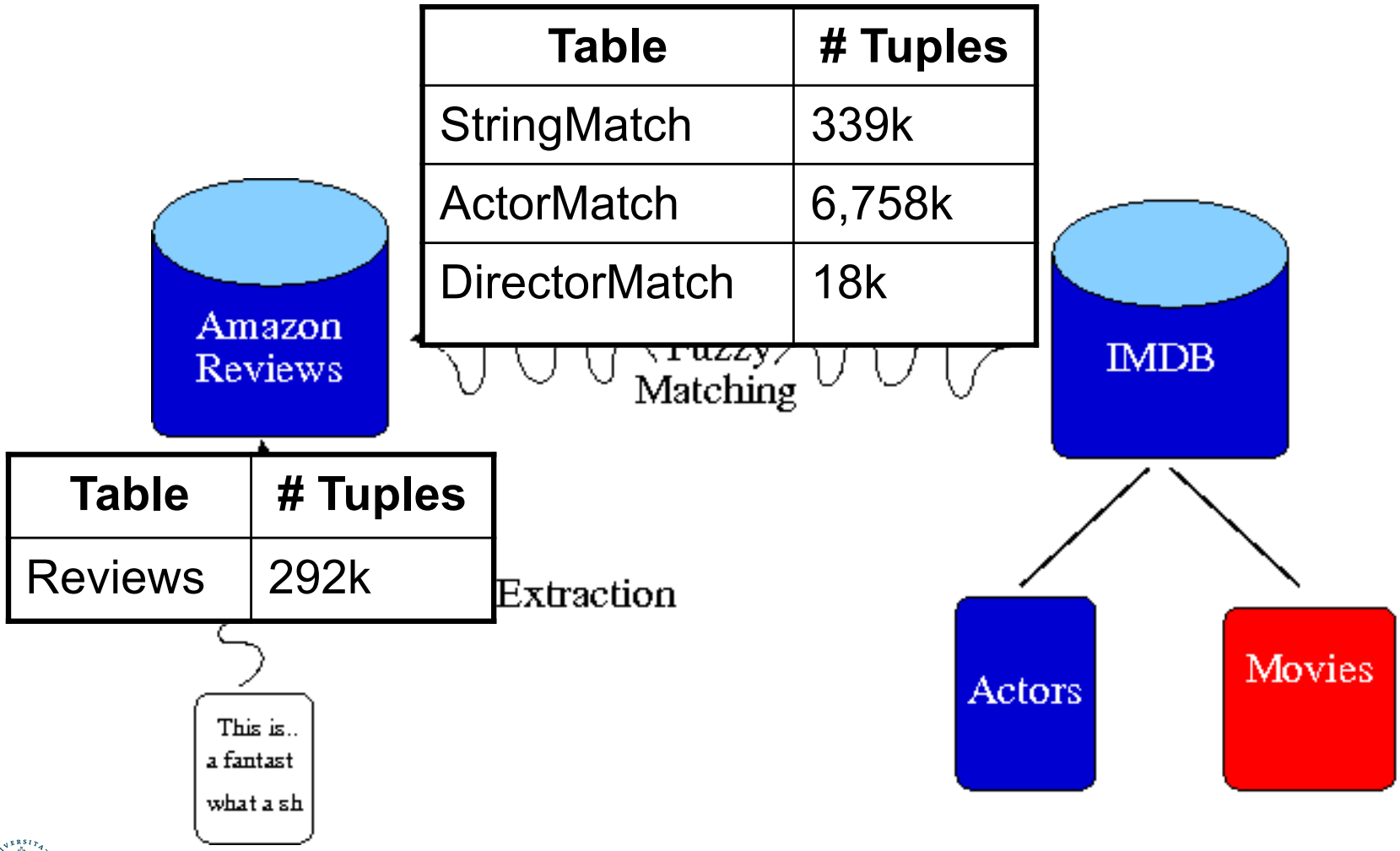


Multisimulation ist 2-approximierend

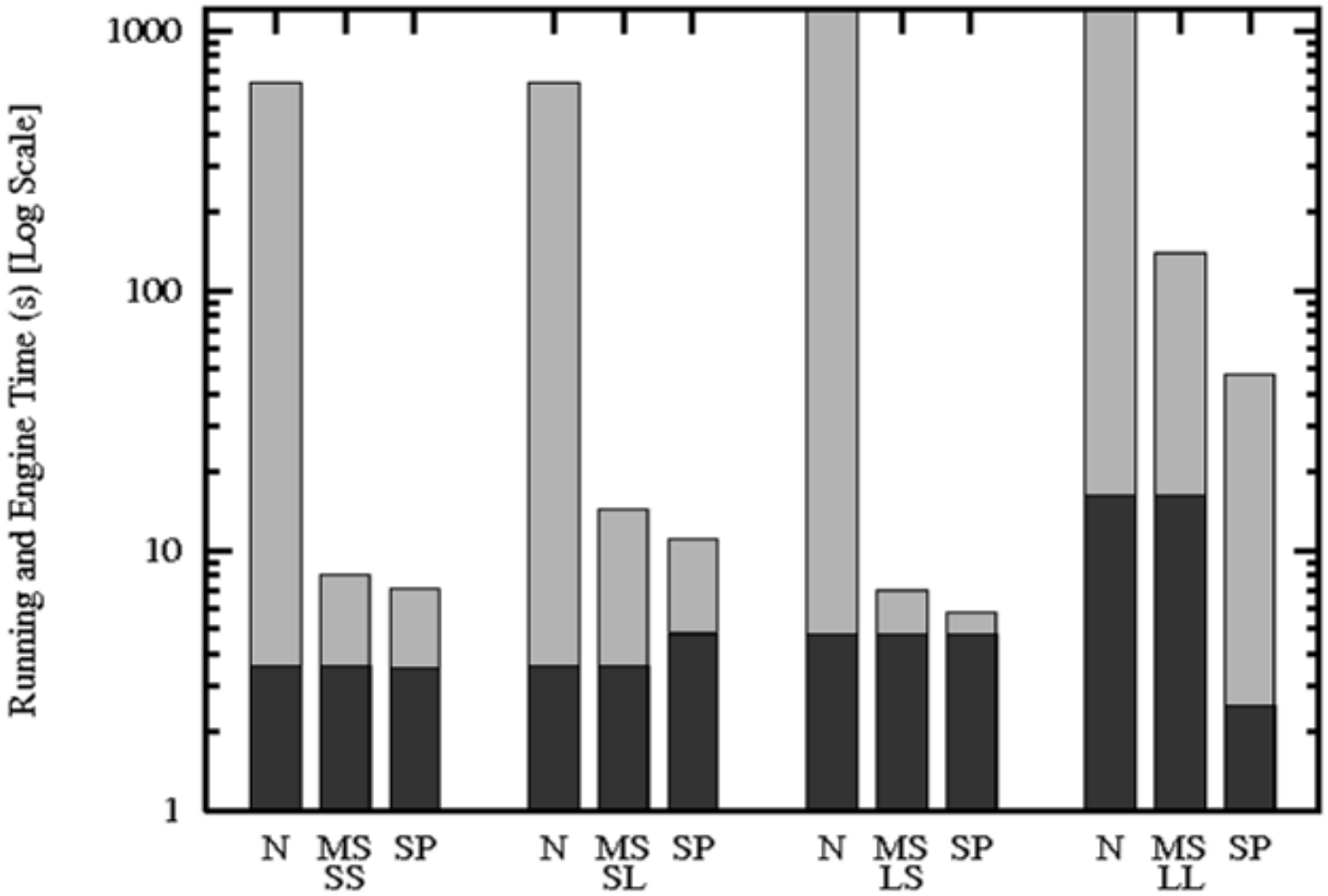
- Theorem [DS07]: Multisimulation führt höchstens zweimal soviele Simulationen aus wie OPT
  - Und: kein deterministischer Algorithmus kann auf beliebigen Probleminstanzen besser arbeiten
- Varianten
  - Top-k-Menge (gezeigt)
  - Anytime (produziere von 1 bis k)
  - Rang (produziere top-k nach Rang sortiert)
  - Alle (alle Intervalle nach Rang sortiert)

Nilesh Dalvi and Dan Suciu. 2007. Efficient query evaluation on probabilistic databases. *The VLDB Journal* 16, 4, 523-544, **2007**.

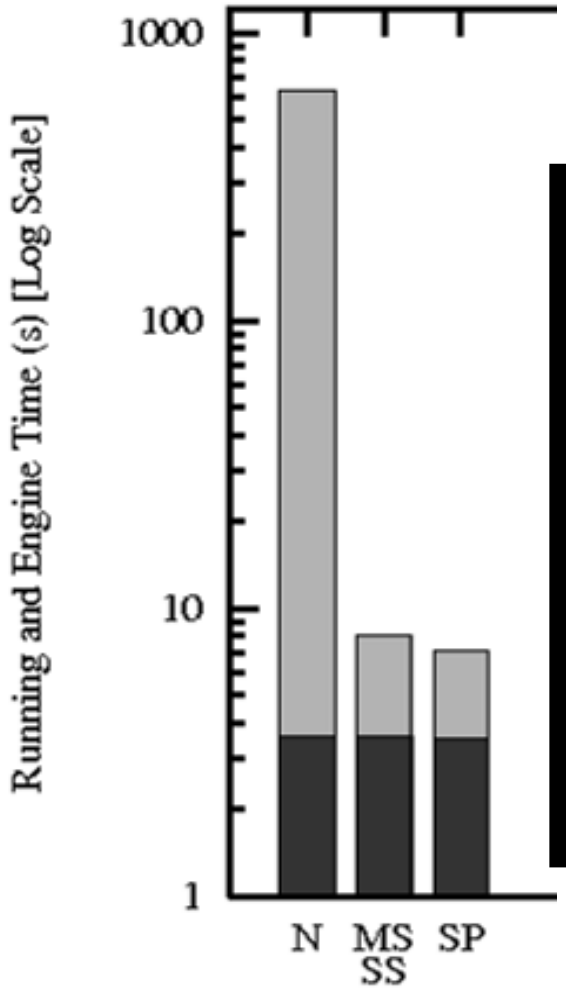
Experimente: Unsichere Tupel



Laufzeiten [**N(aiv), MS, and S(afe) P(lan)**]



Laufzeiten



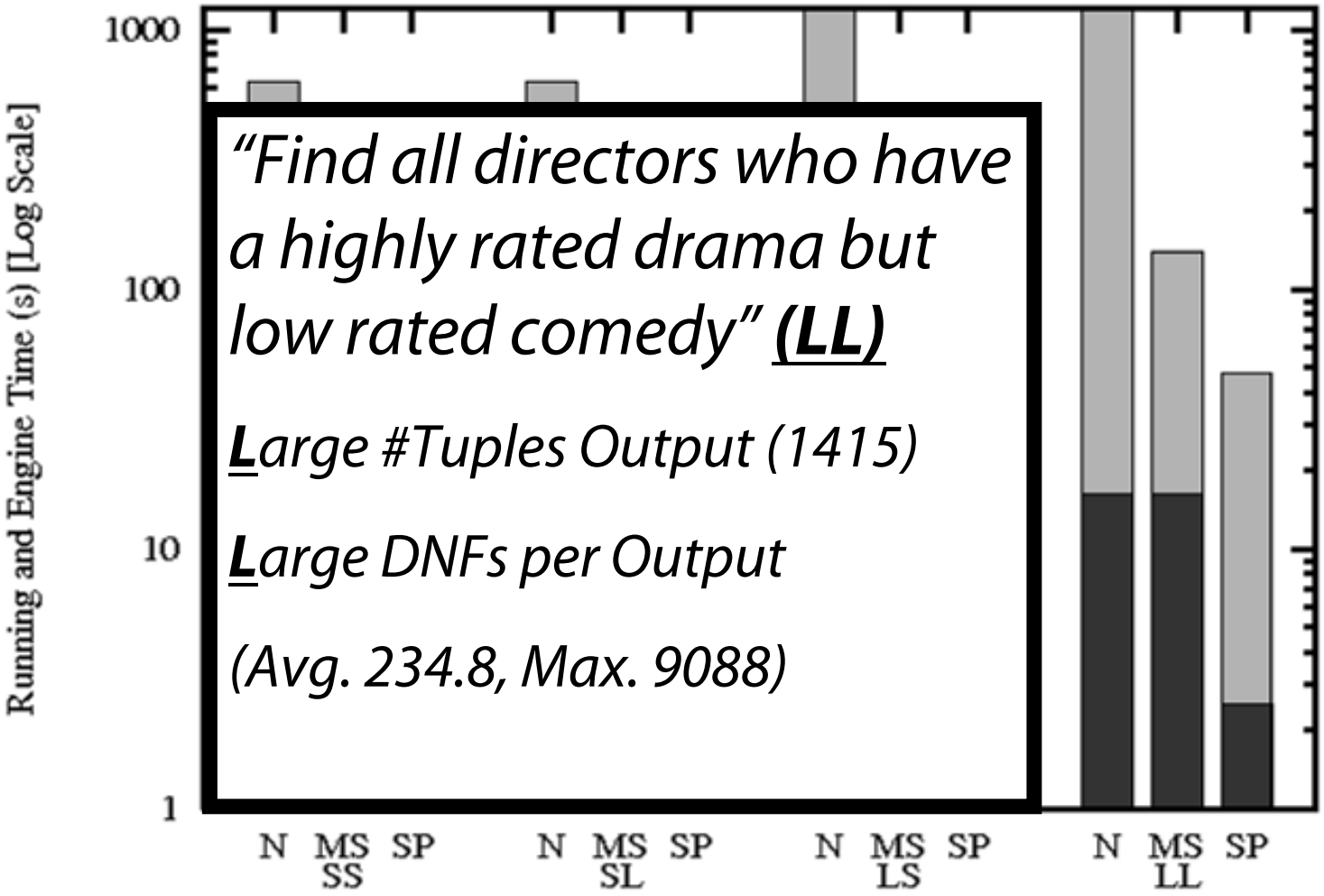
"Find all years in which Anthony Hopkins was in a highly rated movie"
(SS)

S*mall Number of Tuples Output (33)*

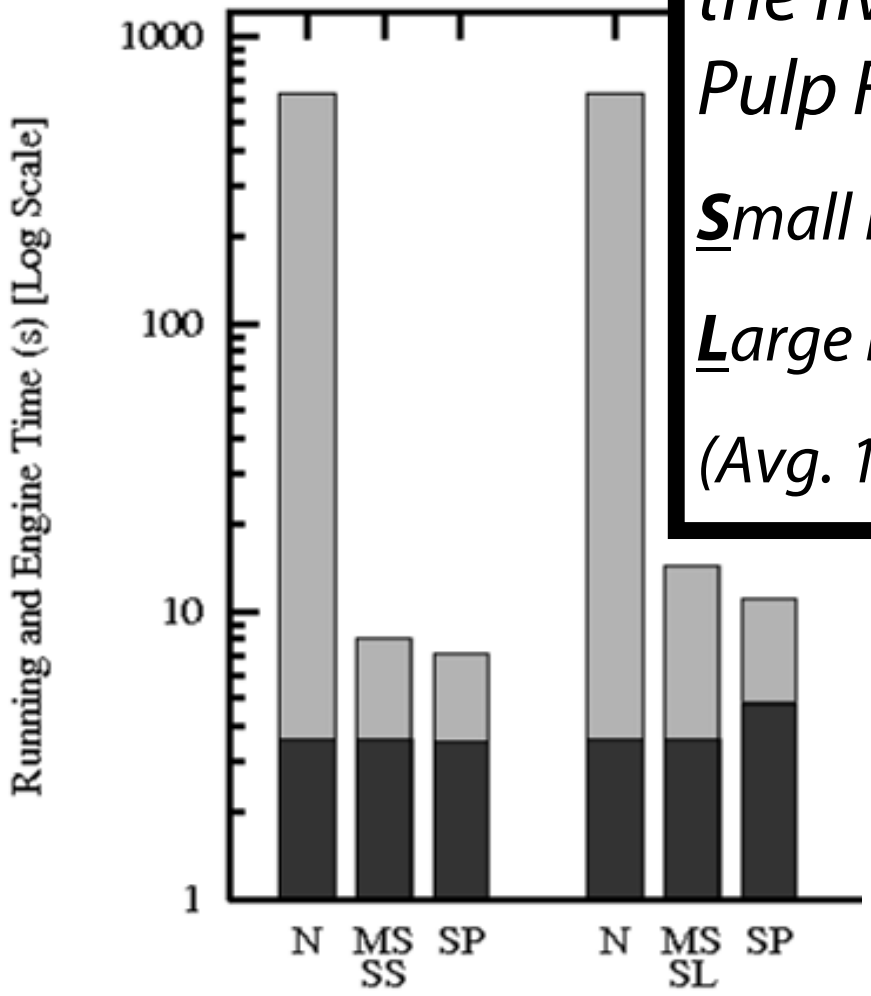
S*mall DNFs per Output*

(Avg. 20.4, Max 63)

Laufzeiten


Running and Engine Time (s) [Log Scale]
1000
100
10
1
N
MS
SS
SP
N
MS
SL
SP
N
MS
LS
SP
N
MS
LL
SP
"Find all directors who have a highly rated drama but low rated comedy" (LL)
Large #Tuples Output (1415)
Large DNFs per Output
(Avg. 234.8, Max. 9088)

Laufzeiten



"Find all actors in Pulp Fiction who appeared in two very bad movies in the five years before appearing in Pulp Fiction" ***(SL)***

S*mall Number of Tuples Output (33)*

L*arge DNFs per Output*

(Avg. 117.7,Max 685)

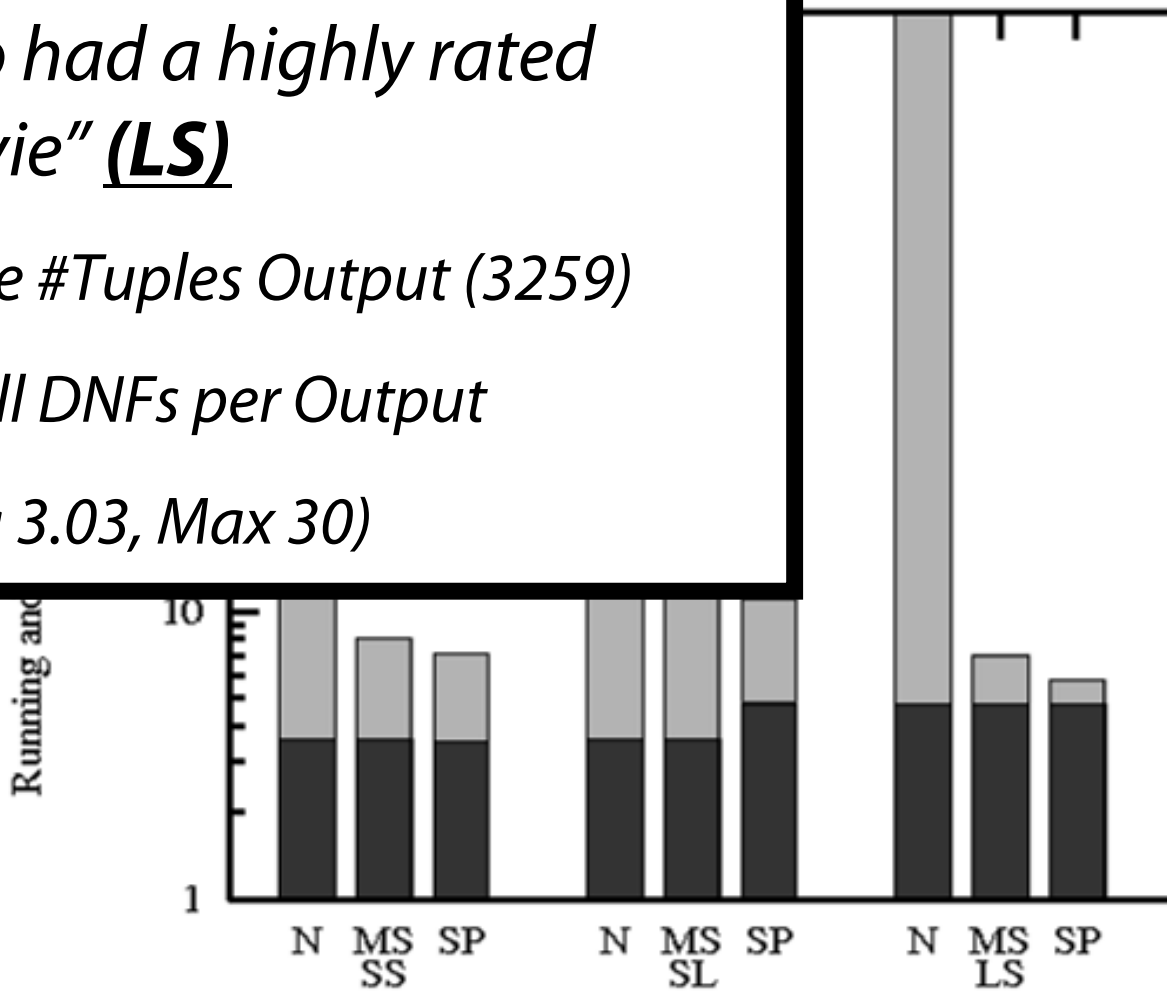
Laufzeiten

"Find all directors in the 80s who had a highly rated movie" ***(LS)***

L*arge #Tuples Output (3259)*

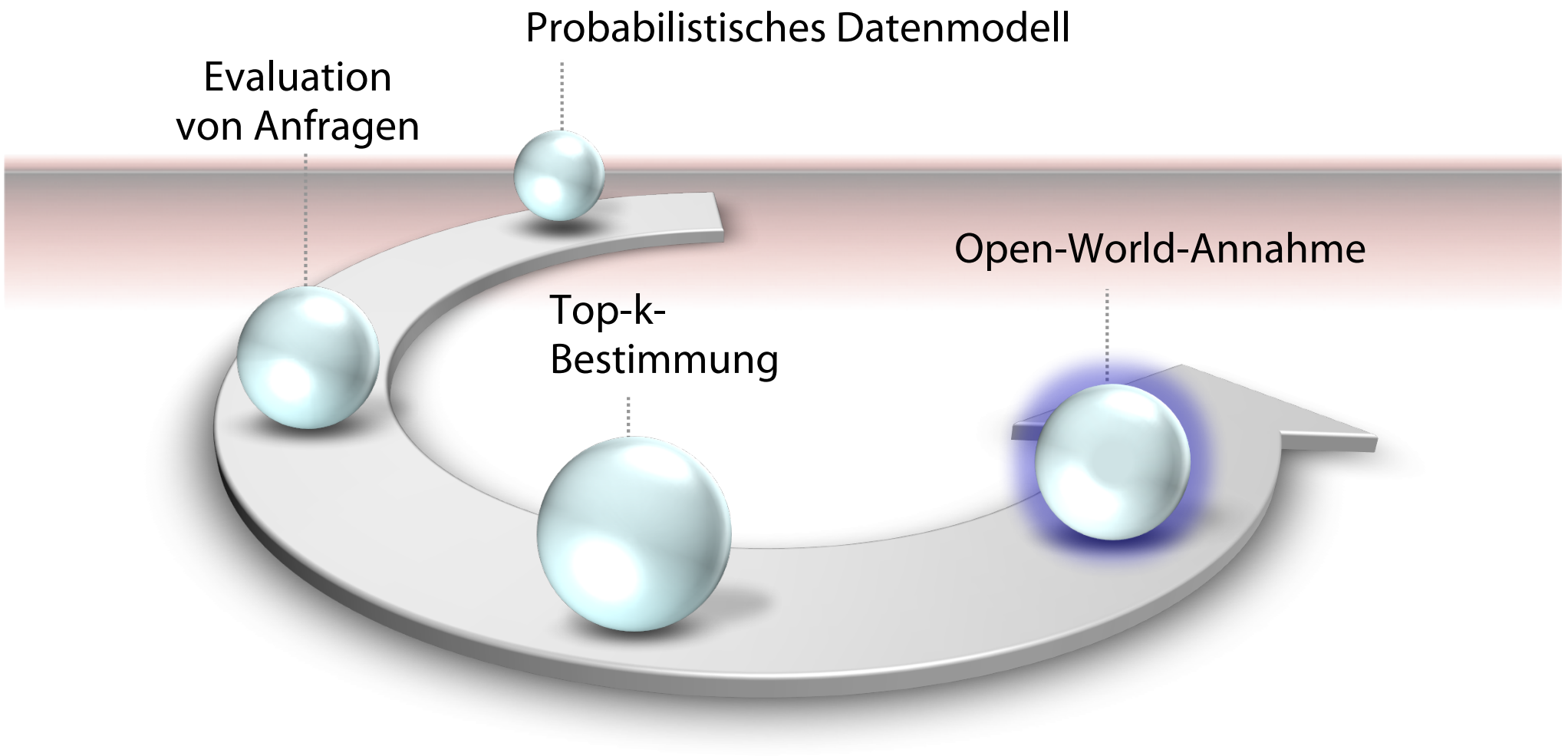
S*mall DNFs per Output*

(Avg 3.03, Max 30)



Non-Standard-Datenbanken

Probabilistische Datenbanken



Danksagung

Die nachfolgenden Präsentationen sind aus einem Vortrag "Open World Probabilistic Databases" von Ismail Ilkan Ceylan, Adnan Darwiche und Guy Van den Broeck

İsmail İlkan Ceylan, Adnan Darwiche, and Guy Van den Broeck. Open-world probabilistic databases. In *Proc. Knowledge Representation and Reasoning* (KR'16), pp. 339-348, **2016**.

CWA vs OWA

Ein Flug taucht nicht in Flüge-Datenbank auf

Der Flug findet nie statt!

In einer Patientenakte ist eine Penicillin-Allergie nicht erwähnt

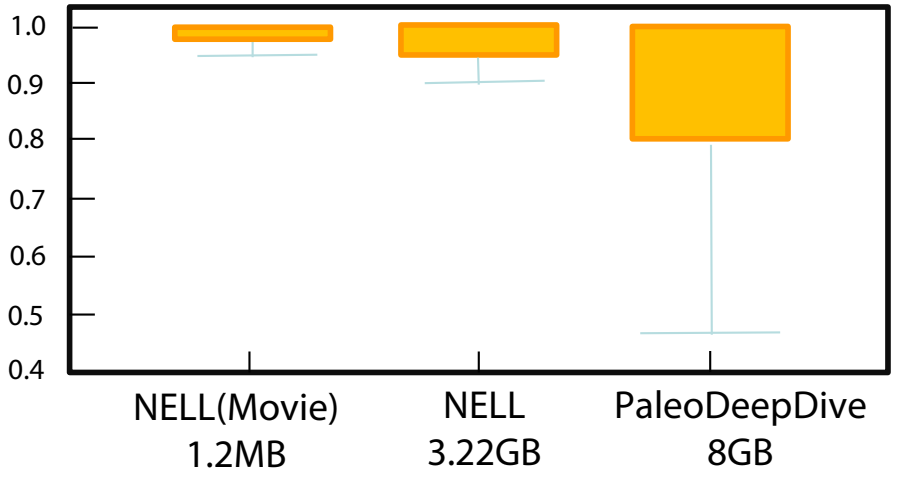
Der Patient leidet nicht an einer Penicillin-Allergie !

Einführung

- Unter der Annahme der geschlossenen Welt (Closed-World Assumption, CWA) wird als falsch angenommen, was nicht als wahr beweisbar ist
- Unter der Annahme der offenen Welt (Open-World Assumption, OWA) wird das, was nicht beweisbar ist, als unbekannt angenommen
- Das klassische PDB-Modell verwendet CWA
  - Fakten nicht in der DB haben Wahrscheinlichkeit 0
- Offene PDBs:
  - Fakten nicht in der DB haben "unbekannte" Wahrscheinlichkeit

Klasssische Anwendung

- Faktenextraktion aus Texten (DeepDive, Nell, Yago)
- Fakten beschrieben mit Sicherheitswerten
  - Deutung: Wie wahrscheinlich ist es, dass Faktum wahr?
- Verteilung der Sicherheitswerte:



- Vervollständigung? Unmöglich!

Probleme bei CWA in PDBs: Beispiele

Couple		P
arquette	cox	0.6
pitt	jolie	0.8
thornton	jolie	0.6
pitt	aniston	0.9
kunis	kutcher	0.7

Inmovie		P
w.Smith	ali	0.9
w.Smith	sharktable	0.8
j.james	ali	0.6
arquette	scream	0.7
pitt	mr ms smith	0.5
jolie	mr ms smith	0.7
jolie	sharktable	0.9

$$Q_1(x, y) = \text{Inmovie}(x, z), \text{Inmovie}(y, z), \text{Couple}(x, y)$$

$$Q_2 = \text{Inmovie}(x, z), \text{Inmovie}(y, z), \text{Couple}(x, y)$$

$Q_1(\text{pitt}, \text{jolie})$

Q_2

Erwartet:

$$P\left(Q_1(\text{pitt}, \text{jolie})\right) < P(Q_2)$$

Gefunden:

$$P\left(Q_1(\text{pitt}, \text{jolie})\right) = P(Q_2) = 0.28$$

Probleme bei CWA in PDBs: Beispiele

Couple		P
arquette	cox	0.6
pitt	jolie	0.8
thornton	jolie	0.6
pitt	aniston	0.9
kunis	kutcher	0.7

Inmovie		P
w.Smith	ali	0.9
w.Smith	sharktable	0.8
j.james	ali	0.6
arquette	scream	0.7
pitt	mr ms smith	0.5
jolie	mr ms smith	0.7
jolie	sharktable	0.9

$$Q_1(x, y) = \text{Inmovie}(x, z), \text{Inmovie}(y, z), \text{Couple}(x, y)$$

$Q1(\text{w. smith}, \text{j. james})$

$Q1(\text{thornton}, \text{aniston})$

Erwartet:

$$P\left(Q_1(\text{w. smith}, \text{j. james})\right) > P(Q1(\text{thornton}, \text{aniston}))$$

Gefunden:

$$P\left(Q_1(\text{w. smith}, \text{j. james})\right) = P(Q1(\text{thornton}, \text{aniston})) = 0$$

Probleme bei CWA in PDBs: Beispiele

Couple		P
arquette	cox	0.6
pitt	jolie	0.8
thornton	jolie	0.6
pitt	aniston	0.9
kunis	kutcher	0.7

Inmovie		P
w.Smith	ali	0.9
w.Smith	sharktable	0.8
j.james	ali	0.6
arquette	scream	0.7
pitt	mr ms smith	0.5
jolie	mr ms smith	0.7
jolie	sharktable	0.9

$$Q_1(x, y) = \text{Inmovie}(x, z), \text{Inmovie}(y, z), \text{Couple}(x, y)$$

$Q_1(\text{w. smith}, \text{j. james})$

$\text{Inmovie}(x, y) \wedge \neg\, \text{Inmovie}(x, y)$

$\text{Inmovie}(x, y) \wedge \neg \text{Inmovie}(x, y)$ ist nicht erfüllbar, wird aber gleich bewertet wie $Q_1(\text{w. smith}, \text{j. james})$

OpenPDBs

Inmovie		P
w.Smith	ali	0.9
w.Smith	sharktable	0.8
j.James	ali	0.6
arquette	scream	0.7
pitt	mr ms smith	0.5
jolie	mr ms smith	0.7
jolie	sharktable	0.9

Couple		P
arquette	cox	0.6
pitt	jolie	0.8
thornton	jolie	0.6
pitt	aniston	0.9
kunis	kutcher	0.7

Domain D

cruise
depp
hayek
pitt
j.James
.
.
Mission Impossible
Troy
Kingdom of Heaven

$\lambda \in [0, 1]$

Offene Tupel für $\lambda = 0.3$:
(Inmovie(pitt, Troy), 0.3)
(Inmovie(hayek, Mission Impossible), 0.15)

OpenPDBs

- Eine offene PDB P_λ ist ein Paar $(\mathcal{P}, \lambda)$,
  wobei $\mathcal{P}$ eine probabilistische DB ist und $\lambda \in [0, 1]$
- Für jedes Tupel nicht in $\mathcal{P}$ fügen wir ein Tupel $\langle t : p \rangle$
  hinzu für irgendein $p \in [0, \lambda]$

- P_λ induziert eine Menge von Wahrscheinlichkeitsverteilungen $K_{P\lambda}$
  - Intervall-basierte Wahrscheinlichkeitsangaben für
    offene Tupel

OpenPDBs

- Das Wahrscheinlichkeitsintervall

  einer Booleschen Anfrage Q an P_λ ist

$$K_{P\lambda}(Q) = [\underline{P}_{P\lambda}(Q), \overline{P}_{P\lambda}(Q)]$$

  wobei:

$$\underline{P}_{P\lambda}(Q) = \min_{P \in KP\lambda} P(Q) \quad , \quad \overline{P}_{P\lambda}(Q) = \max_{P \in KP\lambda} P(Q)$$

Beispiele - CWA vs OWA

$$Q_3 = \mathrm{Ac}(\mathrm{patt}), \mathrm{Workedfor}(\mathrm{patt}, \mathrm{hwicke}), \mathrm{Di}(\mathrm{hwicke})$$

$$Q_4 = \mathrm{Ac}(\mathrm{patt}), \mathrm{Workedfor}(\mathrm{patt}, \mathrm{x}), \mathrm{Di}(\mathrm{x})$$

Erwartet:

$$\bar{\mathrm{P}}\,(Q_4) > \bar{\mathrm{P}}(Q_3)$$

NELL Datenbank:

Gefunden:

$$\bar{\mathrm{P}}\,(Q_4) = \bar{\mathrm{P}}(Q_3) = 0$$

In einer offenen Welt:

Gefunden: für $\lambda = 0.3$

$$\bar{\mathrm{P}}\,(Q_4) = 0.82\,, \quad \bar{\mathrm{P}}(Q_3) = 0.51$$

Ac stands for actor, patt for Pattinson, Di for director, hwicke for Hardwicke, Mov for movie, and trainsp for Trainspotting

Beispiele - CWA vs OWA

$$Q_5 = Ac(x), Inmovie(x, trainsp), Mov(trainsp), \neg Di(x)$$

Erwartet:

$$\bar{\mathrm{P}}\,(Q_5) > 0$$

NELL Datenbank:

Found:

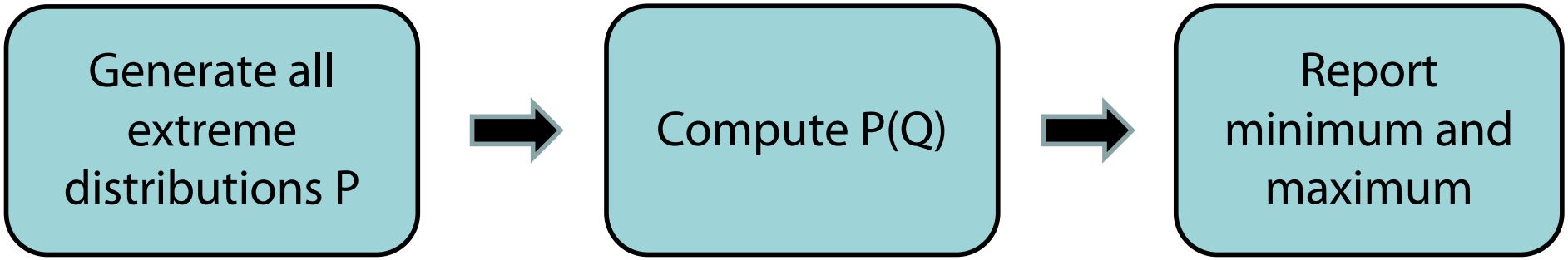
$$\bar{\mathrm{P}}\,(Q_5) = 0$$

In einer offenen Welt:

Found: for $\lambda = 0.3$

$$\bar{\mathrm{P}}\,(Q_5) = 0.78$$

Naiver Algorithmus



Exponentiell in der Anzahl der Open-World-Tupel

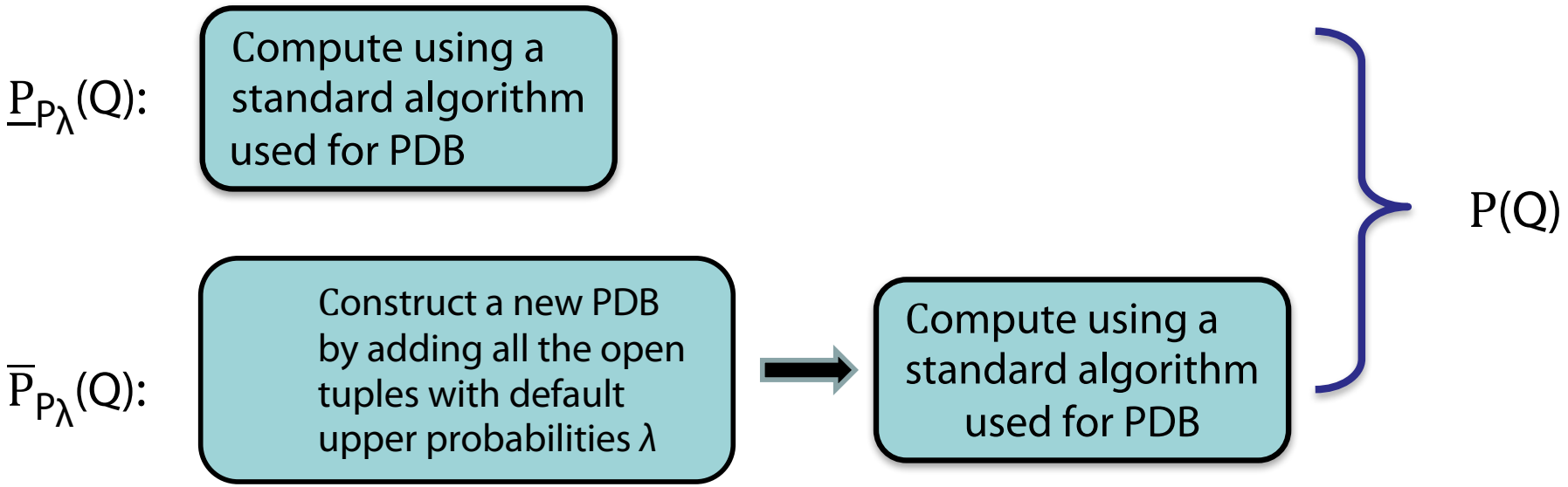
Naiver Algorithmus für UCQs

Beispiel

$$\text{Exist m, a } [\underbrace{\text{Inmovie(a, m)}}\ \text{AND}\ \underbrace{\text{Ac(a)}}]$$

- Obere Grenze: Jedes Tupel in Inmovie und Ac hat eine maximale Wahrscheinlichkeit
- Untere Grenze: Jedes Tupel in Inmovie und Ac hat eine minimale Wahrscheinlichkeit
- Monotonie für UCQs vereinfacht Auswertung

Naiver Algorithmus for UCQs



Bei der Auswertung von $\overline{P}_{P\lambda}(Q)$ wächst die PDB polynomiell in der Domänengröße

$Lift_O^R$ Algorithmus

CNF:
$(R(x) \vee S(y,z)) \wedge (S(x,y) \vee (\mathrm{T}(\mathrm{x}))$

$Lift_O^R$ (Q, P, λ, D) - abbreviated by L(Q, P)

Input: CNF Q , probability tuples Ƥ, threshold λ and domain D

Output: The upper probability $\overline{\mathrm{P}}_{(p,\,\lambda)}(Q)$ over domain D

Step 0: Base of Recursion

if Q is a single ground atom t then

if ⟨t : p⟩ ∈ Ƥ then return p else return λ

Step 1: Rewriting of Query

Convert Q to union of CNFs: $Q_{UCNF} = Q_1 \vee \ldots \vee Q_m$

Example: $(R(x) \vee S(y,z)) \wedge (S(x,y) \vee (\mathrm{T}(\mathrm{x}))$

➔ $((R(x)) \wedge (S(x,y) \vee (\mathrm{T}(\mathrm{x}))) \vee ((S(y,z)) \wedge (S(x,y) \vee \big(\mathrm{T}(\mathrm{x})\big))$

$Lift_O^R$ Algorithmus

$Q_1 \perp Q_2$ if Q_1, Q_2 doesn't share any relational symbols

Step 2: Decomposable Disjunction
if $m > 1$ and $Q_{UCNF} = Q_1 \vee Q_2$ where $Q_1 \perp Q_2$ then
- $q_1 \leftarrow L(Q_1, P|_{Q1})$ and $q_2 \leftarrow L(Q_2, P|_{Q2})$
- return $1 - (1 - q1) \cdot (1 - q2)$

Step 3: Inclusion-Exclusion
if $m > 1$ but Q_{UCNF} has no independent Q_i then
- return $\sum_{s \subseteq \{1,\dots,m\}} (-1)^{|s|+1} \cdot L(\bigwedge_{i \in s} Q_i, P|_{\wedge i \in s\, Q_i})$

Step 4: Decomposable Conjunction
if $Q = Q_1 \wedge Q_2$ where $Q_1 \perp Q_2$ then
- return $L(Q_1, P|_{Q1}) \cdot L(Q2, P|_{Q2})$

Specifically, $\mathcal{P}|_Q$ denotes the subset of $\mathcal{P}$ that talks about the predicates that appear in Q.

$Lift_O^R$ Algorithmus

Step 5: Decomposable Universal Quantifier

if Q has a separator variable x then

- let T be all constants as x-argument in P
- $q_c \leftarrow \Pi_{t \in T} L(Q[x/t], P|_{x=t})$
- $q_o \leftarrow L(Q[x/t], \varnothing)$ for some $t \in D \setminus T$
- return $q_c \cdot q_o^{|D \setminus T|}$

A separator is a variable that appears in every atom in Q

An x-argument is an argument that hold a separator variable x

Forall a,m [Inmovie(a,m) OR Ac(a)]

Inmovie	
Pitt	Troy
Butler	300
deCaprio	Inception

Ac
Pitt
Cruise
Hayek

Step 6: Fail

$T = \{$Pitt, Butler, deCaprio, Cruise, Heyek$\}$

$Lift_O^R$ Algorithmus für UCQs

$Lift_O^R$ berechnet die Wahrscheinlichkeiten für UCQs

Daten-Komplexität:

Monotone UCQs werden auf OpenPDBs in PTime evaluiert, wenn sie auf PDBs in PTime evaluiert werden und umgekehrt.

Zusammenfassung

- Einfache PDBs erfüllen nicht (mehr) die heutigen Erwartungen (Vervollständigungen nicht möglich)
- OpenPDBs setzen die Annahme der offenen Welt um
- Vorgestellt wurde ein effizienter Algorithmus zur Auswertung von UCQs in OpenPDBs
- Nächste Schritte:
  - Erhöhung der Ausdrucksstärke (> UCQs)
  - Unendliche Domäne

Ismail Ilkan Ceylan, Adnan Darwiche and Guy Van den Broeck. Open-World Probabilistic Databases: An Abridged Report, In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), Sister Conference Best Paper Track, **2017**.