Introduction to Probabilistic Data Management

Evgeny Kharlamov   Pierre Senellart

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

TELECOM ParisTech

Bases de données avancées, October 26, 2011

Part I: Uncertainty in the Real World

Uncertain data

Numerous sources of uncertain data:

- Measurement errors
- Data integration from contradicting sources
- Imprecise mappings between heterogeneous schemata
- Imprecise automatic process (information extraction, natural language processing, etc.)
- Imperfect human judgment

Uncertain data

Numerous sources of uncertain data:

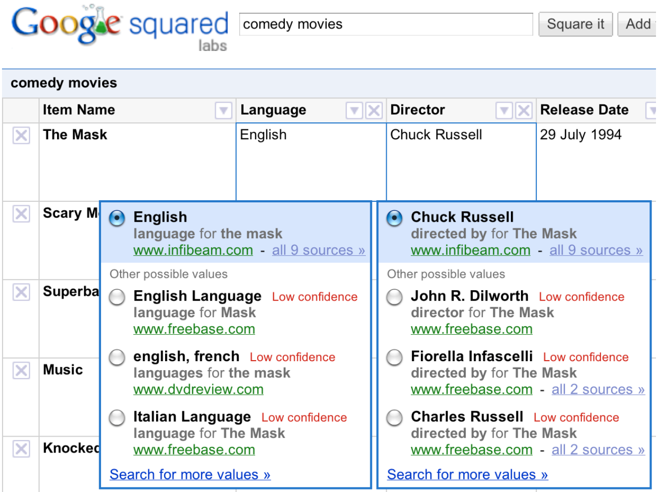
- Measurement errors
- Data integration from contradicting sources
- Imprecise mappings between heterogeneous schemata
- Imprecise automatic process (information extraction, natural language processing, etc.)
- Imperfect human judgment

Use case: Web information extraction

instance	iteration	date learned	confidence
arabic, egypt	406	08-sep-2011	(Seed) 100.0
chinese, republic_of_china	439	24-oct-2011	100.0
chinese, singapore	421	21-sep-2011	(Seed) 100.0
english, britain	439	24-oct-2011	100.0
english, canada	439	24-oct-2011	(Seed) 100.0
english, england001	439	24-oct-2011	100.0
arabic, morocco	422	23-sep-2011	100.0
cantonese, hong_kong	406	08-sep-2011	100.0
english, uk	436	19-oct-2011	100.0
english, south_vietnam	427	27-sep-2011	99.9
french, morocco	422	23-sep-2011	99.9
greek, turkey	430	07-oct-2011	99.9

Never-ending Language Learning (NELL, CMU),
`http://rtw.ml.cmu.edu/rtw/kbbrowser/`

Use case: Web information extraction



Google Squared (terminated), screenshot from [Fink et al., 2011]

Use case: Web information extraction

Subject	Predicate	Object	Confidence
Elvis Presley	diedOnDate	1977-08-16	97.91%
Elvis Presley	isMarriedTo	Priscilla Presley	97.29%
Elvis Presley	influences	Carlo Wolff	96.25%

YAGO, `http://www.mpi-inf.mpg.de/yago-naga/yago`

Uncertainty in Web information extraction

- The information extraction system is imprecise
- The system has some confidence in the information extracted, which can be:
  - a probability of the information being true (e.g., conditional random fields)
  - an ad-hoc numeric confidence score
  - a discrete level of confidence (low, medium, high)
- What if this uncertain information is not seen as something final, but is used as a source of, e.g., a query answering system?

Different types of uncertainty

Two dimensions:

- Different types:
  - Unknown value: NULL in an RDBMS
  - Alternative between several possibilities: either A or B or C
  - Imprecision on a numeric value: a sensor gives a value that is an approximation of the actual value
  - Confidence in a fact as a whole: cf. information extraction
  - Structural uncertainty: the schema of the data itself is uncertain
- Qualitative (NULL) or Quantitative (95%, low-confidence, etc.) uncertainty

Managing uncertainty

Objective

Not to pretend this imprecision does not exist, and manage it as rigorously as possible throughout a long, automatic and human, potentially complex, process.

Managing uncertainty

Objective

Not to pretend this imprecision does not exist, and manage it as rigorously as possible throughout a long, automatic and human, potentially complex, process.

Especially:

- Represent all different forms of uncertainty
- Use probabilities to represent quantitative information on the confidence in the data
- Query data and retrieve uncertain results
- Allow adding, deleting, modifying data in an uncertain way
- Bonus (if possible): Keep as well lineage/provenance information, so as to ensure traceability

Why probabilities?

- Not the only option: fuzzy set theory [Galindo et al., 2005], Dempster-Shafer theory [Zadeh, 1986]
- Mathematically rich theory, nice semantics with respect to traditional database operations (e.g., joins)
- Some applications already generate probabilities (e.g., statistical information extraction or natural language probabilities)
- In other cases, we “cheat” and pretend that (normalized) confidence scores are probabilities: see this as a first-order approximation

Objective of this tutorial

- Present data models for uncertain data management in general, and probabilistic data management in particular:
  - relational
  - XML
- Show how these models can be queried: algorithms, complexity, approximation techniques…
- Discuss the problem of updating a probabilistic database

Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML

Possible worlds semantics

Possible world: A regular (deterministic) relational or XML database

Incomplete database: (Compact) representation of a set of possible worlds

Probabilistic database: (Compact) representation of a probability distribution over possible worlds, either:

- finite: a set of possible worlds, each with their probability
- continuous: more complicated, requires defining a σ-algebra, and a measure for the sets of this σ-algebra

Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML

The relational model

- Data stored into tables
- Every table has a precise schema (type of columns)
- Adapted when the information is very structured

Patient	Examin. 1	Examin. 2	Diagnosis
A	23	12	α
B	10	23	β
C	2	4	γ
D	15	15	α
E	15	17	β

Codd tables, a.k.a. SQL NULLs

Patient	Examin. 1	Examin. 2	Diagnosis
A	23	12	α
B	10	23	$\perp_1$
C	2	4	γ
D	15	15	$\perp_2$
E	$\perp_3$	17	β

- Most simple form of incomplete database
- Widely used in practice, in DBMS since the mid-1970s!
- All NULLs ($\perp$) are considered distinct
- Possible world semantics: all (infinitely many under the open world assumption) possible completions of the table
- In SQL, three-valued logic, weird semantics:

```
SELECT * FROM Tel WHERE tel_nr = '333' OR tel_nr <> '333'
```

C-tables [Imieliński and Lipski, 1984]

Patient	Examin. 1	Examin. 2	Diagnosis	Condition
A	23	12	α	
B	10	23	$\perp_1$	
C	2	4	γ	
D	$\perp_2$	15	$\perp_1$	
E	$\perp_3$	17	β	$18 < \perp_3 < \perp_2$

- NULLs are labeled, and can be reused inside and across tuples
- Arbitrary correlations across tuples
- Closed under the relational algebra (Codd tables only closed under projection and union)
- Every set of possible worlds can be represented as a database with c-tables

Tuple-independent databases (TIDs)

Patient	Examin. 1	Examin. 2	Diagnosis	Probability
A	23	12	α	0.9
B	10	23	β	0.8
C	2	4	γ	0.2
C	2	14	γ	0.4
D	15	15	α	0.6
D	15	15	β	0.4
E	15	17	β	0.7
E	15	17	α	0.3

- Allow representation of the confidence in each row of the table
- Impossible to express dependencies across rows
- Very simple model, well understood

Block-independent databases (BIDs)

Patient	Examin. 1	Examin. 2	Diagnosis	Probability
A	23	12	α	0.9
B	10	23	β	0.8
C	2	4	γ	0.2
C	2	14	γ	0.4
D	15	15	β	0.6
D	15	15	α	0.4
E	15	17	β	0.7
E	15	17	α	0.3

- The table has a primary key: tuples sharing a primary key are mutually exclusive (probabilities must sum up to ≤ 1)
- Simple dependencies (exclusion) can be expressed, but not more complex ones

Probabilistic c-tables [Green and Tannen, 2006]

Patient	Examin. 1	Examin. 2	Diagnosis	Condition
A	23	12	α	w_1
B	10	23	β	w_2
C	2	4	γ	w_3
C	2	14	γ	$\neg w_3 \wedge w_4$
D	15	15	β	w_5
D	15	15	α	$\neg w_5 \wedge w_6$
E	15	17	β	w_7
E	15	17	α	$\neg w_7$

- The w_i's are Boolean random variables
- Each w_i has a probability of being true (e.g., $\Pr(w_1) = 0.9$)
- The w_i's are independent
- Any finite probability distribution of tables can be represented using probabilistic c-tables

Two actual PRDBMS: Trio and MayBMS

Two main probabilistic relational DBMS:

Trio [Widom, 2005] Various uncertainty operators: unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations. See example later on.

MayBMS [Koch, 2009] Implementation of the probabilistic c-tables model. In addition, uncertain tables can be constucted using a `REPAIR-KEY` operator, similar to BIDs.

Two actual PRDBMS: Trio and MayBMS

Two m

```
test=# select * from R;
 dummy | weather | ground |  p
-------+---------+--------+------
 dummy | rain    | wet    | 0.35
 dummy | rain    | dry    | 0.05
 dummy | no rain | wet    |  0.1
 dummy | no rain | dry    |  0.5
(4 rows)

test=# create table S as
repair key Dummy in R weight by P;
SELECT
test=# select Ground, conf() from S group by Ground;
 ground | conf
--------+------
 dry    | 0.55
 wet    | 0.45
(2 rows)
```

own

ible

ter on.

Ma

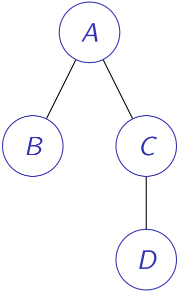
bles

d using

Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML

The semistructured model and XML




```
<a>
  <b>...</b>
  <c>
    <d>...</d>
  </c>
</a>
```

- Tree-like structuring of data
- No (or less) schema constraints
- Allow mixing tags (structured data) and text (unstructured content)
- Particularly adapted to tagged or heterogeneous content

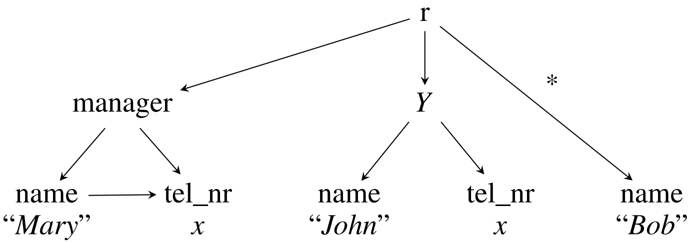
Why Probabilistic XML?

- Extensive literature about probabilistic relational databases [Dalvi et al., 2009, Widom, 2005, Koch, 2009]
- Different typical querying languages: conjunctive queries vs tree-pattern queries (possibly with joins)
- Cases where a tree-like model might be appropriate:
  - No schema or few constraints on the schema
  - Independent modules annotating freely a content warehouse
  - Inherently tree-like data (e.g., mailing lists, parse trees) with naturally occurring queries involving the descendant axis

Remark

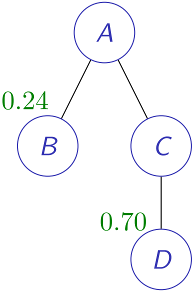
Some results can be transferred from one model to the other. In other cases, connection much trickier!

Incomplete XML [Barceló et al., 2009]



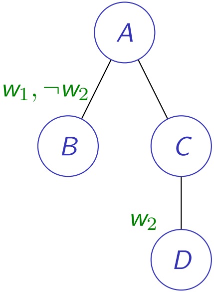
- Models all XML documents where these patterns exist (i.e., this subtree can be matched)
- Can be used for query answering, etc.

Simple probabilistic annotations



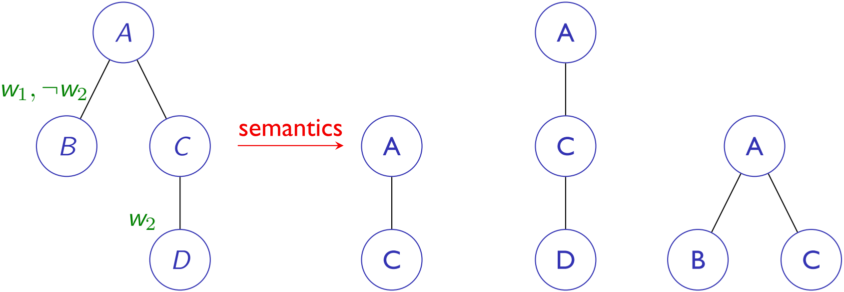
- Probabilities associated to tree nodes
- Express parent/child dependencies
- Impossible to express more complex dependencies
- $\Rightarrow$ some sets of possible worlds are not expressible this way!

Annotations with event variables



Event	Prob.
w_1	0.8
w_2	0.7

Annotations with event variables



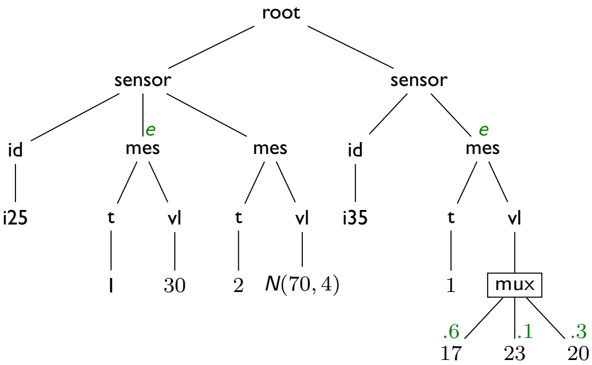
Event	Prob.
w_1	0.8
w_2	0.7

$p_1 = 0.06$ $p_2 = 0.70$ $p_3 = 0.24$

- Expresses arbitrarily complex dependencies
- Obviously, analogous to probabilstic c-tables

A general probabilistic XML model

[Abiteboul et al., 2009]

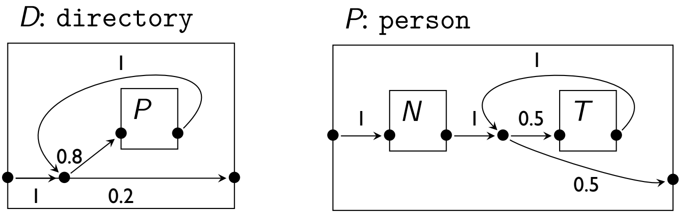


- *e*: event “it did not rain” at time I
- mux: mutually exclusive options
- $N(70, 4)$: normal distribution

- Compact representation of a set of possible worlds
- Two kinds of dependencies: global (*e*) and local (mux)
- Generalizes all previously proposed models of the literature

Recursive Markov chains [Benedikt et al., 2010]

```
<!ELEMENT directory (person*)>
<!ELEMENT person (name,phone*)>
```



- Probabilistic model that extends PXML with local dependencies
- Allows generating documents of unbounded width or depth

Part III: Querying Probabilistic Databases

- Semantics and goals
- Queries over relational probabilistic DBs
- Queries over XML probabilistic DBs

Semantics Of Query Answering: Example

Person

name	city	probability
Ivan	Moscow	0.3
Jean	Paris	0.8
Pedro	Madrid	0.4

Query:
SELECT name FROM Person

Semantics Of Query Answering: Example

Person

name	city	probability
Ivan	Moscow	0.3
Jean	Paris	0.8
Pedro	Madrid	0.4

Query:
SELECT name FROM Person

Pr = 0.3*0.8*0.4

name	city
Ivan	Moscow
Jean	Paris
Pedro	Madrid

Pr = 0.3*0.2*0.4

name	city
Ivan	Moscow
Pedro	Madrid

...

Semantics Of Query Answering: Example

Person

name	city	probability
Ivan	Moscow	0.3
Jean	Paris	0.8
Pedro	Madrid	0.4

Query:
SELECT name FROM Person

Pr = 0.3*0.8*0.4

name	city
Ivan	Moscow
Jean	Paris
Pedro	Madrid

Pr = 0.3*0.2*0.4

name	city
Ivan	Moscow
Pedro	Madrid

...

Possible answers: ({Ivan, Juan, Pedro}, 0.3*0.8*0.4),
({Ivan, Pedro}, 0.3*0.2*0.4), ...

Possible tuples: (Ivan, 0.3), (Jean, 0.8), (Pedro, 0.4)

Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Possible Tuples Semantics

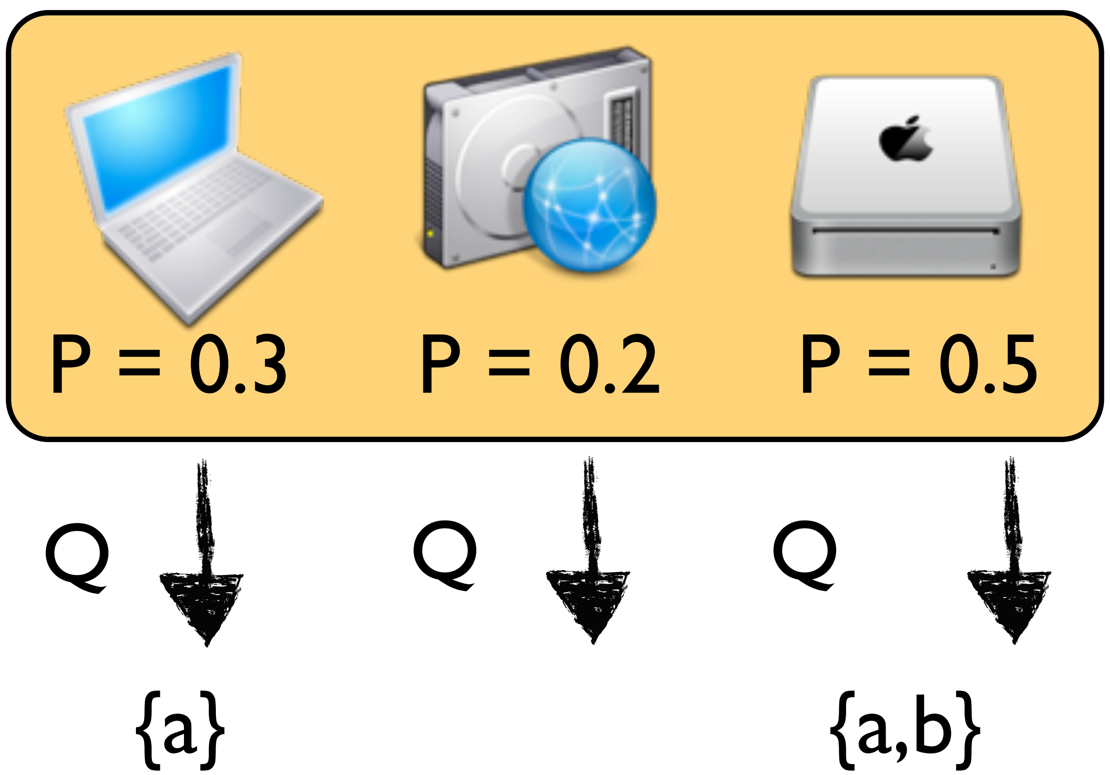
Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Semantics Of Query Answering

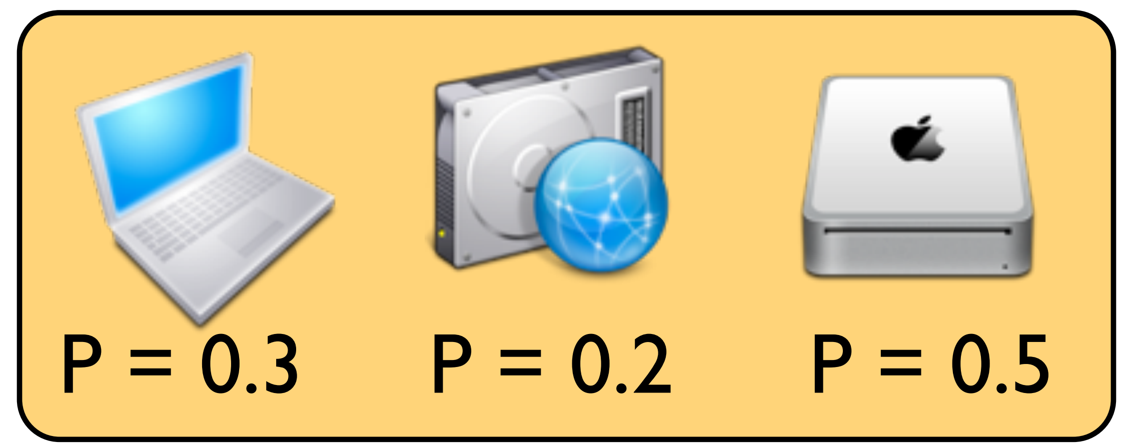
Possible Answers Semantics

Probabilistic DB:



Possible Tuples Semantics

Probabilistic DB:



Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Q Q Q

{a} {a,b}

Answer: ({a}, 0.3); ({a,b}, 0.5)

Possible Tuples Semantics

Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Q Q Q

{a} {a,b}

Answer: ({a}, 0.3); ({a,b}, 0.5)

Probability distribution on sets of tuples

Possible Tuples Semantics

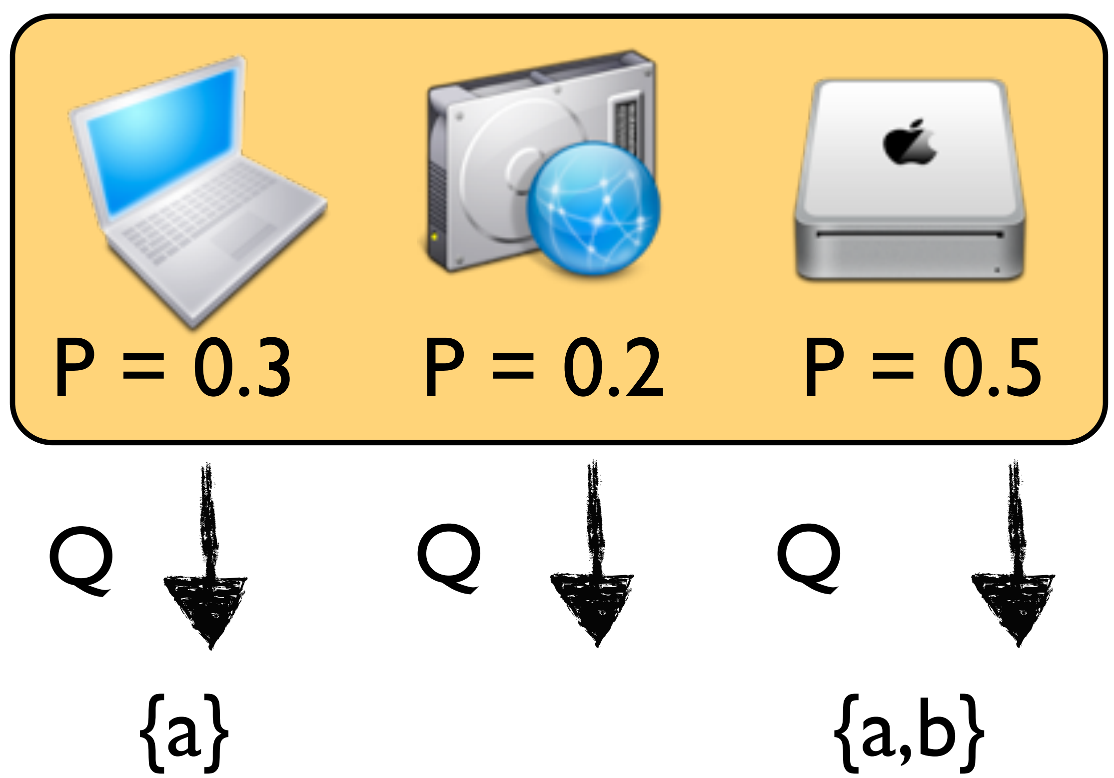
Probabilistic DB:

P = 0.3 P = 0.2 P = 0.5

Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

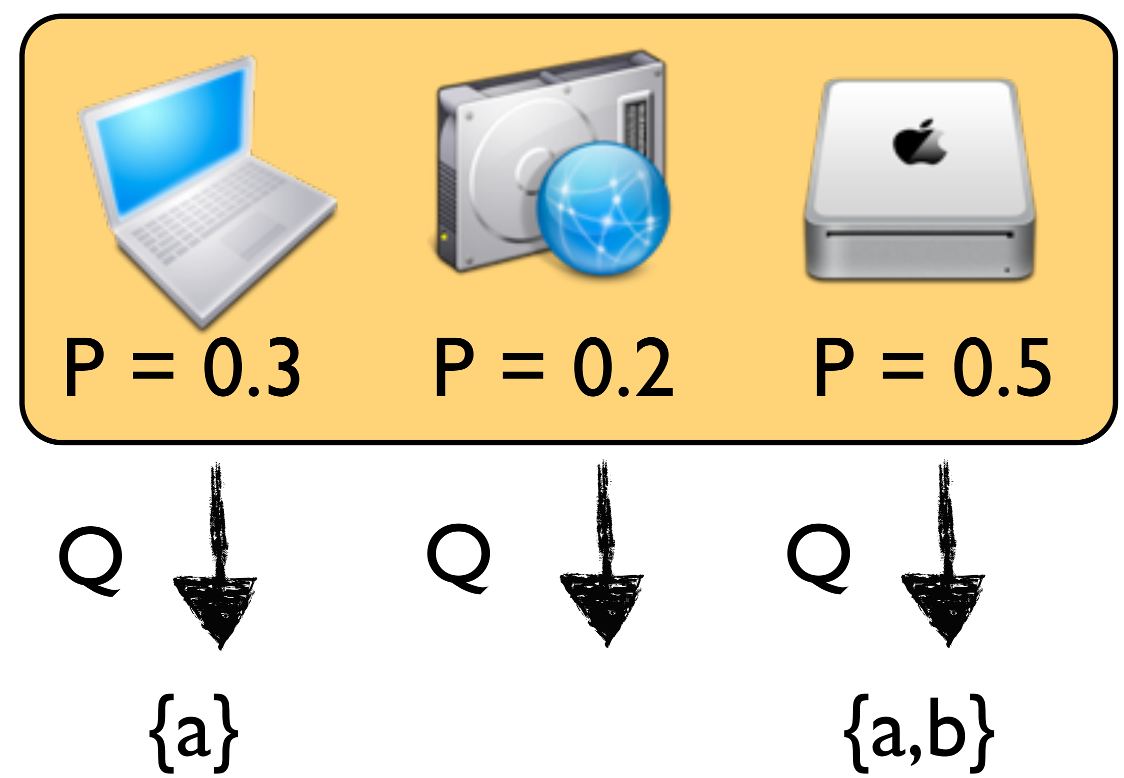


Answer: ({a}, 0.3); ({a,b}, 0.5)

Probability distribution on
sets of tuples

Possible Tuples Semantics

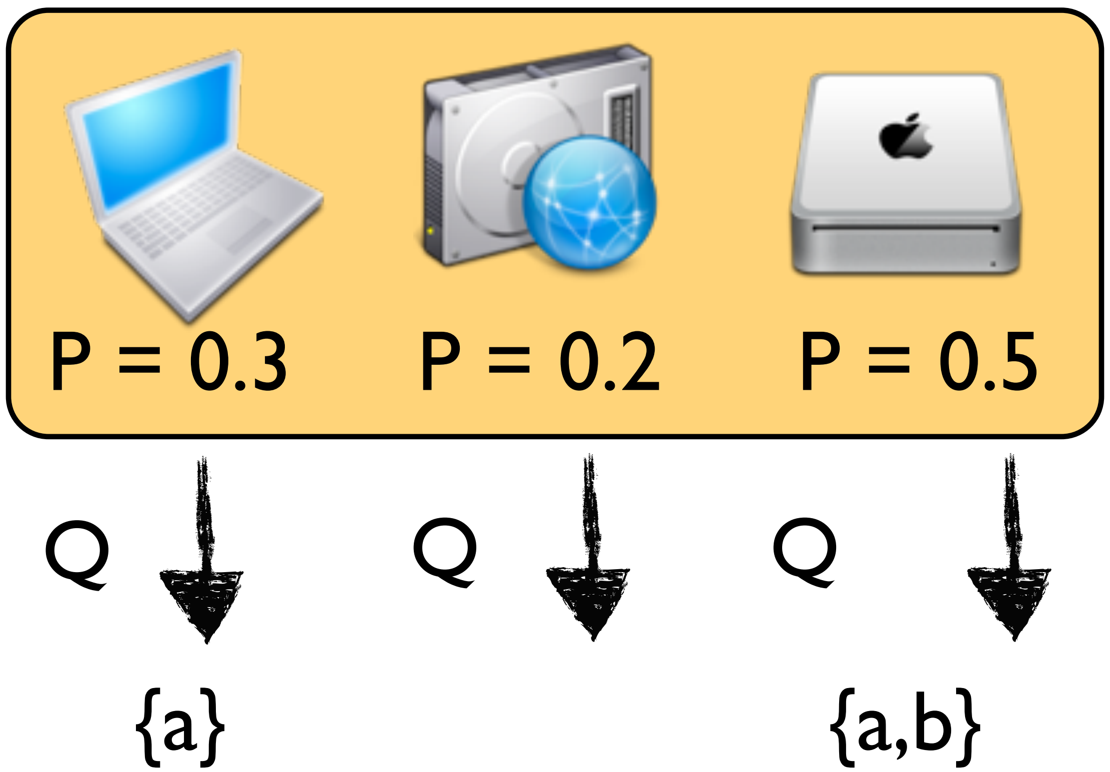
Probabilistic DB:



Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

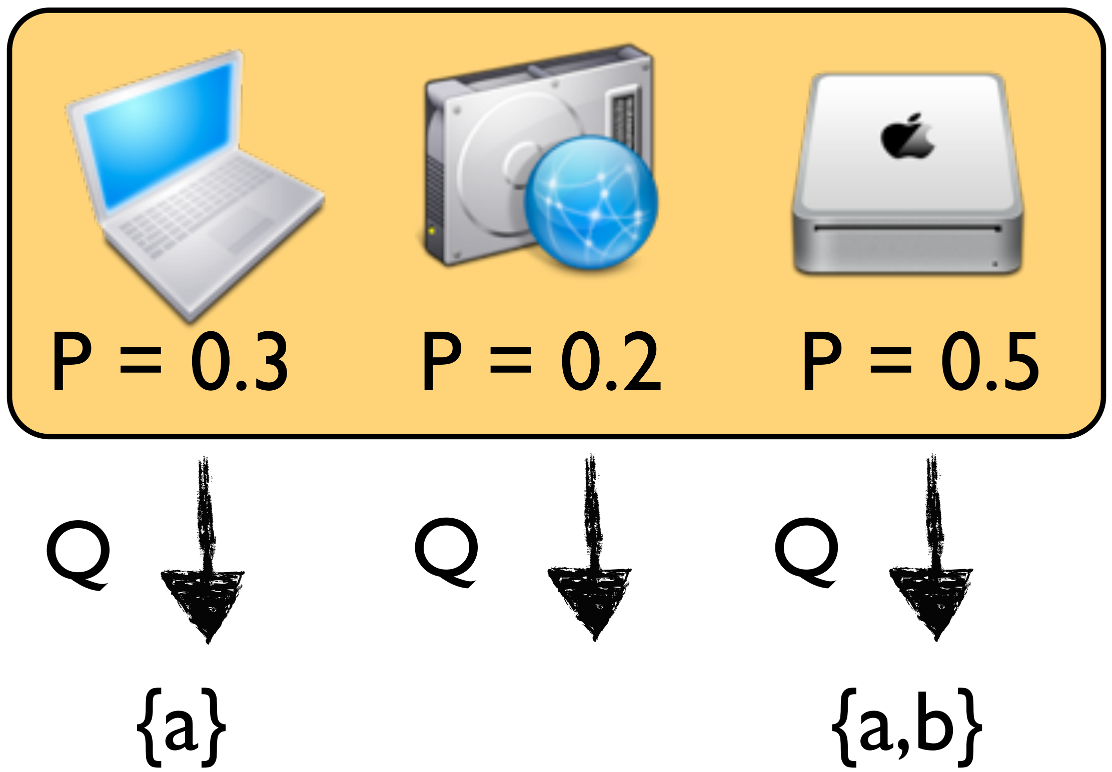


Answer: ({a}, 0.3); ({a,b}, 0.5)

Probability distribution on sets of tuples

Possible Tuples Semantics

Probabilistic DB:

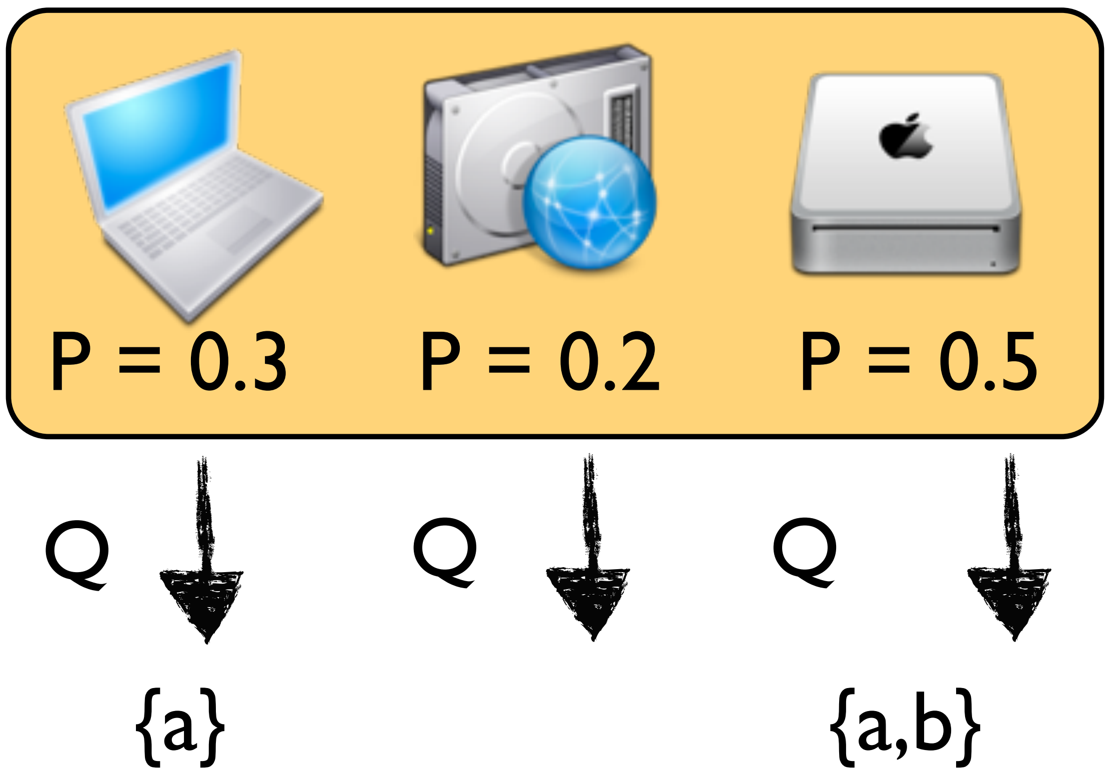


Answer: (a, 0.8), (b, 0.5)

Semantics Of Query Answering

Possible Answers Semantics

Probabilistic DB:

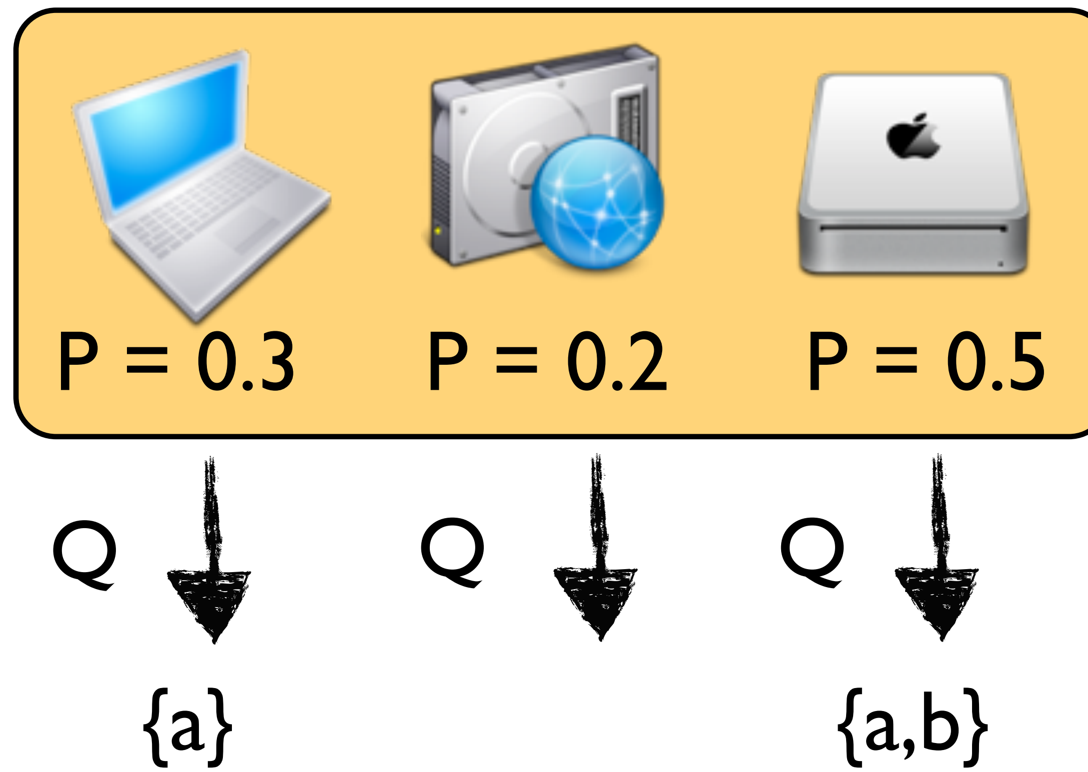


Answer: ({a}, 0.3); ({a,b}, 0.5)

Probability distribution on sets of tuples

Possible Tuples Semantics

Probabilistic DB:



Answer: (a, 0.8), (b, 0.5)

Probability distribution on tuples

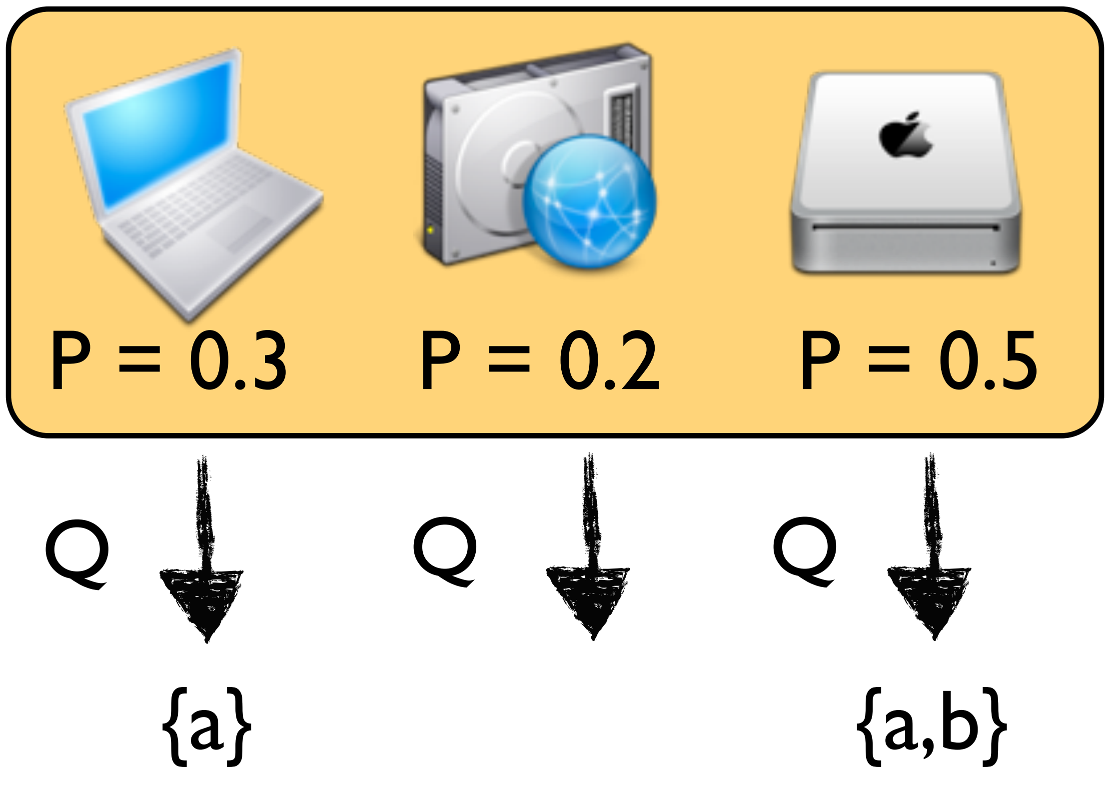
Possible Answer vs Possible Tuple Semantics

[Dalvi,Suciu'09]

- Possible answers semantics:
  - Precise
  - Can be used to compose queries
  - Difficult user interface
- Possible tuples semantics:
  - Less precise, but simple; sufficient for most apps
  - Cannot be used to compose queries
  - Simple user interface

Goals of Query Answering

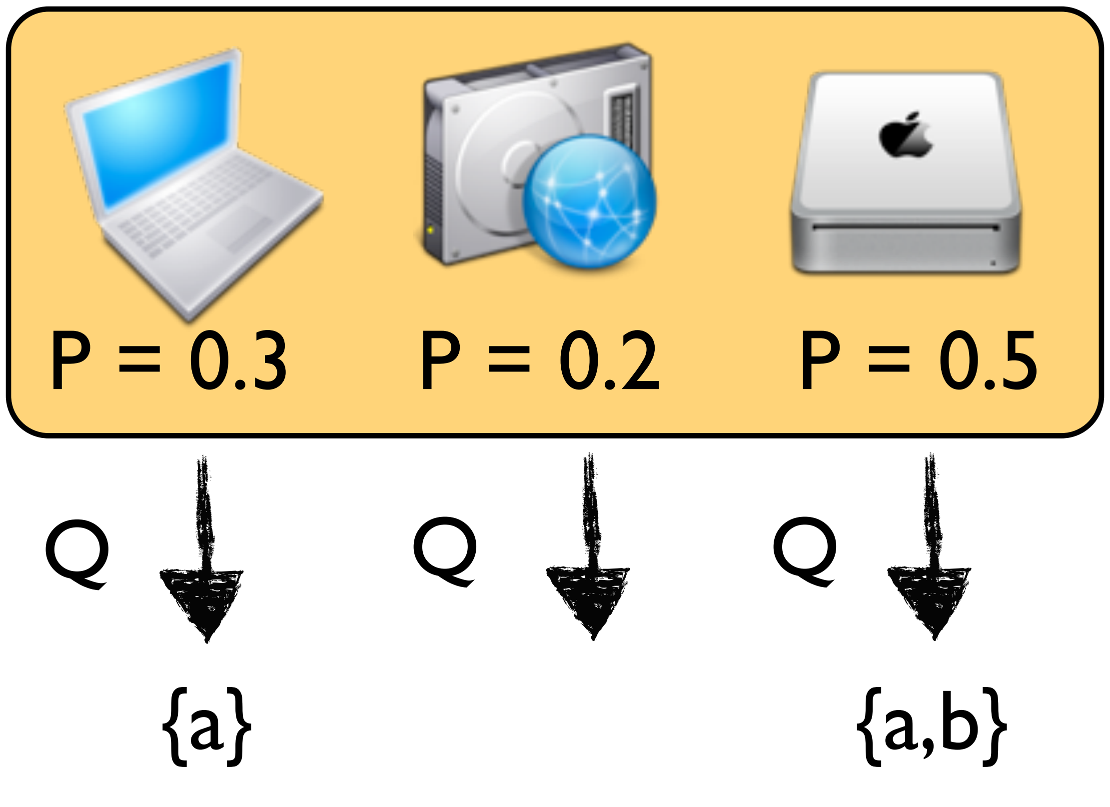
Probabilistic DB:



Answer: (a, 0.8), (b, 0.5)

Goals of Query Answering

Probabilistic DB:

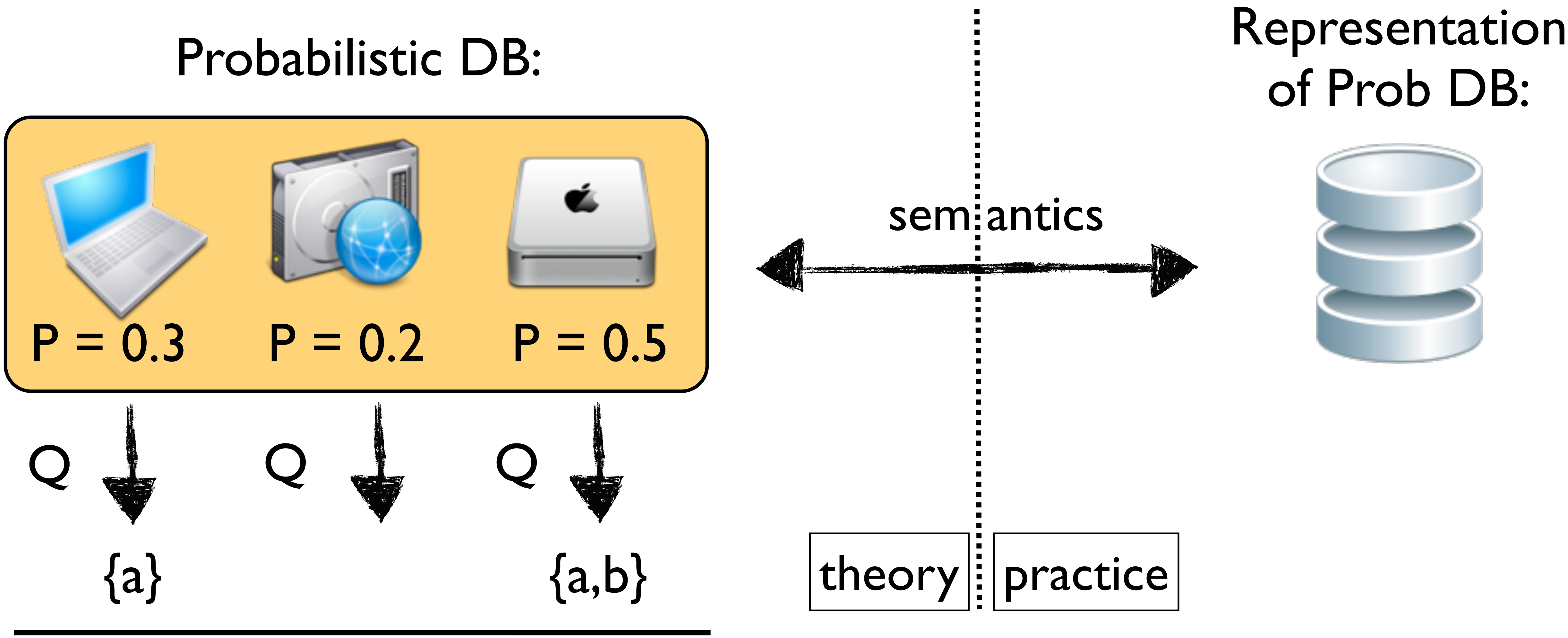


Answer: (a, 0.8), (b, 0.5)

theory | practice

- There may be EXP many worlds ➜ naive evaluation is exponential
- Can we do better?

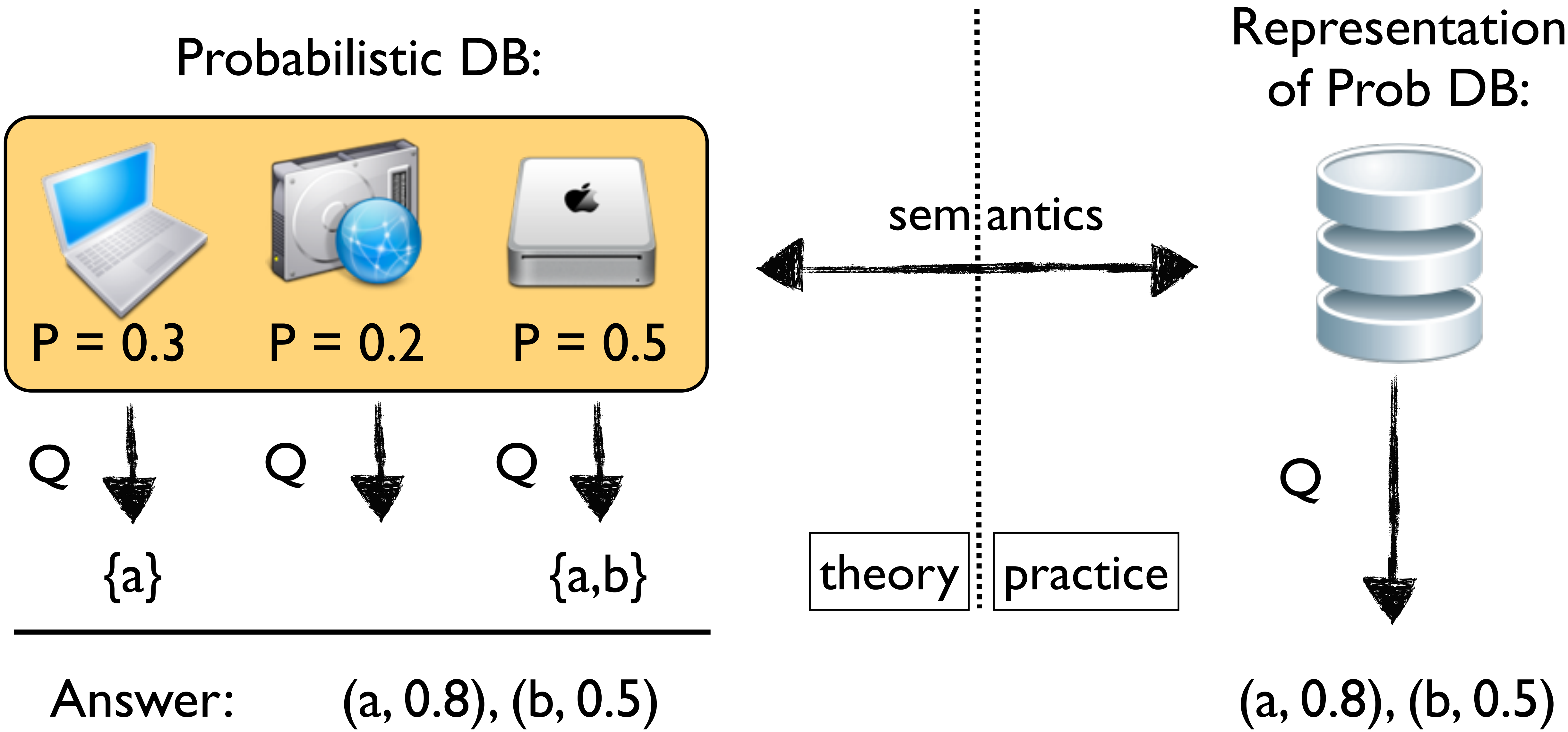
Goals of Query Answering



Answer: (a, 0.8), (b, 0.5)

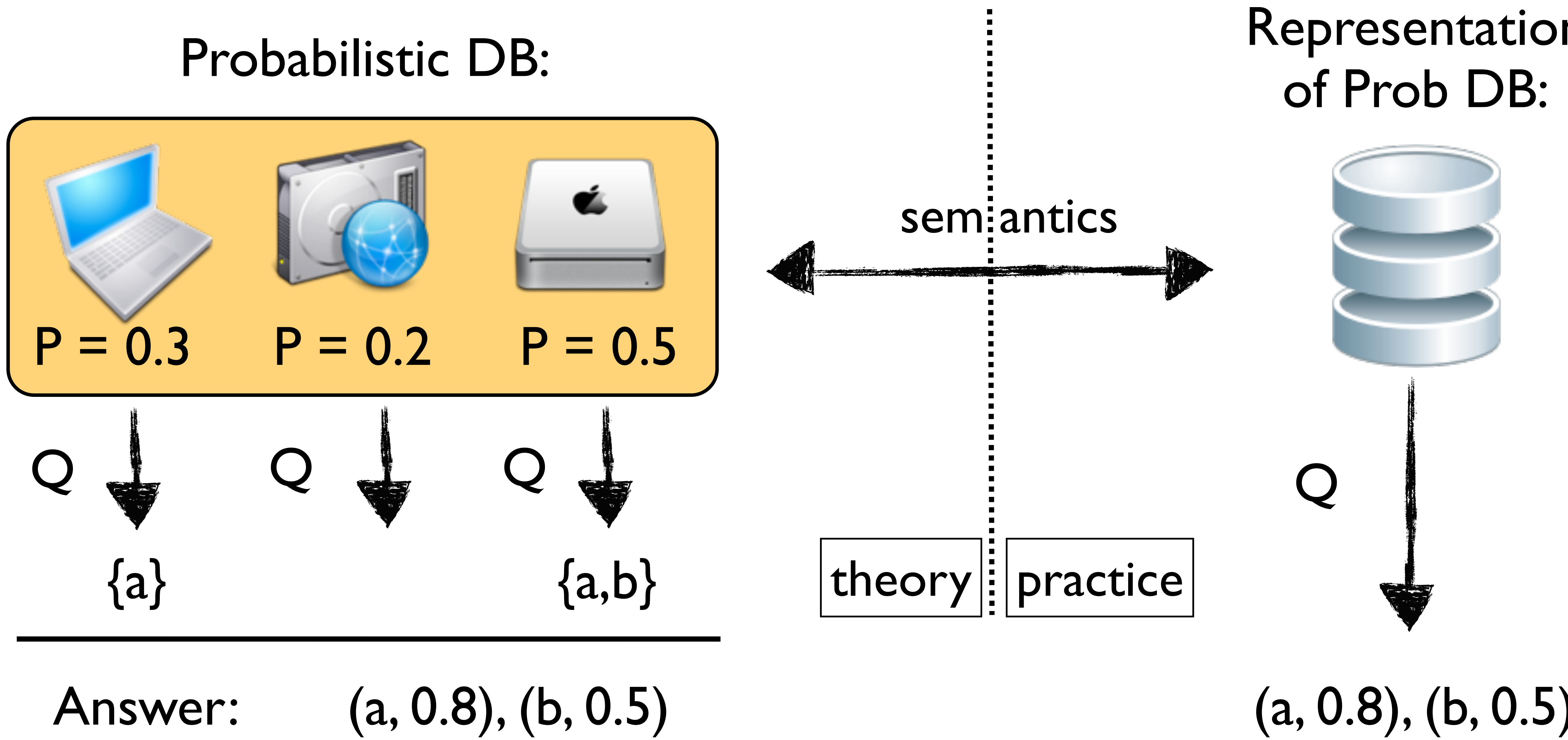
- There may be EXP many worlds ➜ naive evaluation is exponential
- Can we do better?

Goals of Query Answering



- There may be EXP many worlds ➜ naive evaluation is exponential
- Can we do better?

Goals of Query Answering



- There may be EXP many worlds ➜ naive evaluation is exponential
- Can we do better?
- Goal: to find out how to query representation system directly

Part III: Querying Probabilistic Databases

- Semantics and goals
- Queries over relational probabilistic DBs
  - Queries in Trio, MayBMS, and Mystiq
  - Query lineage
  - Approximate query evaluation
- Queries over XML probabilistic DBs

Part III: Querying Probabilistic Databases

- Semantics and goals
- Queries over relational probabilistic DBs
  - Queries in Trio, MayBMS, and Mystiq
  - Query lineage
  - Approximate query evaluation
- Queries over XML probabilistic DBs

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Suspects

41	Jimmy
42	Billy || Frank
43	Hank

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

Suspects = π_{person}(Saw ⋈ Drives)

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Correlations are missing.
It's a wrong representation

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

Suspects = π_{person}(Saw ⋈ Drives)

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

Correlations are missing.
It's a wrong representation

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Correlations are missing.
It's a wrong representation

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Correlations are missing.
It's a wrong representation

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

Suspects = π_{person}(Saw ⋈ Drives)

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Correlations are missing.
It's a wrong representation

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

Suspects = π_{person}(Saw ⋈ Drives)

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

Correlations are missing.
It's a wrong representation

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

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

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

This is the right representation

Query Answering in Trio

[Widom'05]
[Benjelloun&al'06]

Saw

ID	witness	car
21	Cathy	Honda || Mazda

Drivers

ID	person	car
31	Jimmy	Toyota || Mazda
32	Billy || Frank	Honda
33	Hank	Honda

Suspects = π_{person}(Saw ⋈ Drives)

Possible answers semantics

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

This is the right representation

Lineage

Suspects

41	Jimmy	?
42	Billy || Frank	?
43	Hank	?

Lineage:

L(41) = (21,2), (31,2)

L(42,1) = (21,1), (32,1); L(42,2) = (21,1), (32,2)

L(43) = (21,1), (33)

- Lineage or provenance:
  - Meat to explain where the data comes from
  - Internal lineage: comes from data itself
  - External lineage: someone tells us
- Without lineage Trio system is not closed under queries (as we saw on the previous example)

Trio Data Model with Lineage

- Uncertainty-Lineage Databases: ULDBs
  - Alternatives
  - '?' (Maybe) Annotations
  - Confidences
  - Lineage

General Lineage: Examples of Operators (1)

Drivers

ID	person	car	Lineage
31	Jimmy	Toyota	$x \wedge y$
32	Jimmy	Honda	y
33	Hank	Honda	$x \vee z$

Saw

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

Project = π_{person} (Drives)

Select = $\sigma_{\text{car="honda"}}$ (Drives)

Project

person	Lineage
Jimmy	$(x \wedge y) \vee y$
Hank	$x \vee z$

Select

person	car	Lineage
Jimmy	Honda	y
Hank	Honda	$x \vee z$

General Lineage: Examples of Operators (1)

Drivers

ID	person	car	Lineage
31	Jimmy	Toyota	$x \wedge y$
32	Jimmy	Honda	y
33	Hank	Honda	$x \vee z$

Saw

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2   Pr(z is true) = 0.8
Pr(y is true) = 0.4   Pr(w is true) = 0.5

Project = π_{person} (Drives)

Select = $\sigma_{\text{car="honda"}}$ (Drives)

Project

person	Lineage
Jimmy	$(x \wedge y) \vee y$
Hank	$x \vee z$

Select

person	car	Lineage
Jimmy	Honda	y
Hank	Honda	$x \vee z$

General Lineage: Examples of Operators (2)

Drivers

ID	person	car	Lineage
31	Jimmy	Toyota	$x \wedge y$
32	Jimmy	Honda	y
33	Hank	Honda	$x \vee z$

Saw

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2   Pr(z is true) = 0.8
Pr(y is true) = 0.4   Pr(w is true) = 0.5

Join = Saw $\bowtie_{car}$ Drives

Several = $\pi_{person}(\sigma_{person=\text{"Hank"}}$ (Saw $\bowtie_{car}$ Drives)

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

Several

person	Lineage
Hank	$(x \vee z) \wedge w$

General Lineage: Examples of Operators (2)

Drivers

ID	person	car	Lineage
31	Jimmy	Toyota	$x \wedge y$
32	Jimmy	Honda	y
33	Hank	Honda	$x \vee z$

Saw

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2   Pr(z is true) = 0.8
Pr(y is true) = 0.4   Pr(w is true) = 0.5

Join = Saw $\bowtie_{car}$ Drives

Several = $\pi_{person}(\sigma_{person=\text{"Hank"}}$ (Saw $\bowtie_{car}$ Drives)

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

Several

person	Lineage
Hank	$(x \vee z) \wedge w$

General Lineage: Examples of Operators (3)

Saw-day

ID	witness	car	Lineage
31	Cathy	Honda	z
32	Bob	BMW	$y \wedge w$

Saw-night

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2   Pr(z is true) = 0.8
Pr(y is true) = 0.4   Pr(w is true) = 0.5

Union = Saw-day ∪ Saw-night

Difference = Saw-day \ Saw-night

Union

witness	car	Lineage
Cathy	Honda	$z \vee w$
Bob	BMW	$y \wedge w$

Difference

witness	car	Lineage
Cathy	Honda	$z \wedge (\neg w)$
Bob	BMW	$y \wedge w$

General Lineage: Examples of Operators (3)

Saw-day

ID	witness	car	Lineage
31	Cathy	Honda	z
32	Bob	BMW	$y \wedge w$

Saw-night

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2 Pr(z is true) = 0.8

Pr(y is true) = 0.4 Pr(w is true) = 0.5

Union = Saw-day ∪ Saw-night

Difference = Saw-day \ Saw-night

Union

witness	car	Lineage
Cathy	Honda	$z \vee w$
Bob	BMW	$y \wedge w$

Difference

witness	car	Lineage
Cathy	Honda	$z \wedge (\neg w)$
Bob	BMW	$y \wedge w$

MayBMS

- Since MayBMS is essentially probabilistic c-tables query evaluation is based on
  - computation of lineage
  - computation of the probability that a tuple to be in the answer is the probability of the tuple's lineage

General Lineage

[Widom'05]

- Types of Lineage:
  - Conjunctive lineage: sufficient for most operations
  - Disjunctive lineage: for duplicate-elimination
  - Negative lineage: for difference
  - Boolean formulas: general case after several queries

Query Probabilities from Lineage

Join = Saw $\bowtie_{car}$ Drives

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

Theorem: SPJUD-query evaluation over PrRBDs with boolean-formulas lineage is #P-hard, i.e. intractable

Query Probabilities from Lineage

Join = Saw $\bowtie_{car}$ Drives

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

- Pr(Jimmy $\in$ (Saw $\bowtie_{car}$ Drives)) = $Pr(y \wedge w)$ = Pr(y) x Pr(w) = 0.4 x 0.5 =0.2

Theorem: SPJUD-query evaluation over PrRBDs with boolean-formulas lineage is #P-hard, i.e. intractable

Query Probabilities from Lineage

Join = Saw $\bowtie_{car}$ Drives

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

- Pr(Jimmy $\in$ (Saw $\bowtie_{car}$ Drives)) = $Pr(y \wedge w) = Pr(y) \times Pr(w) = 0.4 \times 0.5 = 0.2$
- Pr(Hank $\in$ (Saw $\bowtie_{car}$ Drives)) = $Pr((x \vee z) \wedge w))$

$$= Pr(x \vee z) \times Pr(w)$$
$$= [\, Pr(x) + Pr(z) - Pr(x \wedge z)\,] \times 0.5$$
$$= [\, Pr(x) + Pr(z) - Pr(x) \times Pr(z)\,] \times 0.5$$
$$= [\, 0.2 + 0.8 - 0.2 \times 0.8\,] \times 0.5 = 0.42$$

Theorem: SPJUD-query evaluation over PrRBDs with boolean-formulas lineage is #P-hard, i.e. intractable

Query Probabilities from Lineage

Join = Saw $\bowtie_{car}$ Drives

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

Join

person	car	witness	Lineage
Jimmy	Honda	Cathy	$y \wedge w$
Hank	Honda	Cathy	$(x \vee z) \wedge w$

- Pr(Jimmy $\in$ (Saw $\bowtie_{car}$ Drives)) = $Pr(y \wedge w) = Pr(y) \times Pr(w) = 0.4 \times 0.5 = 0.2$
- Pr(Hank $\in$ (Saw $\bowtie_{car}$ Drives)) = $Pr((x \vee z) \wedge w))$

$$= Pr(x \vee z) \times Pr(w)$$
$$= [\ Pr(x) + Pr(z) - Pr(x \wedge z)\ ] \times 0.5$$
$$= [\ Pr(x) + Pr(z) - Pr(x) \times Pr(z)\ ] \times 0.5$$
$$= [\ 0.2 + 0.8 - 0.2 \times 0.8\ ] \times 0.5 = 0.42$$

In general:
Pr(lineage) = Pr(φ)
where φ is a prop. formula

Theorem: SPJUD-query evaluation over PrRBDs with boolean-formulas lineage is #P-hard, i.e. intractable

#P Functions

- Probability computation is a function and not a decision problem
- Usually complexity is studied for decision problems: P(x) = yes/no
- Complexity classes for probability computation are
  for classes of functions
- #P functions: f(x) = n
  - there is a PTIME non-deterministic Turing machine M_f
  - n = the number of accepting runs of M_f on x, i.e., of $M_f(x)$
- #P functions are counting counterparts of NP decision problems
- Example of #P-complete function:
  #2DNF: count number of evaluations for 2DNF propositional formulas
- #P-comp. functions are counter counterparts of NP-comp. problems

Can Queries Evaluation be Easy?

Theorem: SPJUD-query evaluation over PrRBDs with boolean-formulas lineage is #P-hard, i.e. intractable

- This means that evaluation of SQL queries over PrRDBs cannot be efficient in general
- Practical cases?

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

- Conjunctive queries (SPJ):

  e.g. Q(x) :- Person(x) $\wedge$ Works_for(x, "Irish Pub") $\wedge$ Married_to(x,y) $\wedge$ Nurse(x,y)

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

- Conjunctive queries (SPJ):

  e.g. Q(x) :- Person(x) ∧ Works_for(x,"Irish Pub") ∧ Married_to(x,y) ∧ Nurse(x,y)

- Self join: the same predicate occurs more than once:

  Q_Fr(x) :- Friends(x,y) ∧ Works_for(x,"Irish Pub") ∧ Works_for(y,"Temple Bar")

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

- Conjunctive queries (SPJ):
  e.g. Q(x) :- Person(x) $\wedge$ Works_for(x, "Irish Pub") $\wedge$ Married_to(x,y) $\wedge$ Nurse(x,y)

- Self join: the same predicate occurs more than once:
  Q_Fr(x) :- Friends(x,y) $\wedge$ Works_for(x, "Irish Pub") $\wedge$ Works_for(y,"Temple Bar")

- Hierarchical query: $\Sigma(x)$ - predicates where x occur.
  If x, y occur in Q, then $\Sigma(x) \cap \Sigma(y) = \emptyset$, or $\Sigma(x) \subseteq \Sigma(y)$ or $\Sigma(y) \subseteq \Sigma(x)$

  Q_Fr is hierarchical:
  $\Sigma(x)$ = { Friends, Works_for } and $\Sigma(y)$ = { Friends, Works_for }

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

- Conjunctive queries (SPJ):
  e.g. Q(x) :- Person(x) $\wedge$ Works_for(x,"Irish Pub") $\wedge$ Married_to(x,y) $\wedge$ Nurse(x,y)

- Self join: the same predicate occurs more than once:
  Q_Fr(x) :- Friends(x,y) $\wedge$ Works_for(x,"Irish Pub") $\wedge$ Works_for(y,"Temple Bar")

- Hierarchical query: Σ(x) - predicates where x occur.
  If x, y occur in Q, then $\Sigma(x) \cap \Sigma(y) = \emptyset$, or $\Sigma(x) \subseteq \Sigma(y)$ or $\Sigma(y) \subseteq \Sigma(x)$

  Q_Fr is hierarchical:
  Σ(x) = { Friends, Works_for } and Σ(y) = { Friends, Works_for }

Theorem: Computation of probabilities of query answers is polynomial time for queries that are:
- without self-joins, and
- hierarchical

TIDs and Conjunctive Queries

[Dalvi&Suciu'04]

- Conjunctive queries (SPJ):
  e.g. Q(x) …(x,y) ∧ Nurse(x,y)
- Self join: t…
  Q_Fr(x) :- F… "Temple Bar")
- Hierarchi…
  If x, y occ… r Σ(y) ⊆ Σ(x)

  Q_Fr is hie…
  Σ(x) = { Fri…

Hard SPJ query for TIDs:

Q = Person(x) ∧ Works_For(x,y) ∧ Company(y)

- Q without self-joins
- Q is not hierarchical

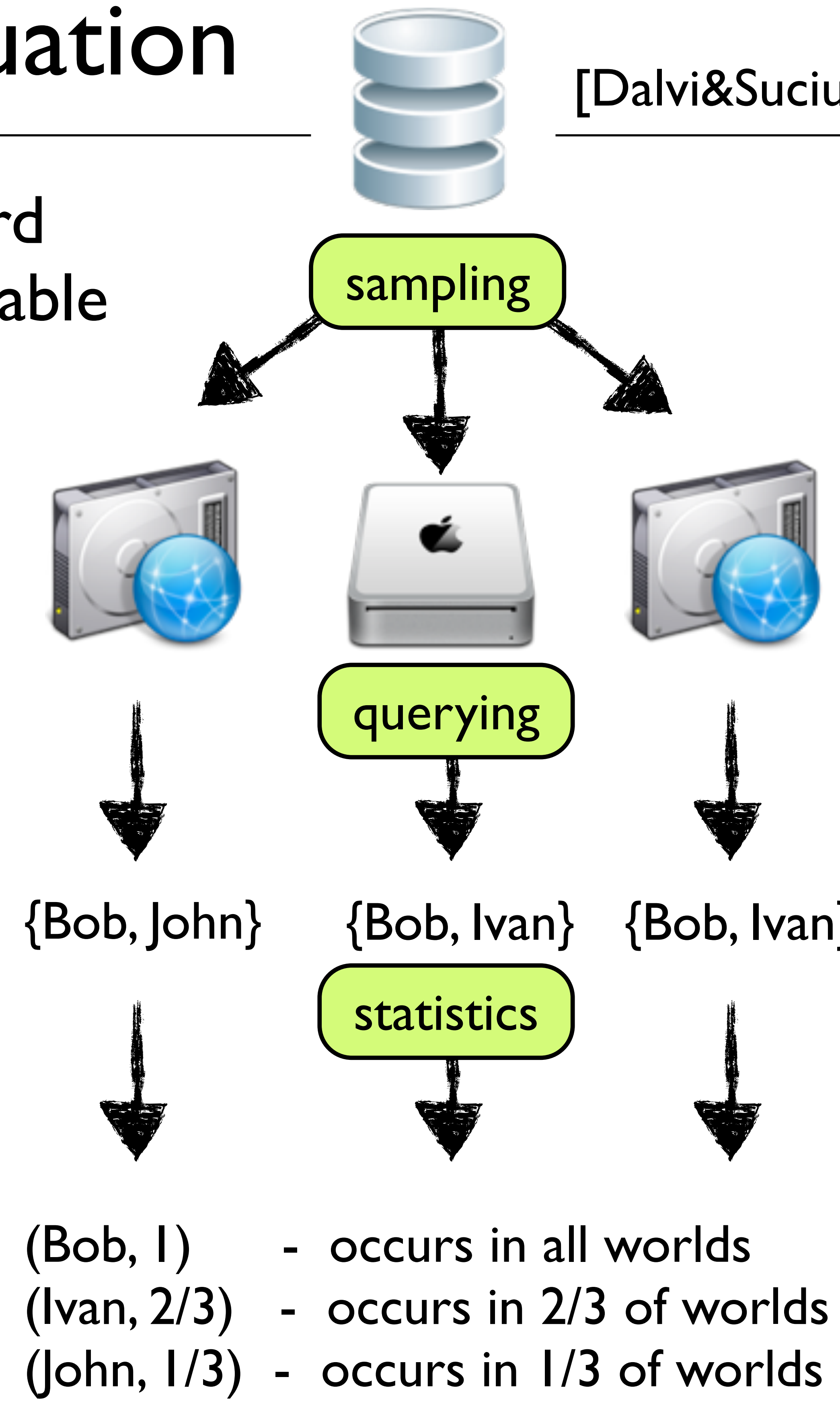
Message:
in most of the cases query evaluation is hard even over a simple TID model

Theorem: Computation of probabilities of query answers is polynomial time for queries that are:
- without self-joins, and
- hierarchical

Approximate Query Evaluation

[Dalvi&Suciu'04]

- In most cases query evaluation is hard
  approximate computation is unavoidable
- Sampling techniques:
  - Given: prob. RDB, query Q
  - Sample DB instances D
  - evaluate Q(D)
  - take all resulting answers ans
  - assign prob.s to ans as
    the frequency of occurrence
- PTIME guarantees for
  additive approximation of
  probabilities



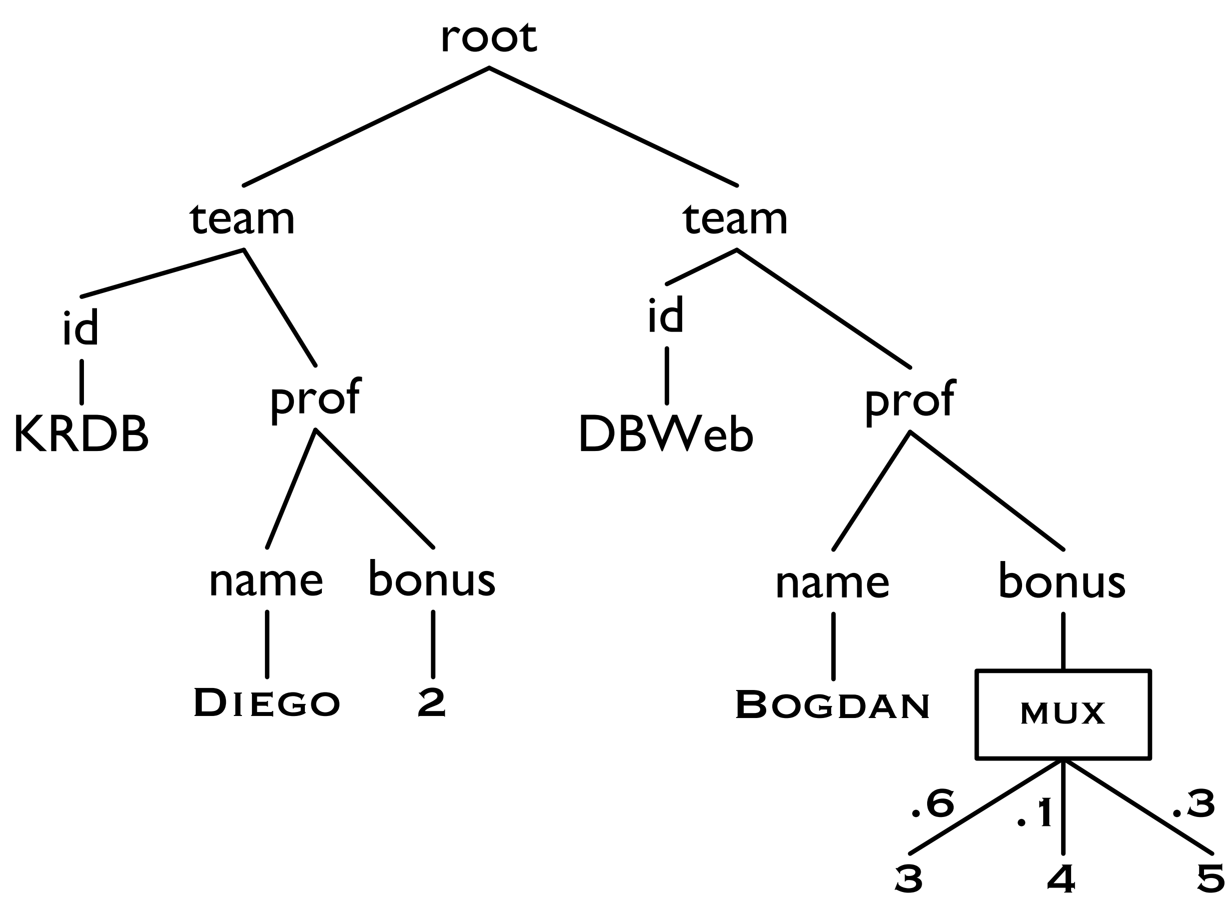
Part III: Querying Probabilistic Databases

- Semantics and goals
- Queries over relational probabilistic DBs
- Queries over XML probabilistic DBs
  - Tree-pattern queries
  - Aggregate queries

Part III: Querying Probabilistic Databases

- Semantics and goals
- Queries over relational probabilistic DBs
- Queries over XML probabilistic DBs
  - Tree-pattern queries
  - Aggregate queries

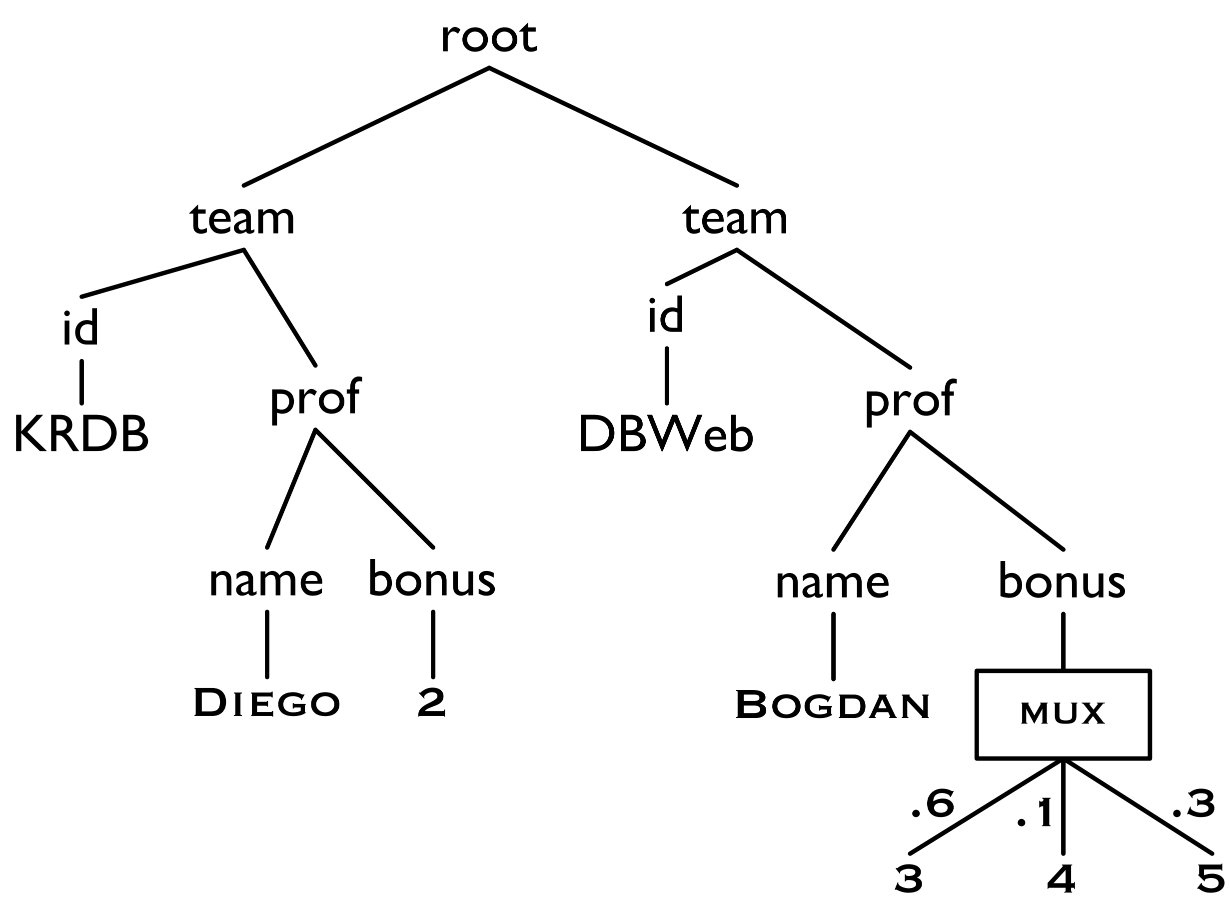
Querying PrXML: Example



Query:

Q: *Is there a bonus of 4?*

Querying PrXML: Example



Query:

Q: *Is there a bonus of 4?*

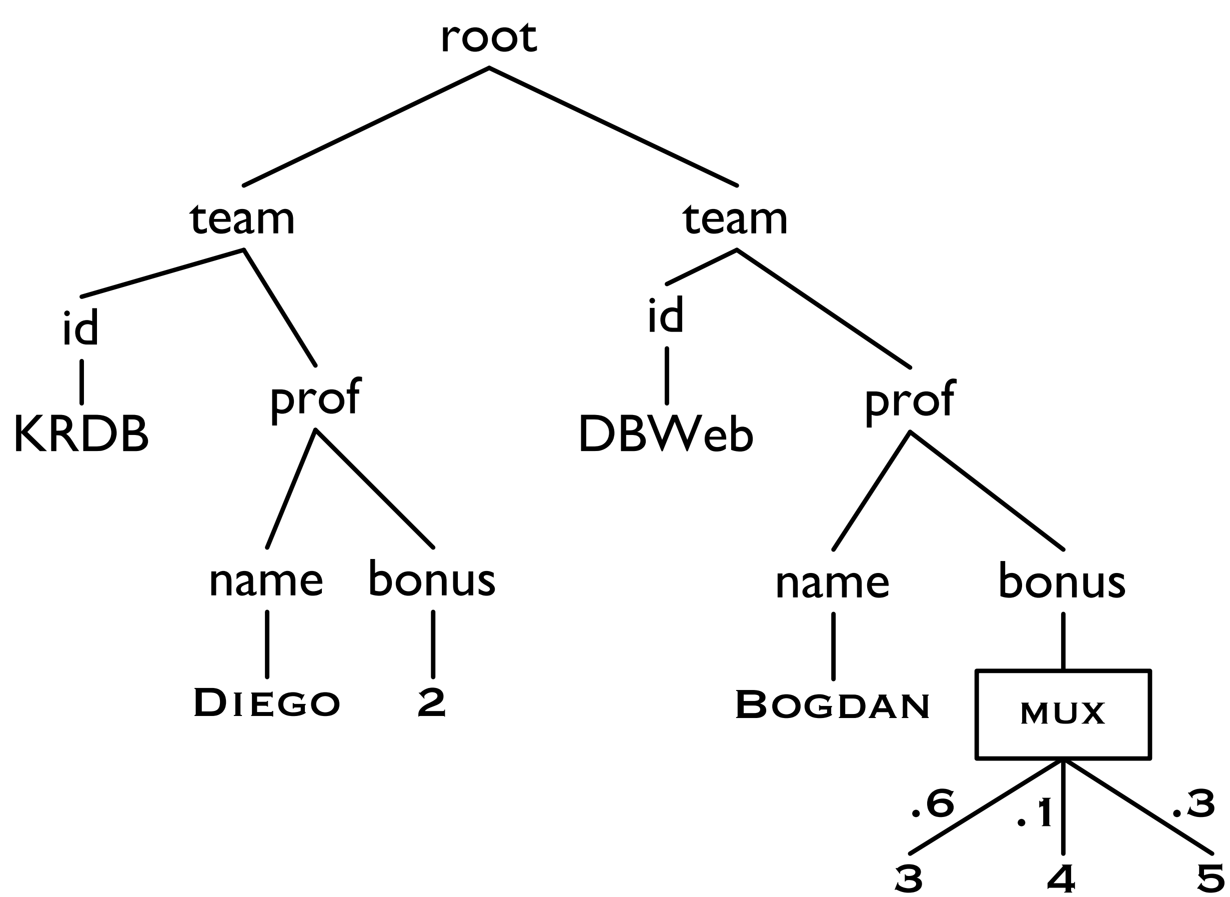
Answers over worlds:

Q(w3) = no, Pr(w3)=0.6

Q(w4) = yes, Pr(w4)=0.1

Q(w5) = no, Pr(w5)=0.3

Querying PrXML: Example



Query:

Q: *Is there a bonus of 4?*

Answers over worlds:

Q(w3) = no, Pr(w3)=0.6

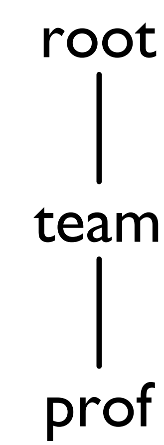
Q(w4) = yes, Pr(w4)=0.1

Q(w5) = no, Pr(w5)=0.3

Query answer over PrXML: { (yes, 0.1), (no, 0.9) }

Tree Pattern Queries

a) Single-Path Queries - SP
Are there professors working for some teams?



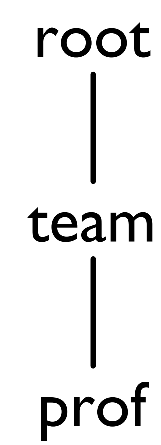
XPath notation:
/root/team/prof

Tree Pattern Queries

a) Single-Path Queries - SP

Are there professors working for some teams?

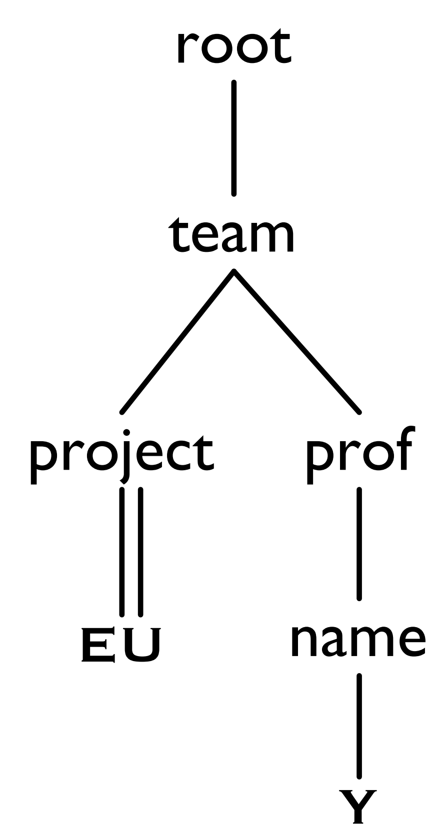
XPath notation:
/root/team/prof



b) Tree-Pattern Queries - TP

Return names of professors working for teams involved in EU projects?

XPath notation:
/root/team[project//EU]/prof/name/*

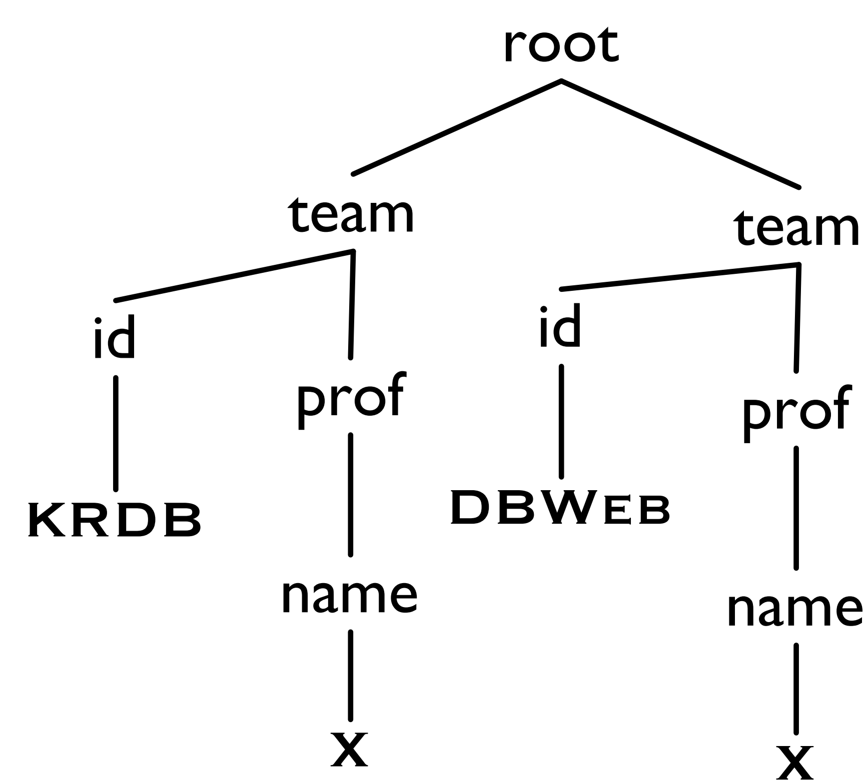


Tree Pattern Queries

c) Tree-Pattern Queries with Joins - TPJ
Are there (names of) professors working for both KRDB and DBWeb?

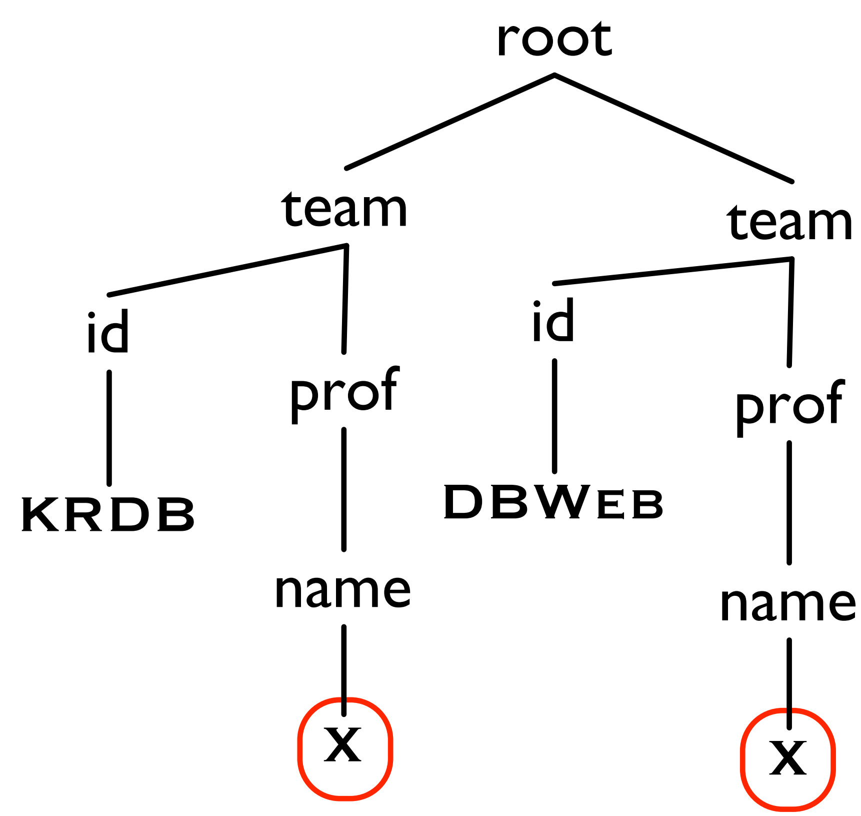
XPath notation:

. //team[id="KRDB"] /prof/name =
. //team[id="DBWeb"]/prof/name



Tree Pattern Queries

c) Tree-Pattern Queries with Joins - TPJ
Are there (names of) professors working for both KRDB and DBWeb?

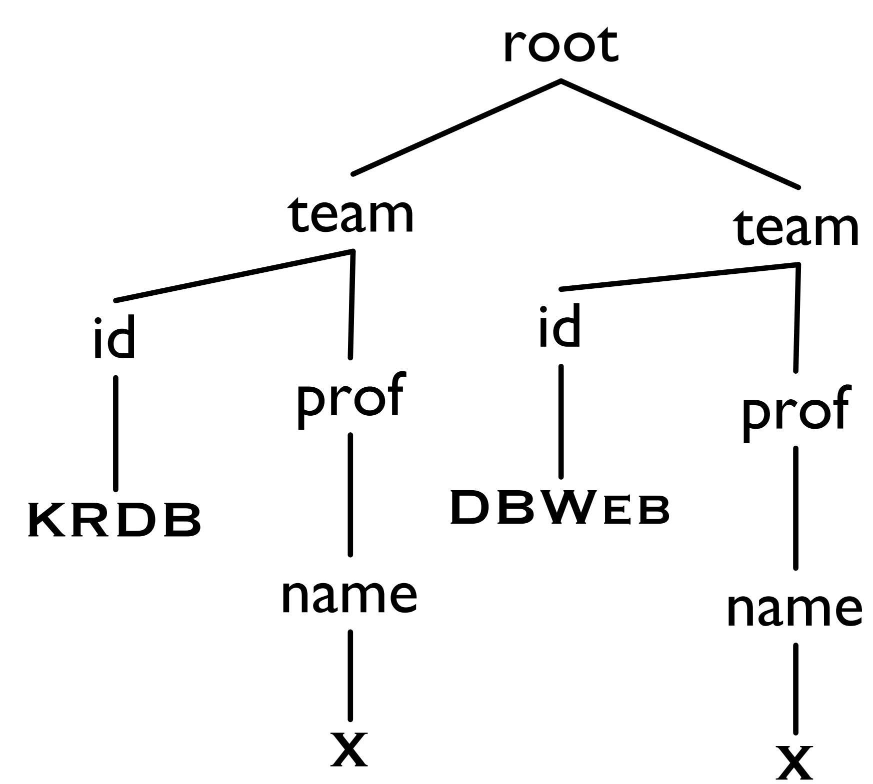


XPath notation:

. //team[id="KRDB"] /prof/name =
. //team[id="DBWeb"]/prof/name

Tree Pattern Queries

c) Tree-Pattern Queries with Joins - TPJ
Are there (names of) professors working for both KRDB and DBWeb?

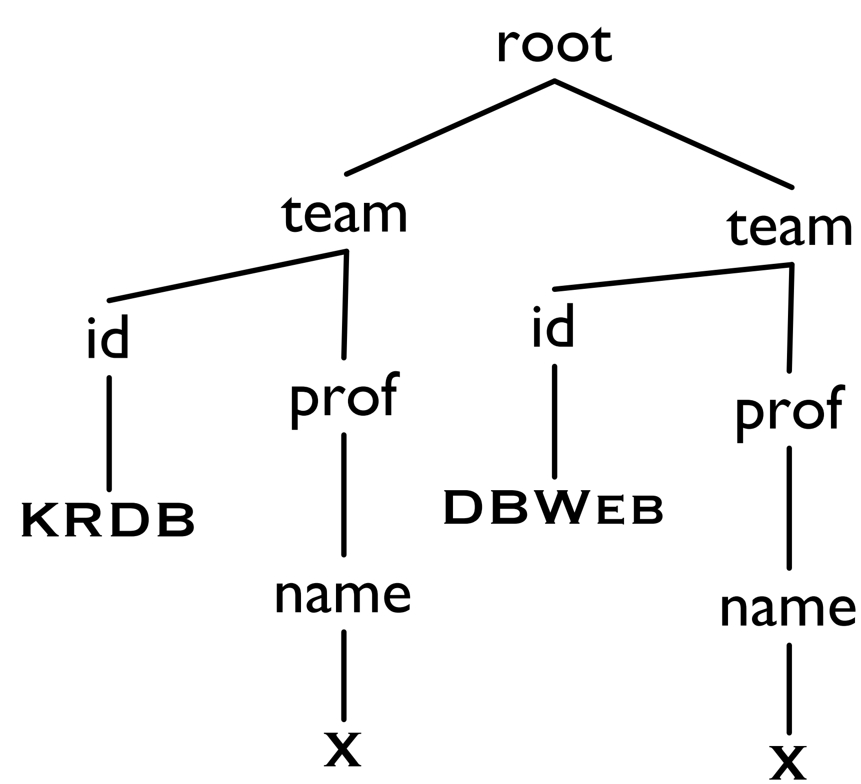


XPath notation:

. //team[id="KRDB"] /prof/name =
. //team[id="DBWeb"]/prof/name

Tree Pattern Queries

c) Tree-Pattern Queries with Joins - TPJ
Are there (names of) professors working for both KRDB and DBWeb?



XPath notation:

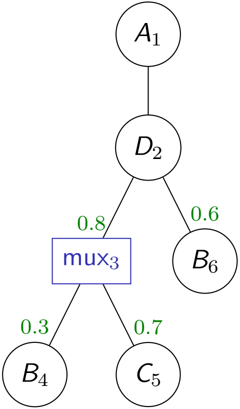
. //team[id=”KRDB”] /prof/name =
. //team[id=”DBWeb”]/prof/name

- TP - in navigational XPath
- TPJ - fragment of XPath 2.0

Algorithm for TP over local dependencies

[Kimelfeld and Sagiv, 2007]

Bottom-up dynamic programming algorithm. Query: /A//B



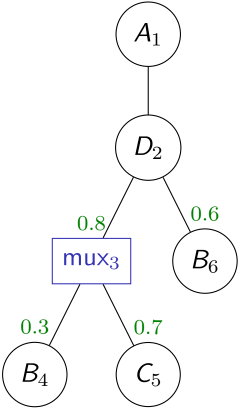
	A_1	D_2	mux_3	B_4	C_5	B_6
/B				1	0	1
//B				1	0	1
/A//B				0	0	0

mux	convex sum
ordinary	inclusion-exclusion

Algorithm for TP over local dependencies

[Kimelfeld and Sagiv, 2007]

Bottom-up dynamic programming algorithm. Query: /A//B



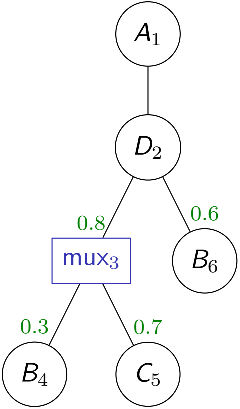
	A_1	D_2	mux_3	B_4	C_5	B_6
/B			0.3	1	0	1
//B			0.3	1	0	1
/A//B			0	0	0	0

mux	convex sum
ordinary	inclusion-exclusion

Algorithm for TP over local dependencies

[Kimelfeld and Sagiv, 2007]

Bottom-up dynamic programming algorithm. Query: /A//B



	A_1	D_2	mux_3	B_4	C_5	B_6
/B		0	0.3	1	0	1
//B		0.696	0.3	1	0	1
/A//B		0	0	0	0	0

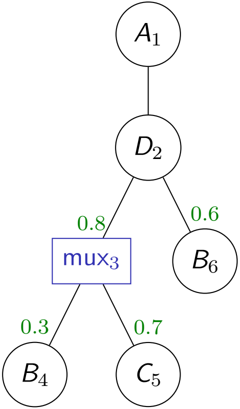
mux	convex sum
ordinary	inclusion-exclusion

$$\begin{aligned}\Pr(D_2 \models //B) &= 1 - (1 - 0.8 \times \Pr(\text{mux}_3 \models /B)) \times (1 - 0.6 \times \Pr(B_6 \models /B)) \\ &= 1 - (1 - 0.8 \times 0.3) \times (1 - 0.6) = 0.696\end{aligned}$$

Algorithm for TP over local dependencies

[Kimelfeld and Sagiv, 2007]

Bottom-up dynamic programming algorithm. Query: /A//B

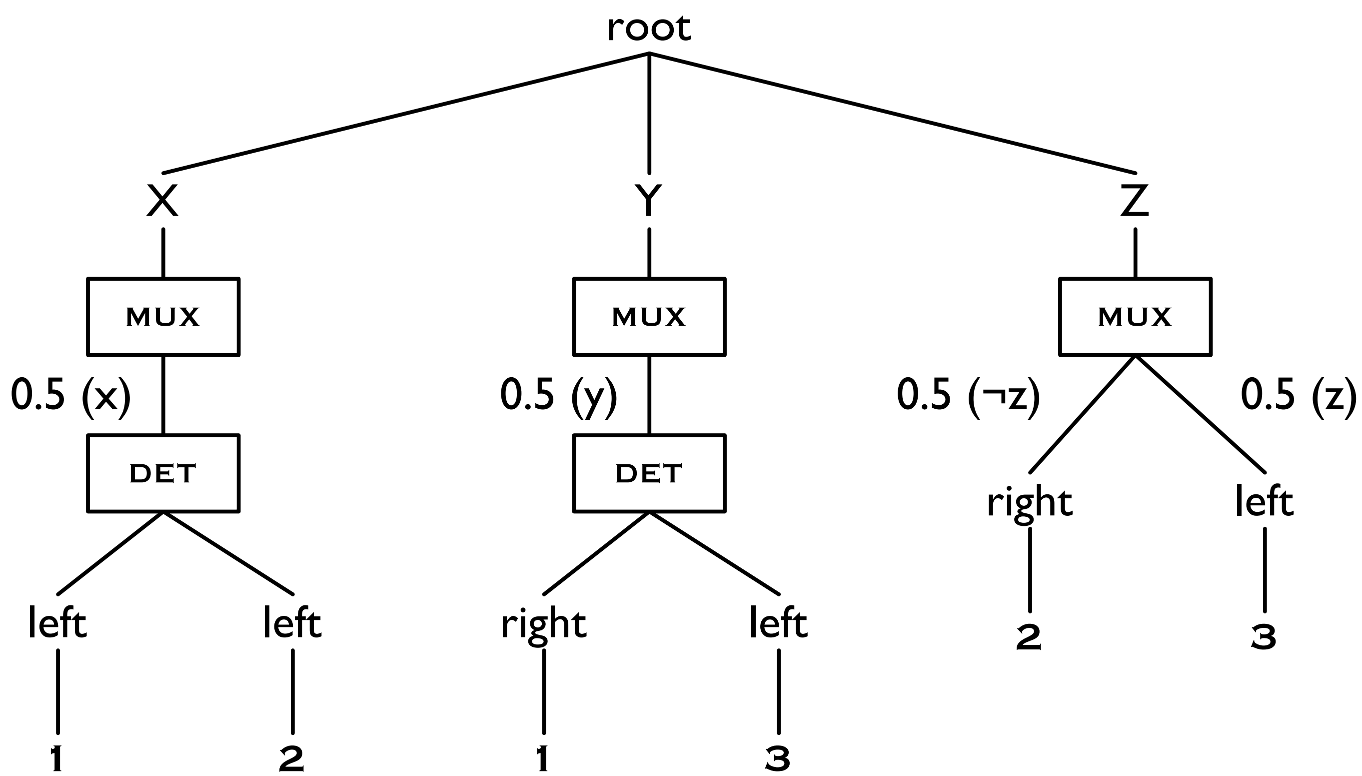


	A_1	D_2	mux_3	B_4	C_5	B_6
/B	0	0	0.3	1	0	1
//B	0.696	0.696	0.3	1	0	1
/A//B	0.696	0	0	0	0	0

mux	convex sum
ordinary	inclusion-exclusion

Hard Query with one Join

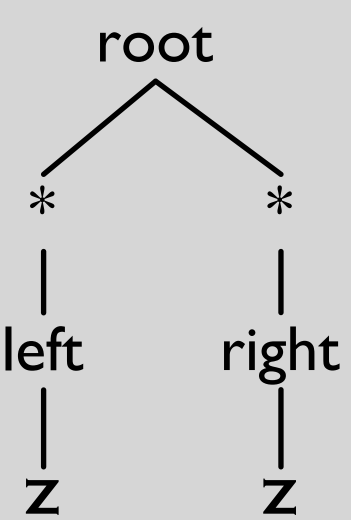
Encoding of 2DNF formula: $(x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge z)$:



Intuition on encoding:

- one root-subdocument for every variable
- one child of MUX gathers negative occurrences of variables
- another child of MUX - negative
- left/0 means left

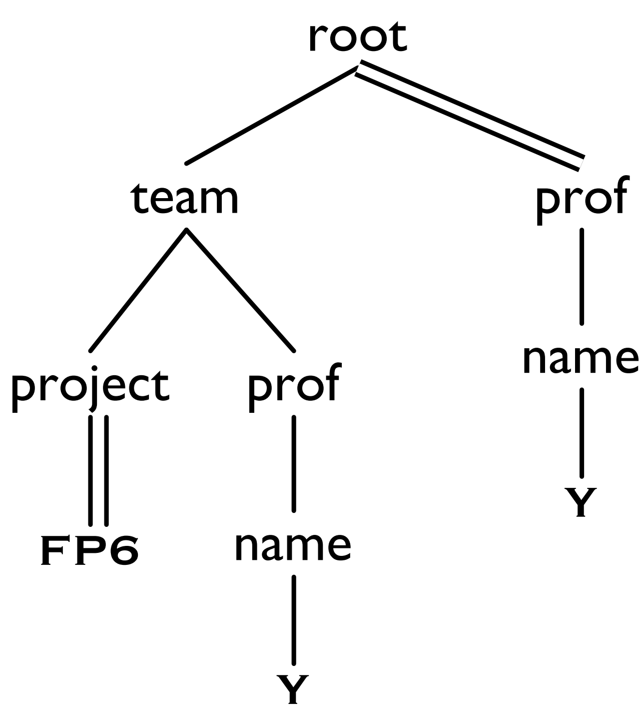
Hard query with joins Q



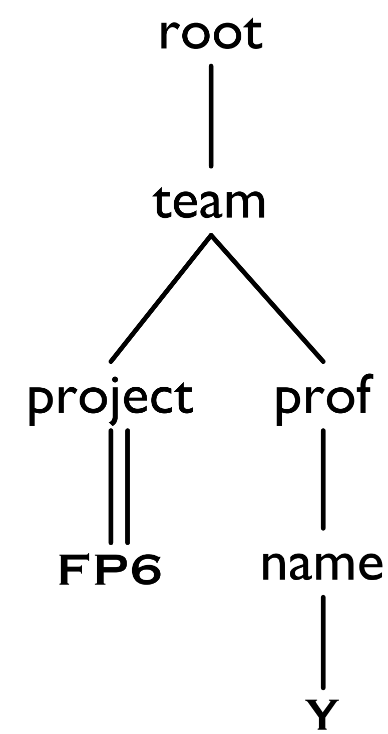
Theorem: (Reduction from #2DNF)
Every 2DNF formula φ can be converted into a PrXML doc D_φ s.t. $Pr(\varphi=true) = C \times Pr(Q \text{ matches } D_\varphi)$

Not all Joins are the Same

- Some joins are fake



equivalent to



Theorem: Let q be a TPJ query with a single join. Then:

- if the join of q is fake, then query evaluation of q over PrXML is PTIME;
- otherwise, it is intractable

Querying PrXML (Data Complexity)

Data \ Queries	Single-Path	Tree-Pattern	Tree-Pattern with Joins
Local PrXML	polynomial		intractable
Global PrXML	intractable		intractable

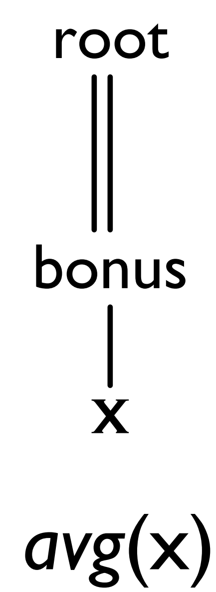
[Kimelfed&al:2007], [Senellart&al:2007]

- Focus on data complexity of functions and not of decision problems
- intractability: #P-hardness = counting counterparts of NP problems.
- Sources of intractability:
  - Global probabilistic dependencies in data
  - Joins in queries
- Practical considerations:
  exact computation only for: local PrXML model + no joins in queries

Aggregate Queries

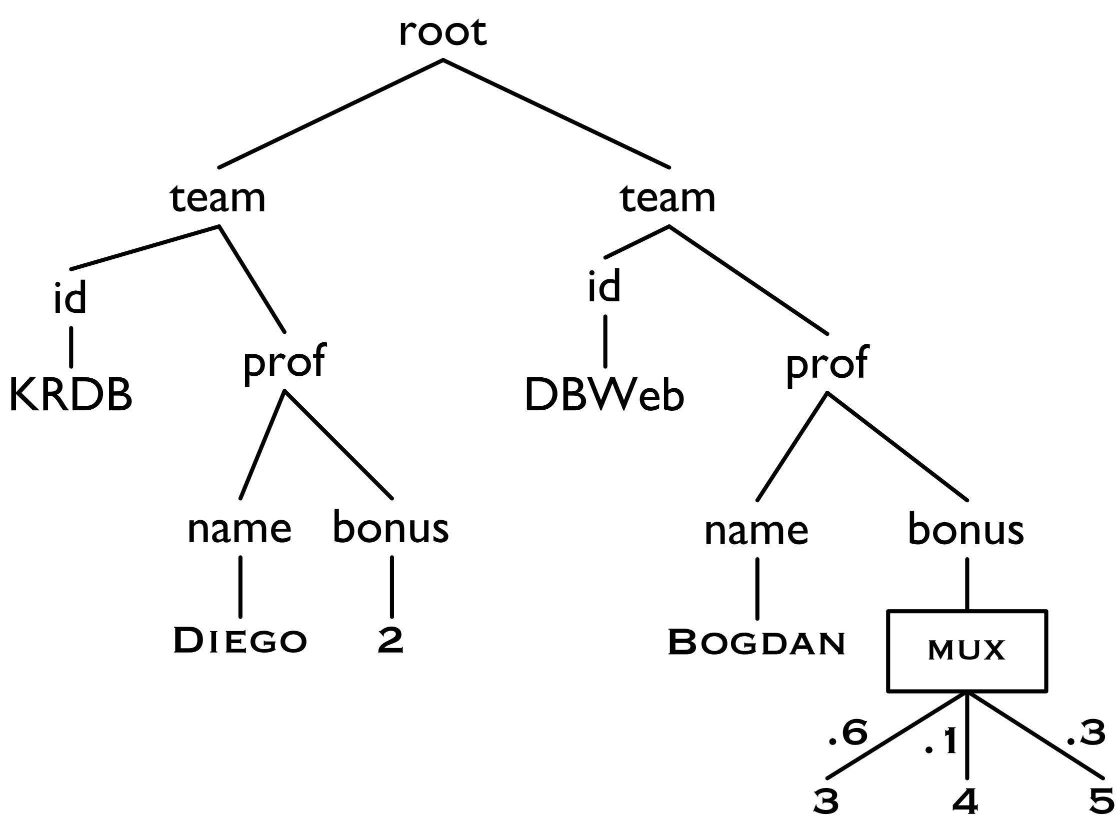
Aggregate Query:

What is the average of bonuses?



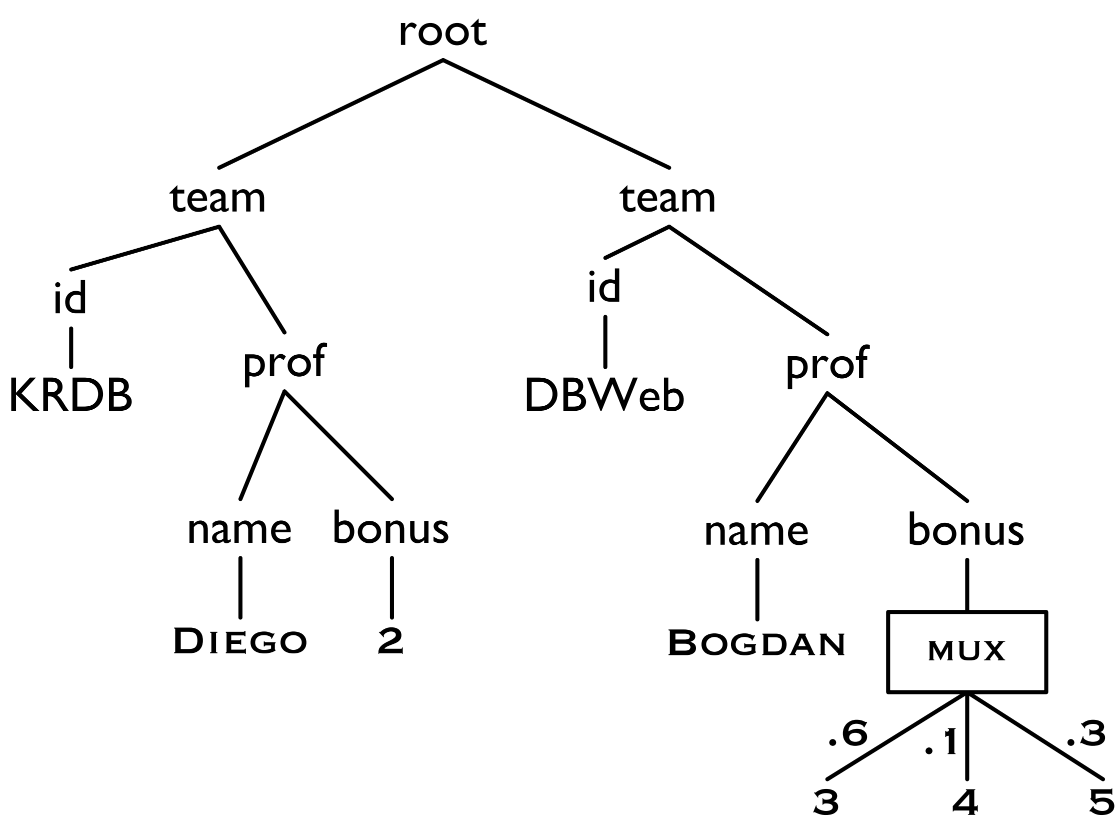
- Extend TPJ queries with aggregate functions
- Aggregate functions:
  sum, *count*, *min*, *max*, *avg*, *countd*

Example of Aggregate Queries over PrXML



Query:
What is the average of bonuses?

Example of Aggregate Queries over PrXML



Query:

What is the average of bonuses?

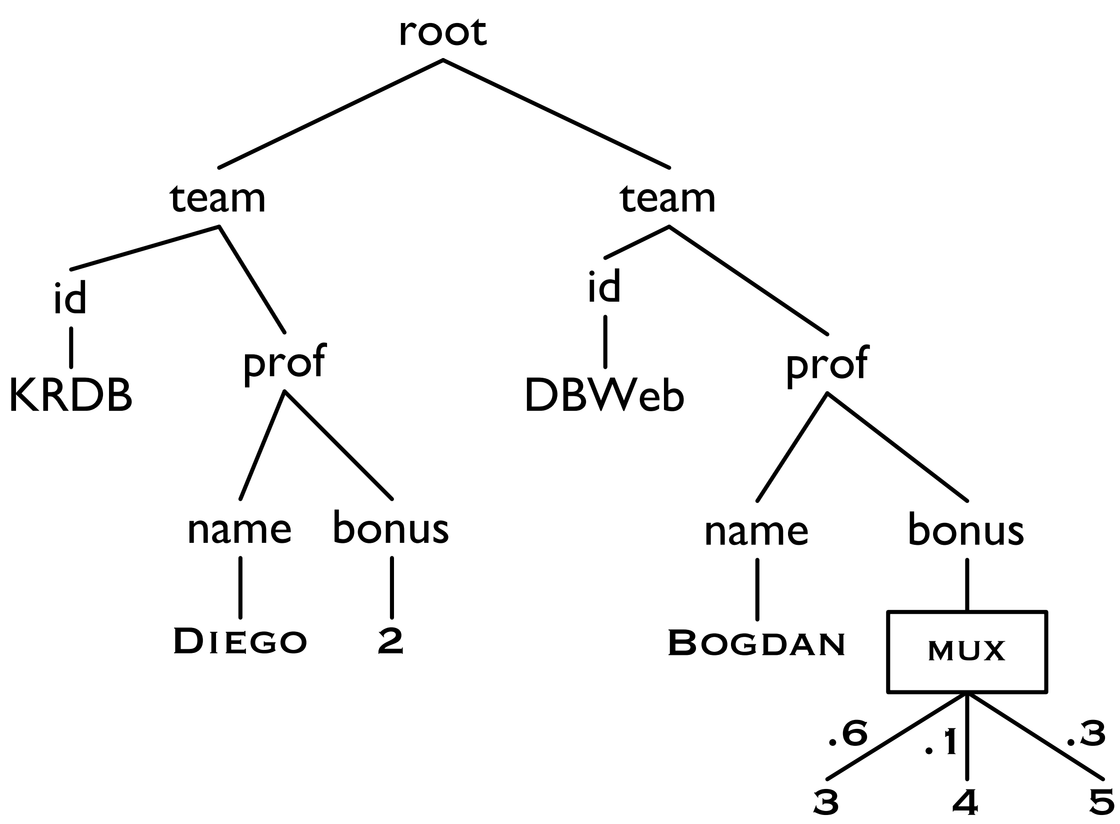
Answers over worlds:

avg(w3) = 2.5,   Pr(w3)=0.6

avg(w4) = 3,   Pr(w4)=0.1

avg(w5) = 3.5,   Pr(w5)=0.3

Example of Aggregate Queries over PrXML



Query Answer over PrXML:
Distribution of aggregate values

Query:
What is the average of bonuses?

Answers over worlds:

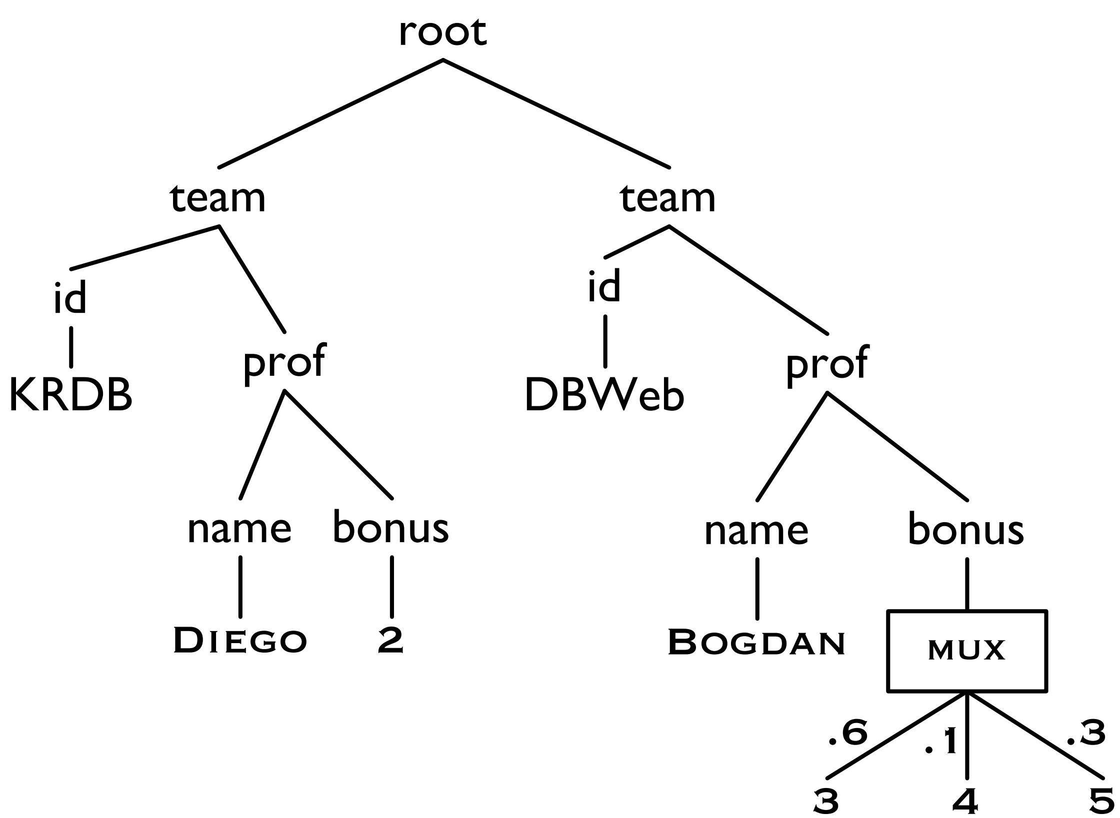
avg(w3) = 2.5, Pr(w3)=0.6

avg(w4) = 3, Pr(w4)=0.1

avg(w5) = 3.5, Pr(w5)=0.3

{ (2.5, 0.6), (3, 0.1), (3.5, 0.3) }

Example of Aggregate Queries over PrXML



Query:
What is the average of bonuses?

Answers over worlds:

avg(w3) = 2.5, Pr(w3)=0.6

avg(w4) = 3, Pr(w4)=0.1

avg(w5) = 3.5, Pr(w5)=0.3

Query Answer over PrXML:
Distribution of aggregate values

{ (2.5, 0.6), (3, 0.1), (3.5, 0.3) }

Problems to study:

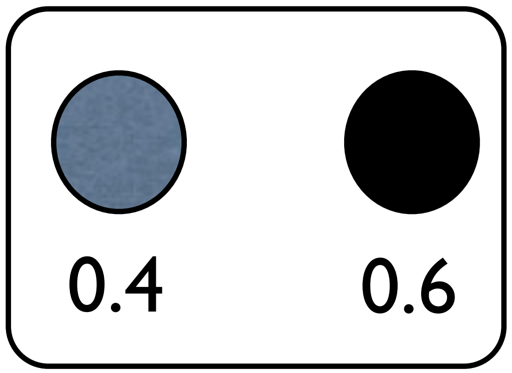
- probability computation: Pr(Q(w)=C)
- moments computation: $E(Q(w)^k)$

Operations on Probability Spaces

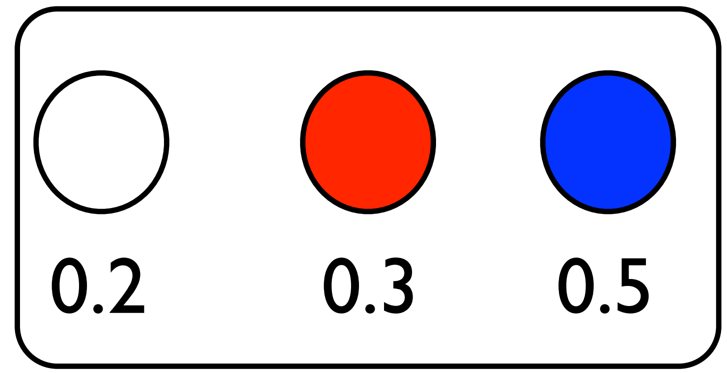
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$



$\Delta_2 =$

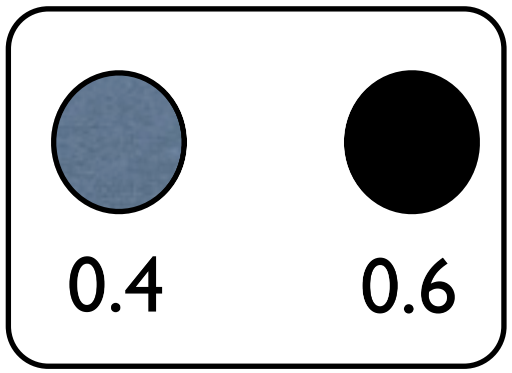


Operations on Probability Spaces

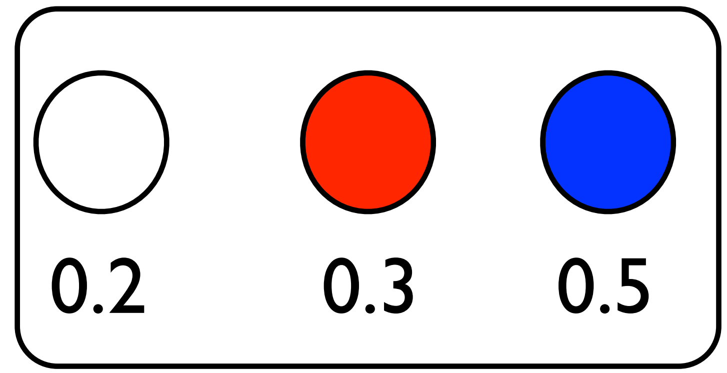
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$



$\Delta_2 =$



Convex Sum:

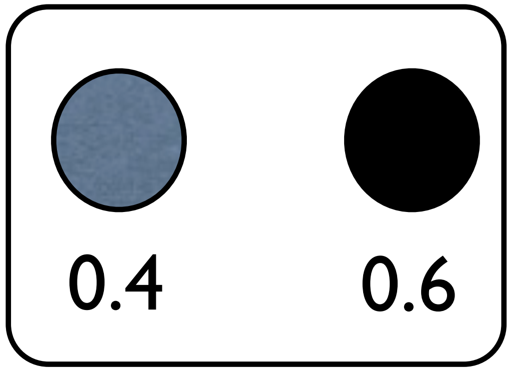
$p \bullet \Delta_1 + q \bullet \Delta_1 =$

Operations on Probability Spaces

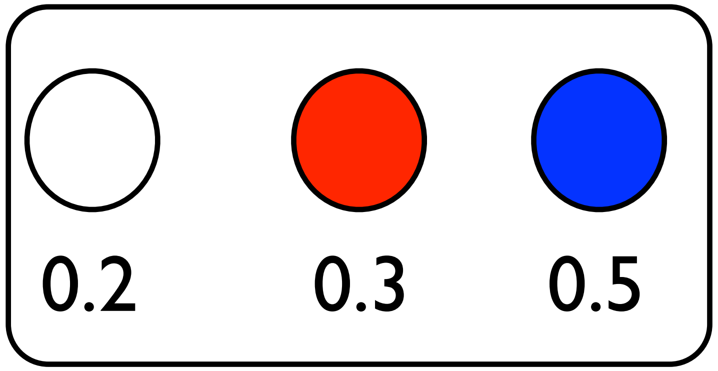
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$

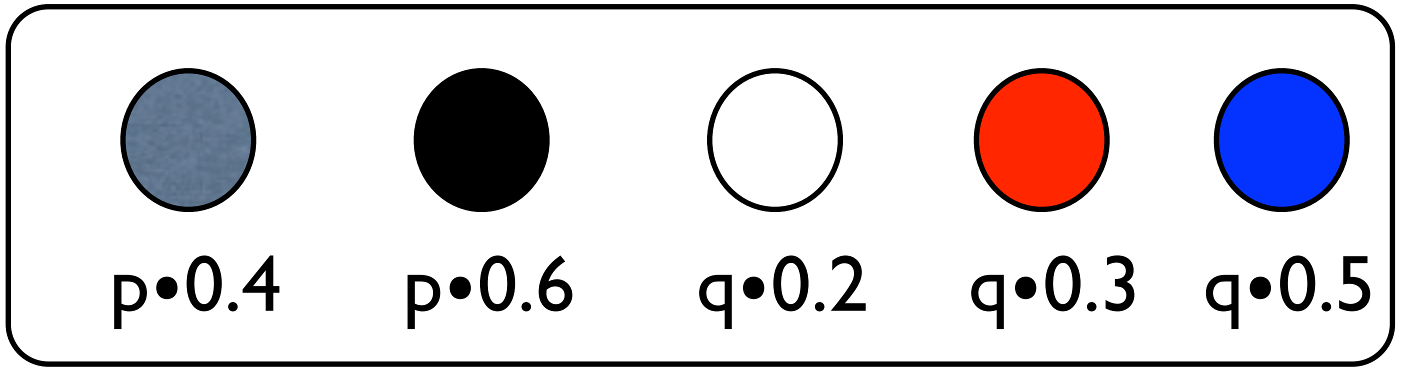


$\Delta_2 =$



Convex Sum:

$p \bullet \Delta_1 + q \bullet \Delta_1 =$

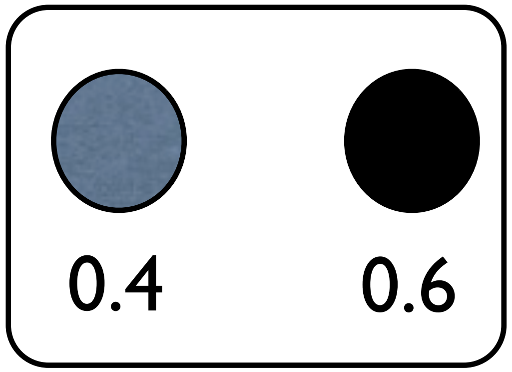


Operations on Probability Spaces

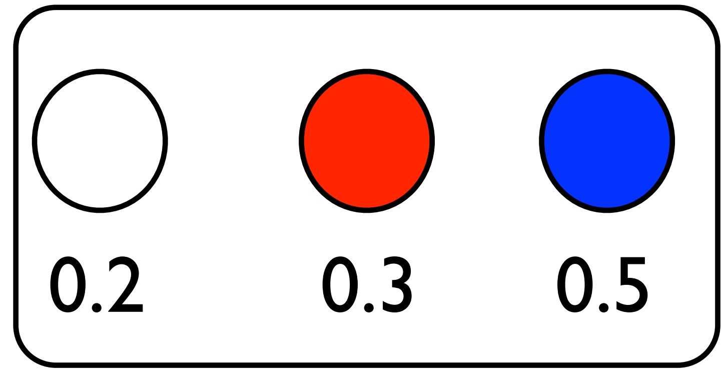
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$

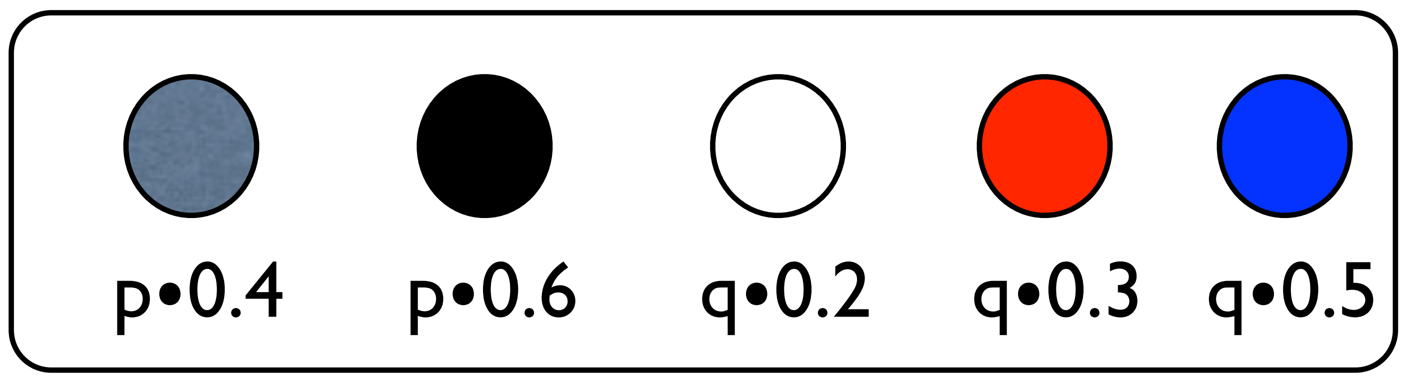


$\Delta_2 =$



Convex Sum:

$p \bullet \Delta_1 + q \bullet \Delta_1 =$



Convolution:

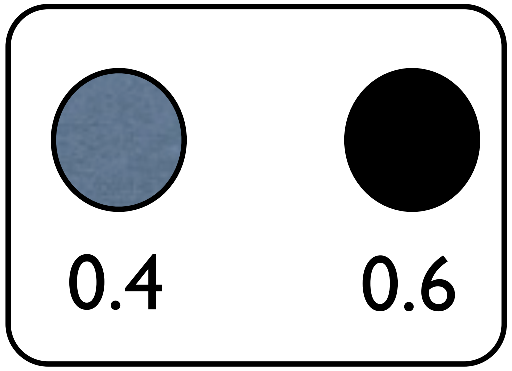
$\Delta_1 \oplus \Delta_1 =$

Operations on Probability Spaces

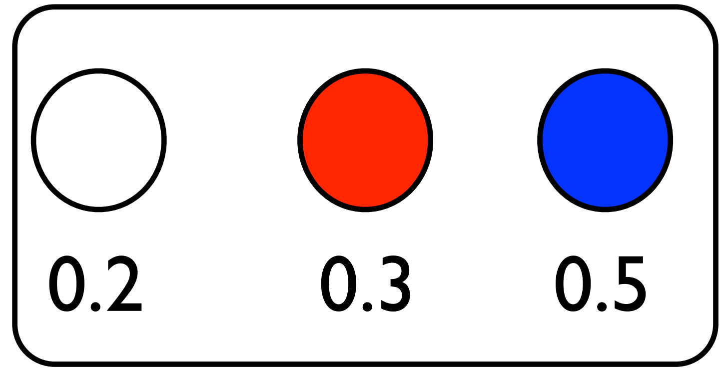
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$

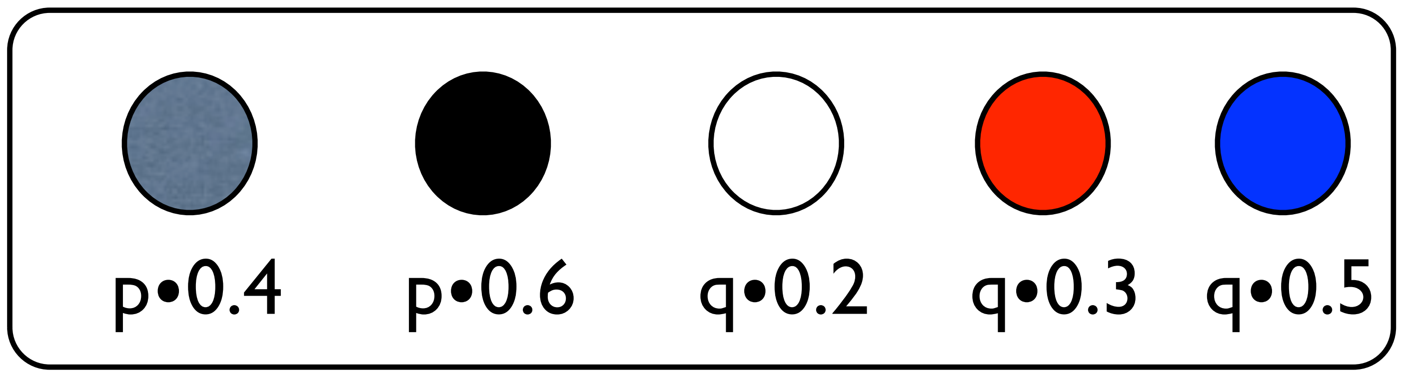


$\Delta_2 =$



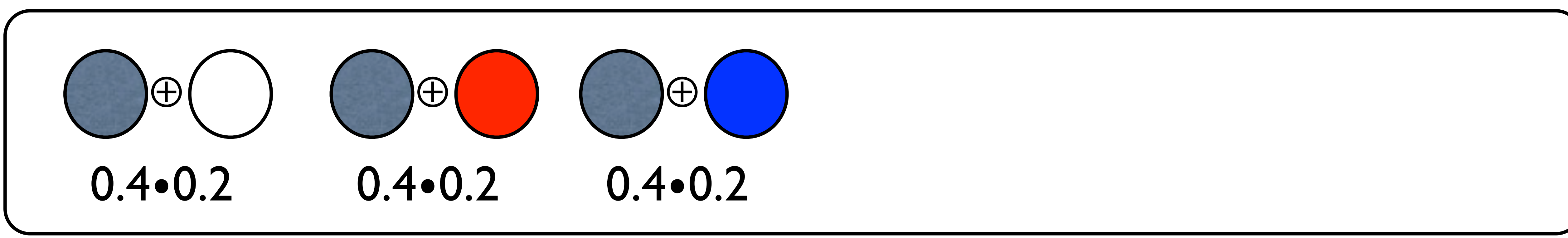
Convex Sum:

$p \bullet \Delta_1 + q \bullet \Delta_1 =$



Convolution:

$\Delta_1 \oplus \Delta_1 =$

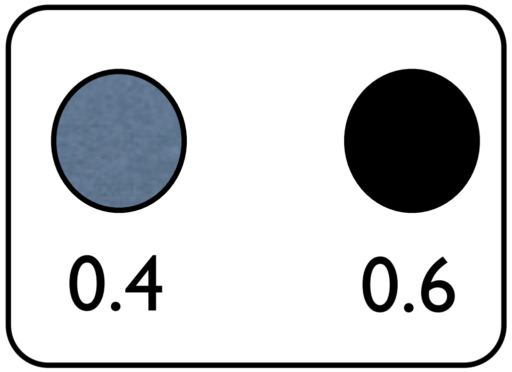


Operations on Probability Spaces

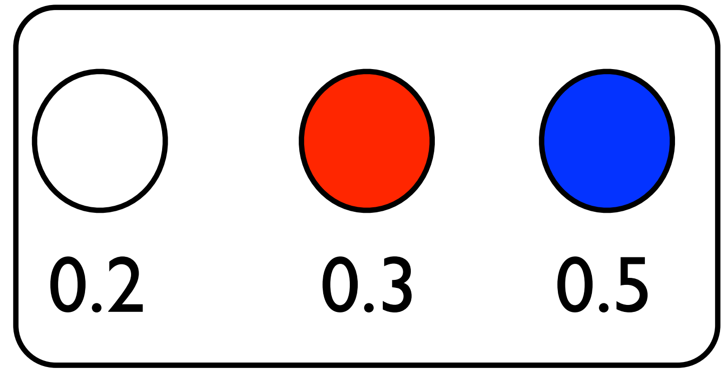
Convex coefficients $P_1,\ldots,P_n$: $P_1+\ldots+P_n = 1$

$\oplus$ operation: in our case it is: sum, count, min, topK, ...

$\Delta_1 =$

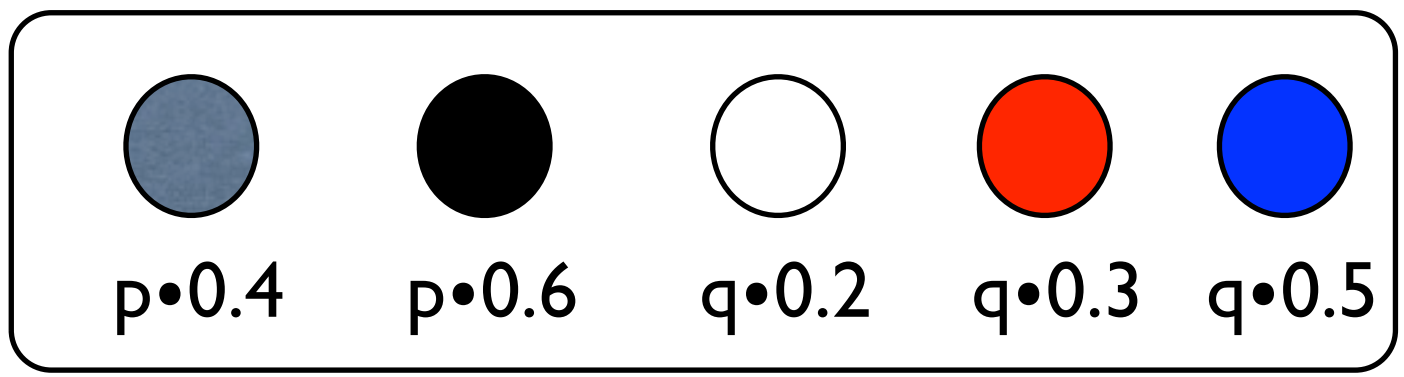


$\Delta_2 =$



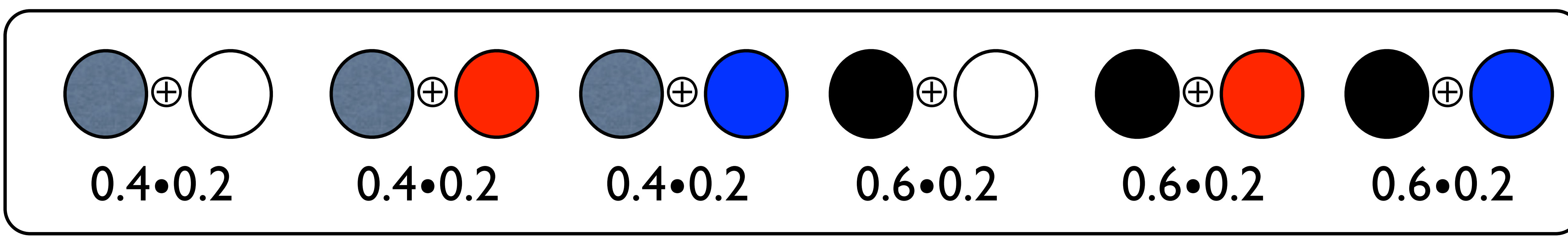
Convex Sum:

$p \bullet \Delta_1 + q \bullet \Delta_1 =$

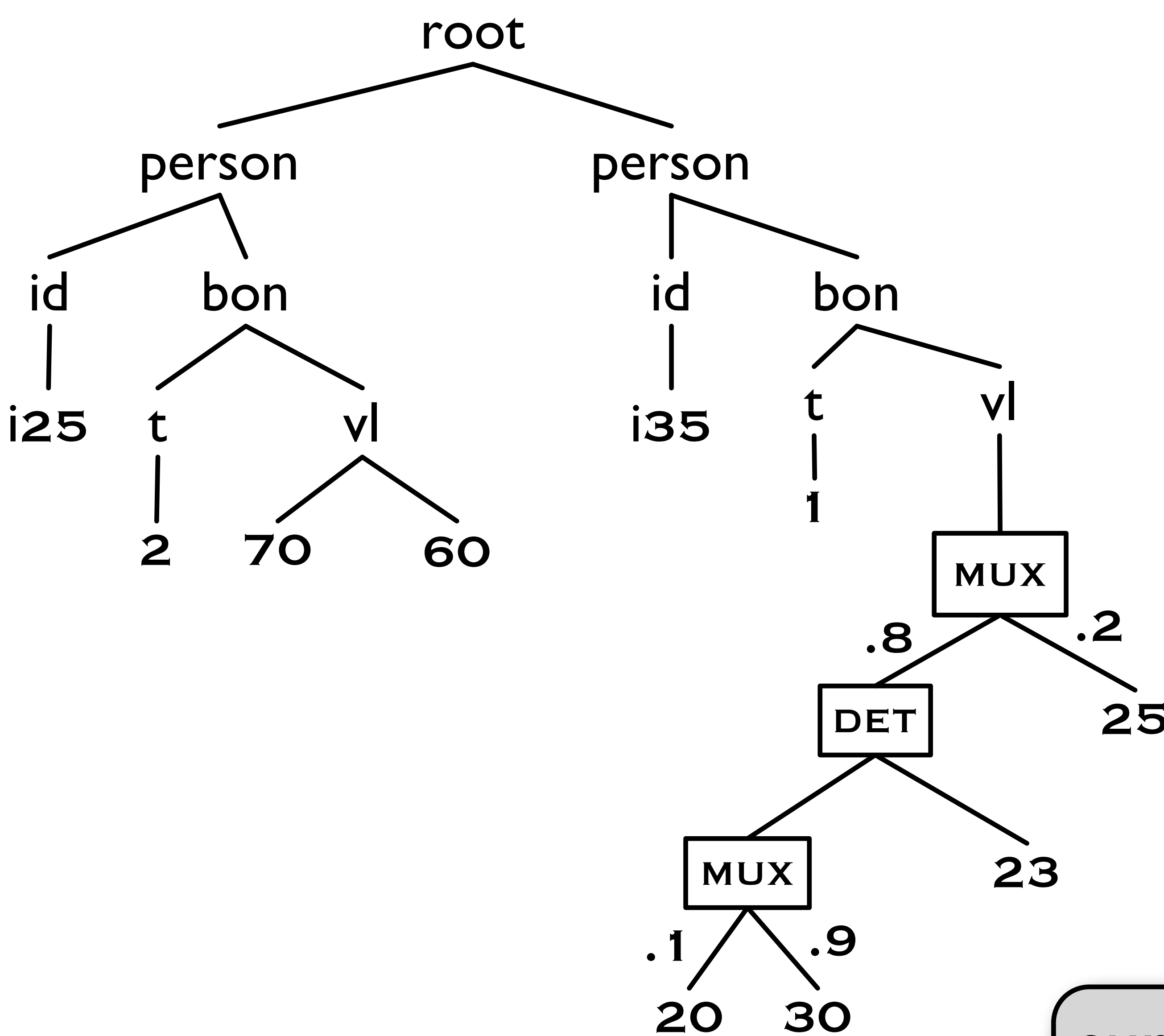


Convolution:

$\Delta_1 \oplus \Delta_1 =$



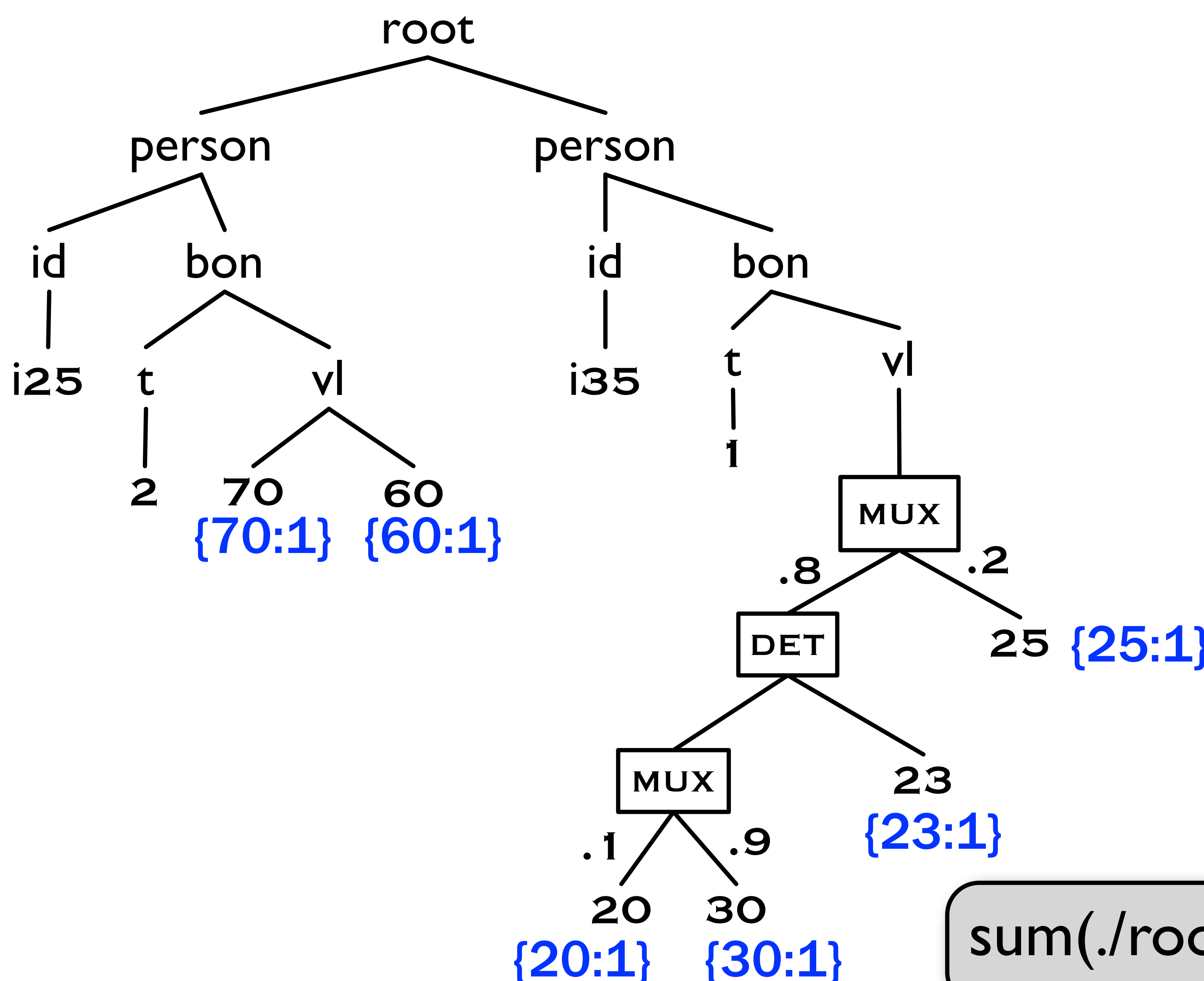
Bottom-up Algorithm for Local PrXML



sum(./root/person/bon/vl/*)

- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

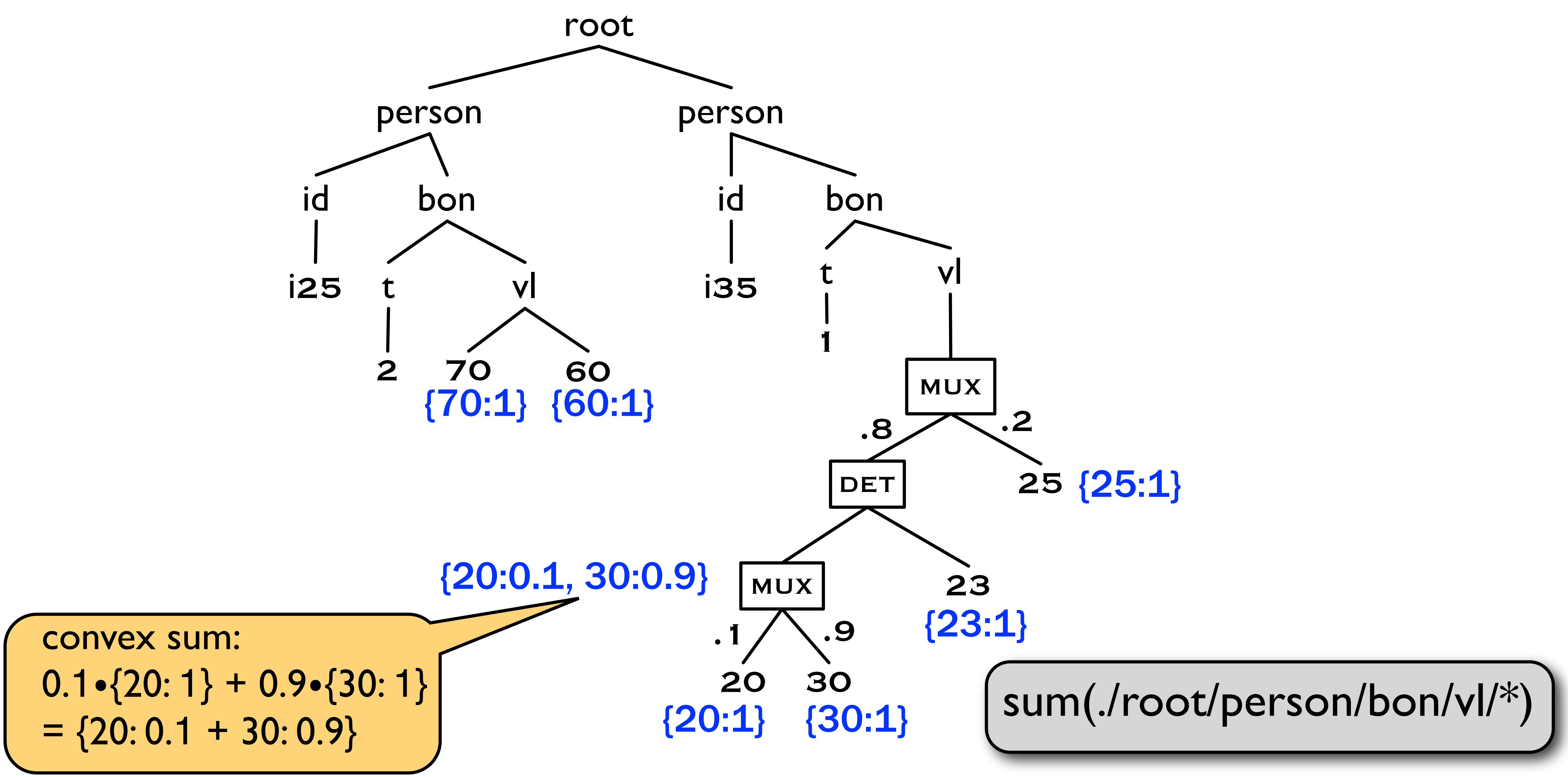
Bottom-up Algorithm for Local PrXML



sum(./root/person/bon/vl/*)

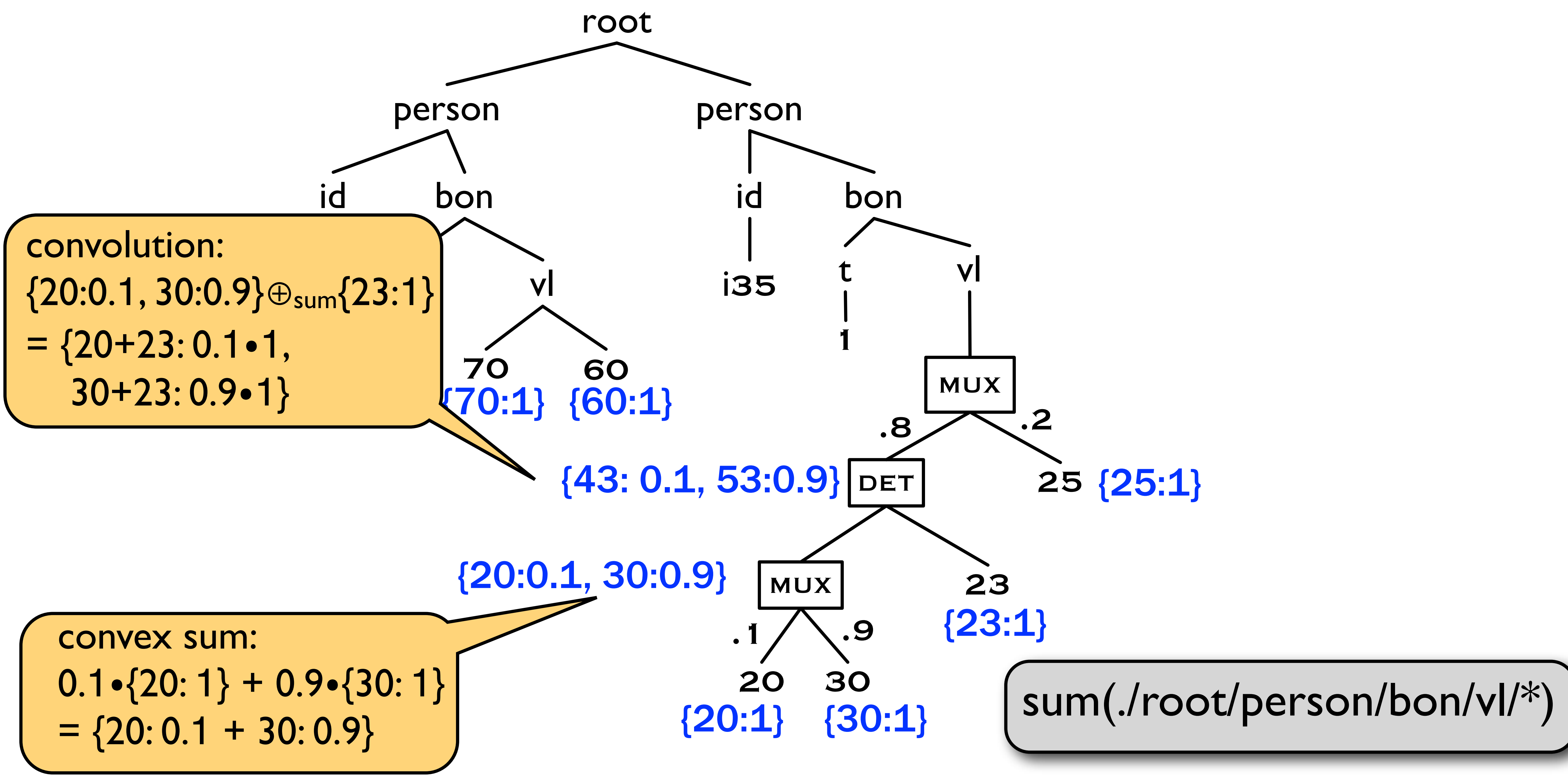
- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



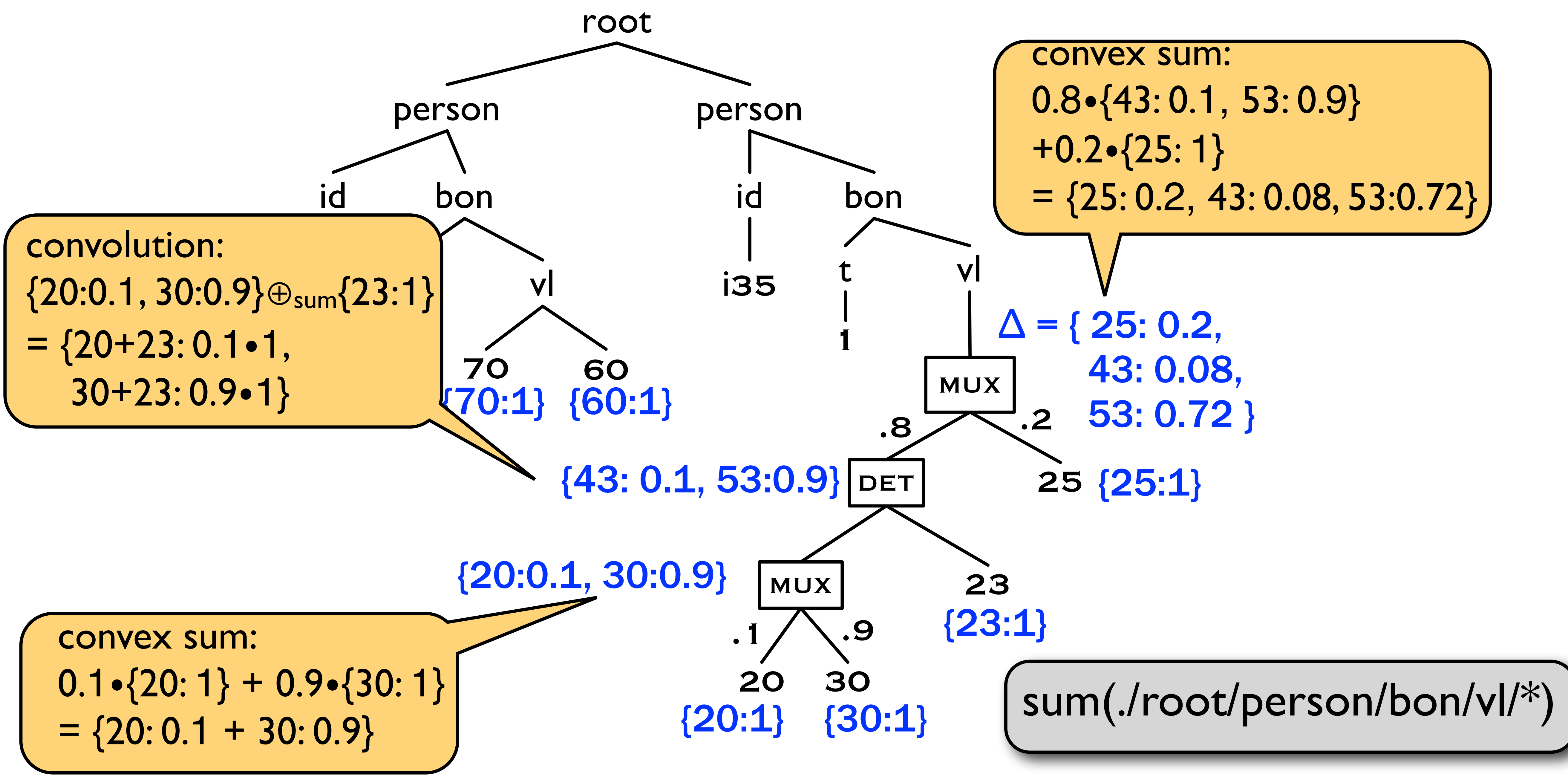
- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



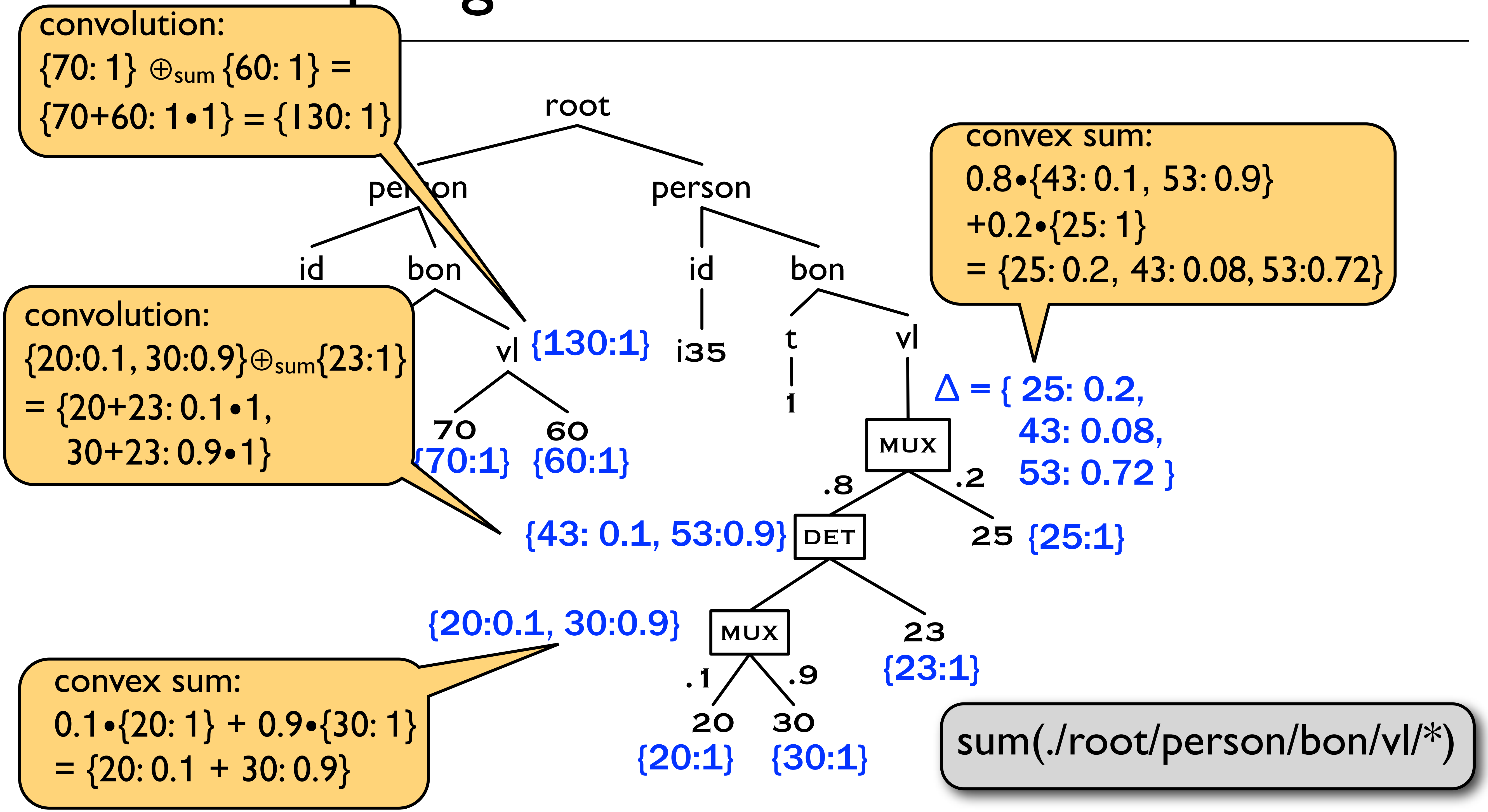
- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



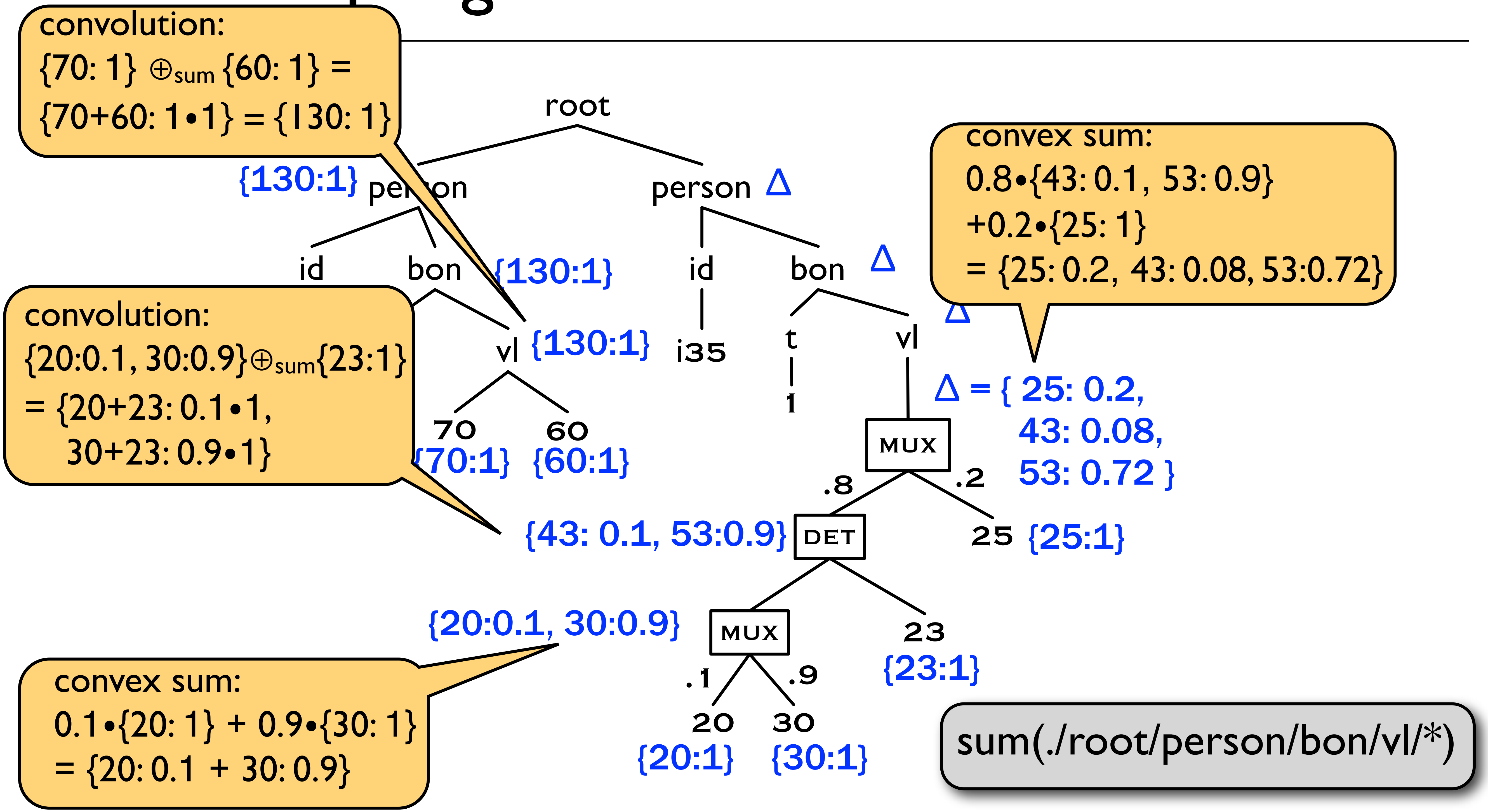
- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



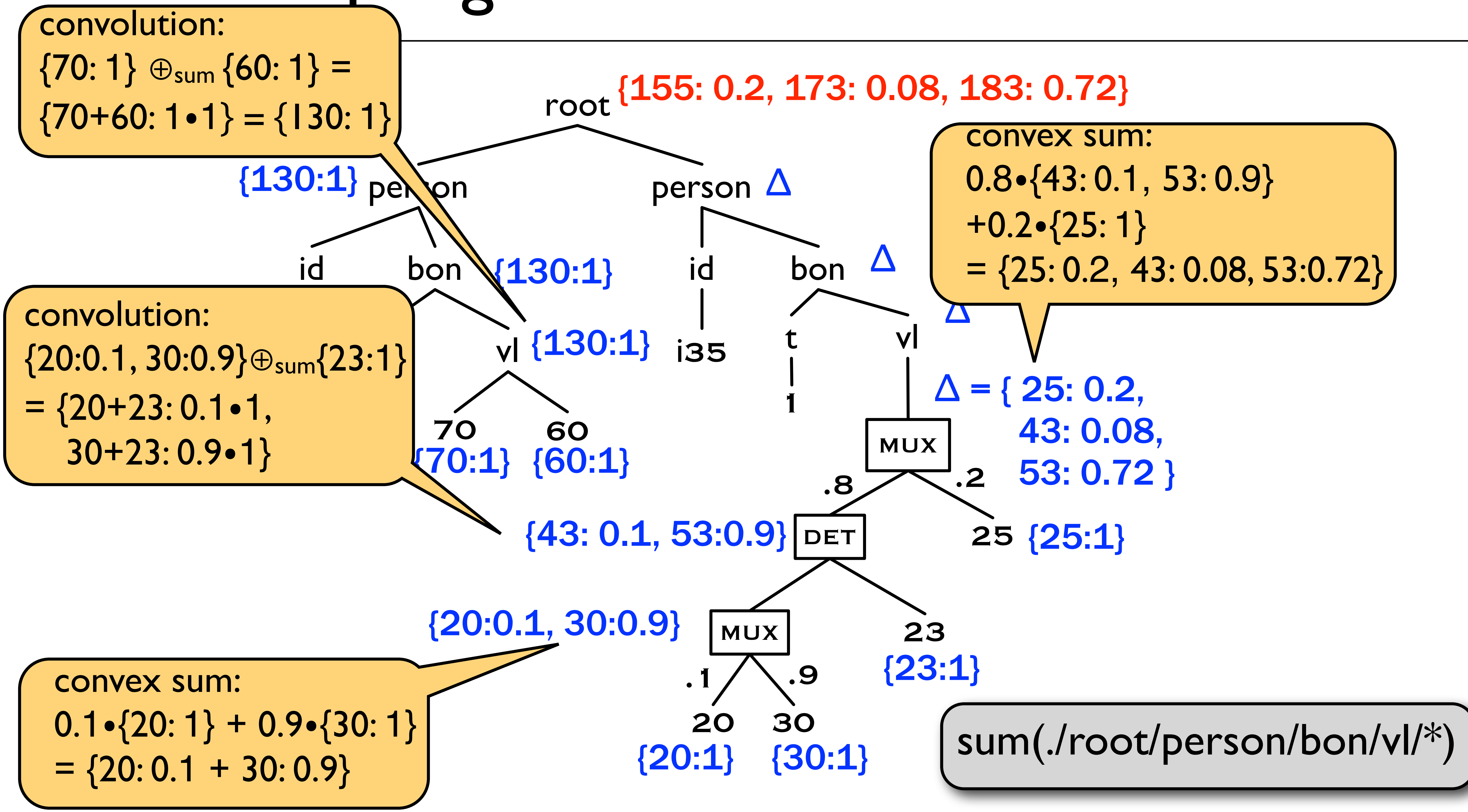
- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Bottom-up Algorithm for Local PrXML



- MUX-node = convex sum of distributions from rooted subtrees
- DET-node, regular node = convolution of distrib. from rooted subtrees

Monoid Functions

- Monoid functions allow for divide-and-conquer strategy:

$$\{|\ 2, 3, 3, 5\ |\} = \{|\ 2, 3\ |\} \cup \{|\ 3, 5\ |\}$$

$$\text{SUM}\ \{|\ 2, 3, 3, 5\ |\} = \text{SUM}\ \{|\ 2, 3\ |\} + \text{SUM}\ \{|\ 3, 5\ |\}$$

- For global PrXML in PTIME only moments of SP w/ count, sum

Monoid Functions

- Monoid functions allow for divide-and-conquer strategy:

$$\{|\ 2, 3, 3, 5\ |\} = \{|\ 2, 3\ |\} \cup \{|\ 3, 5\ |\}$$

$$\text{SUM}\ \{|\ 2, 3, 3, 5\ |\} = \text{SUM}\ \{|\ 2, 3\ |\} + \text{SUM}\ \{|\ 3, 5\ |\}$$

COUNT, SUM, MIN ✓

COUNTD, AVG ✗

- For global PrXML in PTIME only moments of SP w/ count, sum

Monoid Functions

- Monoid functions allow for divide-and-conquer strategy:

$$\{| 2, 3, 3, 5 |\} = \{| 2, 3 |\} \cup \{| 3, 5 |\}$$

$$\text{SUM } \{| 2, 3, 3, 5 |\} = \text{SUM } \{| 2, 3 |\} + \text{SUM } \{| 3, 5 |\}$$

COUNT, SUM, MIN ✓

COUNTD, AVG ✗

Theorem: For aggregate TP-queries with monoid functions over local PrXML bottom-up algorithm is applicable and

- prob. computations is in PTIME in |output distribution|
- moment computation is in PTIME in |input p-document|

- For global PrXML in PTIME only moments of SP w/ count, sum

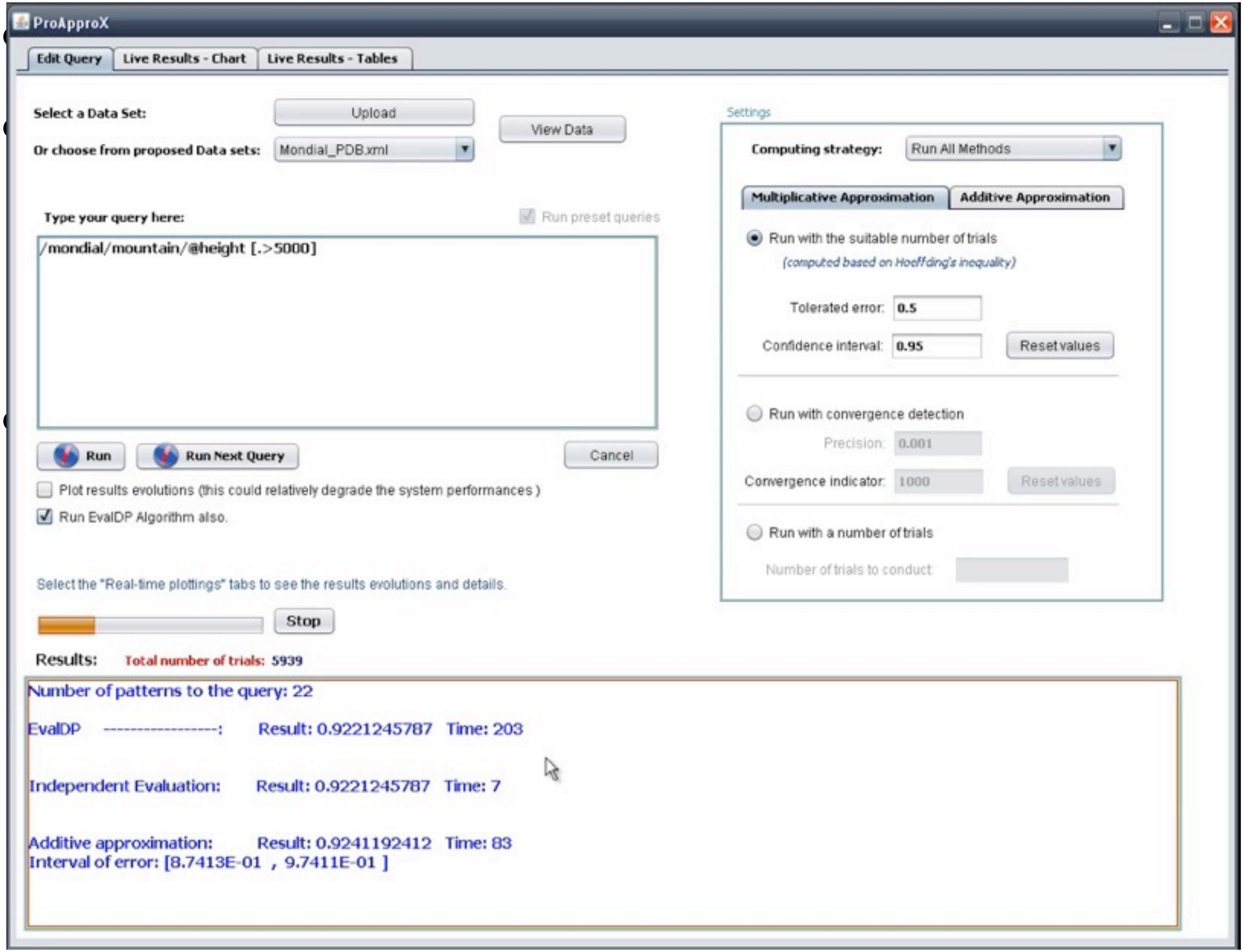
Conclusion on Queries over PrXML

- Value joins in queries are intrinsically intractable
- Global model is intractable for essentially every query
- Aggregation can be easier than querying
  - moments over global PrXML
- Tractable cases for aggregation
  - Distributions: TP + monoid functions over L-PrXML
  - Moments: SP + every considered function over L-PrXML
  - Moments: SP + *sum*, *count* over G-PrXML
- Sampling is unavoidable in many practical cases

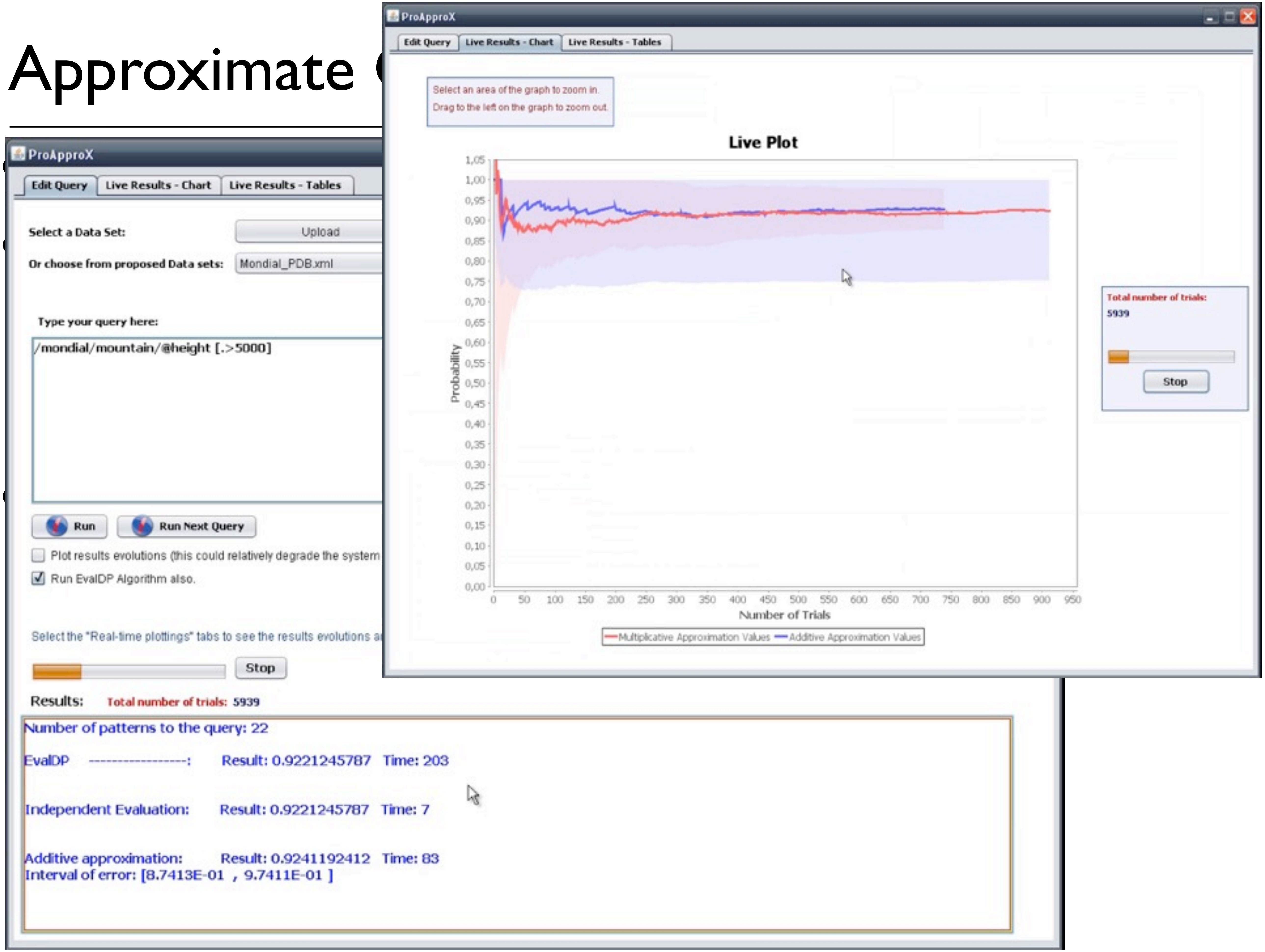
Approximate Query Answering over PrXML

- Use the same sampling idea as in the relation case
- Special case of PrXML:
  - lineage is usually in DNF
    => one can use specialized techniques for probability computation of DNF formulas
- System for query evaluation over PrXML: ProApproX
  it allows for
  - additive approximation
  - multiplicative approximation
  - exact computation

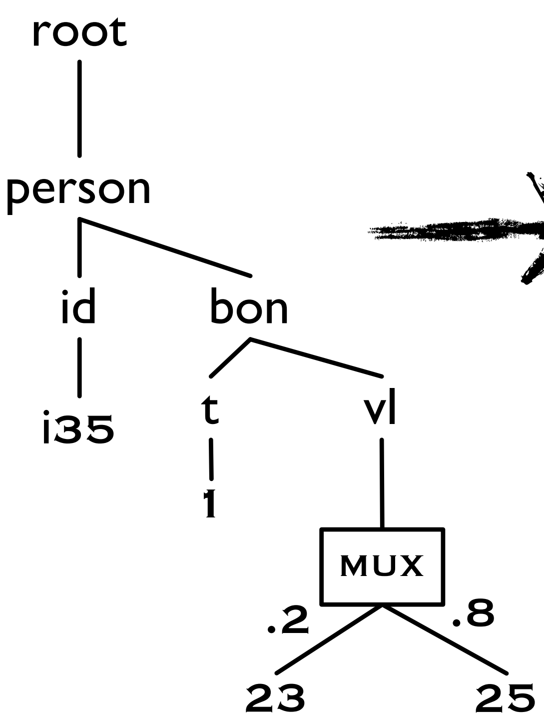
Approximate Query Answering over PrXML



Approximate (



Queries over Relations vs XML



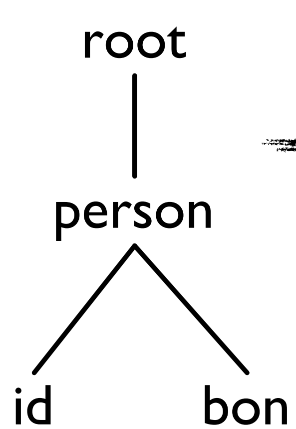
Label

ID	node	label
1	x_1	root
2	x_2	person
3	x_3	vl
4	x_4	23
5	x_5	25
...	...	...

Edge

ID	node	node	prob
1	x_1	x_2	1
2	x_3	x_4	0.2
	x_3	x_5	0.8
...	...	...	...

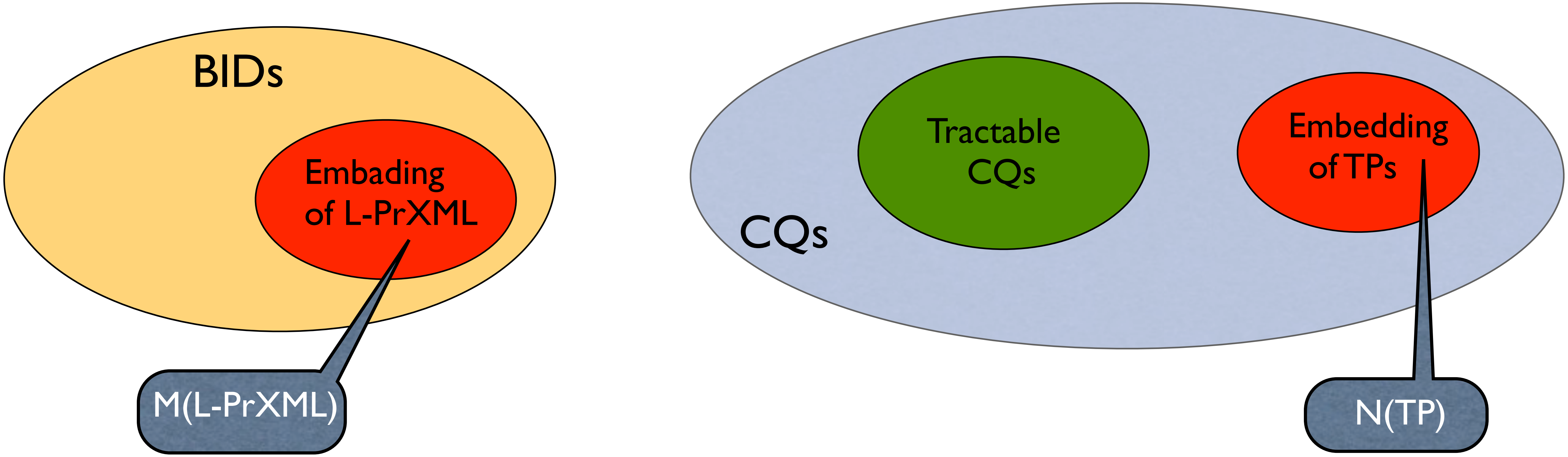
- PrXML can be encoded in BIDs
- Encoding is of specific form



Q:- Label(x_1,root), Edge(x_1, x_2), Label(x_2,person), Edge(x_2, x_3), Label(x_3,id), Edge(x_2, x_4), Label(x_4,bon)

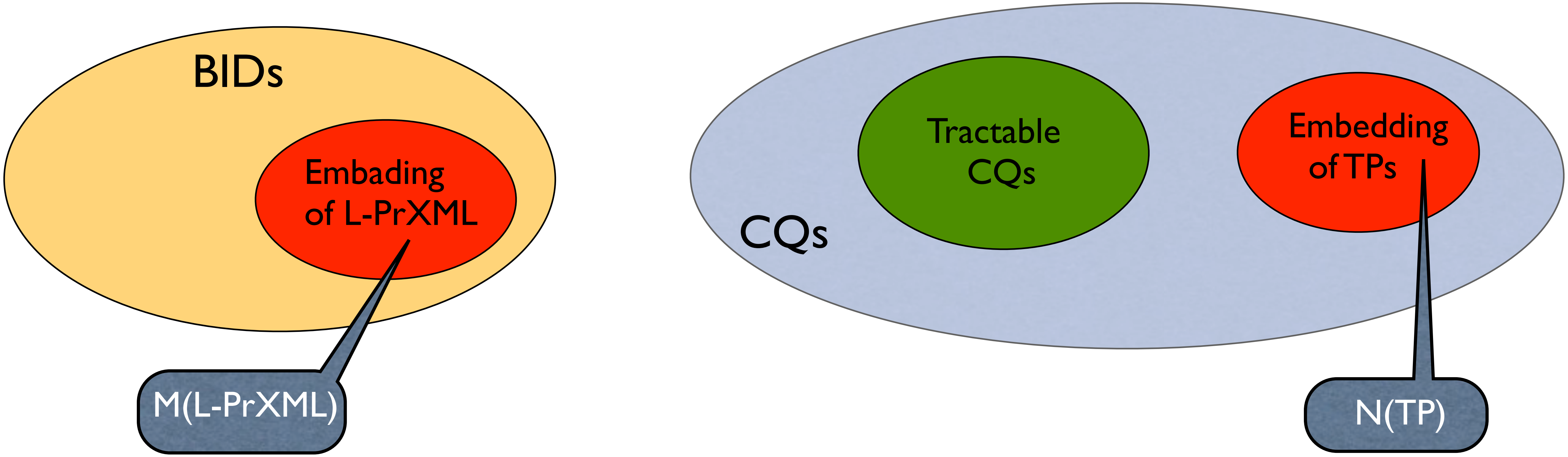
- TP queries can be encoded as CQs
- CQs are
  - hierarchical: $\Sigma(x_i)$ = {Edge, Label}
  - with self joins: same predicate occurs many times

Queries over Relations vs XML



- TP queries are tractable over the whole class Local-PrXML
- N(TP) - intractable for the whole class BID
- N(TP) is tractable for a fragment of BID M(L-PrXML)
- Embedding of both TP and L-PrXML is very specific

Queries over Relations vs XML



- TP queries are tractable over the whole class Local-PrXML
- N(TP) - intractable for the whole class BID
- N(TP) is tractable for a fragment of BID M(L-PrXML)
- Embedding of both TP and L-PrXML is very specific

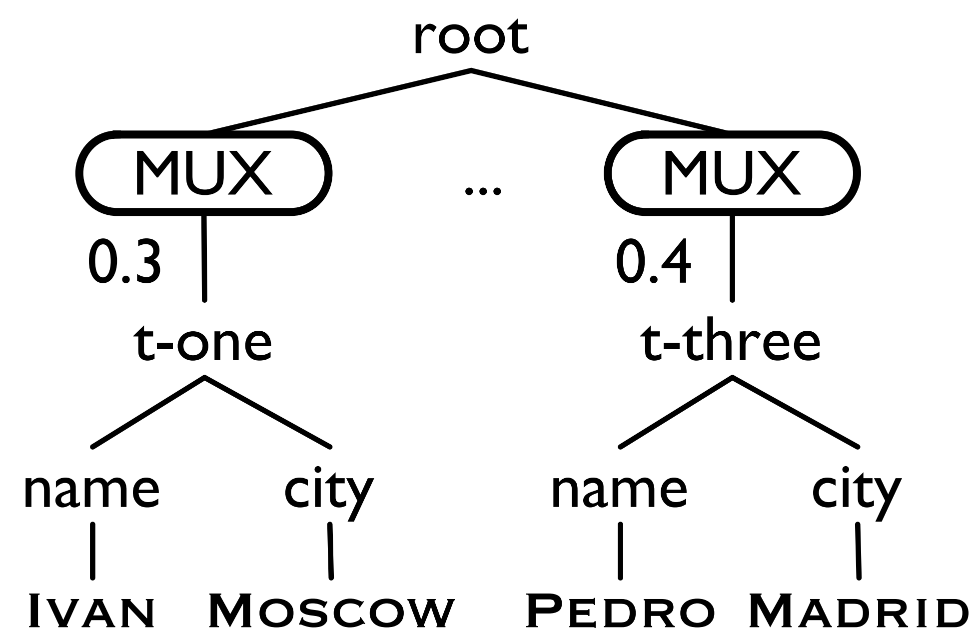
Results are not (easily) translatable from L-PrXML to BIDs.

Relationship between BID & CQ vs. L-PrXML & TP is unclear.

Queries over Relations vs XML

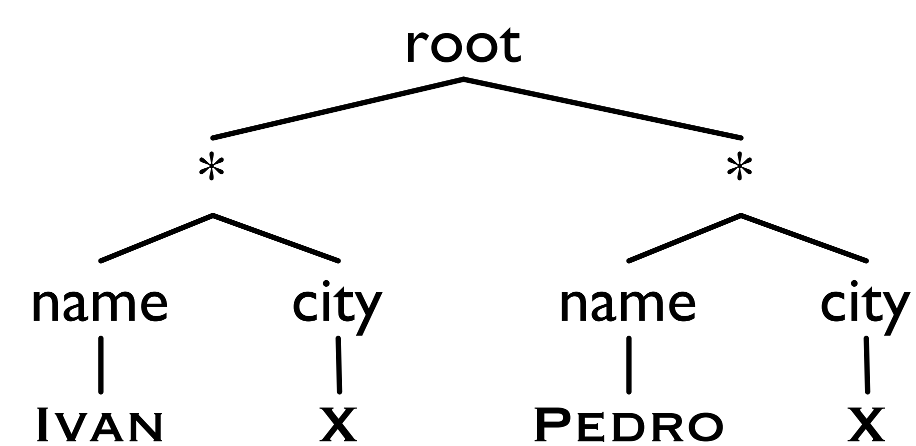
Person

name	city	probability
Ivan	Moscow	0.3
Jean	Paris	0.8
Pedro	Madrid	0.4



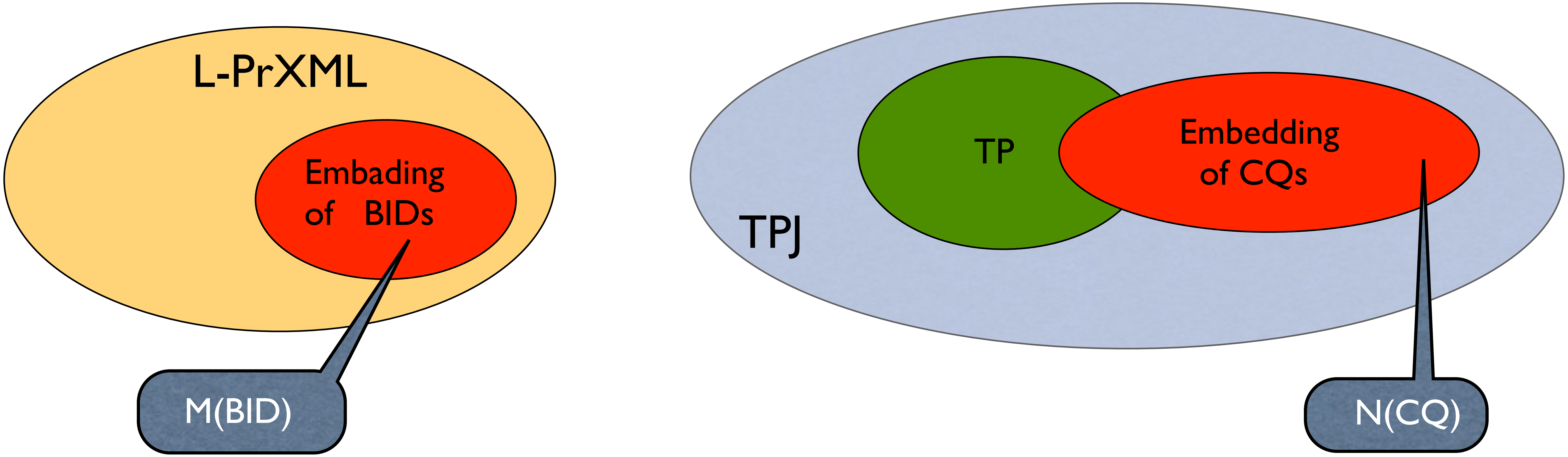
- BIDs can be encoded in PrXML
- Encoding is specific - shallow

Q:- Person(Ivan, x), Person(Jean, x)



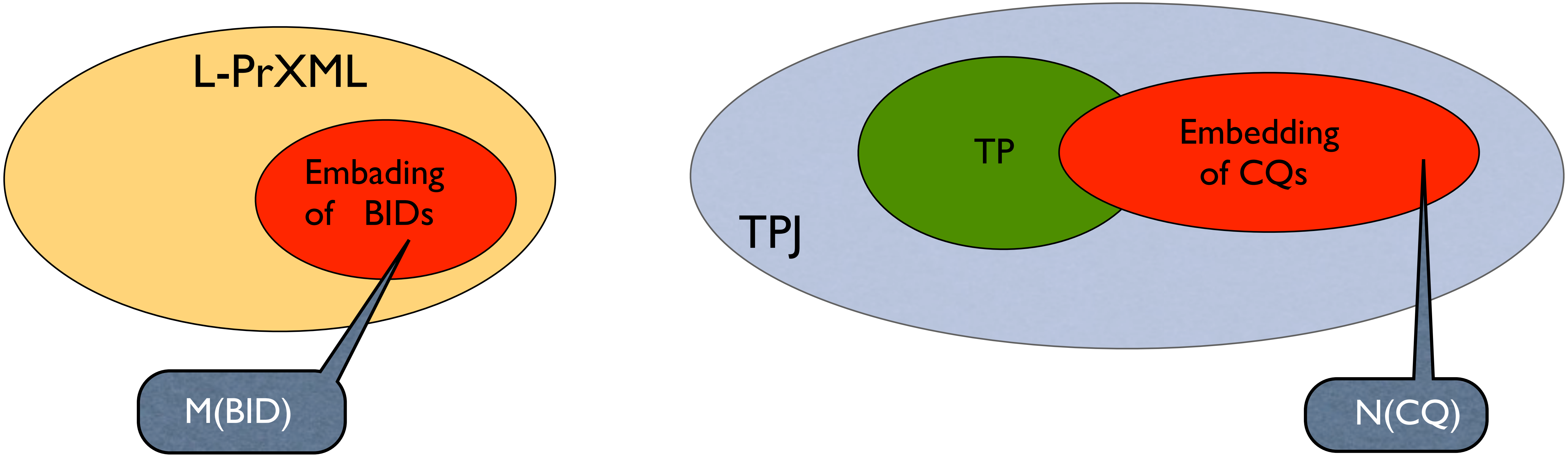
- CQs can be encoded
  - as TP with joins
  - TPs of specific shallow form

Queries over Relations vs XML



- Embedding even CQ that are tractable for BIDs gives TPJs that are intractable over the whole L-PrXML
- N(CQ) - intractable for the whole class BID
- N(CQ) is tractable for a fragment of L-PrXML M(BID)

Queries over Relations vs XML



- Embedding even CQ that are tractable for BIDs gives TPJs that are intractable over the whole L-PrXML
- N(CQ) - intractable for the whole class BID
- N(CQ) is tractable for a fragment of L-PrXML M(BID)

Results are not (easily) translatable from BIDs to L-PrXML.

Relationship between BID & CQ vs. L-PrXML & TP is unclear.

Part IV: Updating Probabilistic Databases

- Updates for relational Probabilistic DBs
- Updates for Pr-XML updates

Updating BIDs

Saw-day

ID	witness	car	probability
31	Cathy	Honda	0.5
32	Bob	BMW	0.3

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8

Updating BIDs

Saw-day

ID	witness	car	probability
31	Cathy	Honda	0.5
32	Bob	BMW	0.3

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8

If a person is a day witness
add they to good-witnesses

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8
22	Bob	BMW	0.3

Updating BIDs

Saw-day

ID	witness	car	probability
31	Cathy	Honda	0.5
32	Bob	BMW	0.3

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8

- Is it a good candidate for the update result?

If a person is a day witness
add they to good-witnesses

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8
22	Bob	BMW	0.3

Updating BIDs

Saw-day

ID	witness	car	probability
31	Cathy	Honda	0.5
32	Bob	BMW	0.3

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8

If a person is a day witness
add they to good-witnesses

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8
22	Bob	BMW	0.3

- Is it a good candidate for the update result?
- No!
  the correlation between Saw-day and Good-witness is missing

Updating BIDs

Saw-day

ID	witness	car	probability
31	Cathy	Honda	0.5
32	Bob	BMW	0.3

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8

If a person is a day witness
add they to good-witnesses

Good-witness

ID	witness	car	probability
21	Cathy	Honda	0.8
22	Bob	BMW	0.3

- Is it a good candidate for the update result?
- No!
  the correlation between Saw-day and Good-witness is missing

Theorem:

BIDs are not closed under updates

Updating in MayBMS

Saw-day

ID	witness	car	Lineage
31	Cathy	Honda	z
32	Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2   Pr(z is true) = 0.8
Pr(y is true) = 0.4   Pr(w is true) = 0.5

If a person is a day witness
add her to good-witnesses

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w
22	Bob	BMW	$y \wedge w$

Updating in MayBMS

Saw-day

ID	witness	car	Lineage
31	Cathy	Honda	z
32	Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

If a person is a day witness add her to good-witnesses

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w
22	Bob	BMW	$y \wedge w$

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

- Is it a good candidate for the update result?

Updating in MayBMS

Saw-day

ID	witness	car	Lineage
31	Cathy	Honda	z
32	Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

If a person is a day witness add her to good-witnesses

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w
22	Bob	BMW	$y \wedge w$

Good-witness

ID	witness	car	Lineage
21	Cathy	Honda	w

Pr(x is true) = 0.2 Pr(z is true) = 0.8
Pr(y is true) = 0.4 Pr(w is true) = 0.5

- Is it a good candidate for the update result?
- Yes!
  we keep correlations between Saw-day and Good-witness

Theorem:

Prob. C-Tables are closed under updates

Limitations of Updates for Rel ProbDBs

Saw

witness	car	Lineage
Cathy	Honda	z
Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

```
UPDATE Saw-day
SET car='VW'
WHERE car = 'Honda'
WITH PROB 0.23
```

Limitations of Updates for Rel ProbDBs

Saw

witness	car	Lineage
Cathy	Honda	z
Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

```
UPDATE Saw-day
SET car='VW'
WHERE car = 'Honda'
WITH PROB 0.23
```

Updated table:

Saw

witness	car	Lineage
Cathy	Honda	$z \wedge \neg v$
Bob	BMW	$y \wedge w$
Cathy	VW	$z \wedge v$

v - new bool. rand. variables

s.t. $Pr(v=true) = 0.23$

Limitations of Updates for Rel ProbDBs

Saw

witness	car	Lineage
Cathy	Honda	z
Bob	BMW	$y \wedge w$

y, z, w - ind. bool. rand. variables

```
UPDATE Saw-day
SET car='VW'
WHERE car = 'Honda'
WITH PROB 0.23
```

Updated table:

Saw

witness	car	Lineage
Cathy	Honda	$z \wedge \neg v$
Bob	BMW	$y \wedge w$
Cathy	VW	$z \wedge v$

v - new bool. rand. variables s.t. Pr(v=true) = 0.23

- Updating single values is problematic
  - value update requires to modify the whole tuple
  - update require tuple duplication
- Value updates are more natural for PrXML

Part IV: Updating Probabilistic Databases

- Updates for relational Probabilistic DBs
- Updates for Pr-XML updates
  - Structure and types
  - Two semantics
  - Updates for continuous PrXML

Part IV: Updating Probabilistic Databases

- Updates for relational Probabilistic DBs
- Updates for Pr-XML updates
  - Structure and types
  - Two semantics
  - Updates for continuous PrXML

Update Operations

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

⇒ We want to insert (delete) data in PXML.

We want to do it conditionally.

Update Operations

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

⇒ We want to insert (delete) data in PXML.

We want to do it conditionally.

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

q - condition query (formally will be defined later)

n - locator of the update

t - the actual new data (tree) to be inserted

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

q - condition query (formally will be defined later)

n - locator of the update

t - the actual new data (tree) to be inserted

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

q - condition query (formally will be defined later)

n - locator of the update

t - the actual new data (tree) to be inserted

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*

- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

q - condition query (formally will be defined later)

n - locator of the update

t - the actual new data (tree) to be inserted

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

q - condition query (formally will be defined later)

n - locator of the update

t - the actual new data (tree) to be inserted

Structure of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*

Update operation (q, n, t): $q^{n,t}$

Inspired by 2 update languages for XML

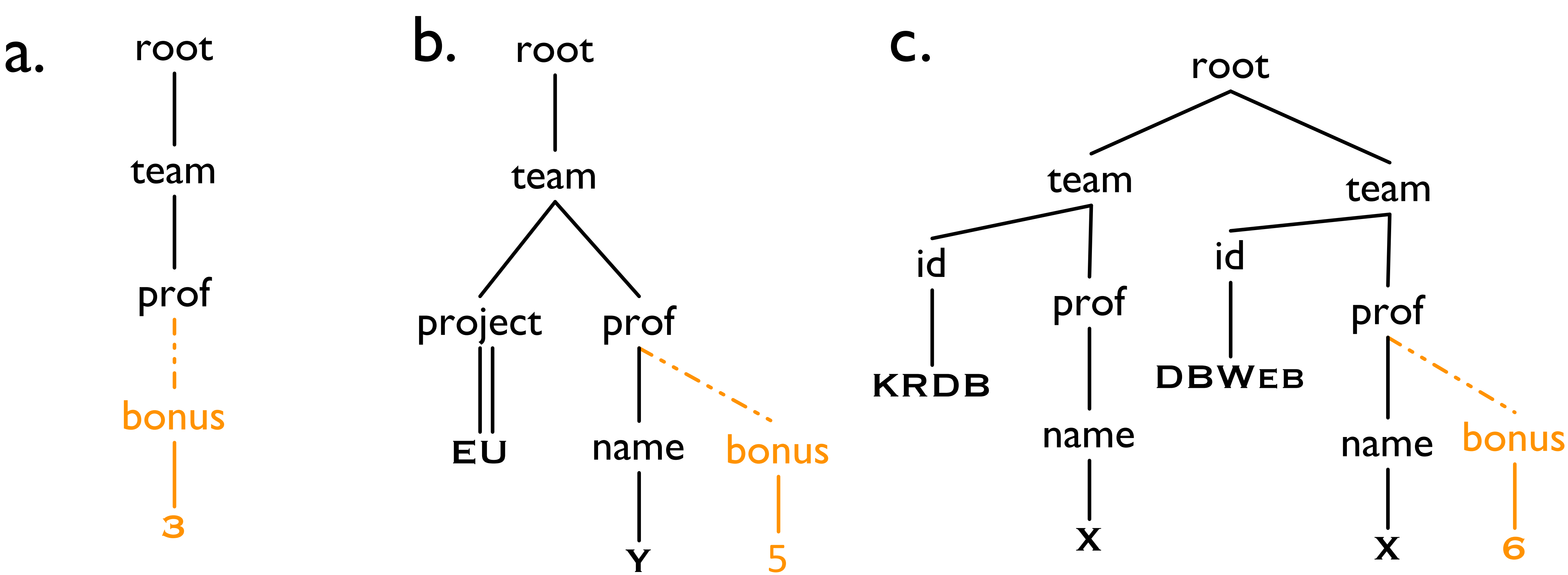
- XUpdate, based on XPath
- XQuery Update Facility, based on XQuery

Types of Updates

a. (Restricted) Single-Path updates - (R)SP

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

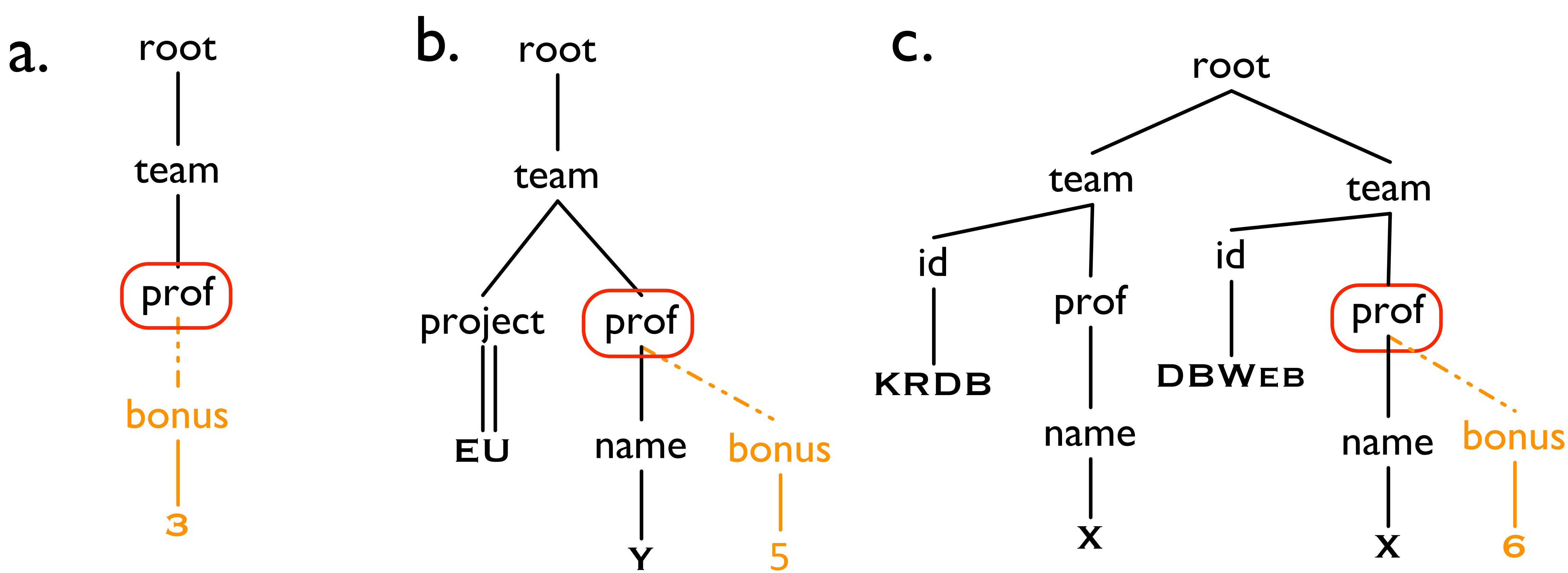


Types of Updates

a. (Restricted) Single-Path updates - (R)SP

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

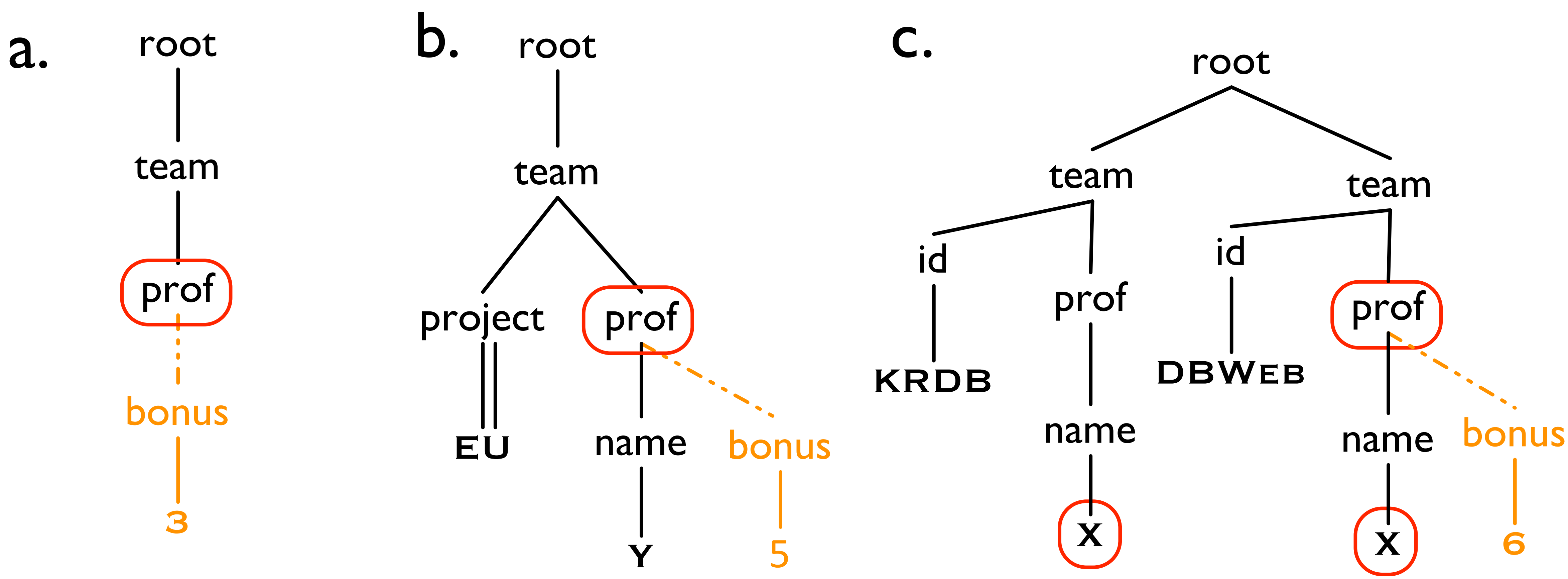


Types of Updates

a. (Restricted) Single-Path updates - (R)SP

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

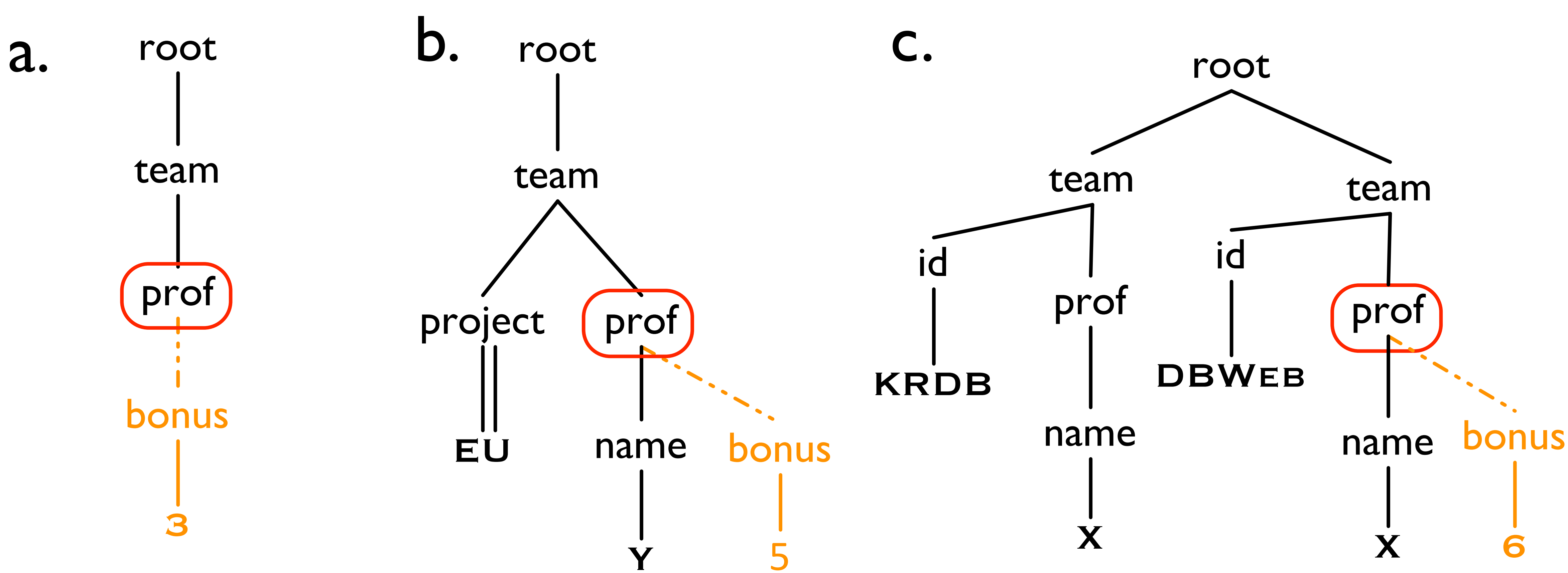


Types of Updates

a. (Restricted) Single-Path updates - (R)SP

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ



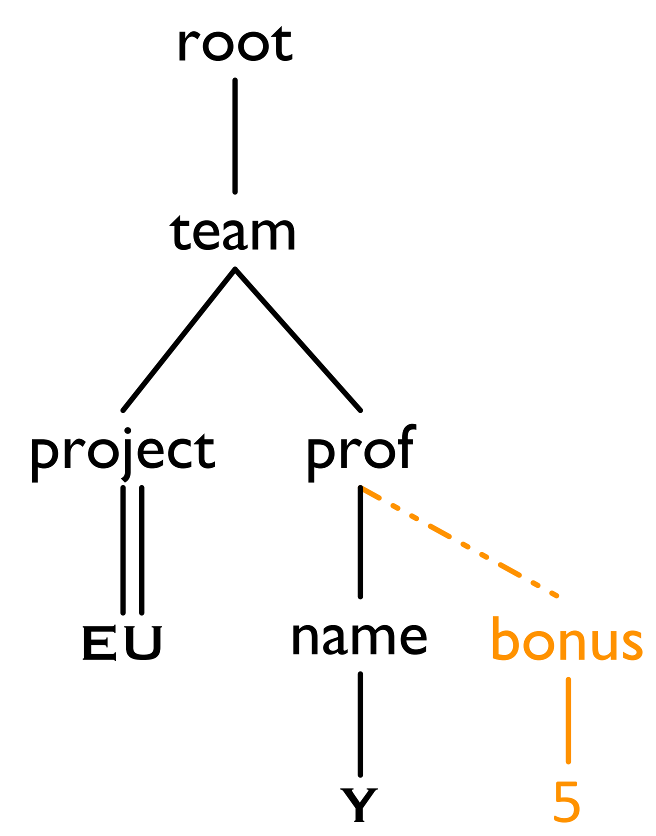
Types of Updates

- *For every professor, insert a bonus of 5 only if her team is in some EU project*
    - Only-if semantics:
      Inserts at most one bonus per professor
- *For every professor, insert a bonus of X for all EU projects with a duration of X years, that her team is involved in*
    - For-all semantics:
      Inserts possibly many bonuses for professors

Semantics of Updates for XML Documents

- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

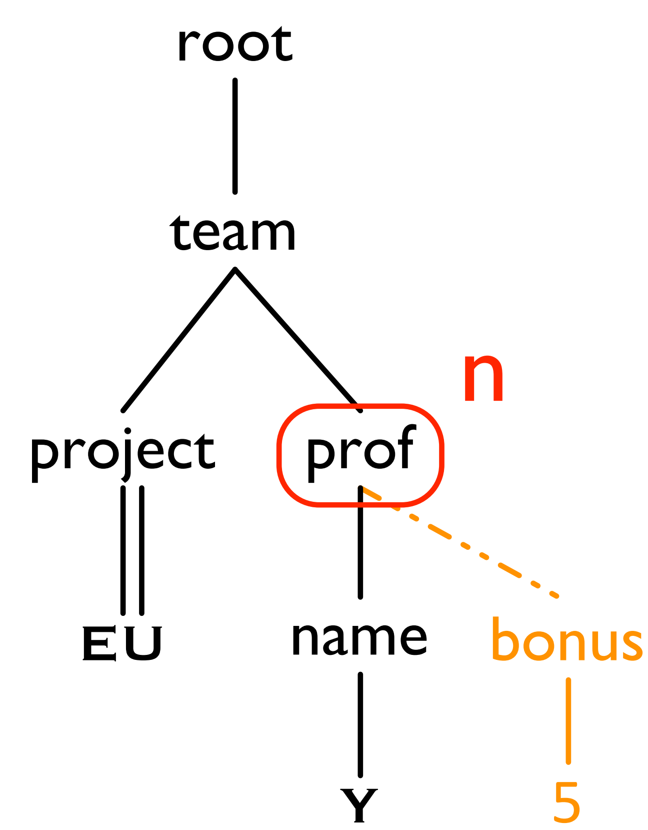
- For-all semantics:
  For every match of n,
  for all k matches of q,
  insert t under n k-times



Semantics of Updates for XML Documents

- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

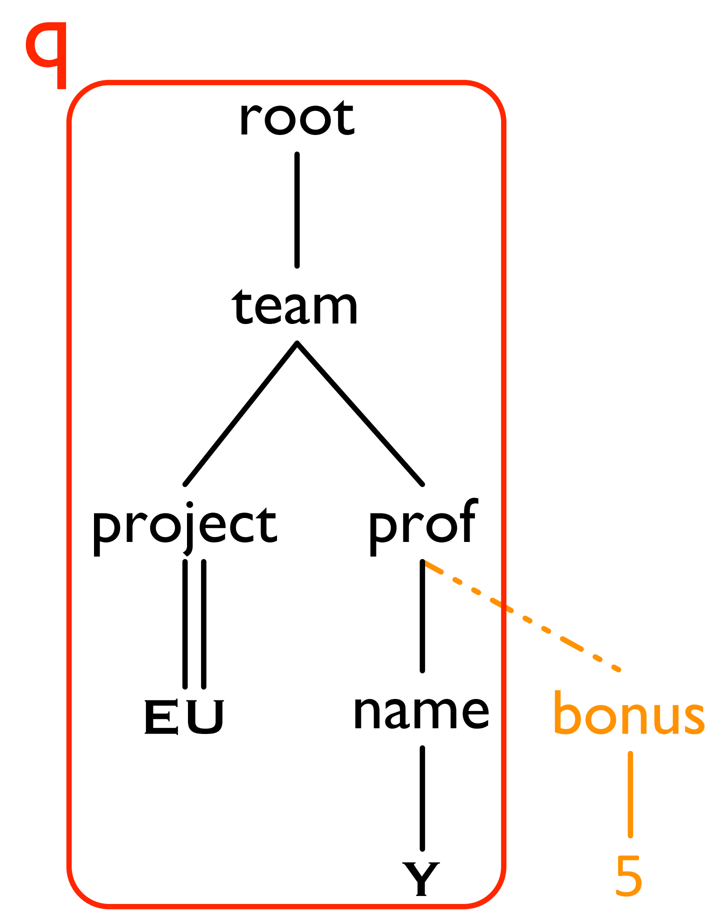
- For-all semantics:
  For every match of n,
  for all k matches of q,
  insert t under n k-times



Semantics of Updates for XML Documents

- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

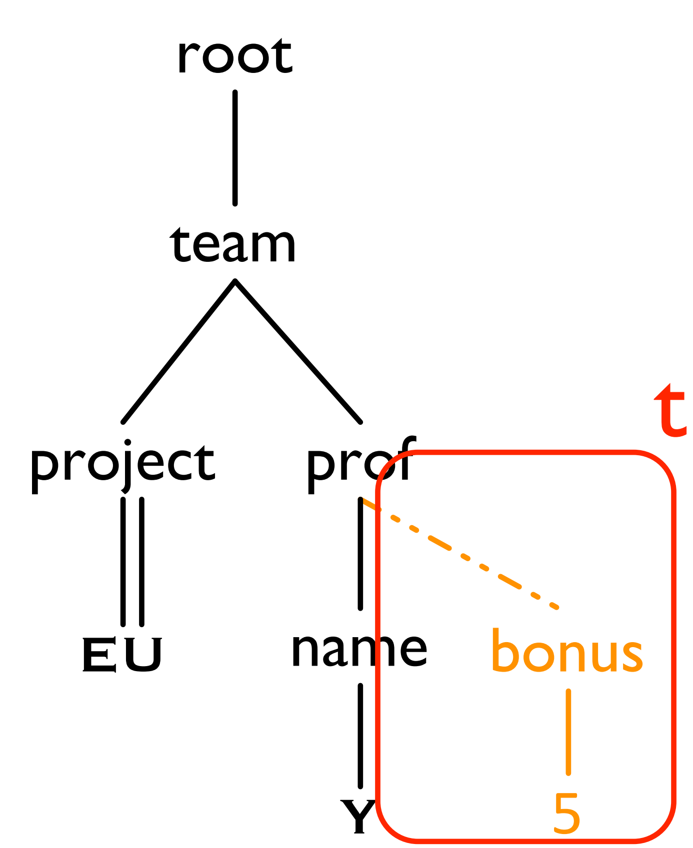
- For-all semantics:
  For every match of n,
  for all k matches of q,
  insert t under n k-times



Semantics of Updates for XML Documents

- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

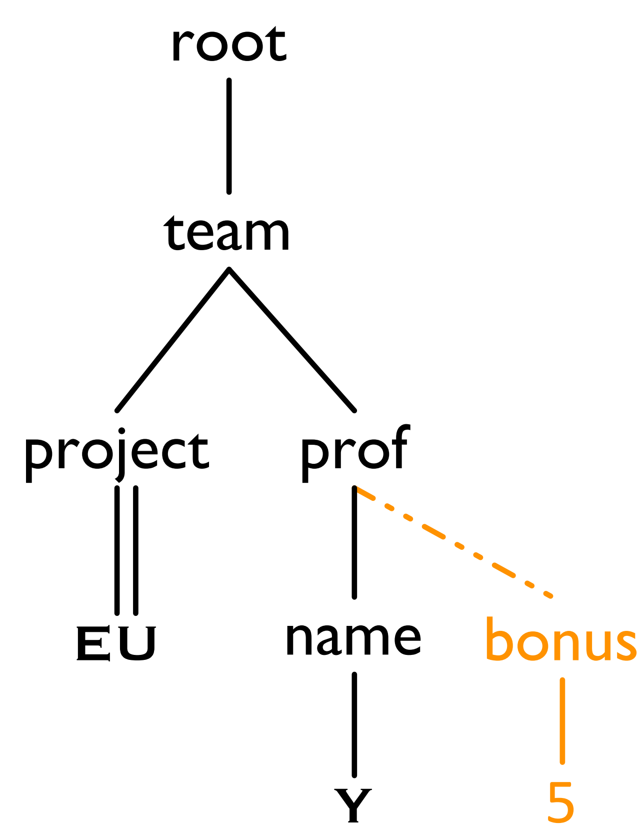
- For-all semantics:
  For every match of n,
  for all k matches of q,
  insert t under n k-times



Semantics of Updates for XML Documents

- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

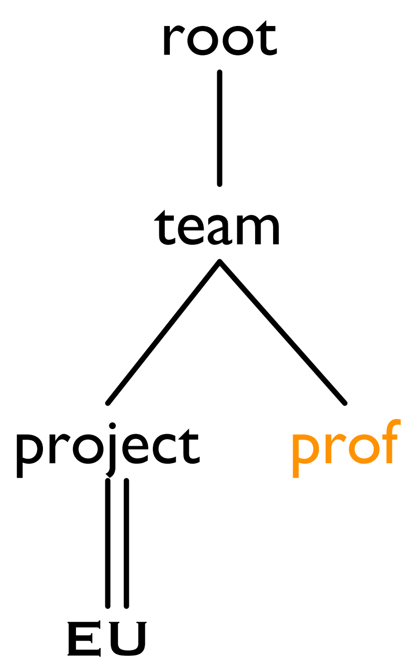
- For-all semantics:
  For every match of n,
  for all k matches of q,
  insert t under n k-times



Deletions

Deletion operation: (q, n)

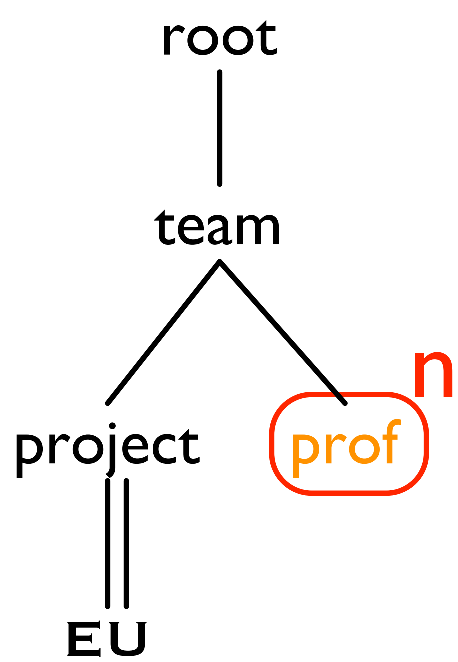
- *Fire a professor if her team is in a EU project*
- For every match of n, if there is a match of q, then delete n and all its descendants
- There is only one semantics for deletions, that is similar to Only-if semantics



Deletions

Deletion operation: (q, n)

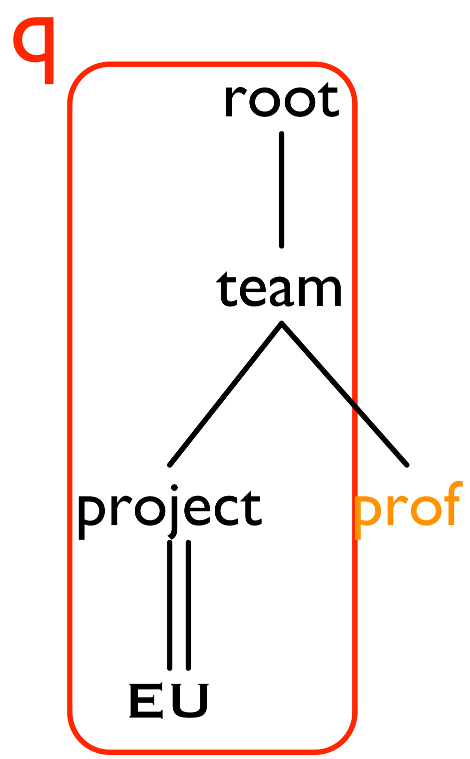
- *Fire a professor if her team is in a EU project*
- For every match of n, if there is a match of q, then delete n and all its descendants
- There is only one semantics for deletions, that is similar to Only-if semantics



Deletions

Deletion operation: (q, n)

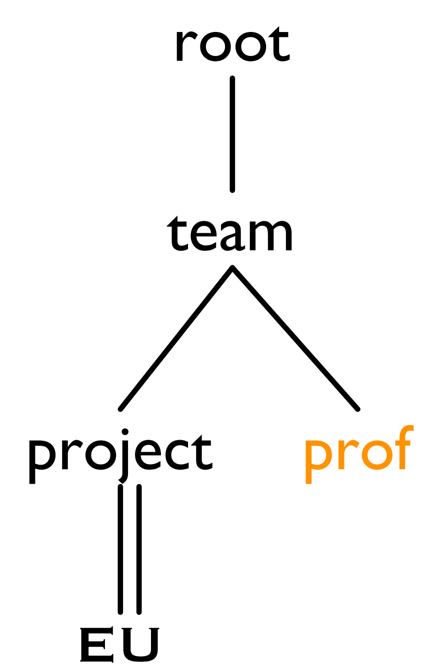
- *Fire a professor if her team is in a EU project*
- For every match of n, if there is a match of q, then delete n and all its descendants
- There is only one semantics for deletions, that is similar to Only-if semantics



Deletions

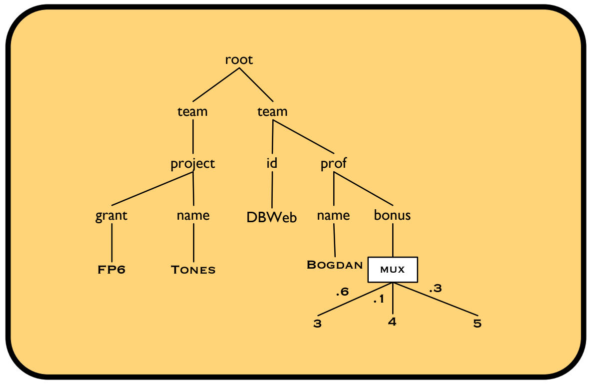
Deletion operation: (q, n)

- *Fire a professor if her team is in a EU project*
- For every match of n, if there is a match of q, then delete n and all its descendants
- There is only one semantics for deletions, that is similar to Only-if semantics

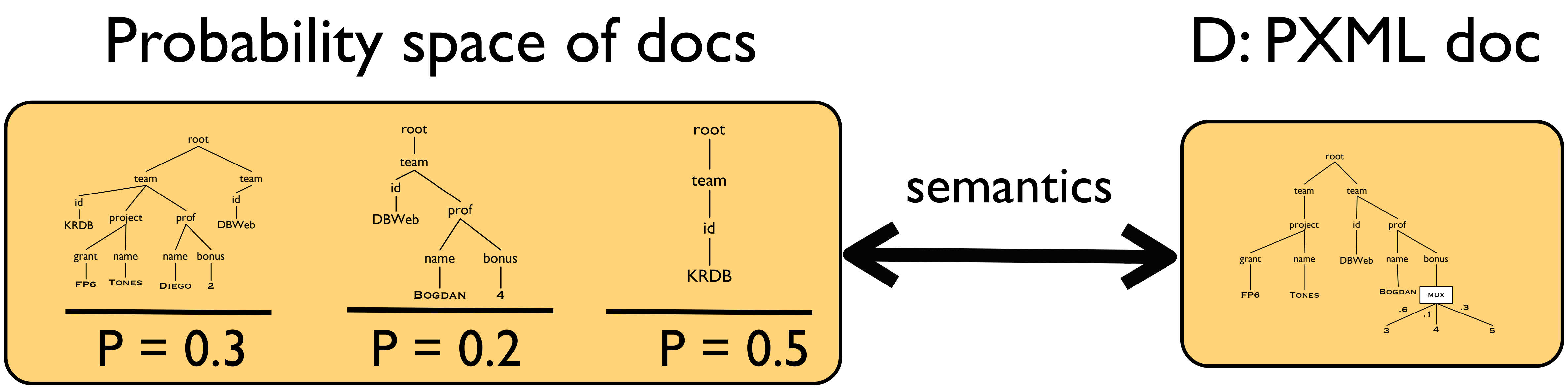


Updating PXML Documents

D: PXML doc



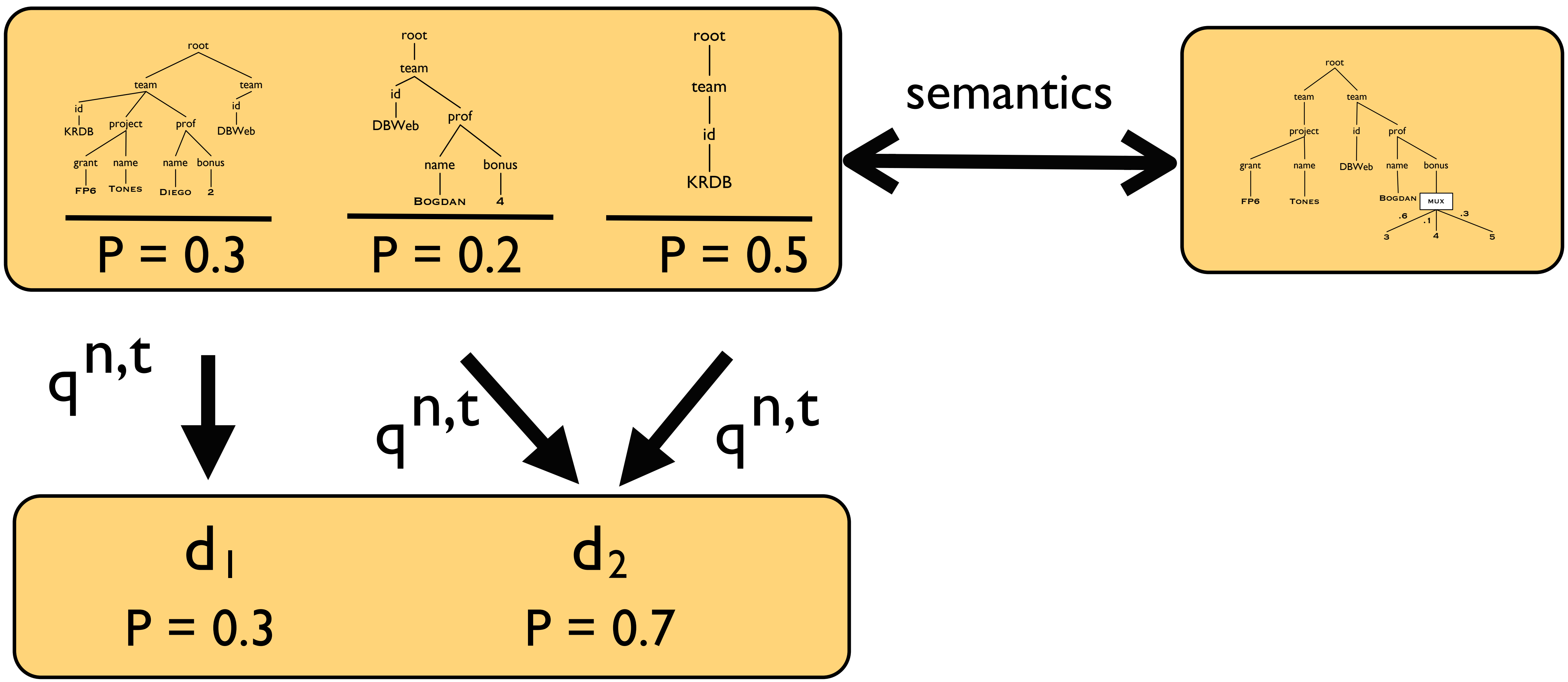
Updating PXML Documents



Updating PXML Documents

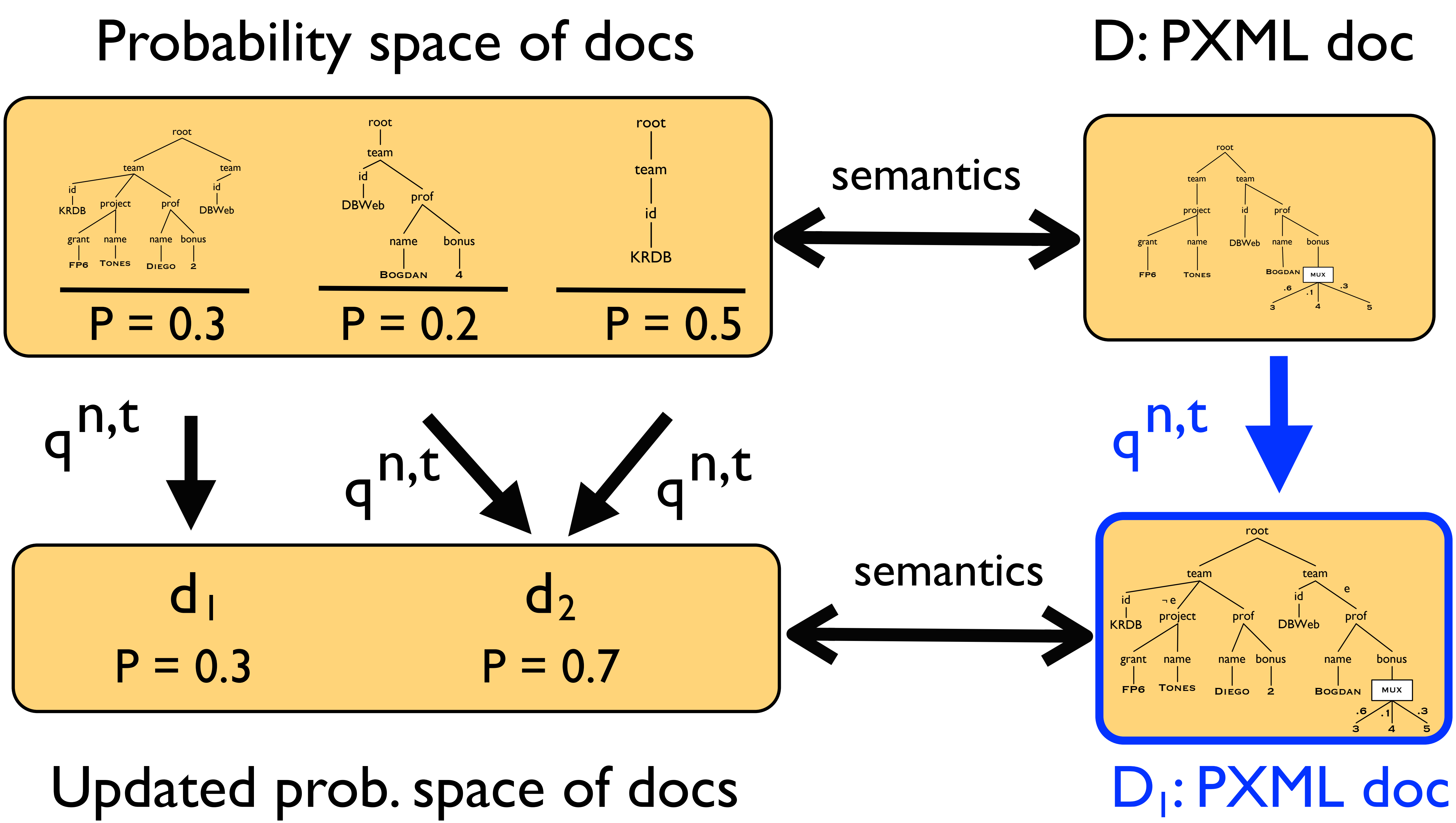
Probability space of docs

D: PXML doc



Updated prob. space of docs

Updating PXML Documents



Problems to Investigate

- Computation of representations of updates
- Given a p-document D and update operation $q^{n,t}$
  - Is it possible to compute a p-document D that represents the update?
  - How hard is the computation?

Only-if Insertions: Data Complexity

<table>
<tr><th>Only-if</th><th>Distr. nodes</th><th>Event conjunct</th><th>Event formulas</th></tr>
<tr><td>RSP</td><td colspan="3">Linear</td></tr>
<tr><td>SP</td><td>P *</td><td rowspan="2">#P-hard</td><td>Linear</td></tr>
<tr><td>TP</td><td>?</td><td rowspan="2">P</td></tr>
<tr><td>TPJ</td><td colspan="2">#P-hard</td></tr>
</table>

* only for queries without descendent edges

- The same table holds for deletions

Only-if Insertions: Data Complexity

Only-if	Distr. nodes	Event conjunct	Event formulas
RSP		Linear	
SP	P*	#P-hard	Linear
TP	?		P
TPJ	#P-hard		

* only for queries without descendent edges

- The same table holds for deletions

Only-if Insertions: Data Complexity

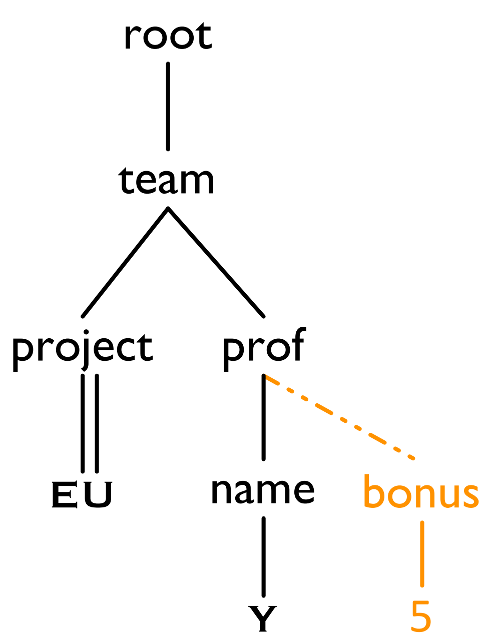
<table>
<tr><th>Only-if</th><th>Distr. nodes</th><th>Event conjunct</th><th>Event formulas</th></tr>
<tr><td>RSP</td><td colspan="3">Linear</td></tr>
<tr><td>SP</td><td>P *</td><td rowspan="2">#P-hard</td><td>Linear</td></tr>
<tr><td>TP</td><td>?</td><td rowspan="2">P</td></tr>
<tr><td>TPJ</td><td colspan="2">#P-hard</td></tr>
</table>

* only for queries without descendent edges

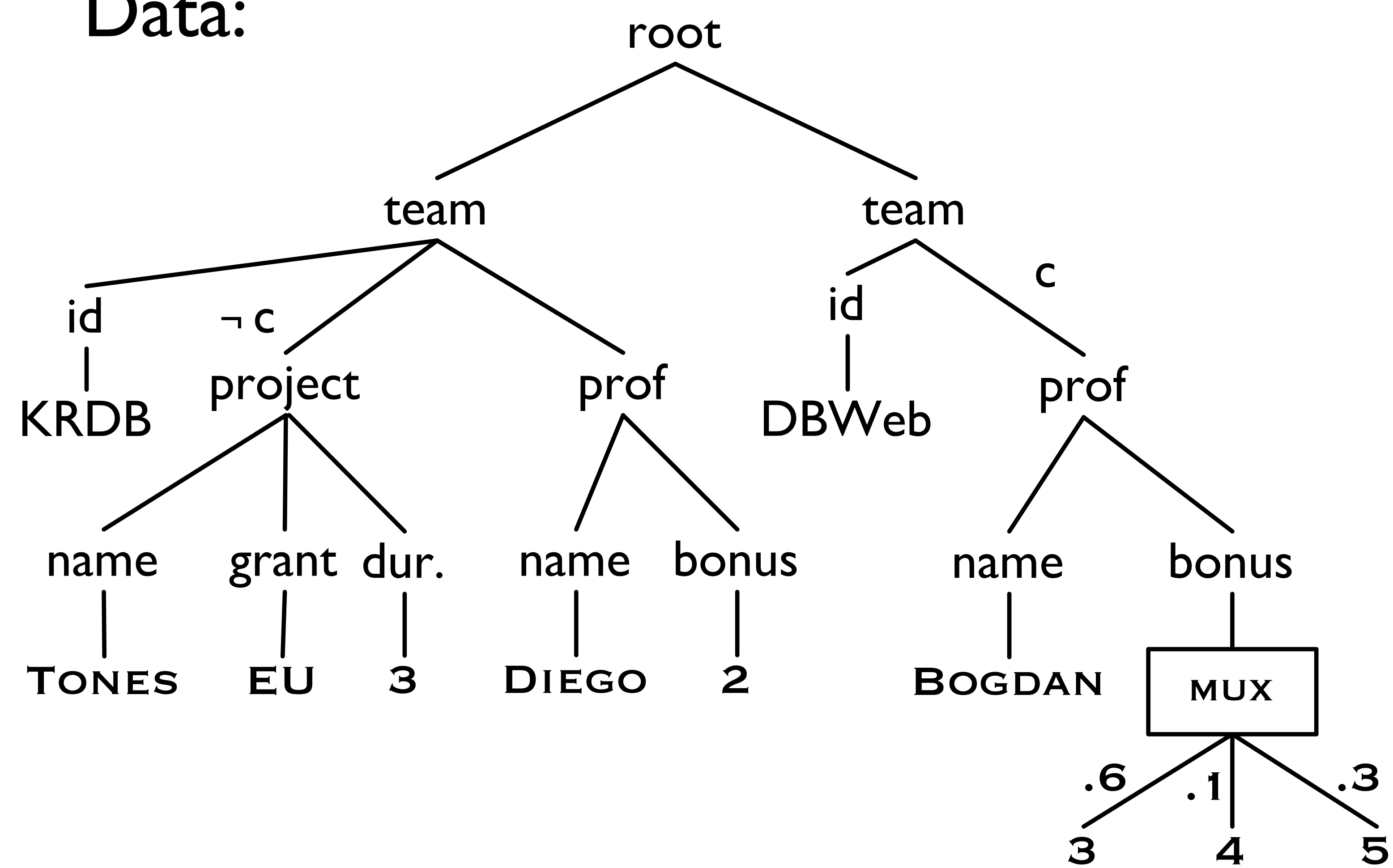
- The same table holds for deletions

Updating PXML: Example

Query:



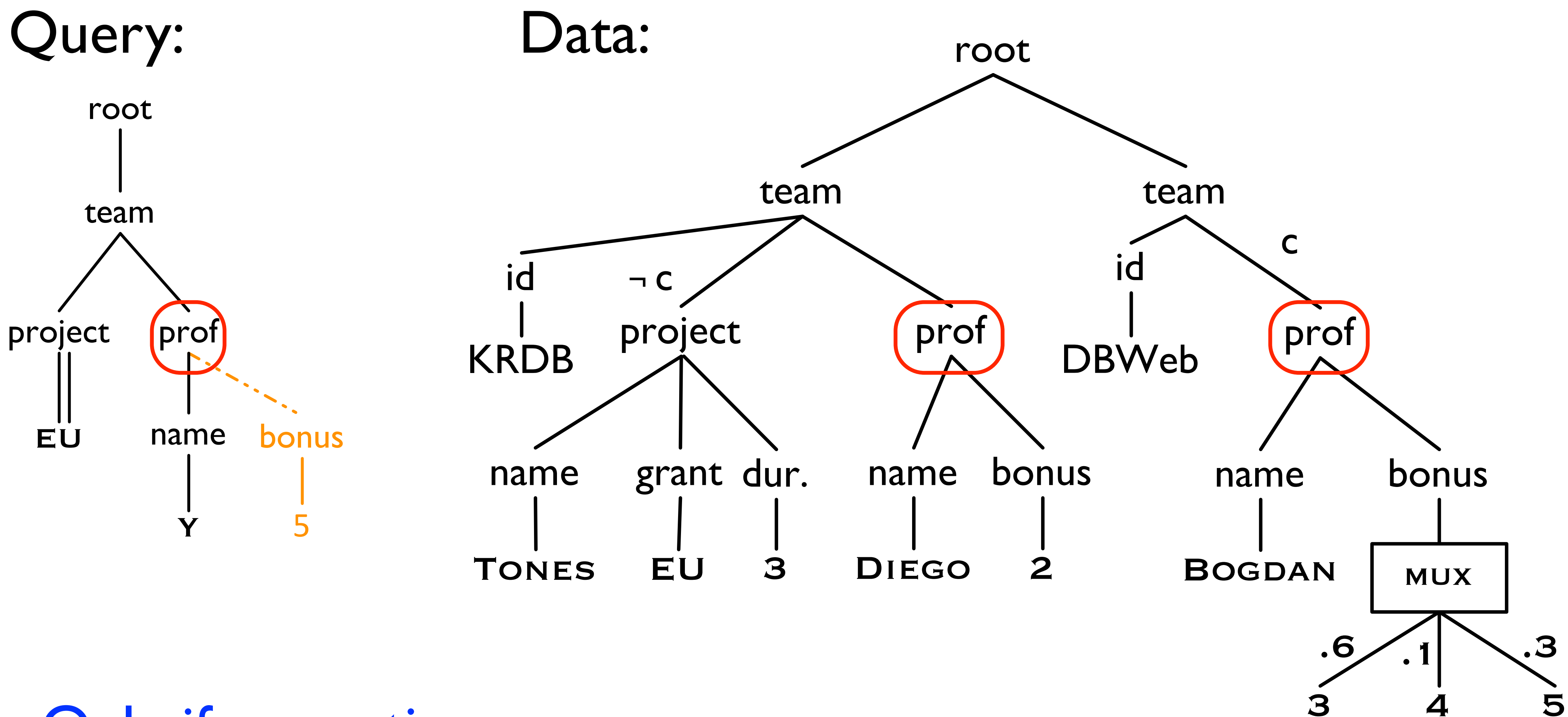
Data:



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

- in this case only-if and for-all semantics coincide

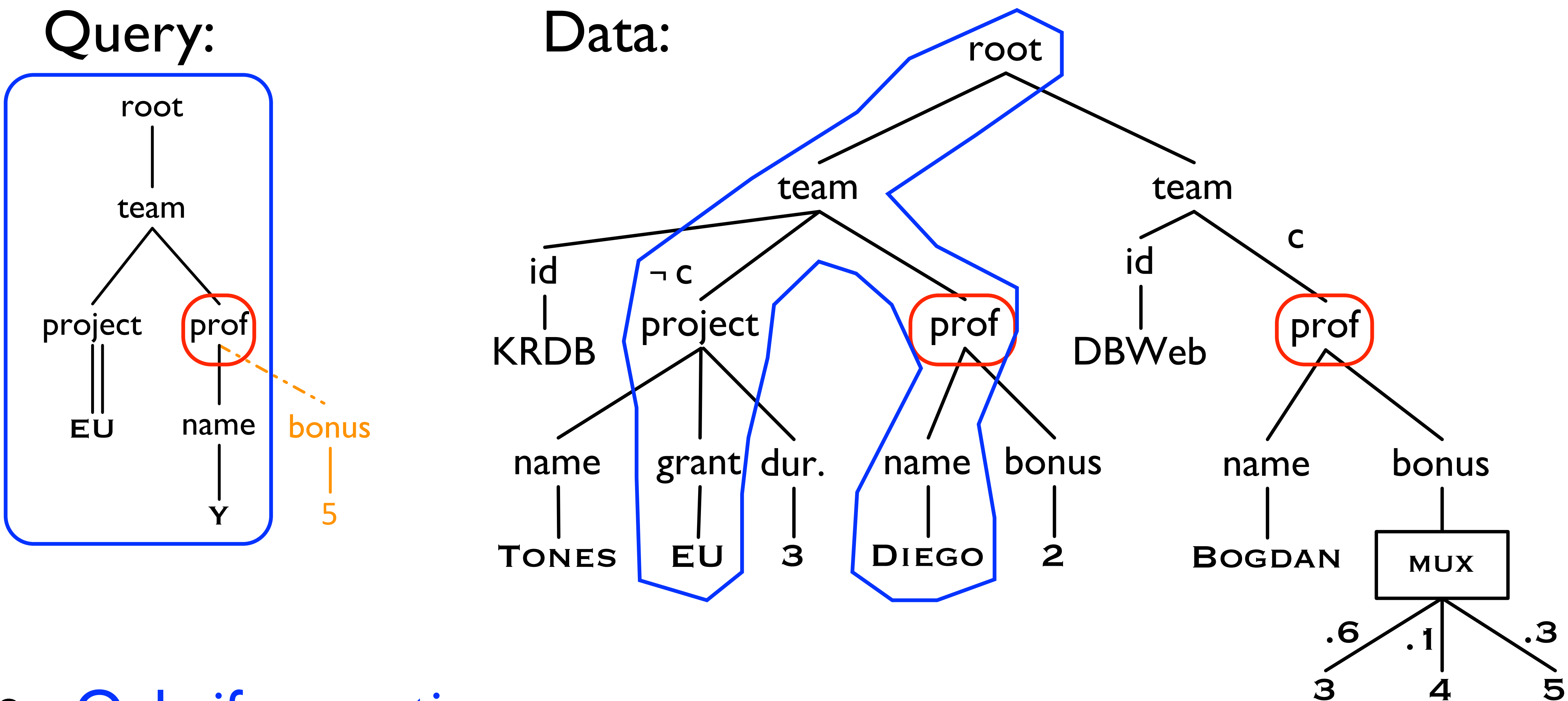
Updating PXML: Example



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

- in this case only-if and for-all semantics coincide

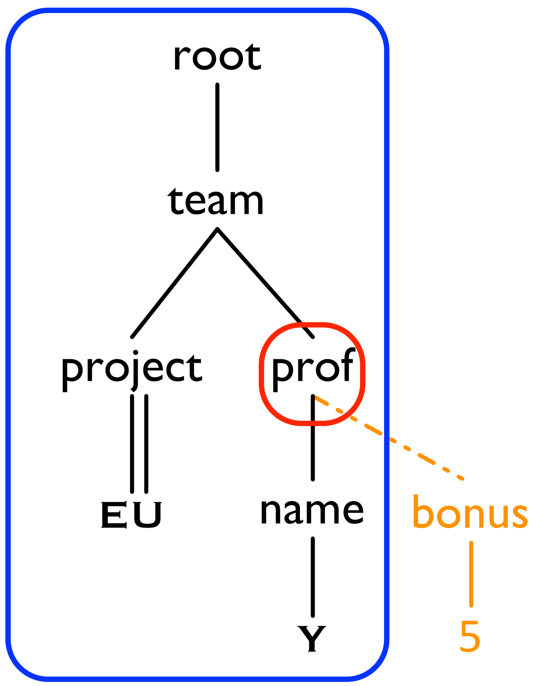
Updating PXML: Example



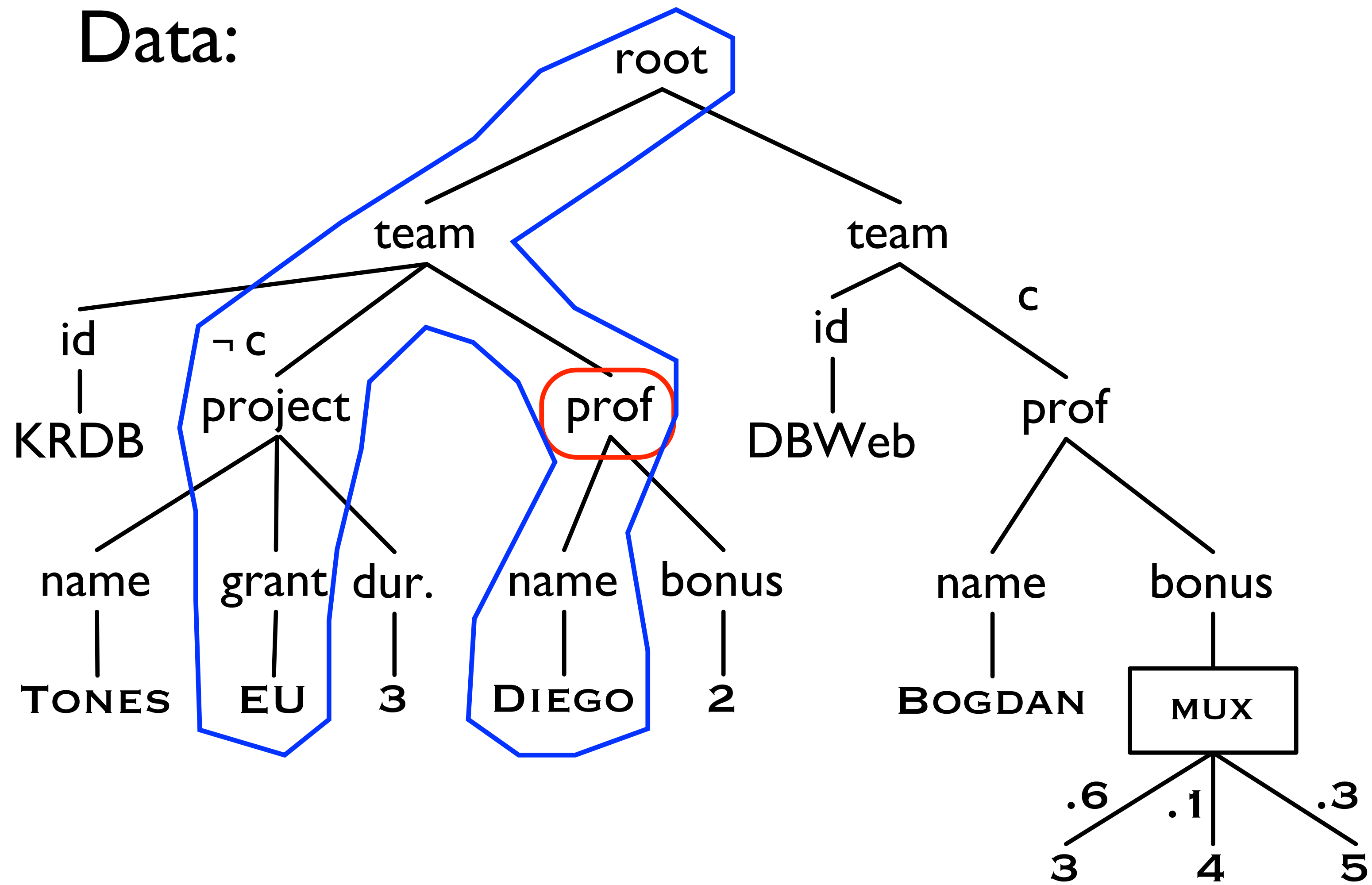
- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n
- in this case only-if and for-all semantics coincide

Updating PXML: Example

Query:



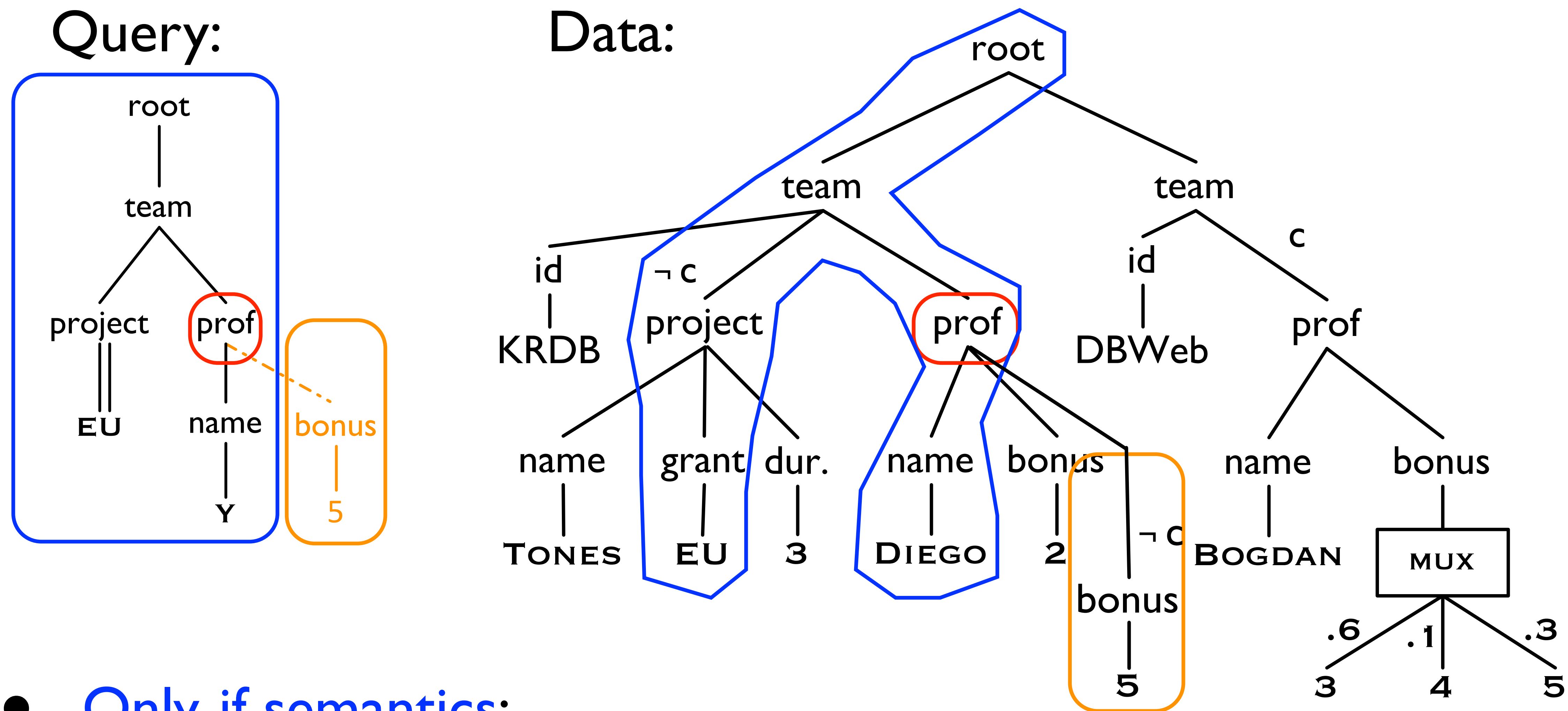
Data:



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

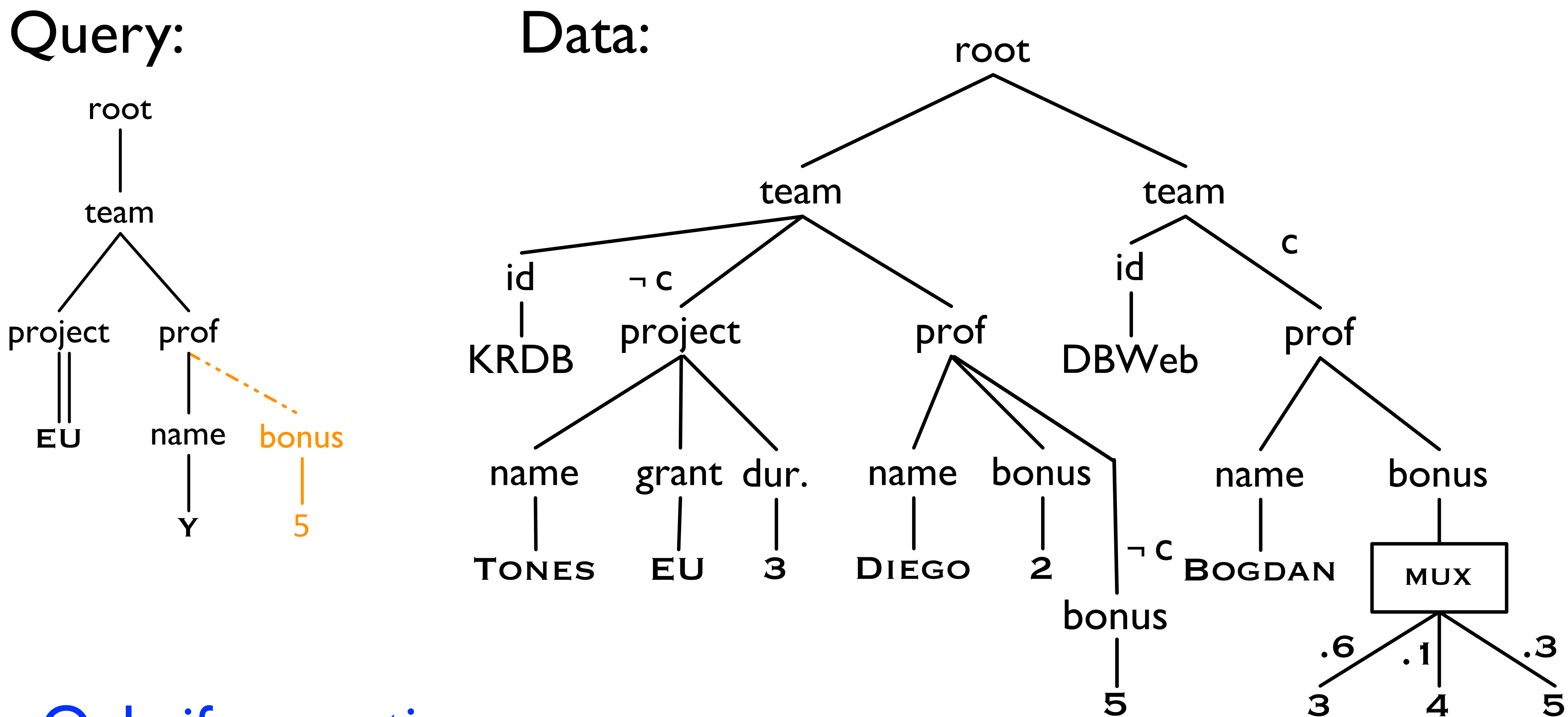
- in this case only-if and for-all semantics coincide

Updating PXML: Example



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n
- in this case only-if and for-all semantics coincide

Updating PXML: Example

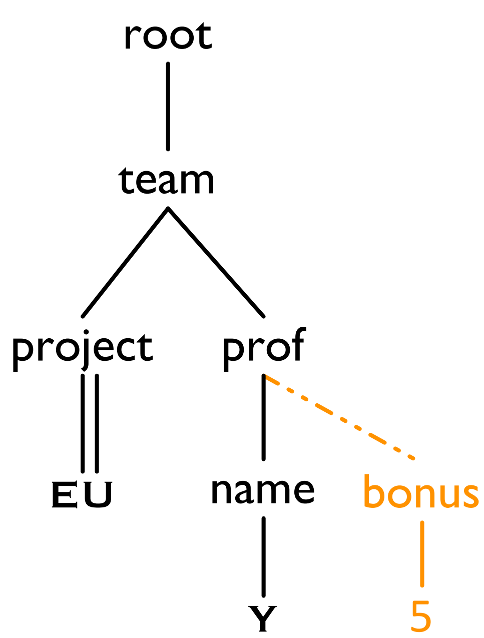


- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n

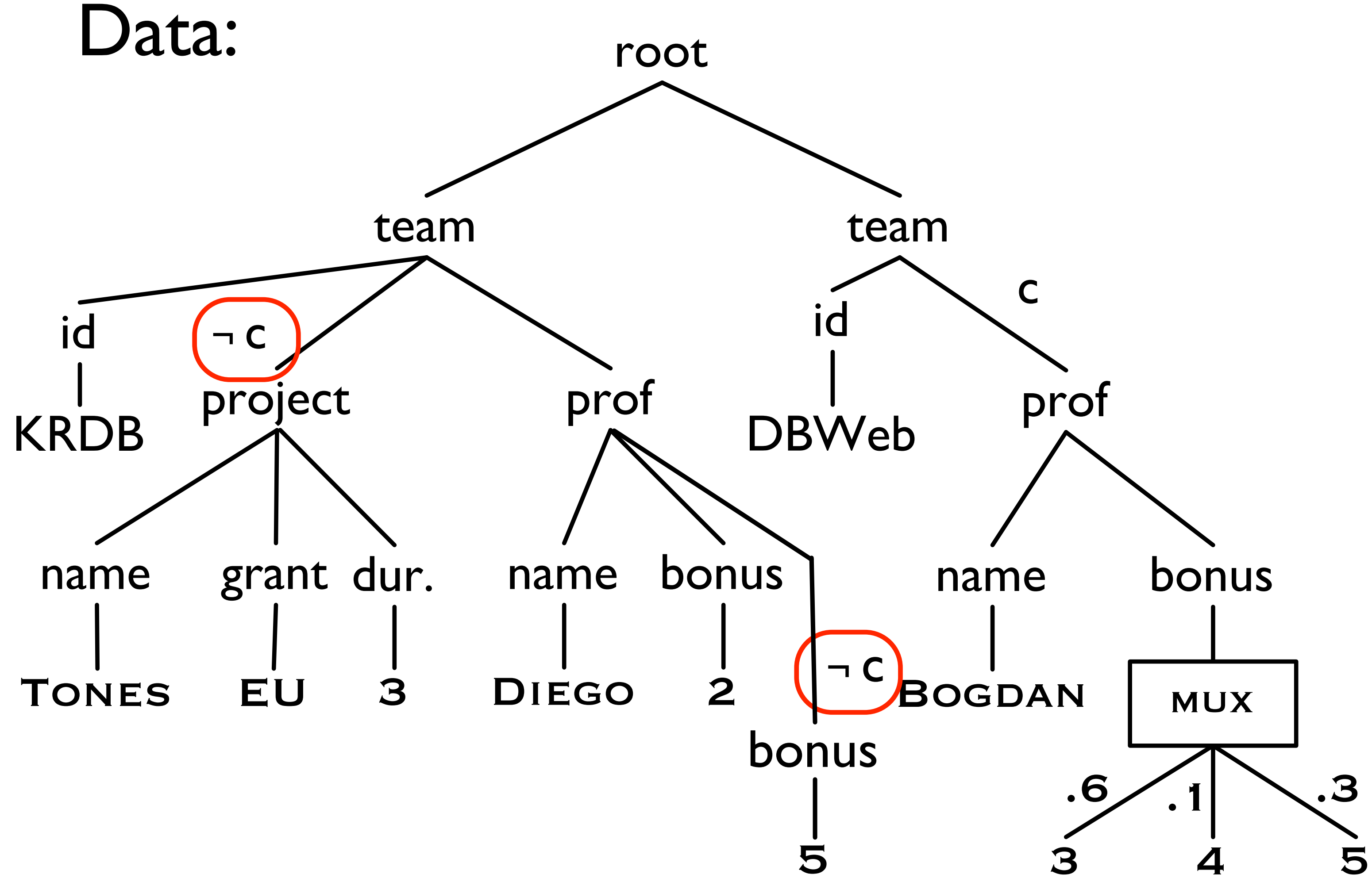
- in this case only-if and for-all semantics coincide

Updating PXML: Example

Query:

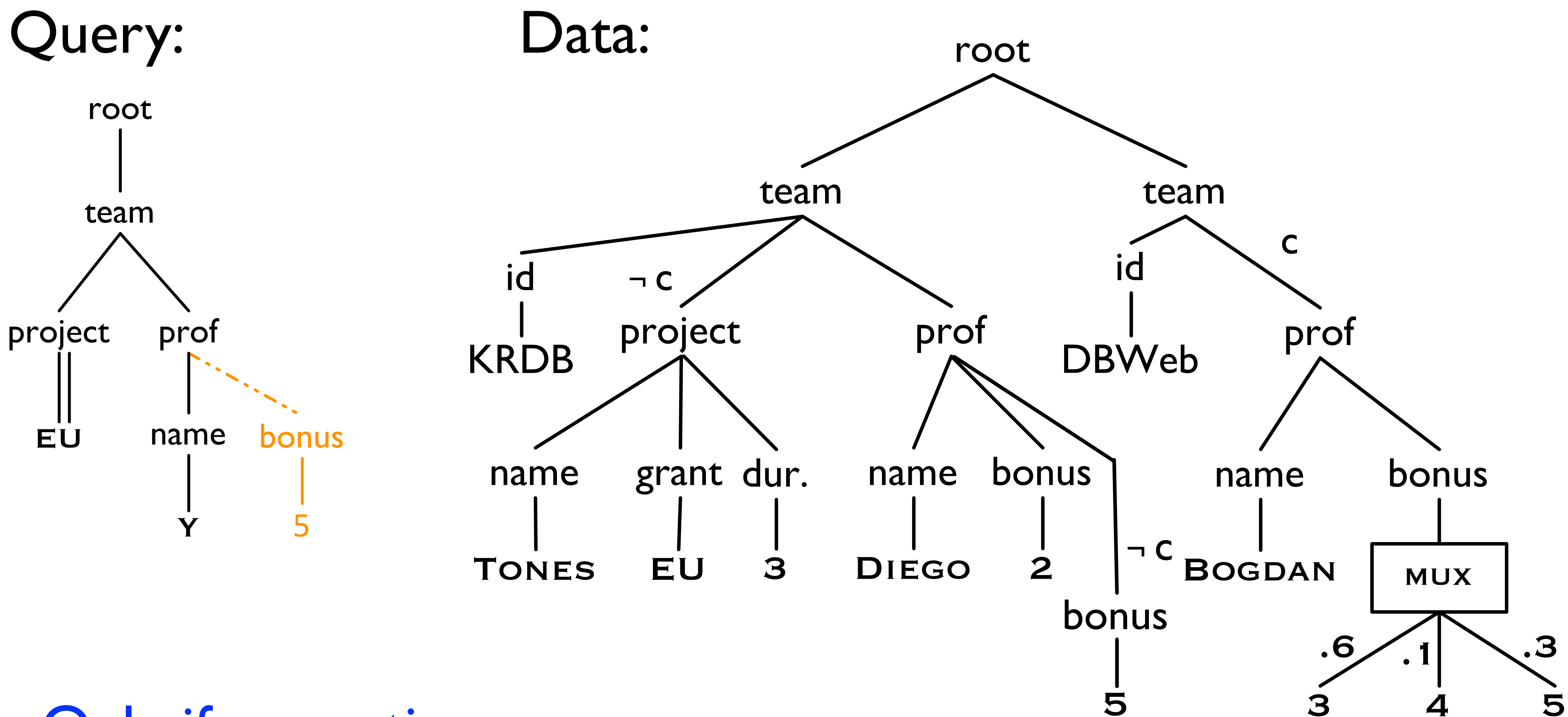


Data:



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n
- in this case only-if and for-all semantics coincide

Updating PXML: Example



- Only-if semantics:
  For every match of n,
  if there is a match of q,
  then insert t under n
- in this case only-if and for-all semantics coincide

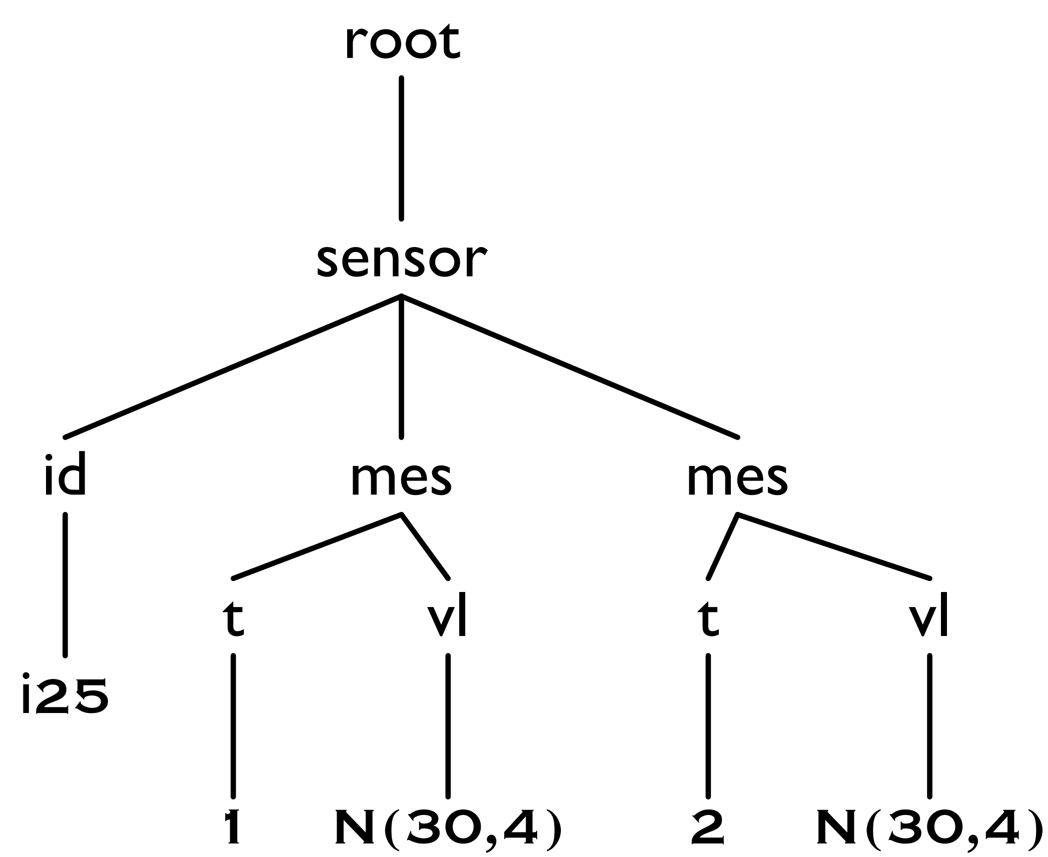
For-all Insertions: Data Complexity

<table>
<tr><th>For-all</th><th>Distributional nodes</th><th>Event conj</th><th>Event formulas</th></tr>
<tr><td>RSP</td><td colspan="3">Linear/P†</td></tr>
<tr><td>SP</td><td>not in PTIME</td><td colspan="2">Linear/P†</td></tr>
<tr><td>TP</td><td>not in PTIME</td><td colspan="2">P</td></tr>
<tr><td>TPJ</td><td>not in PTIME, #P-hard</td><td>P *</td><td>P</td></tr>
</table>

† Linear/P: Linear for queries w/o descendent edges,
Polynomial otherwise

Continuous PXML

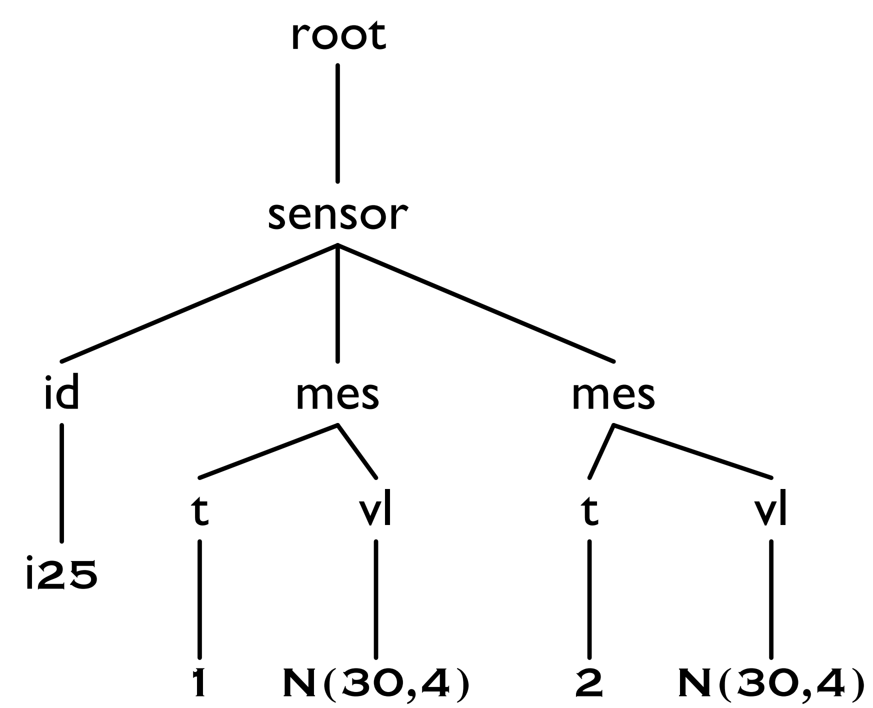
N(30, 4) - Normal distribution



- Probabilistic p-documents with continuous distributions stored on the leaves
- Semantics defined in terms of continuous sets of XML documents

Problems with Updates

- *Insert an alerter "increases"*
  for a sensor only-if
  the second measurement
  is greater than the first one



- probability of the insertion (event) is 1/2
- the update is not representable
  with event formulas and distributions on leaves:
  we need correlations between distributions

Conclusion on PrXML Updates

- Polynomial algorithm for SP update operations without descendent edges
- Results can be generalized to other PXML models and probabilistic updates
- Continuous PXML: problems are highlighted

Part V: To go further

Systems

Trio `http://infolab.stanford.edu/trio/`, useful to see lineage computation

MayBMS `http://maybms.sourceforge.net/`, full-fledged probabilistic relational DBMS, on top of PostgreSQL, usable for actual applications.

ProApproX `http://www.infres.enst.fr/~souihli/Publications.html` to play with various approximation and exact query evaluation methods for probabilistic XML.

Reading material

- An influential paper on incomplete databases [Imieliński and Lipski, 1984]
- A book on probabilistic relational databases, focused around TIDs/BIDs and MayBMS [Suciu et al., 2011]
- An in-depth presentation of MayBMS [Koch, 2009]
- A gentle presentation of relational and XML probabilistic models [Kharlamov and Senellart, 2011]
- A survey of probabilistic XML [Kimelfeld and Senellart, 2011]

Research directions

- Demonstrating the usefulness of probabilistic databases over ad-hoc approach on concrete applications: Web information extraction, data warehousing, scientific data management, etc.
- Understanding better the connection between probabilistic relational databases and probabilistic XML: why does the picture look so different?
- Understanding under which restrictions on the data (e.g., (hyper)tree-width characteristics) query answering can be tractable.
- Connecting probabilistic databases with probabilistic models in general, e.g., as used in machine learning: Bayesian networks, Makov logic networks, factor graphs, etc.
- Other operations on probabilistic data: mining, deduplication, learning, matching, etc.

Thank you!

ACSI Project

Artifact-Centric Service Interoperation

FP 7 grant, agreement n. 257593

http://www.acsi-project.eu/

Webdam Project

Foundations of Web Data Management

ERC FP7 grant, agreement n. 226513

http://webdam.inria.fr/

Serge Abiteboul, Benny Kimelfeld, Yehoshua Sagiv, and Pierre Senellart. On the expressiveness of probabilistic XML models. *VLDB Journal*, 18(5):1041–1064, October 2009.

Pablo Barceló, Leonid Libkin, Antonella Poggi, and Cristina Sirangelo. XML with incomplete information: models, properties, and query answering. In *Proc. Symposium on Principles of Database Systems (PODS)*, pages 237–246, New York, NY, 2009. ACM.

Michael Benedikt, Evgeny Kharlamov, Dan Olteanu, and Pierre Senellart. Probabilistic XML via Markov chains. *Proceedings of the VLDB Endowment*, 3(1):770–781, September 2010. Presented at the VLDB 2010 conference, Singapore.

Omar Benjelloun, Anish Das Sarma, Alon Y. Halevy, and Jennifer Widom. Uldbs: Databases with uncertainty and lineage. In *VLDB*, pages 953–964, 2006.

Nilesh Dalvi, Chrisopher Ré, and Dan Suciu. Probabilistic databases: Diamonds in the dirt. *Communications of the ACM*, 52(7), 2009.

Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875, 2004.

Robert Fink, Andrew Hogue, Dan Olteanu, and Swaroop Rath. SPROUT^2: a squared query engine for uncertain web data. In *SIGMOD*, 2011.

José Galindo, Angelica Urrutia, and Mario Piattini. *Fuzzy Databases: Modeling, Design And Implementation*. IGI Global, 2005.

Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. In *Proc. EDBT Workshops, IIDB*, Munich, Germany, March 2006.

Tomasz Imieliński and Witold Lipski. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984.

Evgeny Kharlamov and Pierre Senellart. Modeling, querying, and mining uncertain XML data. In Andrea Tagarelli, editor, *XML Data Mining: Models, Methods, and Applications*. IGI Global, 2011.

B. Kimelfeld and Y. Sagiv. Matching twigs in probabilistic XML. In *Proc. VLDB*, Vienna, Austria, September 2007.

Benny Kimelfeld and Pierre Senellart. Probabilistic XML: Models and complexity, September 2011. Preprint.

Christoph Koch. MayBMS: A system for managing large uncertain and probabilistic databases. In Charu Aggarwal, editor, *Managing and Mining Uncertain Data*. Springer-Verlag, 2009.

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

Jennifer Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proc. CIDR*, Asilomar, CA, USA, January 2005.

Lotfi A. Zadeh. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2), 1986.