Scalable RDF Data Management

… with a Touch of Uncertainty

Martin Theobald

University of Luxembourg

Faculty of Sciences, Technology & Communication

Joint work with:

- Hernán Blanco, Maximilian Dylla, Sairam Gurajada, Maarten Van den Heuvel, Iris Miliaraki, Dat Ba Nguyen

Information Extraction

Jeffrey Ullman

From Wikipedia, the free encyclopedia

Jeffrey David Ullman (born November 22, 1942) is a renowned computer scientist. His textbooks on compilers (various editions are popularly known as the Dragon Book), theory of computation (also known as the Cinderella book), data structures, and databases are regarded as standards in their fields.

Jeffrey D. Ullman	
Born	November 22, 1942 (age 69)
Citizenship	American
Nationality	American
Institutions	Stanford University
Alma mater	Columbia University, Princeton University
Doctoral advisor	Arthur Bernstein, Archie McKellar
Doctoral students	Surajit Chaudhuri, Kevin Karplus, David Maier, Harry Mairson, Alberto O. Mendelzon, Jeffrey F. Naughton, Anand Rajaraman, Yehoshua Sagiv, Mihalis Yannakakis
Known for	database theory, database systems, formal language theory
Notable awards	Fellow of the Association for Computing Machinery (1994), ACM SIGMOD Contributions Award (1996), ACM SIGMOD Best Paper Award (1996), Karl V. Karlstrom outstanding educator award (1998), Knuth Prize (2000), ACM SIGMOD Edgar F. Codd Innovations Award (2006), ACM SIGMOD Test of Time Award (2006), IEEE John von Neumann Medal (2010)

Contents [hide]

Early life & Career [edit]

Ullman received a Bachelor of Science degree in Engineering Mathematics from Columbia University in 1963 and his Ph.D. in Electrical Engineering from Princeton University in 1966. He then worked for several years at Bell Labs. From 1969 to 1979 he was a professor at Princeton. Since 1979 he has been a professor at Stanford University, where he is currently the Stanford W. Ascherman Professor of Computer Science (Emeritus). In 1995 he was inducted as a Fellow of the Association for Computing Machinery and in 2000 he was awarded the Knuth Prize. Ullman is also the co-recipient (with John Hopcroft) of the 2010 IEEE John von Neumann Medal, "For laying the foundations for the fields of automata and language theory and many seminal contributions to theoretical computer science."[1]

Ullman's research interests include database theory, data integration, data mining, and education using the information infrastructure. He is one of the founders of the field of database theory, and was the doctoral advisor of an entire generation of students who later became leading database theorists in their own right. He was the Ph.D. advisor of Sergey Brin, one of the co-founders of Google, and served on Google's technical advisory board. He is currently the CEO of Gradiance.

Books [edit]

- *Database Systems: The Complete Book* (with H. Garcia-Molina and J. Widom), Prentice-Hall, Englewood Cliffs, NJ,

DBpedia/YAGO et al.

bornOn(Jeff, 09/22/42)
gradFrom(Jeff, Columbia)
gradFrom(Jeff, Princeton)
hasAdvisor(Jeff, Arthur)
hasAdvisor(Surajit, Jeff)
knownFor(Jeff, Theory)

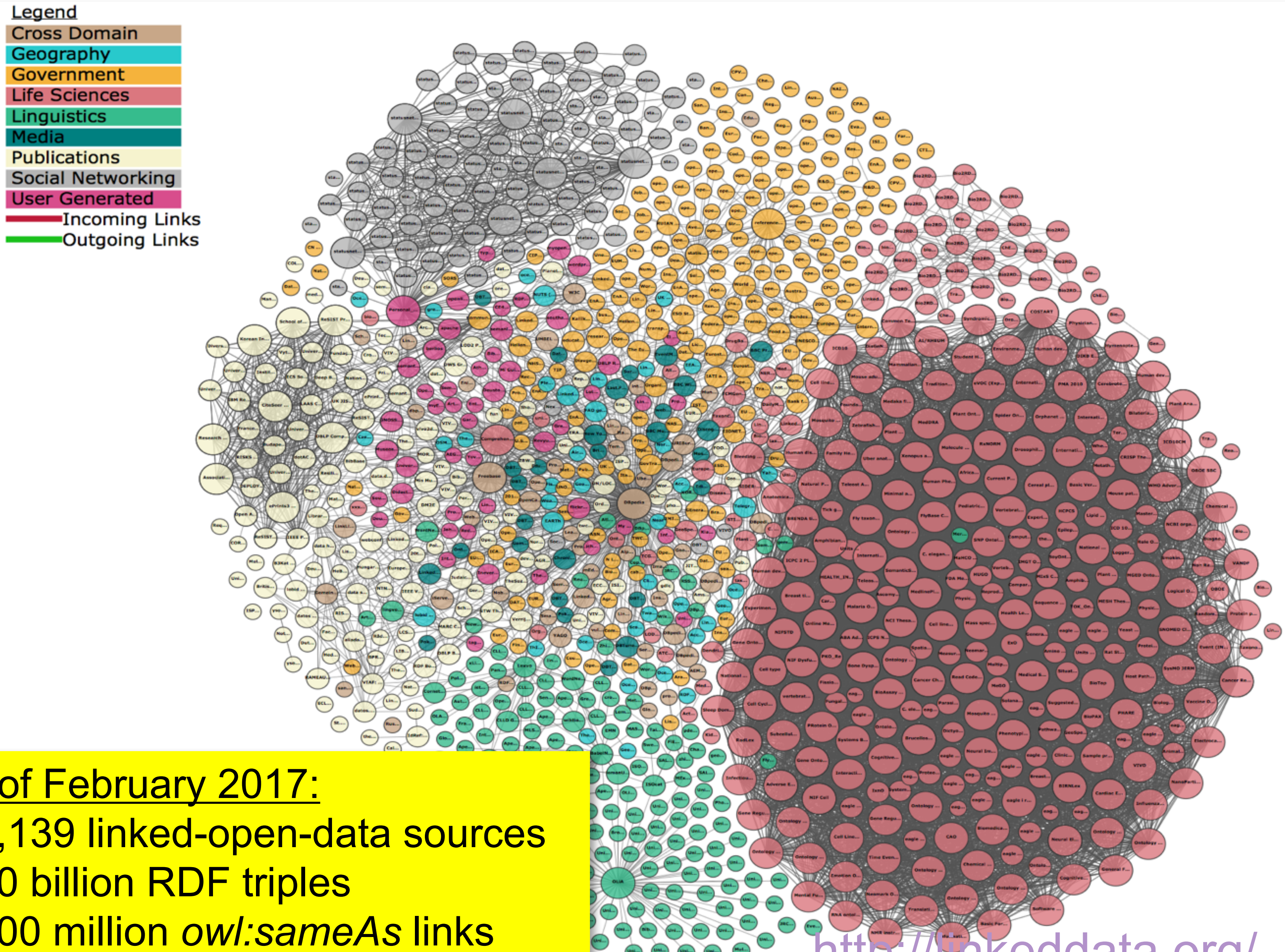
>120 M facts for YAGO3
(from Wikipedia infoboxes)

New fact candidates

author(Jeff, Drag_Book) [0.6]
author(Jeff, Cind_Book) [0.8]
worksAt(Jeff, Bell_Labs) [0.5]
hasAdvisor(Sergej, Jeff) [0.7]
type(Jeff, ACM_Fellow) [0.5]
type(Jeff, CEO) [0.3]

>100's M additional facts
(from Wikipedia free-text)

Linked-Open-Data Cloud



<u>As of February 2017:</u>

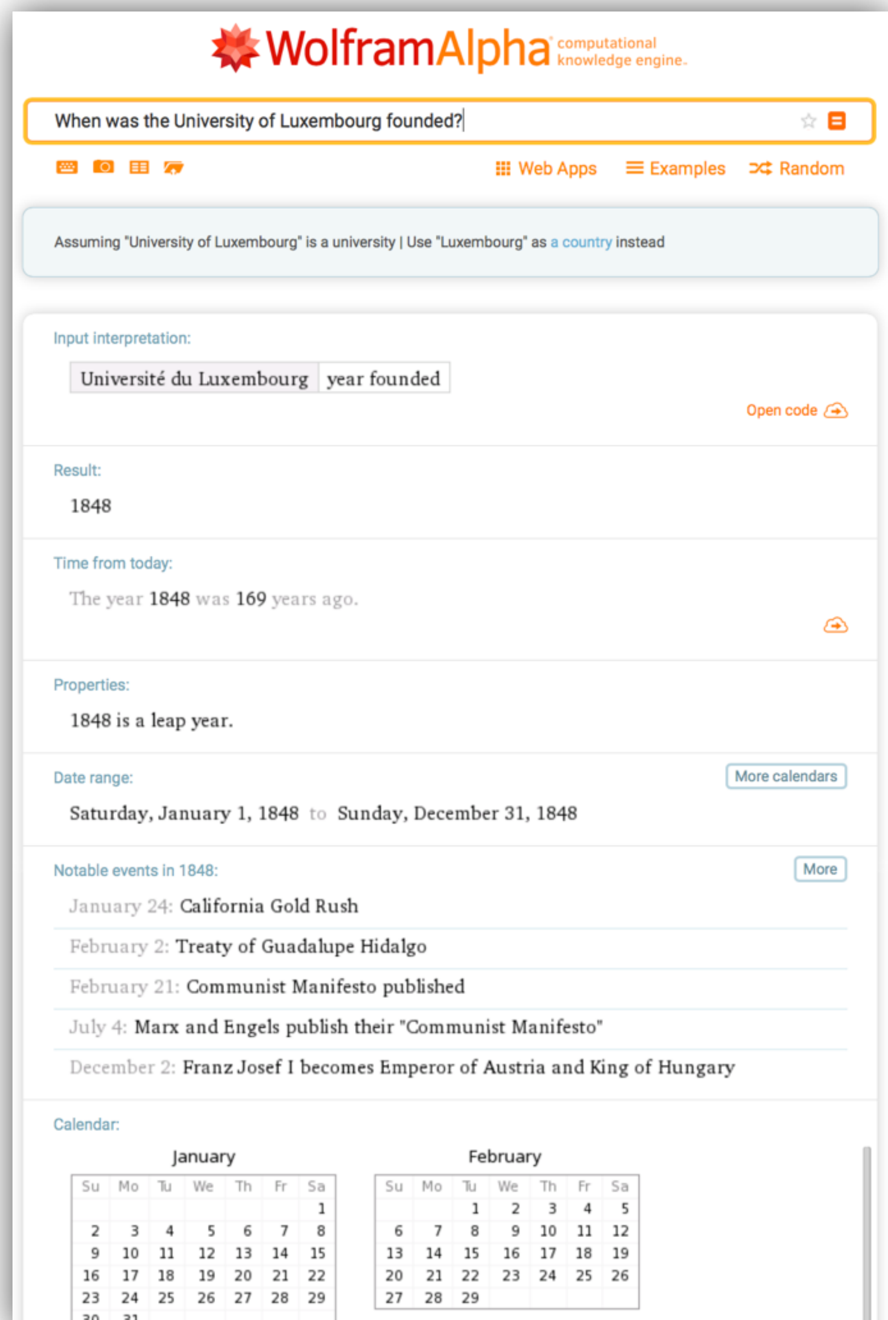
>1,139 linked-open-data sources

>50 billion RDF triples

>500 million *owl:sameAs* links

http://linkeddata.org/

Wolfram Alpha



The ”**Computational Knowledge Engine**”

- Fully implemented in Wolfram-Mathematica
- 10 trillion+ facts
- 50,000+ algorithms and statistical analyses
- 5,000+ templates for visualization and layouts
- 1,000+ domain-specific linguistic analyses

http://www.wolframalpha.com/

IBM Watson: Deep Question Answering

- William Wilkinson's "An Account of the Principalities of Wallachia and Moldavia" inspired this author's most famous novel
- This town is known as "Sin City" & its downtown is "Glitter Gulch”
- As of 2010, this is the only former Yugoslav republic in the EU
- 99 cents got me a 4-pack of Ytterlig coasters from this Swedish chain
- U.S. City: largest airport is named for a World War II Hero; its second largest for a World War II Battle

Question classification & decomposition → **Knowledge back-ends**

WIKIPEDIA
The Free Encyclopedia

DBpedia

freebase™

yago
select knowledge

D. Ferrucci et al.: Building Watson: An Overview of the DeepQA Project. AI Magazine, Fall 2010.

https://www.ibm.com/watson/

RDF-Centered Research Topics

- Information Extraction

  [SIGMOD'09, WebDB'10, PODS'10, WSDM'11, CIKM'12, CLEF/INEX'11/'12, LDOW'14, TACL'16]

- Uncertain RDF Data & Probabilistic Databases

  [ICDE'08, VLDB-J'08, SSDBM'10, BTW'11, CIKM'11, ICDE'13, PVLDB'14, VLDB PhD Workshop'15]

- Scalable RDF Indexing & SPARQL Query Processing

  [SIGMOD'14, SWIM'14, SIGMOD'16]

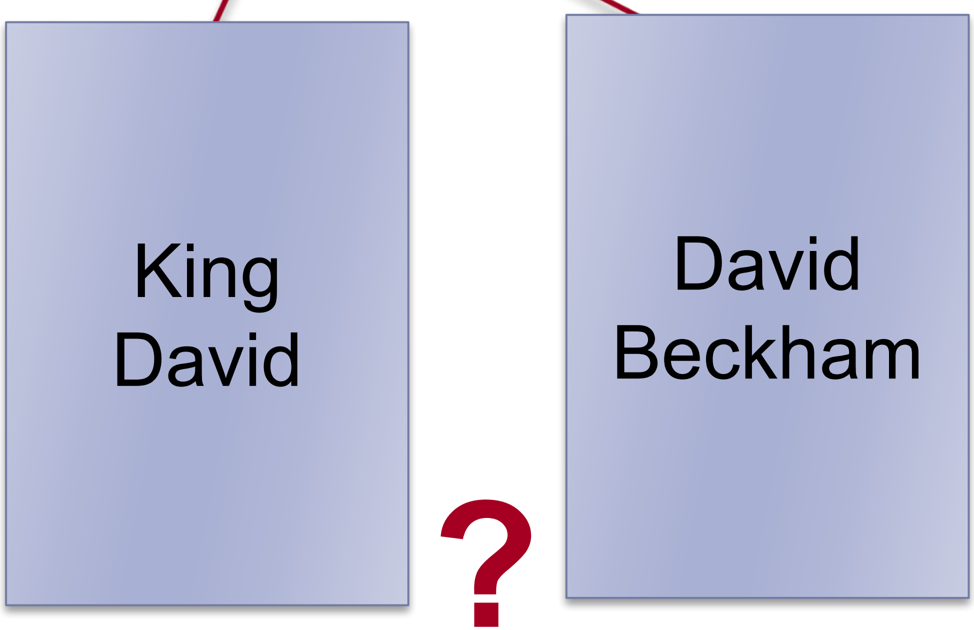
Named-Entity Recognition & Disambiguation

”David played for manu, real, and la galaxy.
His wife posh performed with the spice girls.“

- State-of-art approaches **recognize** named entities and then **disambiguate** these entities in two strictly separated phases.

Named-Entity Recognition & Disambiguation

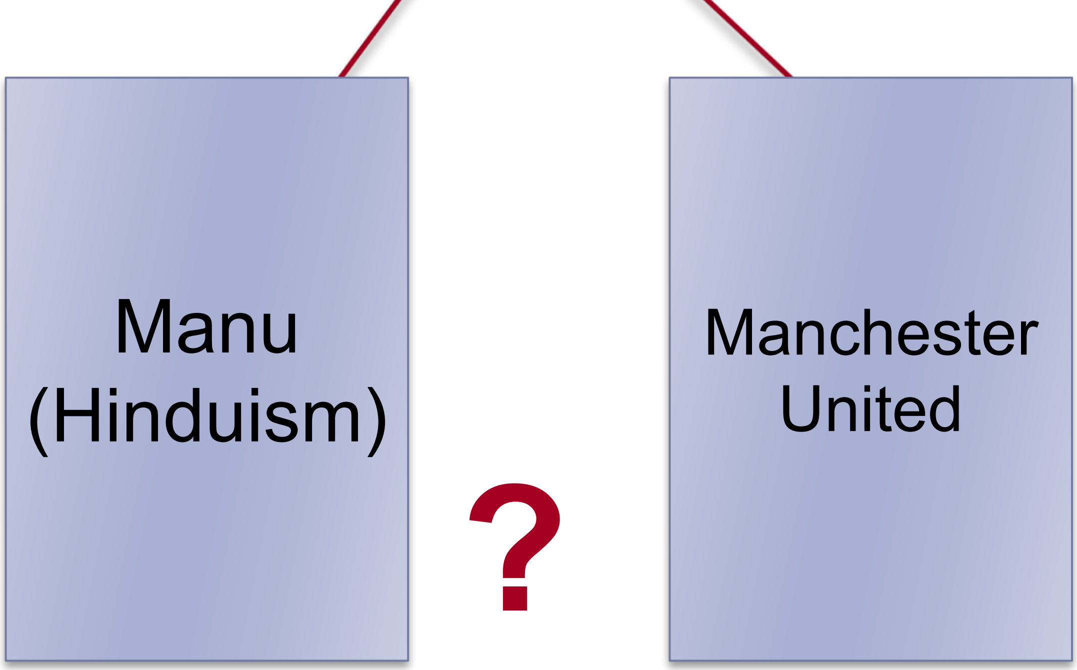
"David played for manu, real, and la galaxy.
His wife posh performed with the spice girls."



- State-of-art approaches **recognize** named entities and then **disambiguate** these entities in two strictly separated phases.

Named-Entity Recognition & Disambiguation

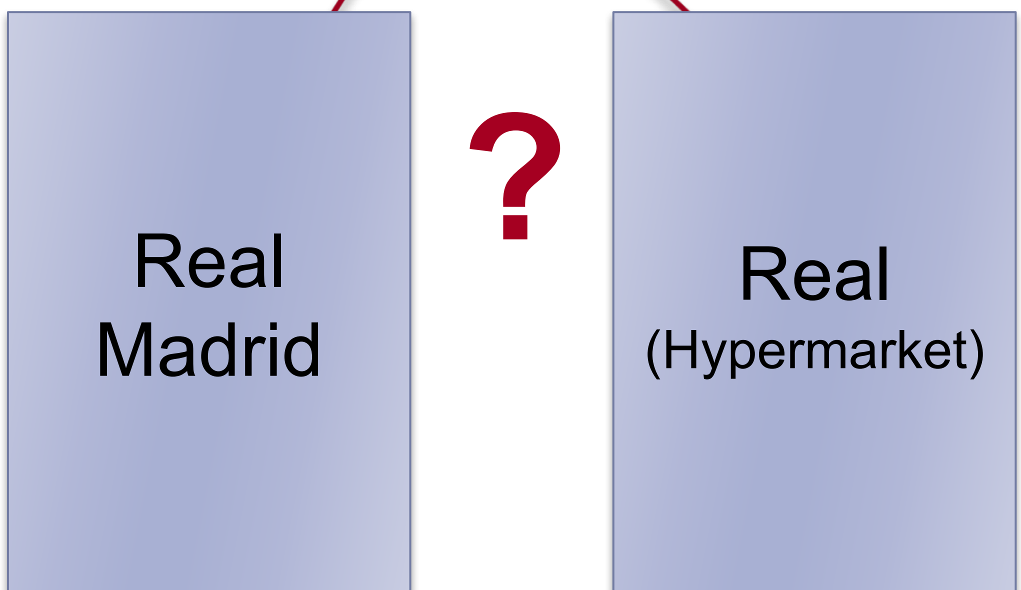
"David played for manu, real, and la galaxy. His wife posh performed with the spice girls."



- State-of-art approaches **recognize** named entities and then **disambiguate** these entities in two strictly separated phases.

Named-Entity Recognition & Disambiguation

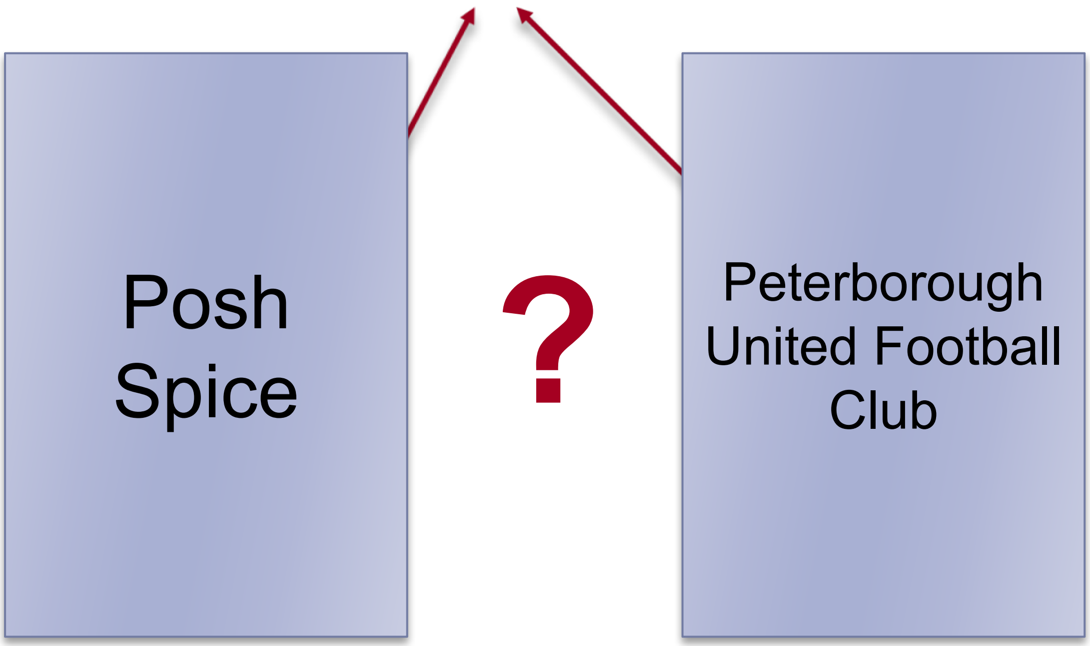
"David played for manu, real, and la galaxy.
His wife posh performed with the spice girls."



- State-of-art approaches **recognize** named entities and then **disambiguate** these entities in two strictly separated phases.

Named-Entity Recognition & Disambiguation

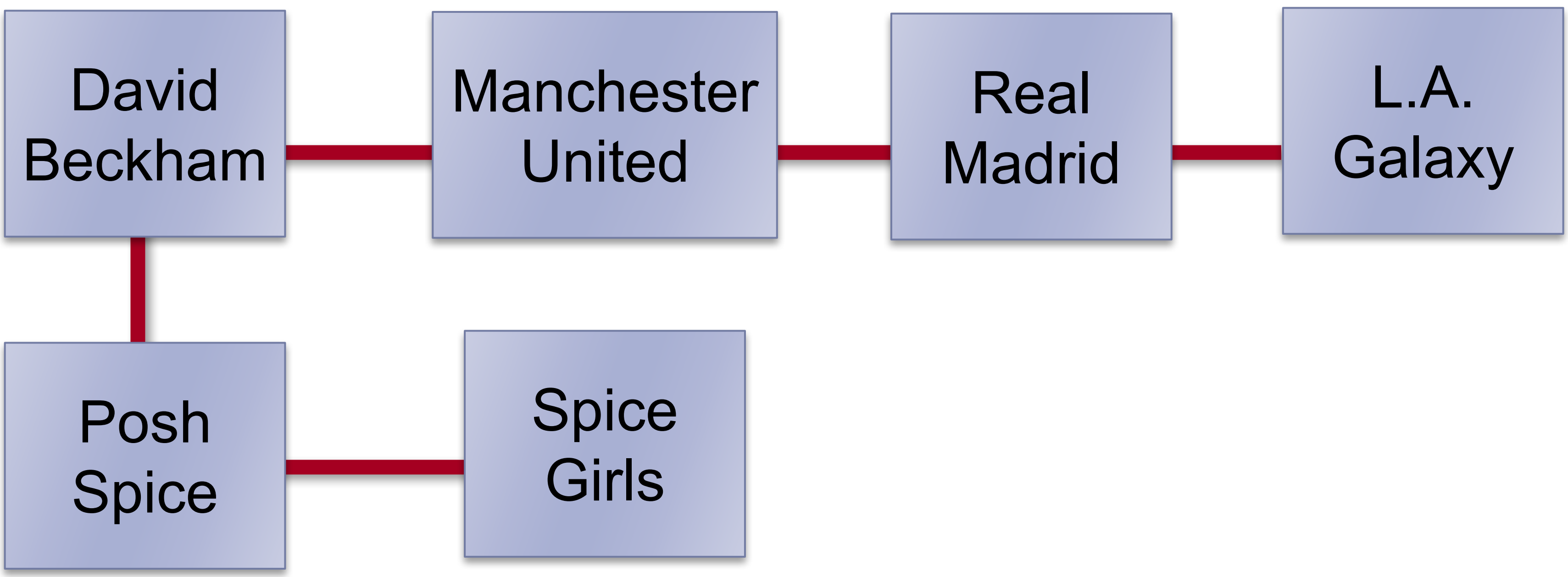
"David played for manu, real, and la galaxy.
His wife posh performed with the spice girls."



- State-of-art approaches **recognize** named entities and then **disambiguate** these entities in two strictly separated phases.

Joint Named-Entity Recognition & Disambiguation

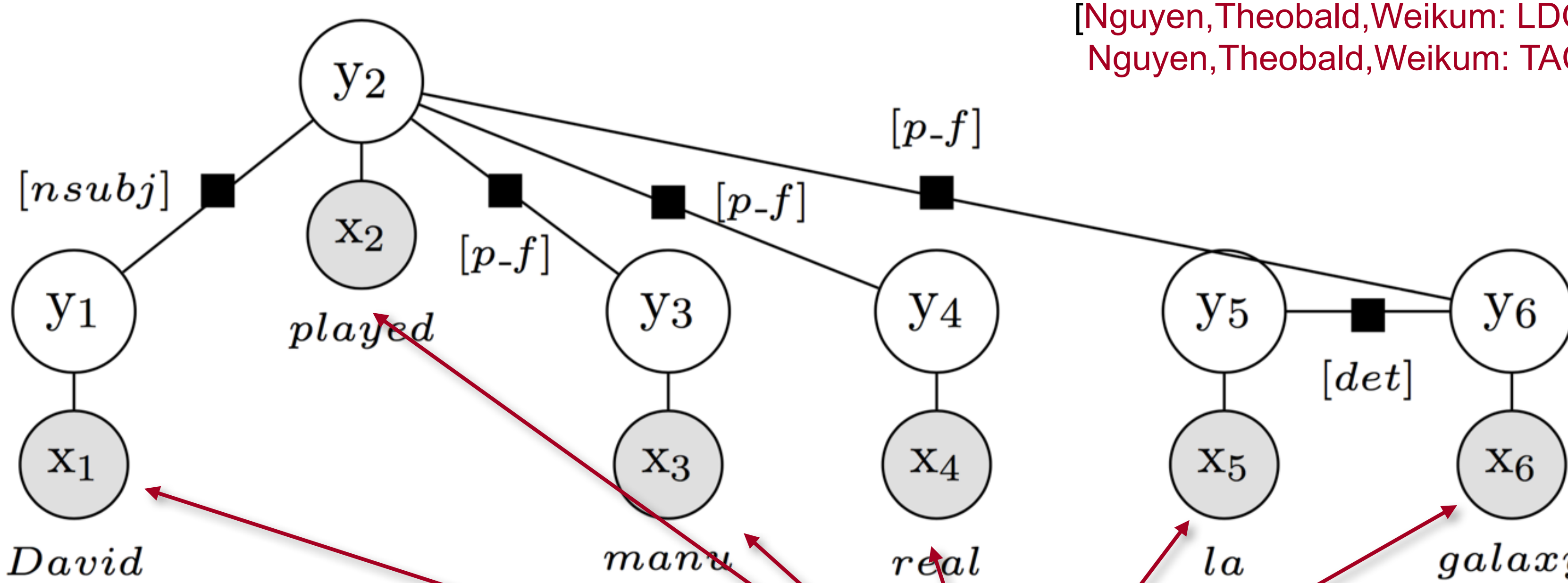
"David played for manu, real, and la galaxy.
His wife posh performed with the spice girls."



- *J*-NERD *jointly* **recognizes** and **disambiguates** named entities with respect to a background knowledge base such as YAGO.

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]

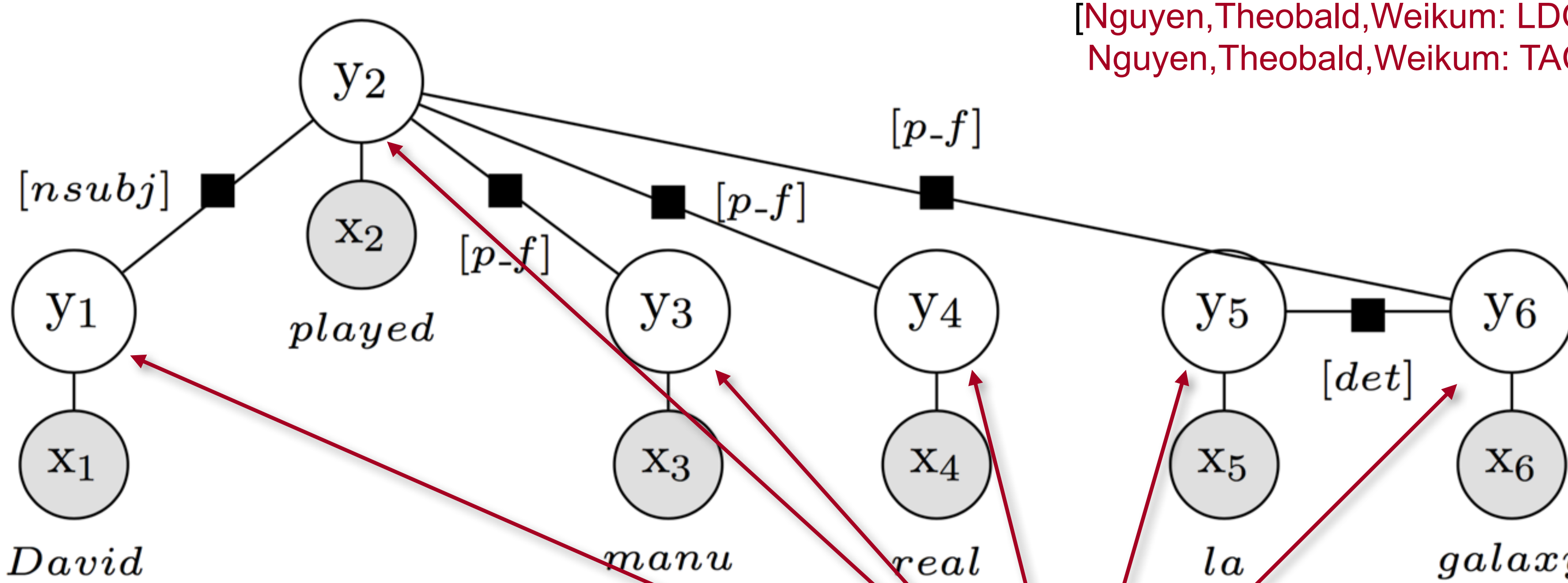


- **Probability distribution** over possible tokens *x* and *combined* NER/D labels *y*

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{A} \mathcal{F}_A(\mathbf{x}_A, \mathbf{y}_A)$$

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]

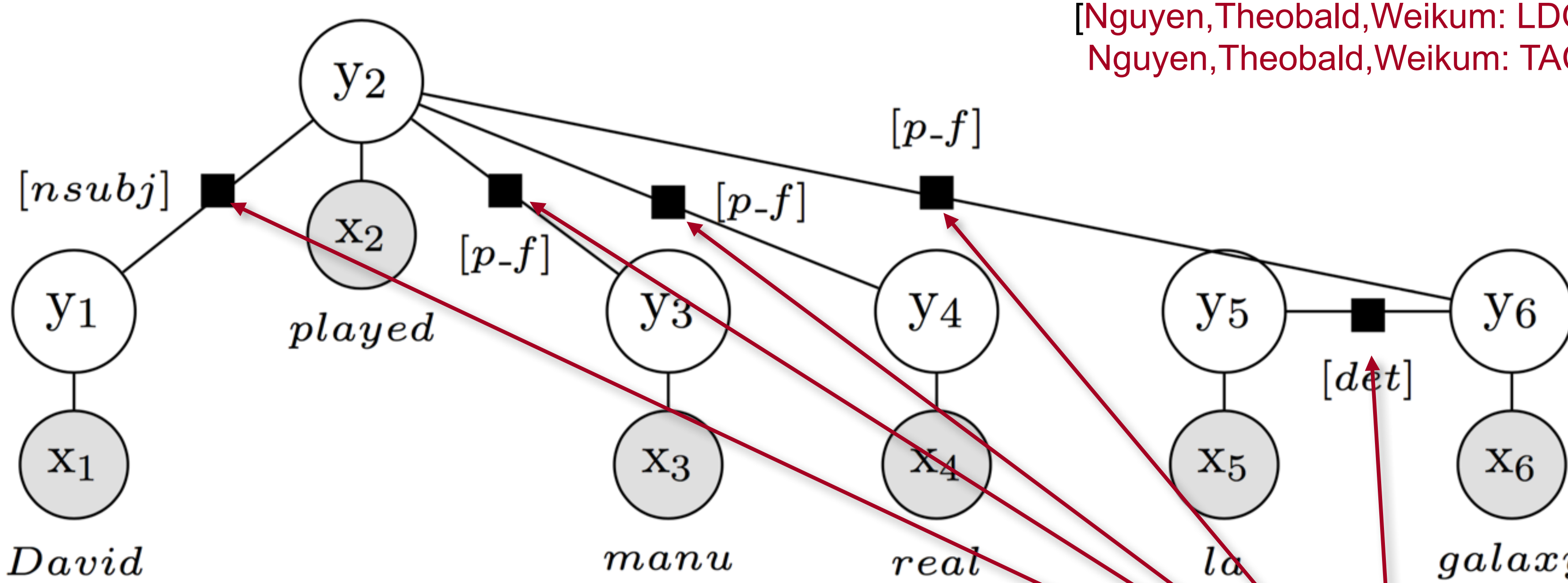


- **Probability distribution** over possible tokens ***x*** and *combined* NER/D labels ***y***

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_A \mathcal{F}_A(\mathbf{x}_A, \mathbf{y}_A)$$

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]

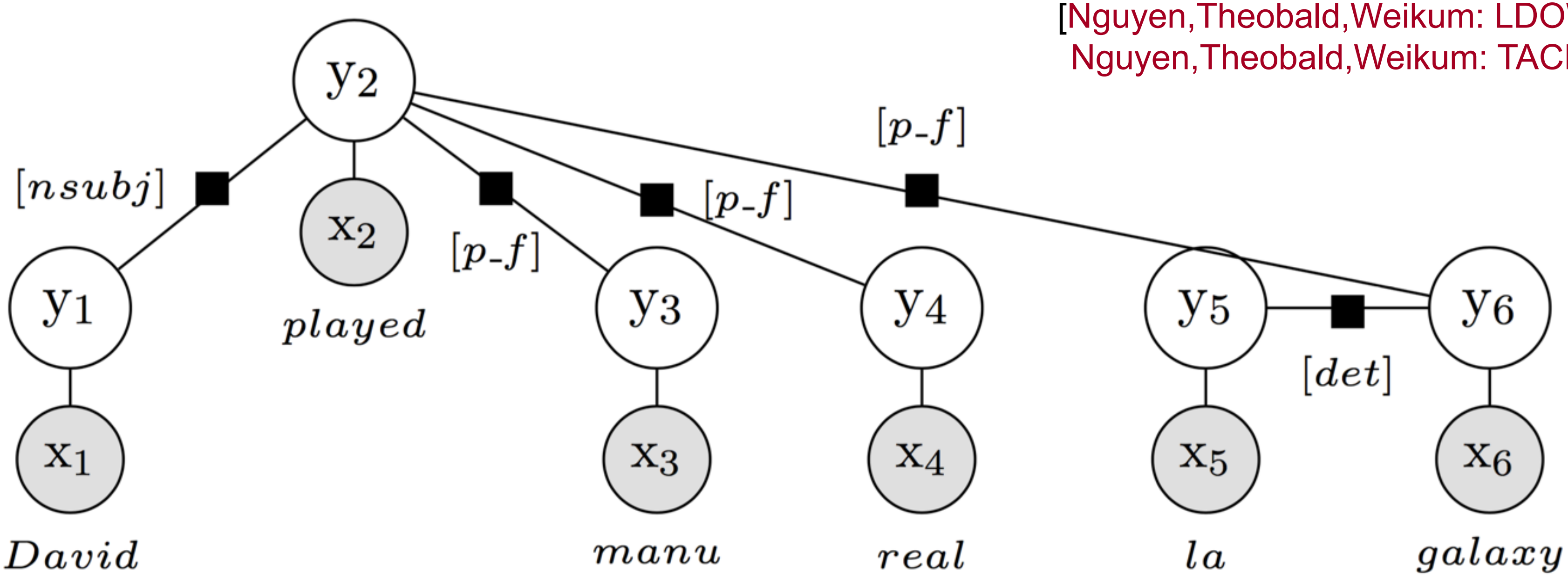


- **Probability distribution** over possible tokens x and *combined* NER/D labels y

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{A} \mathcal{F}_A(\mathbf{x}_A, \mathbf{y}_A)$$

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]



- **Probability distribution** over possible tokens $\boldsymbol{x}$ and *combined* NER/D labels $\boldsymbol{y}$

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{A} \mathcal{F}_A(\mathbf{x}_A, \mathbf{y}_A)$$

- **Probabilistic inference**: find the *most likely* labels $\boldsymbol{y}$, given the observed tokens $\boldsymbol{x}$

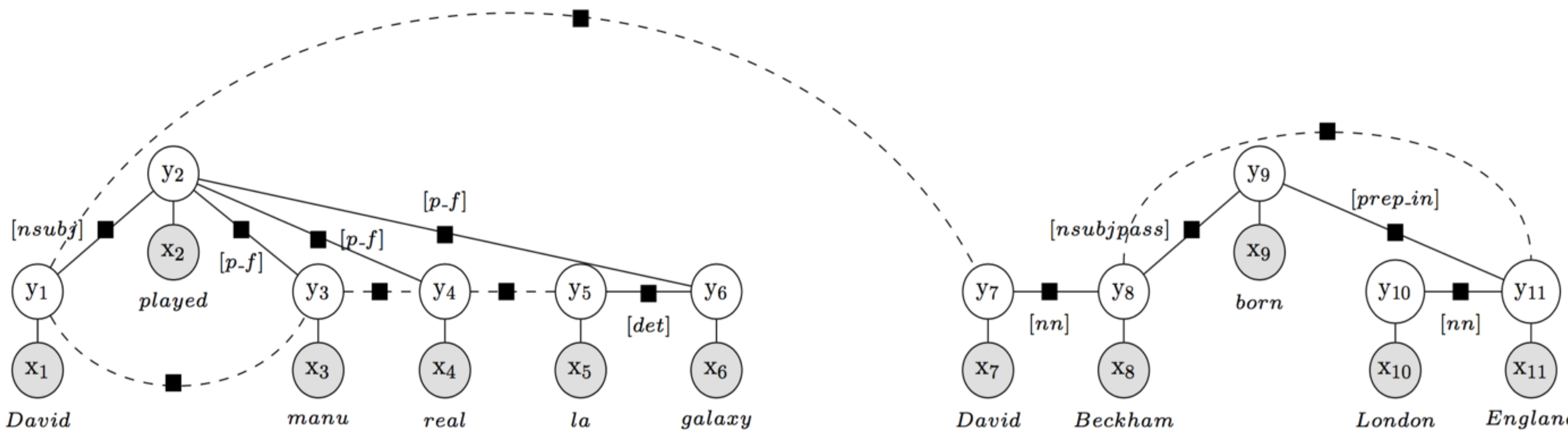
$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x})$$

- **Viterbi algorithm** (dynamic programming) for fast and exact inference

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]

CRF with **cross-sentence** dependencies:



- **Probability distribution** over possible tokens x and *combined* NER/D labels y

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{A} \mathcal{F}_A(\mathbf{x}_A, \mathbf{y}_A)$$

- **Probabilistic inference**: find the *most likely* labels y, given the observed tokens x

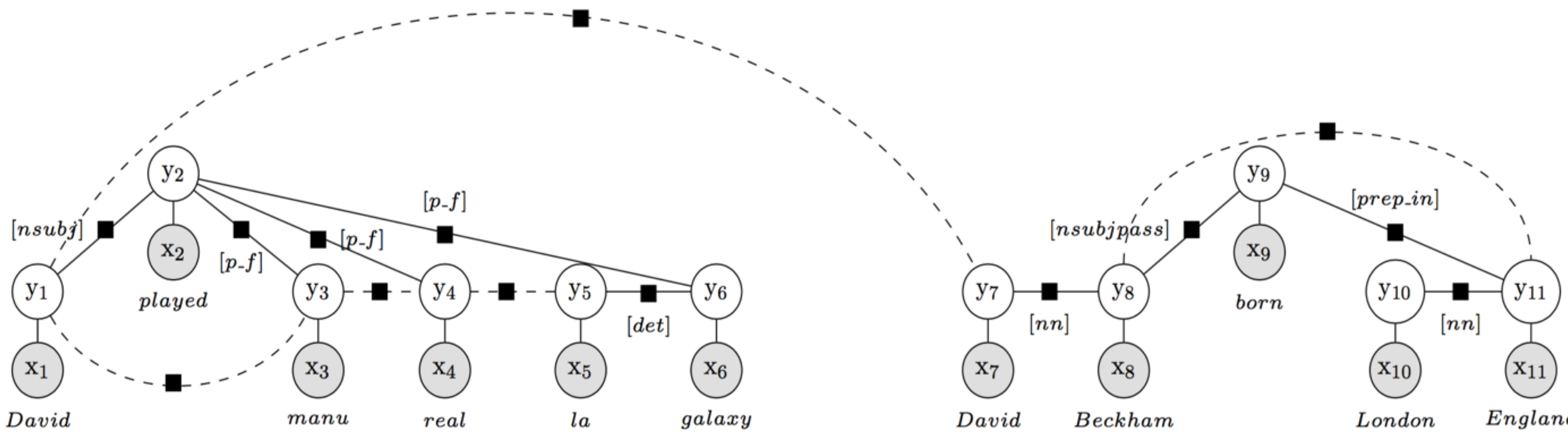
$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x})$$

- General factor graphs: **MCMC-style sampling** for approximate inference

Conditional Random Field in *J*-NERD

[Nguyen,Theobald,Weikum: LDOW'14,
Nguyen,Theobald,Weikum: TACL'16]

CRF with **cross-sentence** dependencies:

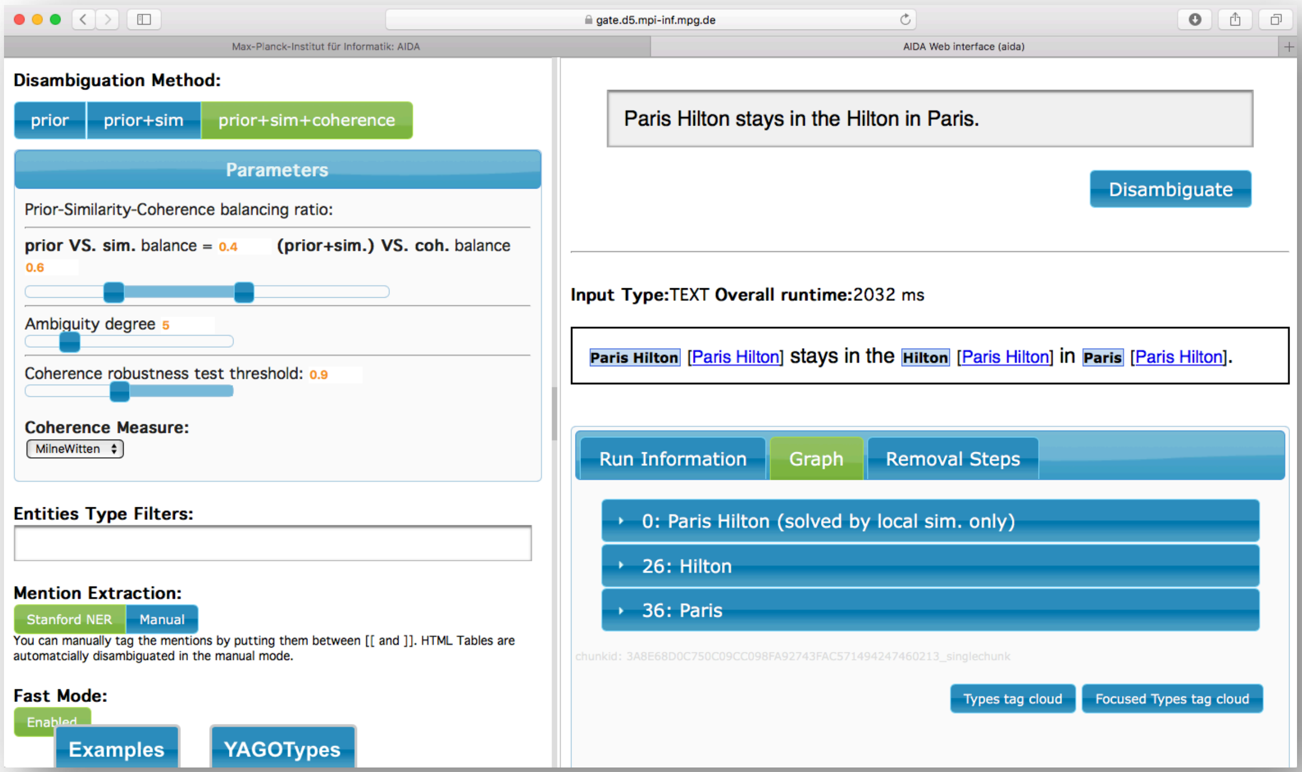


- Evaluation on the **CoNLL newswire collection** with YAGO2 ground-truth annotations (**1,244 labeled articles**)

Method	Prec	Rec	F_1
P-NERD	80.1	75.1	77.5
J-NERD	**81.9**	75.8	**78.7**
AIDA-light	78.7	**76.1**	77.3
TagMe	64.6	43.2	51.8
SpotLight	71.1	47.9	57.3

Ultimate PhD Challenge (I)

”Paris Hilton stays in the Hilton in Paris.“



- All of the current NED tools (incl. AIDA, *J*-NERD, Spotlight, TagMe) get this sentence wrong!
- Humans (usually) get it right, though.

RDF-Centered Research Topics

- Information Extraction

  [SIGMOD'09, WebDB'10, PODS'10, WSDM'11, CIKM'12, CLEF/INEX'11/'12, LDOW'14, TACL'16]

- Uncertain RDF Data & Probabilistic Databases

  [ICDE'08, VLDB-J'08, SSDBM'10, BTW'11, CIKM'11, ICDE'13, PVLDB'14, VLDB PhD Workshop'15]

- Scalable RDF Indexing & SPARQL Query Processing

  [SIGMOD'14, SWIM'14, SIGMOD'16]

Probabilistic Database

A probabilistic database $\mathbf{D}^p$ (compactly) encodes a probability distribution over a finite set of deterministic database instances $\mathbf{D}_i$.

(I) $\mathbf{D_1}$: **0.42**

worksAt(sub, obj)	
Jeff	Stanford
Jeff	Princeton

$\mathbf{D_2}$: **0.18**

worksAt(sub, obj)	
Jeff	Stanford

$\mathbf{D_3}$: **0.28**

worksAt(sub, obj)	
Jeff	Princeton

$\mathbf{D_4}$: **0.12**

worksAt(sub, obj)

- Special Cases:

(II) $\mathbf{D}^p$ tuple-independent

worksAt(sub, obj)		p
Jeff	Stanford	0.6
Jeff	Princeton	0.7

(III) $\mathbf{D}^p$ block-independent

worksAt(sub, obj)		p
Jeff	Stanford	0.6
	Princeton	0.4

Note:
(I) and (II) here are equivalent;
(II) and (III) not!

- **Query Answering Problem:** ("Marginal Probabilities" of Query Answers)

Run query **Q** against each instance $\mathbf{D}_i$; for each answer tuple t_j , $P(t_j)$ is the sum of the probabilities of all instances $\mathbf{D}_i$ where t_j exists.

Flashback: Stanford Trio System

[Widom: CIDR'05]

Uncertainty-Lineage Databases (ULDBs)

1. Alternatives
2. '?' (Maybe) Annotations
3. Confidence values
4. Lineage

Trio's Data Model

[Widom: CIDR'05]

1. Alternatives: uncertainty about value

Saw (witness, *color*, *car*)	
Amy	red, Honda || red, Toyota || orange, Mazda

Three possible instances

Trio's Data Model

[Widom: CIDR'05]

1. Alternatives
2. **'?' (Maybe):** uncertainty about presence

<table>
<tr><th colspan="2">Saw (witness, color, car)</th><th></th></tr>
<tr><td>Amy</td><td>red, Honda || red, Toyota || orange, Mazda</td><td></td></tr>
<tr><td>Betty</td><td>blue, Acura</td><td>?</td></tr>
</table>

Six possible instances

Trio's Data Model

[Widom: CIDR'05]

1. Alternatives
2. '?' (Maybe) Annotations
3. **Confidences:** weighted uncertainty

Saw (witness, *color, car*)		
Amy	***red, Honda*** 0.5 || ***red, Toyota*** 0.3 || ***orange, Mazda*** 0.2	
Betty	***blue, Acura*** 0.6	?

Still six possible instances, but each with a probability

So Far: Data Model is Not Closed

[Widom: CIDR'05]

Saw (witness, *car*)	
Cathy	Honda || Mazda

Drives (*person*, *car*)
Jimmy, Toyota || Jimmy, Mazda
Billy, Honda || Frank, Honda
Hank, Honda

Suspects = $\pi_{person}(\text{Saw} \bowtie \text{Drives})$

Suspects	
Jimmy	?
Billy || Frank	?
Hank	?

CANNOT correctly capture possible result instances

Example with Lineage

ID	Saw (witness, *car*)	
11	Cathy	Honda || Mazda

ID	Drives (*person*, *car*)
21	Jimmy, Toyota || Jimmy, Mazda
22	Billy, Honda || Frank, Honda
23	Hank, Honda

$$\text{Suspects} = \pi_{person}(\text{Saw} \bowtie \text{Drives})$$

ID	Suspects	
31	Jimmy	?
32	Billy || Frank	?
33	Hank	?

$\lambda(31) = (11,2) \wedge (21,2)$

$\lambda(32,1) = (11,1) \wedge (22,1)$; $\lambda(32,2) = (11,1) \wedge (22,2)$

$\lambda(33) = (11,1) \wedge 23$

Example with Lineage

ID	Saw (witness, *car*)	
11	Cathy	Honda || Mazda

ID	Drives (*person*, *car*)
21	Jimmy, Toyota || Jimmy, Mazda
22	Billy, Honda || Frank, Honda
23	Hank, Honda

$$\text{Suspects} = \pi_{\text{person}}(\text{Saw} \bowtie \text{Drives})$$

ID	Suspects	
31	Jimmy	?
32	Billy || Frank	?
33	Hank	?

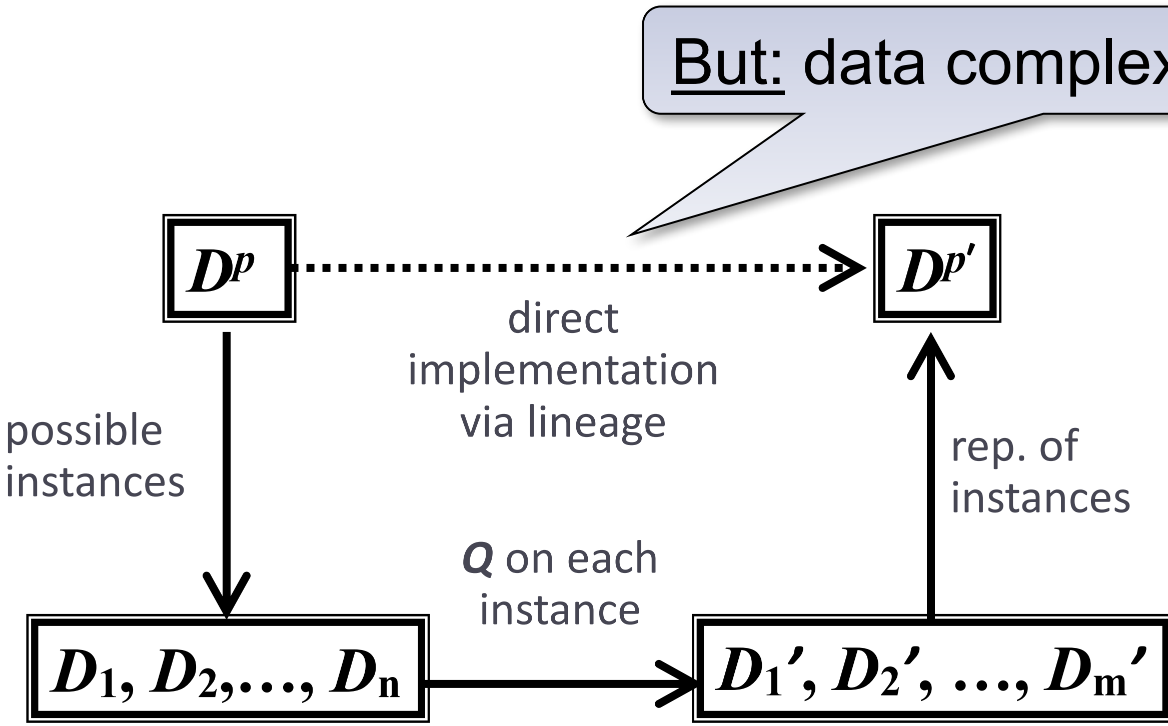
$\lambda(31) = (11,2) \land (21,2)$

$\lambda(32,1) = (11,1) \land (22,1)$; $\lambda(32,2) = (11,1) \land (22,2)$

$\lambda(33) = (11,1) \land 23$

Correctly captures the possible result instances (4)

Operational Semantics



Closure: up-arrow always exists

Completeness: any (finite) set of possible instances can be represented

(will be coming back to this subtlety again later…)

Summary on Trio's Data Model

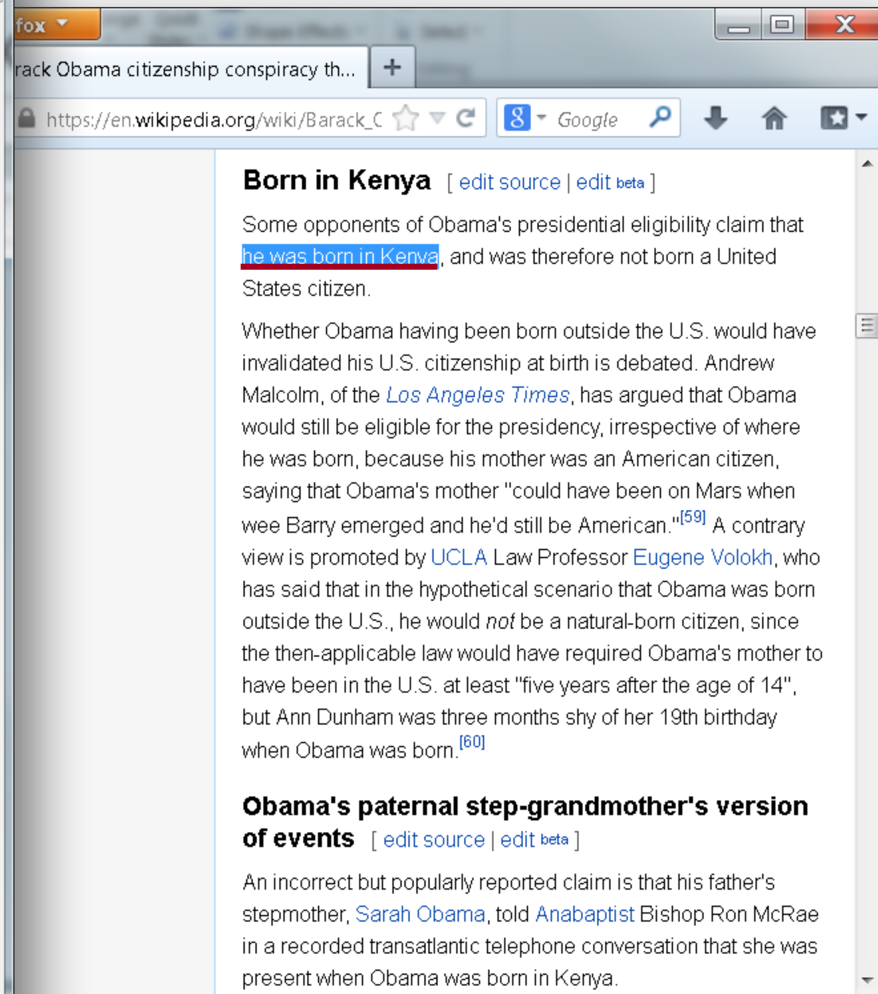
Uncertainty-Lineage Databases (ULDBs)

1. Alternatives
2. '?' (Maybe) Annotations
3. Confidence values
4. Lineage

Theorem: ULDBs are **closed** and **complete**.

Formally studied properties like *minimization*, *equivalence*, *approximation* and *membership* **based on lineage**.
[Benjelloun, Das Sarma, Halevy, Widom, Theobald: VLDB-J. 2008]

… back to Wikipedia



bornIn(Barack, Hawaii)

~~**bornIn(Barack, Kenya)**~~

Soft Rules vs. Hard Rules

(**Soft**) Deduction Rules vs.
(**Hard**) Consistency Constraints

- People may **live in** more than one p…

  $\mathtt{livesIn(x,y)} \Leftarrow \mathtt{marriedTo(x,z)} \wedge \mathtt{lives}\ldots\mathtt{,y)}\ [0.8]$

  $\mathtt{livesIn(x,y)} \Leftarrow \mathtt{hasChild(x,z)} \wedge \mathtt{livesIn(z,y)}\ [0.5]$

Deductive Database:
Datalog, Core of SQL & Relational Algebra, RDF/S, OWL2-RL, etc.

- People are not **born in** different places/on different dates

  $\mathtt{bornIn(x,y)} \wedge \mathtt{bornIn(x,z)} \Rightarrow \mathtt{y{=}z}$

  $\mathtt{bornOn(x,y)} \wedge \mathtt{bornOn(x,z)} \Rightarrow \mathtt{y{=}z}$

- People are not **married to** more tha…

  (at the *same time*, in *most countries*?)

  $\mathtt{marriedTo(x,y,t_1)} \wedge \mathtt{marriedTo(x,z,t_2)} \wedge \mathtt{y{\neq}z}$

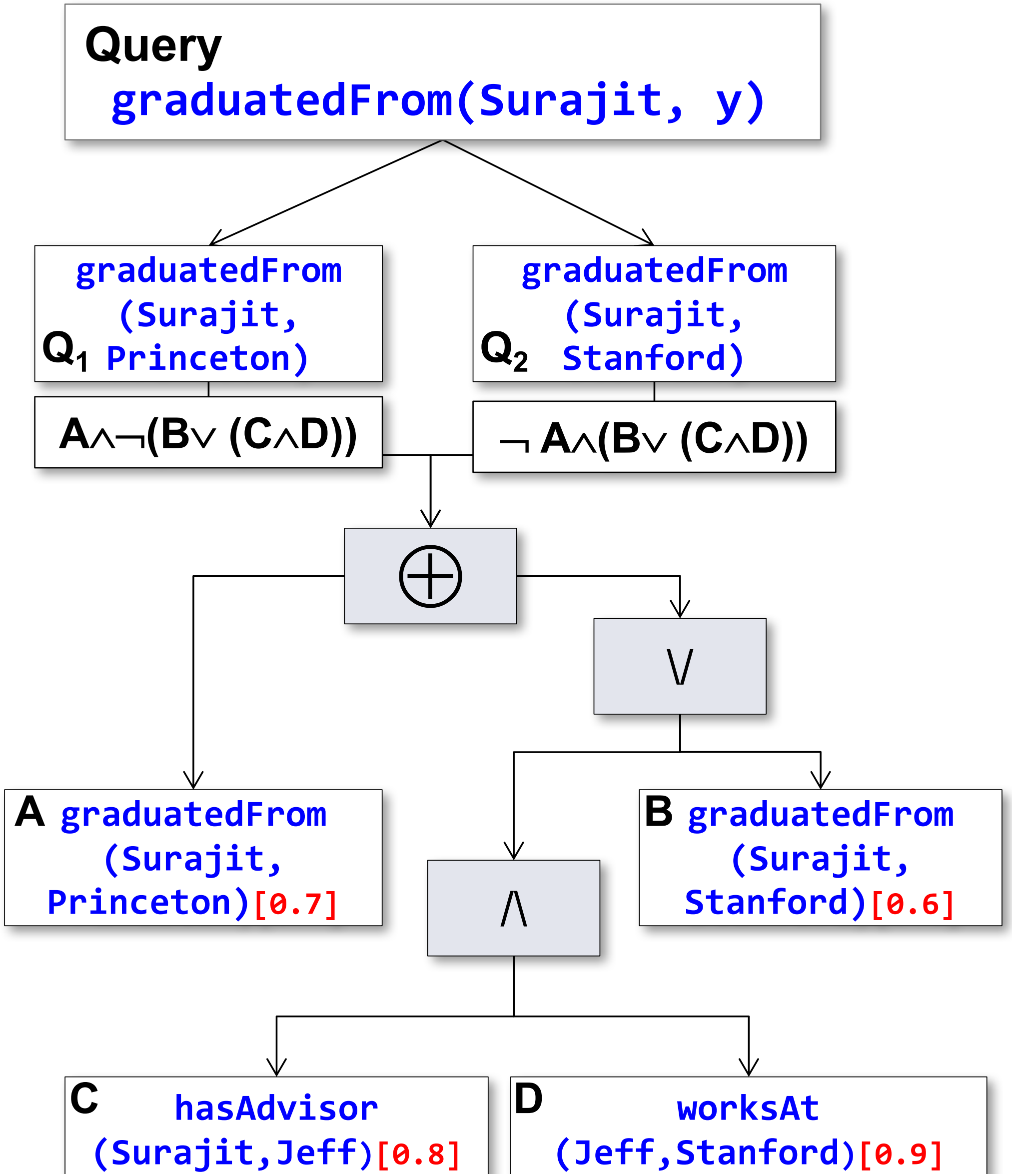
  $\Rightarrow \mathtt{disjoint(t_1,t_2)}$

More General FOL Constraints:
Datalog plus constraints, *owl:FunctionalProperty*, *owl:disjointWith*, etc.

Deductive Grounding w/ Lineage

(SLD Resolution in Datalog/Prolog)

[Yahya,Theobald: RuleML'11, Dylla,Miliaraki,Theobald: ICDE'13]



Rules

hasAdvisor(x,y) ∧
worksAt(y,z)
⇒ graduatedFrom(x,z)

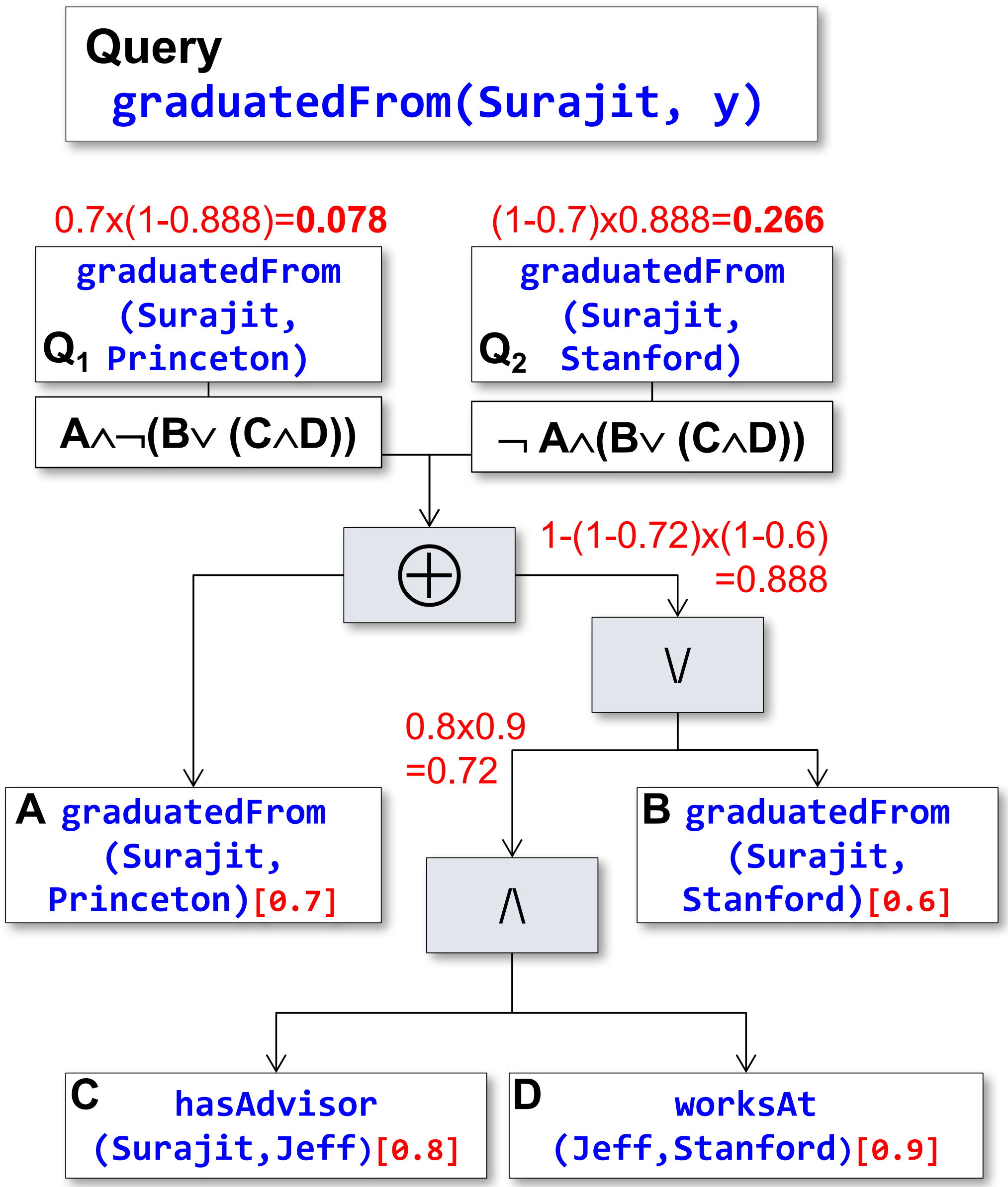
graduatedFrom(x,y) ∧
graduatedFrom(x,z)
⇒ y=z

Base Facts

graduatedFrom(Surajit, Princeton) [0.7]
graduatedFrom(Surajit, Stanford) [0.6]
graduatedFrom(David, Princeton) [0.9]
hasAdvisor(Surajit, Jeff) [0.8]
hasAdvisor(David, Jeff) [0.7]
worksAt(Jeff, Stanford) [0.9]
type(Princeton, University) [1.0]
type(Stanford, University) [1.0]
type(Jeff, Computer_Scientist) [1.0]
type(Surajit, Computer_Scientist) [1.0]
type(David, Computer_Scientist) [1.0]

Lineage & Possible Worlds

[Das Sarma,Theobald,Widom: ICDE'08,
Dylla,Miliaraki,Theobald: ICDE'13]



1) **Deductive Grounding**
- Top-down Datalog evaluation
- Plus tracing the lineage of individual query answers

2) **Lineage DAGs**
- Grounded soft & hard rules
- Base facts with confidences

3) **Probabilistic Inference**

→ Compute marginals:

P(Q): **sum up the probabilities of all possible worlds** that entail the query answers

P(Q|H): drop "impossible worlds"

$$P(Q) := \sum_{\mathcal{W} \models Q} \left(\prod_{X_i \in \mathcal{W}} p(X_i) \prod_{X_i \notin \mathcal{W}} (1 - p(X_i)) \right)$$

Possible Worlds Semantics

$P(Q_1)$=0.0784

$P(Q_2)$=0.2664

$P(Q_1|H)$=0.0784 / 0.412 = 0.1903

$P(Q_2|H)$=0.2664 / 0.412 = 0.6466

A:0.7	B:0.6	C:0.8	D:0.9		Q_2: ¬ A∧(B∨(C∧D))		P(W)
~~1~~	~~1~~	~~1~~	~~1~~		~~0~~		~~0.7x0.6x0.8x0.9 = 0.3024~~
~~1~~	~~1~~	~~1~~	~~0~~		~~0~~		~~0.7x0.6x0.8x0.1 = 0.0336~~
~~1~~	~~1~~	~~0~~	~~1~~		~~0~~		~~… = 0.0756~~
~~1~~	~~1~~	~~0~~	~~0~~		~~0~~		~~… = 0.0084~~
~~1~~	~~0~~	~~1~~	~~1~~		~~0~~		~~… = 0.2016~~
1	0	1	0		0		… = 0.0224
1	0	0	1		0		… = 0.0504
1	0	0	0		0		… = 0.0056
0	1	1	1		1		0.3x0.6x0.8x0.9 = 0.1296
0	1	1	0		1		0.3x0.6x0.8x0.1 = 0.0144
0	1	0	1		1		0.3x0.6x0.2x0.9 = 0.0324
0	1	0	0		1		0.3x0.6x0.2x0.1 = 0.0036
0	0	1	1		1		0.3x0.4x0.8x0.9 = 0.0864
0	0	1	0		0		… = 0.0096
0	0	0	1		0		… = 0.0216
0	0	0	0		0		… = 0.0024

0.0784

0.2664

1.0

0.412

Hard rule H: ¬ A ∨ ¬ (B ∨ (C∧D))

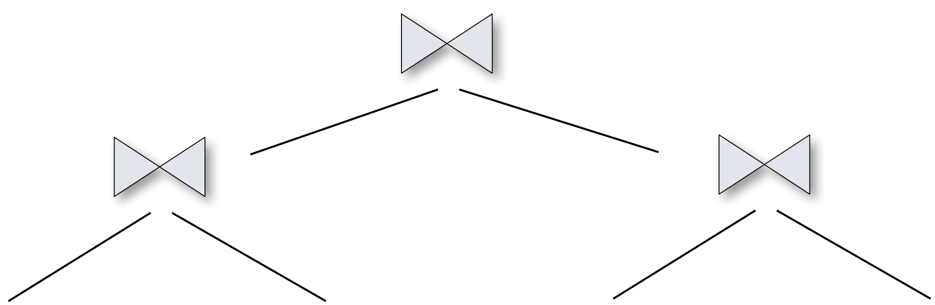
Dichotomy of Queries

[Suciu & Dalvi: SIGMOD'05 Tutorial on "Foundations of Probabilistic Answers to Queries"]

A probabilistic database $\mathbf{D}^p$ (compactly) encodes a probability distribution over a finite set of deterministic database instances $\mathbf{D}_i$.

Is there any professor who works at a university that is located in CA?

```
Q() :- isProfessor(pers), worksAt(pers,uni), located(uni, CA)
```



isProfessor(pers)	p
Jeff	0.9

worksAt(pers, uni)		p
Jeff	Stanford	0.6
Jeff	Princeton	0.7

located(uni, state)		p
Stanford	CA	0.8
Princeton	CA	0.1

Theorem: The query answering problem for the above join query over a tuple-independent probabilistic database is **#P-hard**.

Inference in Probabilistic Databases

Safe query plans [Dalvi & Suciu: VLDB-J'07+J-ACM'12]

- Can propagate confidences along with relational operators.

Read-once functions [Sen et al.: PVLDB'10; Olteanu & Huang: SUM'08]

- Can factorize Boolean formula (in polynomial time) into read-once form, where every variable occurs at most once.

Knowledge compilation [Olteanu et al.: ICDE'10; ICDT'11; VLDB-J'13]

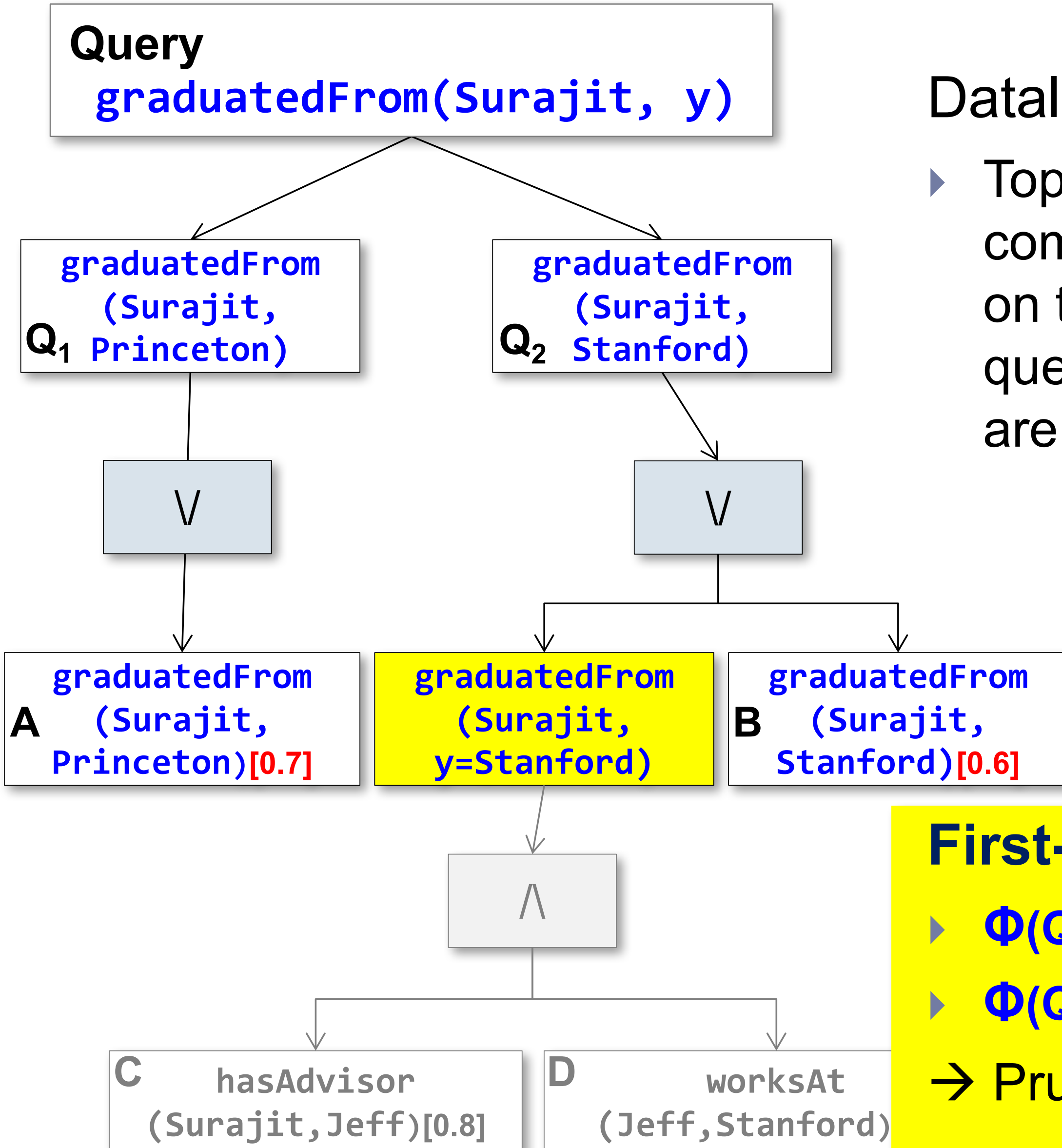
- Can compile Boolean formula into a decision diagram (OBDD/SDD), such that inference resolves to *independent-and* and *independent-or* operations over the decomposed formula.

Top-*k* pruning [Ré, Davli & Suciu: ICDE'07; Karp, Luby & Madras: J-Alg.'89; Olteanu & Wen: ICDE'12]

- Can return top-*k* answers based on *lower* and *upper bounds*, even without knowing their exact marginal probabilities.
- <u>Multi-Simulation:</u> run multiple Markov-Chain-Monte-Carlo (MCMC) simulations in parallel.

Top-*k* Ranking by Marginal Probabilities

[Dylla,Miliaraki,Theobald: ICDE'13]



Datalog/SLD resolution

- Top-down grounding allows us to compute **lower** and **upper bounds** on the marginal probabilities of query answers *even before* rules are fully grounded.

First-order lineage formulas:

- **$\Phi(Q_1) = A$**
- **$\Phi(Q_2) = B \vee \exists y$ gradFrom(Surajit,y)**

→ Prune entire sets of answer candidates represented by Φ.

Bounds for First-Order Formulas

[Dylla,Miliaraki,Theobald: ICDE'13]

Theorem 1:

Given a (partially grounded) **first-order lineage** formula Φ:

$\Phi(Q_2) = B \vee \exists y\ gradFrom(S,y)$

- **Lower bound P_{low}** (for all query answers that can be obtained from grounding Φ): Substitute ∃y gradFrom(S,y) with *false* (or *true* if negated).

  $P_{low}(Q_2) = P(B \vee false) = P(B) = 0.6$

- **Upper bound P_{up}** (for all query answers that can be obtained from grounding Φ): Substitute ∃y gradFrom(S,y) with *true* (or *false* if negated).

  $P_{up}(Q_2) = P(B \vee true) = P(true) = 1.0$

Proof: (sketch)

Substitution of a subformula with *false* reduces the number of models *(possible worlds)* that satisfy Φ; substitution with *true* increases them.

Convergence of Bounds

[Dylla,Miliaraki,Theobald: ICDE'13]

Theorem 2:

Let $\Phi_1,\ldots,\Phi_n$ be a series of first-order lineage formulas obtained from grounding Φ via SLD resolution, and let φ be the *propositional lineage* formula of an answer obtained from this grounding procedure.

Then rewriting each Φ_i according to Theorem 1 into $P_{i,low}$ and $P_{i,up}$ creates a **monotonic series** of lower and upper bounds that **converges to P(φ)**.

$$0 = P(\mathit{false}) \leq P(B \vee \mathit{false}) = 0.6 \leq P(B \vee (C \wedge D)) = 0.888$$
$$\leq P(B \vee \mathit{true}) = P(\mathit{true}) = 1$$

Proof: (sketch, via induction)

Substitution of *true* with a formula reduces the number of models that satisfy Φ; substitution of *false* with a formula increases this number.

Top-*k* Stopping Condition

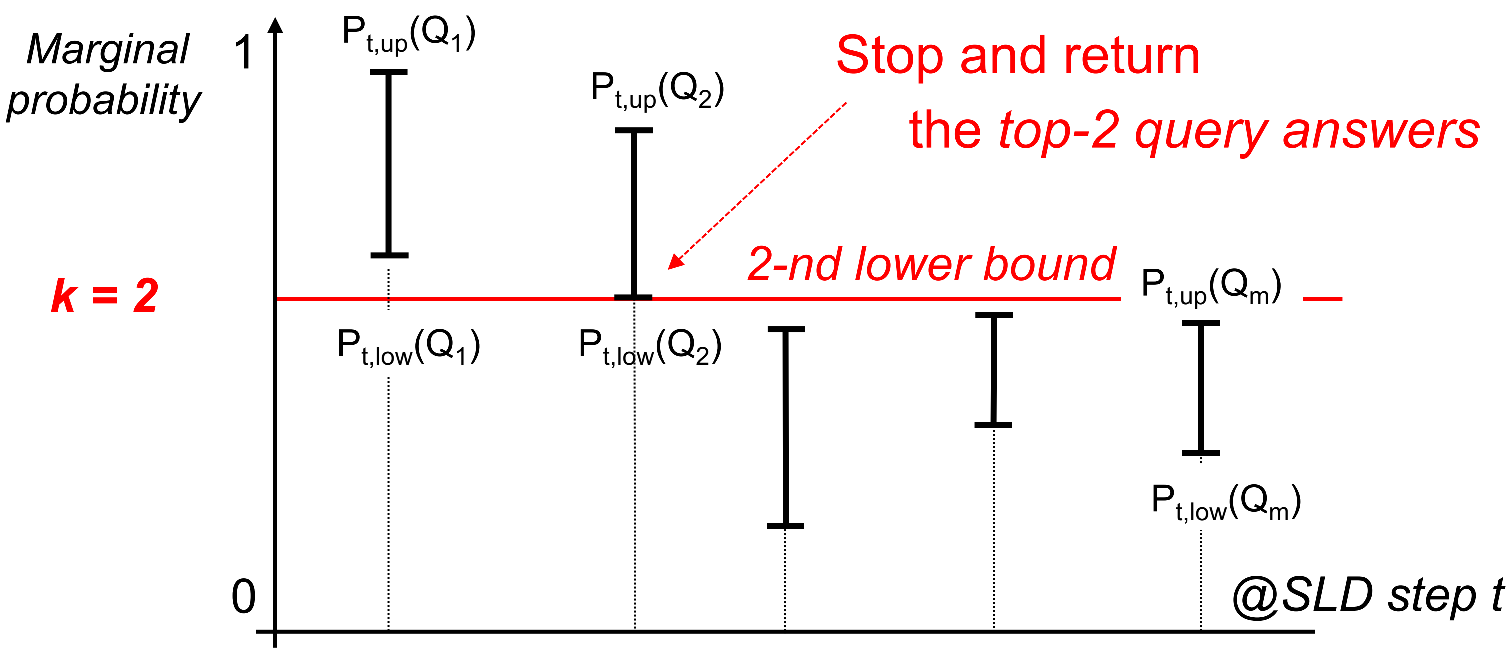
[Fagin et al.'01; Balke,Kießling'02; Dylla,Miliaraki,Theobald: ICDE'13]

"Fagin's Algorithm"

- Maintain two disjoint queues:

  Top-k sorted by P_{low} and *Candidates* sorted by P_{up}
- Return the *top-k* queue at the *t*-th grounding step when:

  $P_{i,low}(Q_k) \mid_{Qk \in Top\text{-}k} > P_{i,up}(Q_j) \mid_{Qj \in Candidates}$

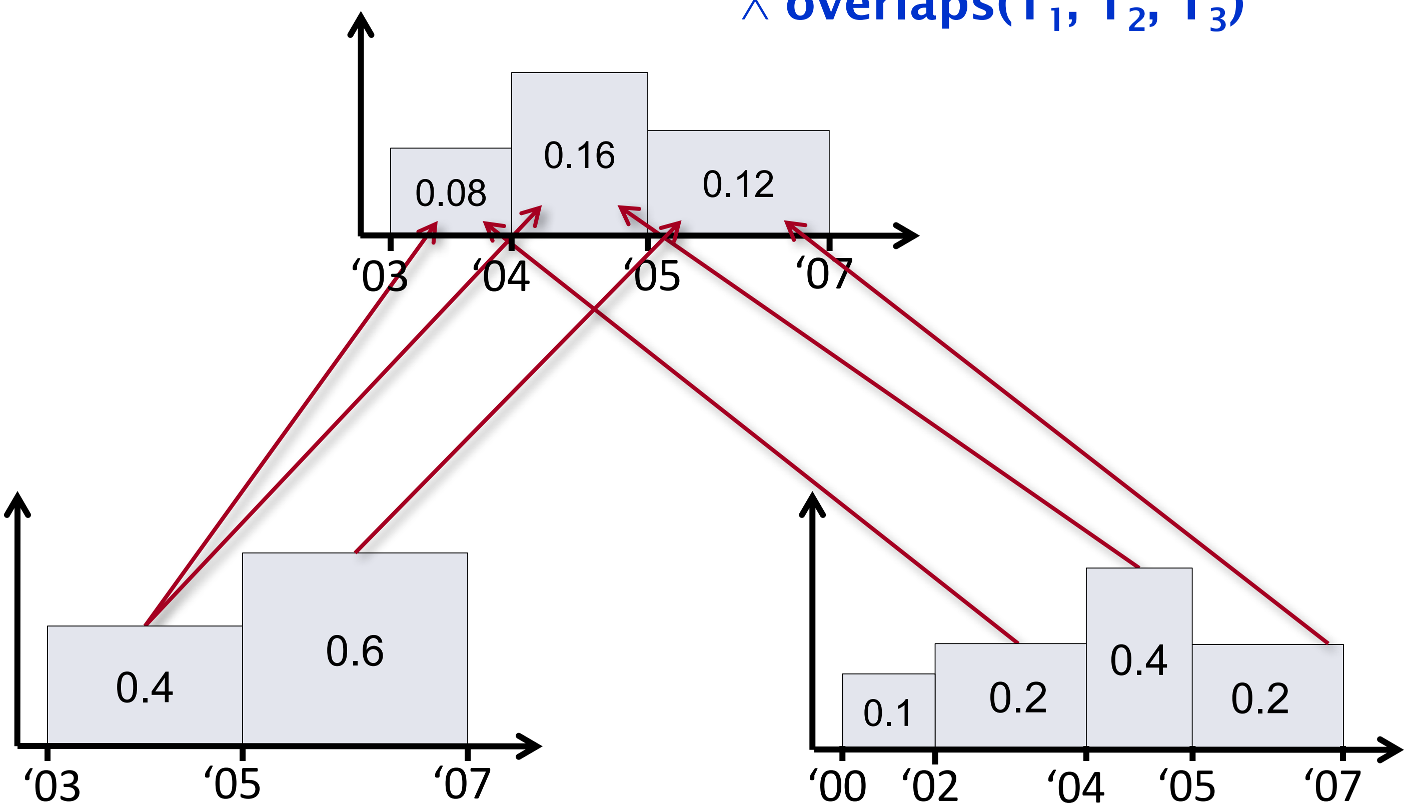


Temporal-Probabilistic Database

[Wang,Yahya,Theobald: MUD'10; Dylla,Miliaraki,Theobald: PVLDB'13]

Derived Facts

teamMates(Beckham, Ronaldo, T_3) ? $\Leftarrow$ **playsFor(Beckham, Real, T_1)** $\wedge$ **playsFor(Ronaldo, Real, T_2)** $\wedge$ **overlaps(T_1, T_2, T_3)**



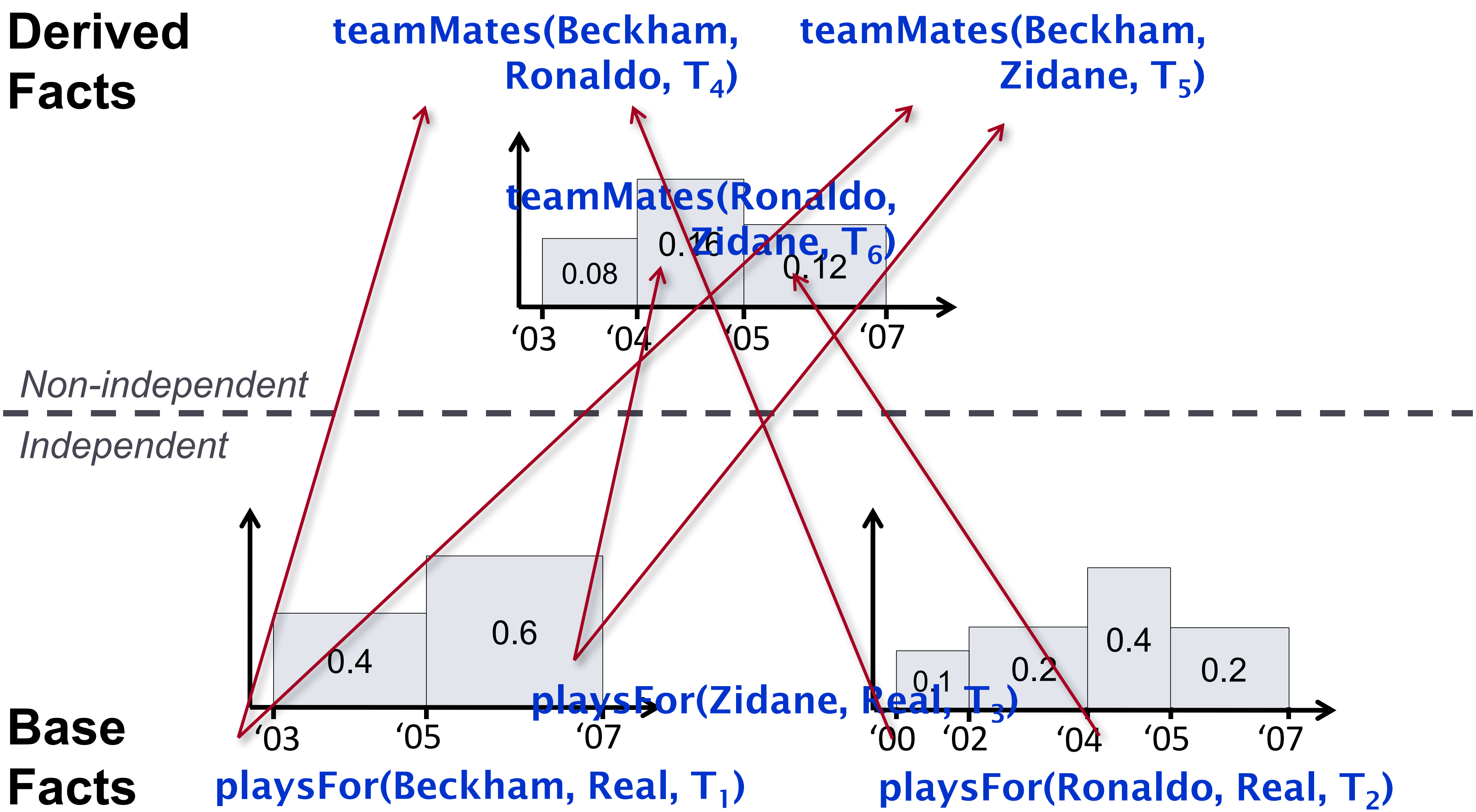
Base Facts

playsFor(Beckham, Real, T_1) **playsFor(Ronaldo, Real, T_2)**

Example using the Allen predicate **overlaps**

Inference in Temporal-Probabilistic Databases

[Wang,Yahya,Theobald: MUD'10; Dylla,Miliaraki,Theobald: PVLDB'13]



Inference in Temporal-Probabilistic Databases

[Wang,Yahya,Theobald: MUD'10; Dylla,Miliaraki,Theobald: PVLDB'13]

Derived Facts

teamMates(Beckham, Ronaldo, T_4)

teamMates(Beckham, Zidane, T_5)

teamMates(Ronaldo, Zidane, T_6)

Need Lineage!

Non-independent

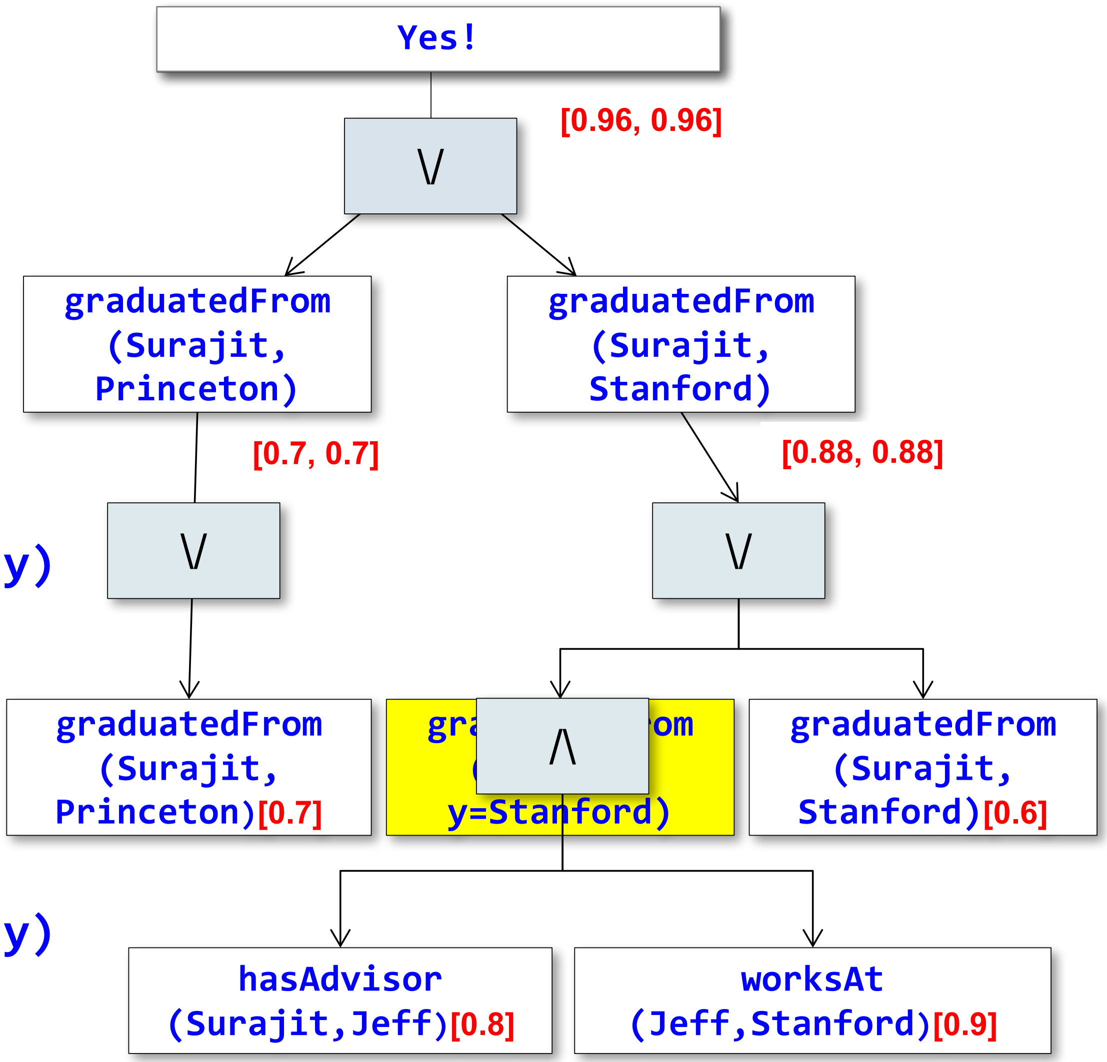
Independent

- **Closed** and **complete** representation model (incl. lineage)
- **Temporal alignment** is **linear** in the number of input intervals
- **Probabilistic inference** per interval remains **#P-hard**
- Inference requires lineage decompositions, top-*k* pruning, or Monte Carlo approximations (Luby-Karp for DNF, MCMC-style sampling)

Ultimate PhD Challenge (II)

Lifted inference with **fully integrated relational and probabilistic optimization** for arbitrary SQL queries!

- **Query answering**:
  `graduatedFrom(Surajit, y)`
- **Boolean queries**:
  `∃y graduatedFrom(Surajit, y)`



RDF-Centered Research Topics

- Information Extraction

  [SIGMOD'09, WebDB'10, PODS'10, WSDM'11, CIKM'12, CLEF/INEX'11/'12, LDOW'14, TACL'16]

- Uncertain RDF Data & Probabilistic Databases

  [ICDE'08, VLDB-J'08, SSDBM'10, BTW'11, CIKM'11, ICDE'13, PVLDB'14, VLDB PhD Workshop'15]

- Scalable RDF Indexing & SPARQL Query Processing

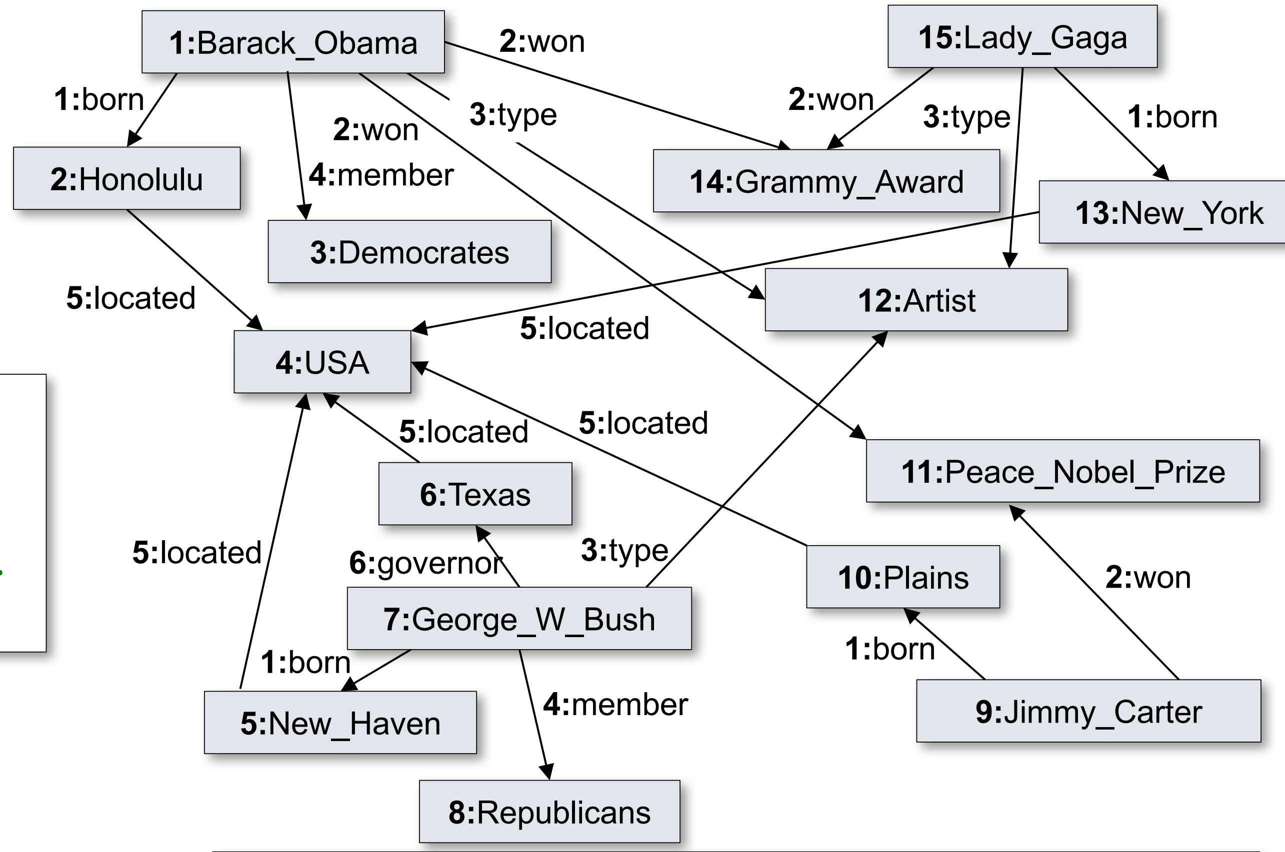
  [SIGMOD'14, SWIM'14, SIGMOD'16]

RDF & SPARQL

RDF Data:

1. Relation(s)
2. Logical facts
3. Directed graphs

```
SELECT ?p, ?c, ?a
WHERE {
   ?p <born> ?c.
   ?c <located> <USA>.
   ?p <won> ?a }
```



SPARQL Queries:

1. Relational joins
2. Conjunction of first-order lite
3. Subgraph isomorphism que

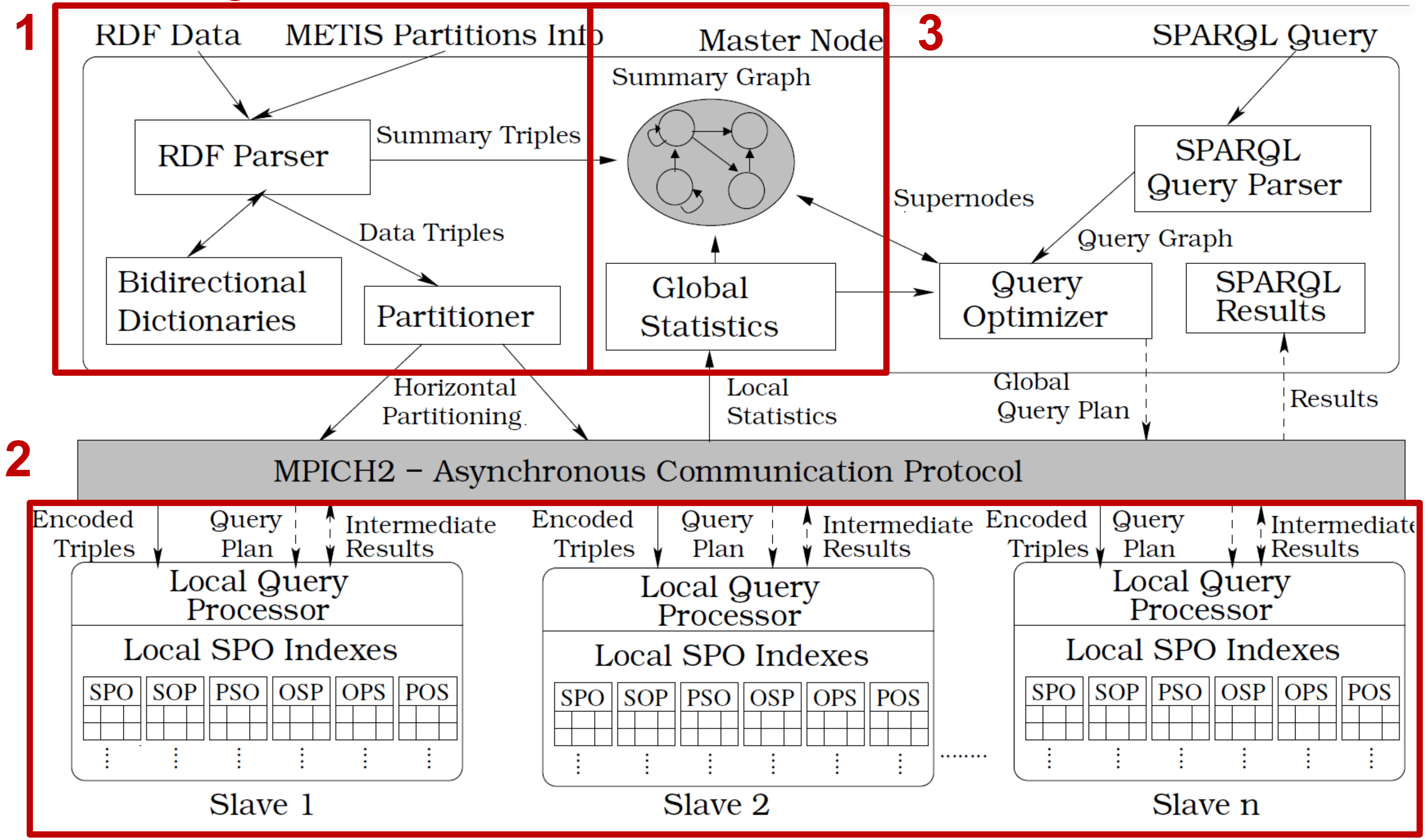
```
Barack_Obama, Honolulu, Grammy_Award.
Barack_Obama, Honolulu, Peace_Nobel_Prize.
Lady_Gaga, New_York, Grammy_Award.
Jimmy_Carter, Plains, Peace_Nobel_Prize.
```

Data complexity of core SPARQL: *polynomial*

Combined data & query complexity: *exponential* *(same as SQL w/o recursion)*

RDF Indexing

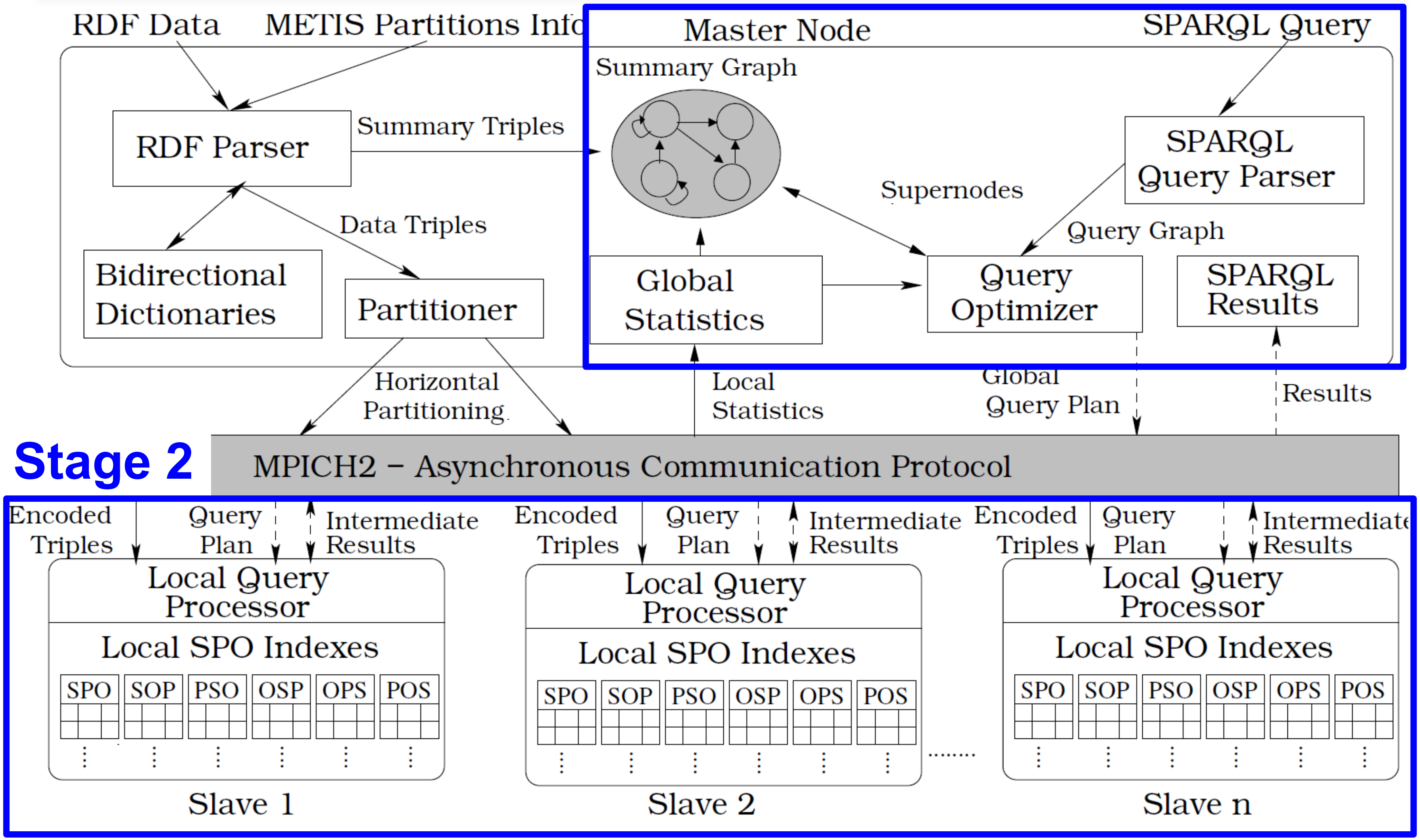
TriAD Architecture



SPARQL Query Processing

TriAD Architecture

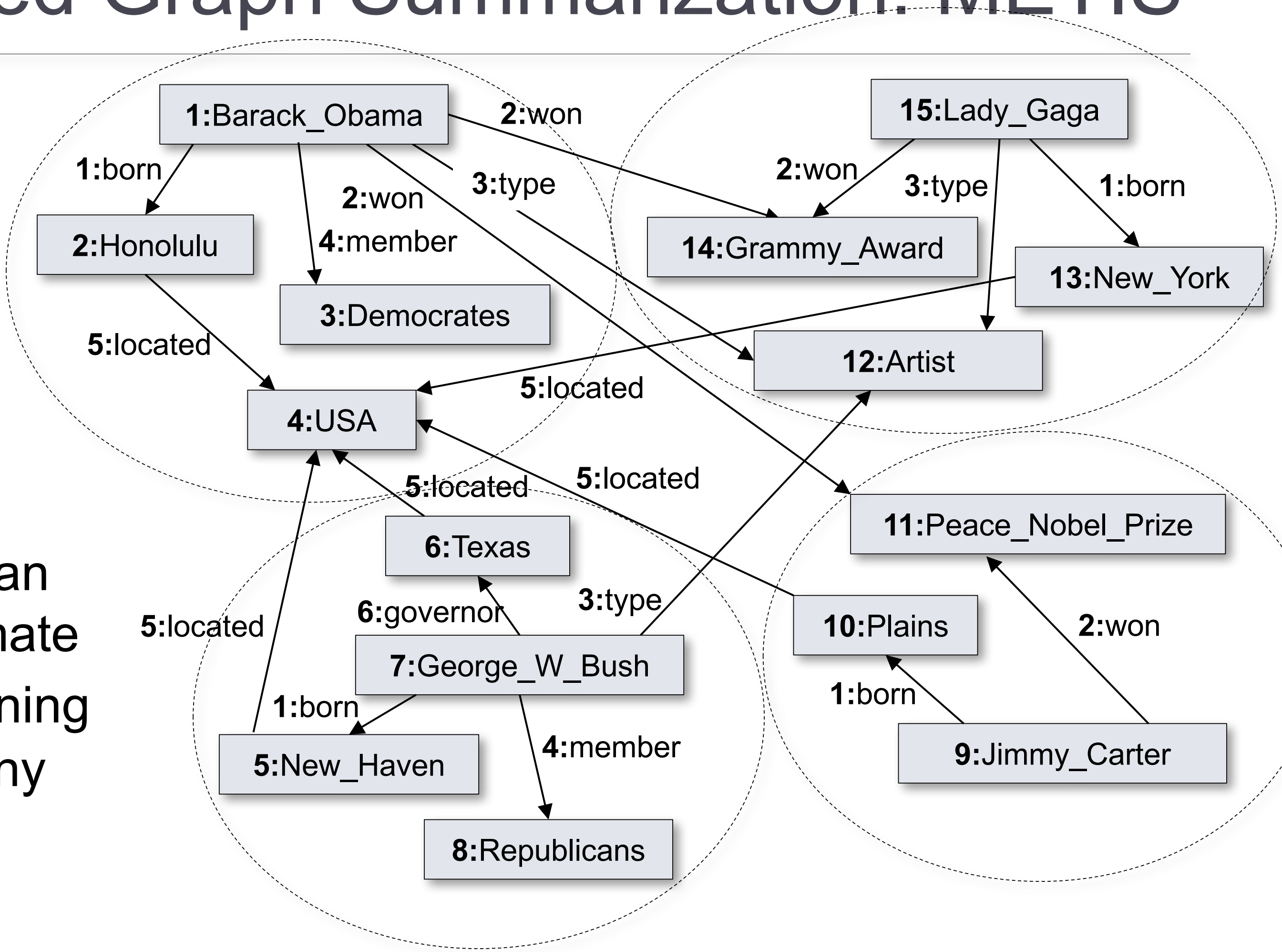
Stage 1



→ TriAD follows a very **classical master-slave architecture**; however with a **direct (asynchronous) communication** among all slaves at query time.

Locality-Based Graph Summarization: METIS

$$k = 4$$



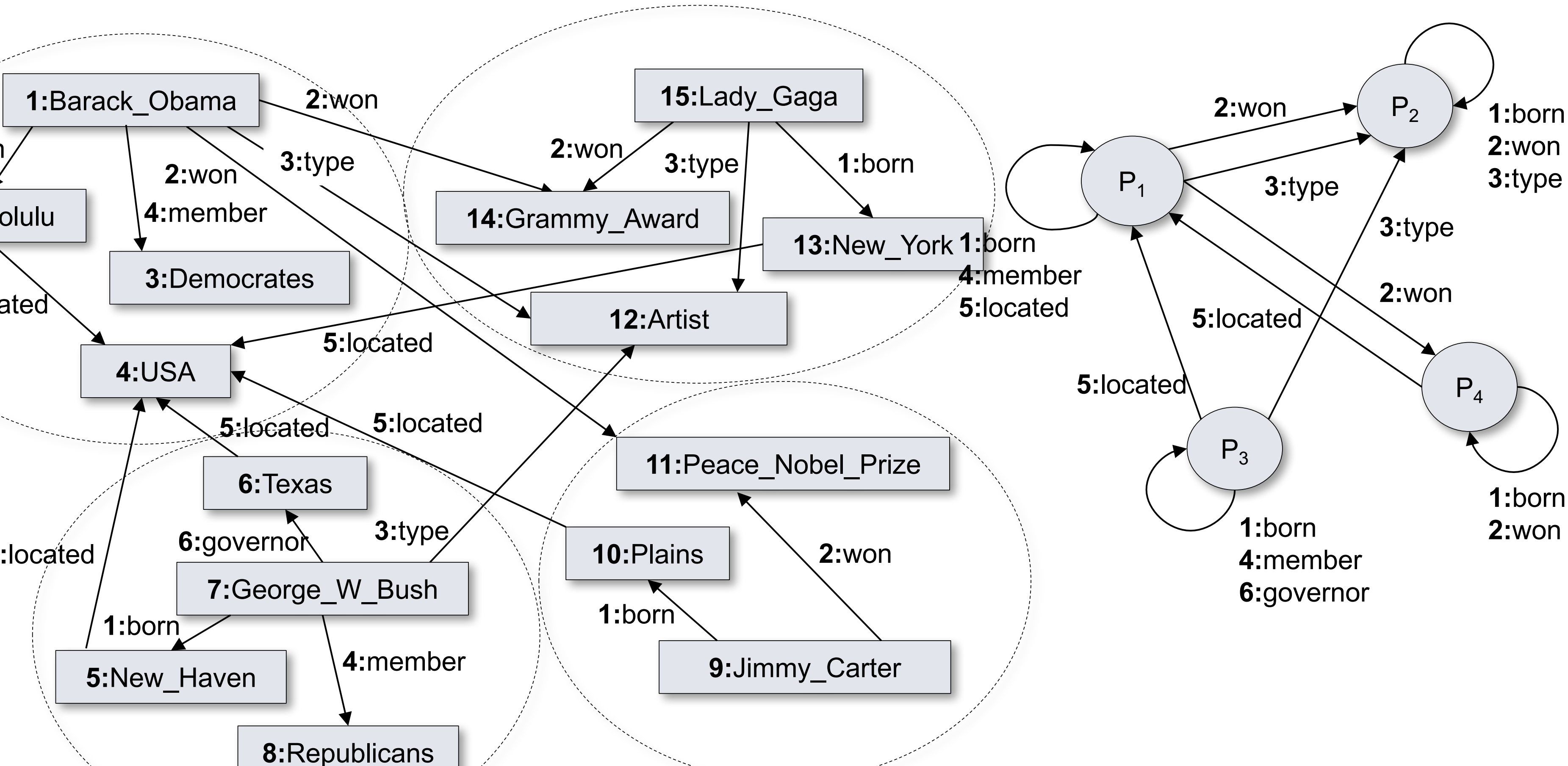
METIS

- Tools like METIS can efficiently approximate a min-k-cut partitioning for graphs with many millions of nodes/edges.

Min-k-Cut

- For a desired amount of k evenly sized partitions, assign each node in the RDF data graph to exactly one partition, such that the number of cut edges among those partitions is minimized.

Summary Graph



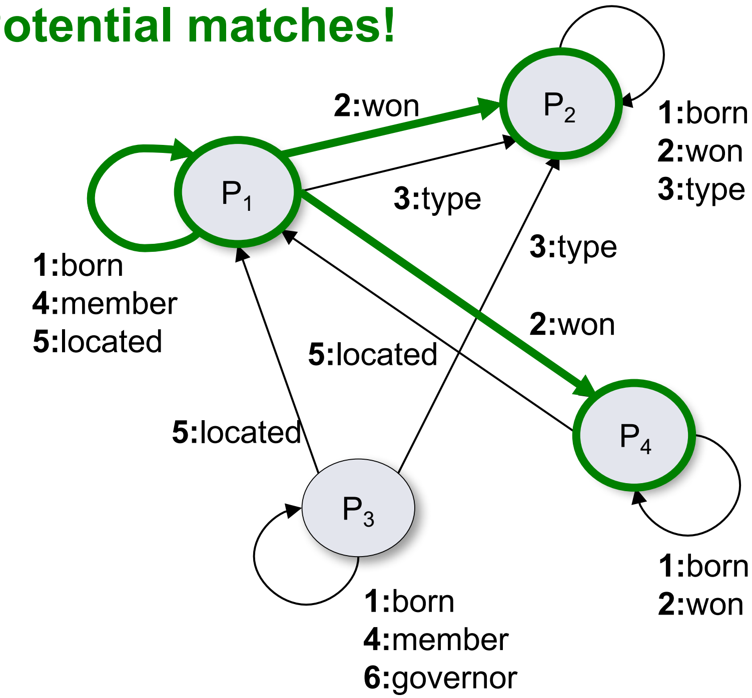
RDF Summary Graph

- Drop all nodes and edges inside the partitions
- Keep only inter-partition edges
- Introduce self-loop edges for intra-partition edges

Querying the Summary Graph

```
SELECT ?c, ?a
WHERE {
   <Barack_Obama> <born> ?c.
   ?c <located> <USA>.
   <Barack_Obama> <won> ?a }
```

Potential matches!



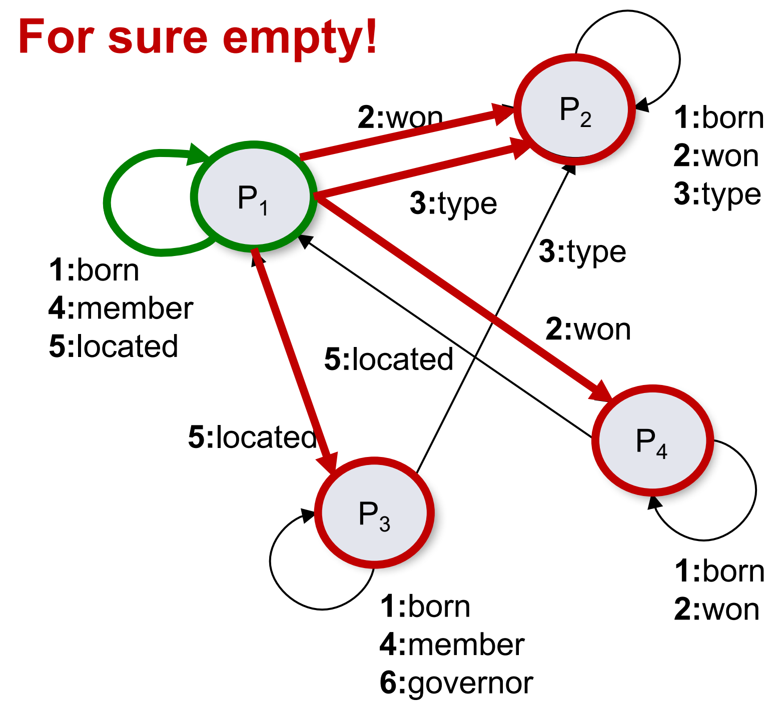
<u>Global Dictionary:</u>

Barack_Obama	$\rightarrow P_1$
USA	$\rightarrow P_1$
Lady_Gaga	$\rightarrow P_2$
Peace_Nobel_Prize	$\rightarrow P_4$

…

Querying the Summary Graph

```
SELECT ?c, ?s
WHERE {
   <Barack_Obama> <born> ?c.
   ?c <located> <USA>.
   <Barack_Obama> <governor> ?s
}
```



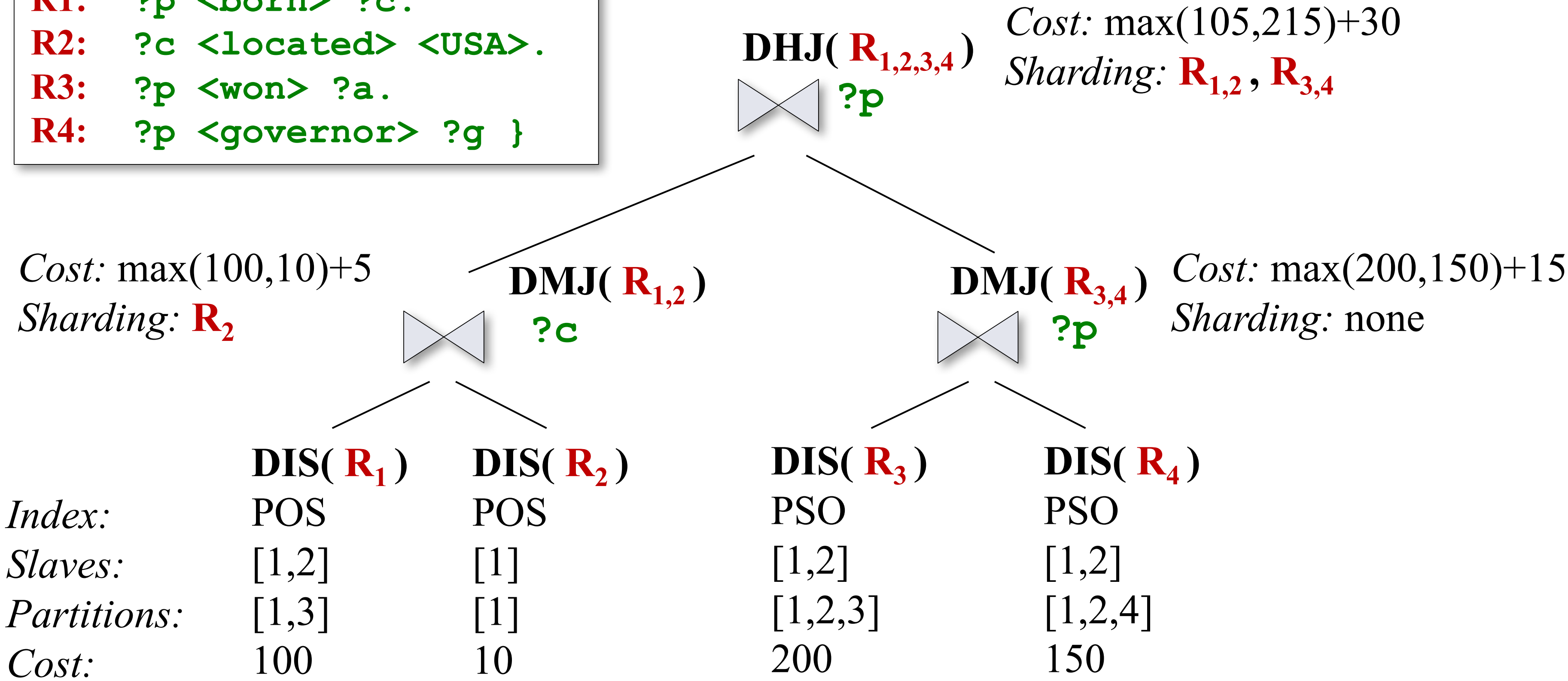
Global Dictionary:

`Barack_Obama` → P_1
`USA` → P_1
`Lady_Gaga` → P_2
`Peace_Nobel_Prize` → P_4
…

- Summary graph guarantees **no false negatives** (i.e., "missed results"); the subsequent processing of the query against the pruned data graph also ensures **no false positives**.
- Facilitates **join-ahead pruning** by skipping over irrelevant partitions.

Example Query Plan

```
SELECT ?p, ?c, ?a, ?g
WHERE {
R1:  ?p <born> ?c.
R2:  ?c <located> <USA>.
R3:  ?p <won> ?a.
R4:  ?p <governor> ?g }
```

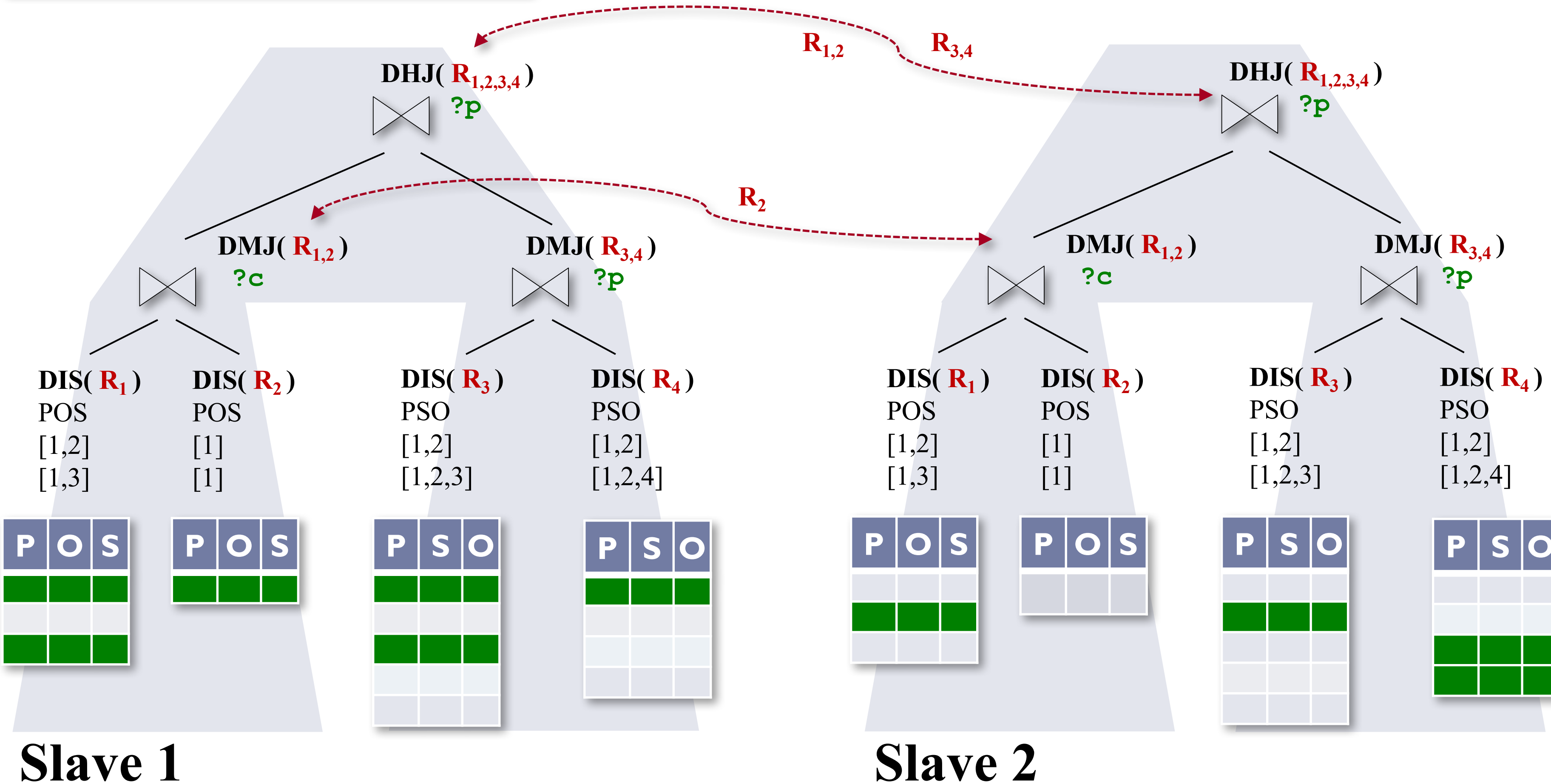


- A copy of the same query plan is shipped to all slaves:
  - DIS operators (leafs) are augmented with **locality** and **pruning information**.
  - 6 SPO permutations allow the usage of **DMJ op's at the first level of joins**.

Distributed & Multithreaded Query Execution

```
SELECT ?p, ?c, ?a, ?g
WHERE {
R1:  ?p <born> ?c.
R2:  ?c <located> <USA>.
R3:  ?p <won> ?a.
R4:  ?p <governor> ?g }
```

- All slaves **concurrently** and **asynchronously** process the same query plan, but each over **disjoint partitions** of the SPO permutation indexes.



Experiments

TriAD is implemented in C++ using GCC 4.4, Boost-1.5 & MPICH2.

All experiments were run on a proprietary cluster with 32 x 48 GB RAM, 2 quad-core XENON CPUs and a 1GBit Ethernet connection.

- **LUBM – Lehigh University Benchmark**

  Scale Factor 160: 28 Mio RDF triples → 16 GB data → 3 GB index

  Scale Factor 10240: 1.8 Bio RDF triples → 730 GB data → 150 GB index

- **BTC – Billion Triples Challenge (2012)**

  DBpedia/Yago/Freebase: 1.4 Bio RDF triples → 231 GB data → 130 GB index

- **WSDTS – Waterloo SPARQL Diversity Test Suite**

  Scale Factor 1000: 109 Mio RDF triples → 15 GB data → 9.1 GB index

9 Competitors: RDF-3x (MPII), MonetDB (U-Amsterdam), BitMat (Rensselaer Polytech), TripleBit (U-Huazhong/U-Georgia), Hadoop-RDF-3x (Yale), Apache Hadoop / Spark (UC Berkeley), SHARD (open-source), Trinity.RDF (MSR)

Benchmark Results

	TriAD	TriAD-SG (200K)	Trinity.RDF	SHARD	H-RDF-3X (cold)	H-RDF-3X (warm)	4store (cold)	4store (warm)	RDF-3X (cold)	RDF-3X (warm)	BitMat (cold)	BitMat (warm)
Q1	7,631	**2,146**	12,648	6.9E5	2.3E6	1.7E5	aborted	aborted	1.9E6	1.8E6	17,339	11,295
Q2	1,663	2,025	6,018	2.1E5	5.3E5	4,095	1.1E5	15,113	6.3E5	17,835	2.4E5	1.8E5
Q3	4,290	**1,647**	8,735	4.7E5	2.2E6	1.3E5	aborted	aborted	1.7E6	1.7E6	8,429	2,679
Q4	2.1	**1.3**	5	3.9E5	166	1	1,903	12	243	3	aborted	aborted
Q5	**0.5**	0.7	4	97,545	85	1	2,429	12	99	1	472	338
Q6	69	**1.4**	9	1.8E5	5.8E5	23,440	3,572	9	913	287	7,796	5,377
Q7	14,895	16,863	31,214	3.9E5	2.3E6	2.1E5	aborted	aborted	6.5E5	46,262	71,157	36,905
Geo-Mean	249	**106**	450	3.0E5	91,378	2,406	-	-	31,345	2,991	-	-

LUBM-10240: Query Processing Times in Milliseconds (*ms*)

	#Results	TriAD	TriAD-SG (200K)	H-RDF-3X (cold)	H-RDF-3X (warm)	RDF-3X (cold)	RDF-3X (warm)
Q1	1	1.5	**0.3**	49	6	297	4
Q2	1	61					
Q3	1	**1**					
Q4	0	**0.6**					
Q5	5	51					
Q6	0	0.5					
Q7	0	50					
Q8	292	128					
Geo.-Mean	–	7.4					

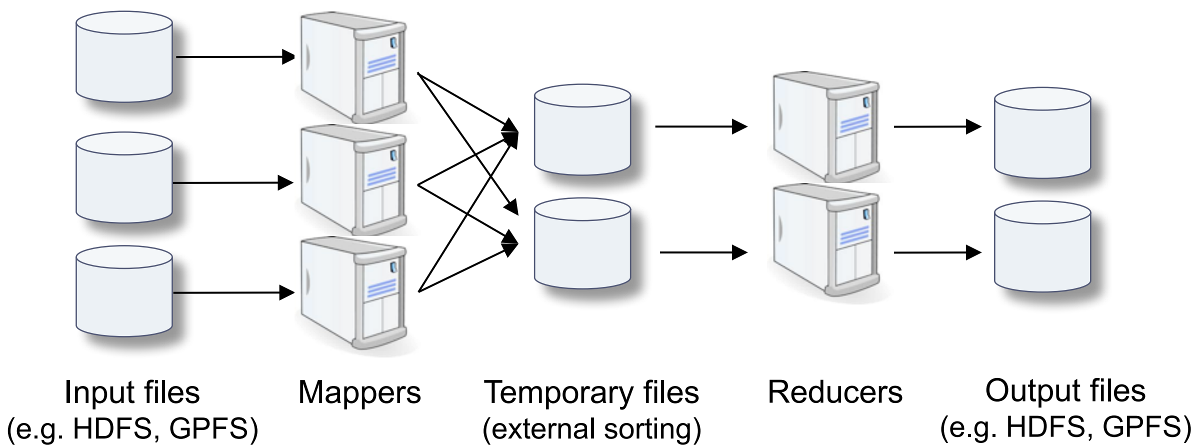
BTC: Query Pr

	#Slaves	L1-L5 (Geo.-Mean)	S1-S7 (Geo.-Mean)	F1-F5 (Geo.-Mean)	C1-C3 (Geo.-Mean)
TriAD	1	**2**	**2**	94	494
TriAD-SG(75K)	1	8	4	35	767
TriAD	5	**2**	3	**29**	**270**
SHARD	5	3.2E5	5.8E5	7.1E5	7.7E5
RDF-3X (cold)	1	10,066	167	1,749	6,610
RDF-3X (warm)	1	18	**2**	41	354
MonetDB (cold)	1	3530	10,459	timeout	timeout
MonetDB (warm)	1	171	744	timeout	timeout

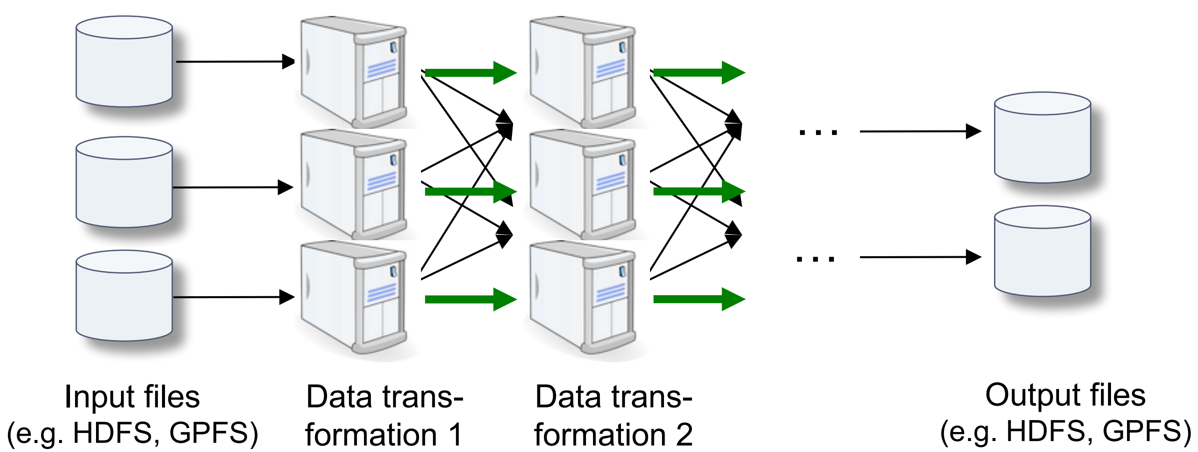
WSDTS-1000: Query Processing Times (*ms*)

Ultimate PhD Challenge (III)

- From **Map** & **Reduce**



- over **Synchronous Dataflows** to **Asynchronous Dataflows**!



Summary

Information Extraction

- Natural-Language Processing & Understanding
- Named-Entity Recognition & Disambiguation
- Extraction of N-Ary Relations
- Knowledge-Graph Construction, Integration & Maintenance

Uncertain Data

- Probabilistic & Temporal Data(base) Models
- Data Integration & Cleaning
- Model- & Dissociation-based Bounds
- Scalable Probabilistic Inference

Big Data

- Big Data Analytics
- Distributed Graph Engines
- Real-Time Dataflows & Stream Processing
- Message Passing & Asynchronous Protocols

References

- Sairam Gurajada, Martin Theobald: Distributed Set Reachability. SIGMOD 2016
- Dat Ba Nguyen, Martin Theobald, Gerhard Weikum: Joint Named Entity Resolution and Disambiguation with Rich Linguistic Features. TACL Vol. 4, 2016
- Hernán Blanco: Scaling Probabilistic Databases. VLDB PhD Workshop 2015
- Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, Gerhard Weikum: AIDA-light: High-Throughput Named-Entity Disambiguation. LDOW 2014
- Sairam Gurajada, Stephan Seufert, Iris Miliaraki, Martin Theobald: TriAD: A Distributed Shared-Nothing RDF Engine based on Asynchronous Message Passing. SIGMOD 2014
- Maximilian Dylla, Martin Theobald, Iris Miliaraki: Querying and Learning in Probabilistic Databases. Reasoning Web 2014
- Maximilian Dylla, Iris Miliaraki, Martin Theobald: A Temporal-Probabilistic Database Model for Information Extraction. PVLDB 6(14), 2014
- Maximilian Dylla, Iris Miliaraki, Martin Theobald: Top-k Query Processing in Probabilistic Databases with Non-Materialized Views. ICDE 2013
- Ndapandula Nakashole, Mauro Sozio, Fabian Suchanek, Martin Theobald: Query-Time Reasoning in Uncertain RDF Knowledge Bases with Soft and Hard Rules. VLDS 2012
- Mohamed Yahya, Martin Theobald: D2R2: Disk-Oriented Deductive Reasoning in a RISC-Style RDF Engine. RuleML 2011
- Timm Meiser, Maximilian Dylla, Martin Theobald: Interactive Reasoning in Uncertain RDF Knowledge Bases. CIKM 2011
- Ndapandula Nakashole, Martin Theobald, Gerhard Weikum: Scalable Knowledge Harvesting with High Precision and High Recall. WSDM 2011
- Maximilian Dylla, Mauro Sozio, Martin Theobald: Resolving Temporal Conflicts in Inconsistent RDF Knowledge Bases. BTW 2011
- Ndapandula Nakashole, Martin Theobald, Gerhard Weikum: Find your Advisor: Robust Knowledge Gathering from the Web. WebDB 2010
- Anish Das Sarma, Martin Theobald, Jennifer Widom: LIVE: A Lineage-Supported Versioned DBMS. SSDBM 2010
- Anish Das Sarma, Martin Theobald, Jennifer Widom: Exploiting Lineage for Confidence Computation in Uncertain and Probabilistic Databases. ICDE 2008
- Omar Benjelloun, Anish Das Sarma, Alon Y. Halevy, Martin Theobald, Jennifer Widom: Databases with uncertainty and lineage. VLDB J. 17(2), 2008