Agrégation de documents XML probabilistes

Serge Abiteboul [1], T.-H. Hubert Chan [2], Evgeny Kharlamov [1,3]
Werner Nutt [3], Pierre Senellart [4]

[1] INRIA Saclay – Île-de-France
[2] The University of Hong-Kong
[3] Free University of Bozen-Bolzano
[4] Télécom ParisTech

Incomplete Databases

An incomplete database D contains many instances

$$D = \{ d_1,..., d_n,...\}$$

Query q(x), constant c

- c is a certain answer for q if $c \in q(d_i)$ for all $d_i \in D$

- c is a possible answer for q if $c \in q(d_i)$ for some $d_i \in D$

Many ways to represent incomplete databases

# Probabilistic Databases

Incomplete database $D = \{ d_1,..., d_n \}$

- with probabilities $Pr(d_i) > 0$ for each instance
- such that $Pr(d_1) + ... + Pr(d_n) = 1$

Query q returns constant c with probability p if

$$p = \sum_{c \in q(d_i)} Pr(d_i)$$

- Mainly studied in the relational setting
- Imprecise data on the Web $\Rightarrow$ Probabilistic XML

Personnel Data, Instance 1
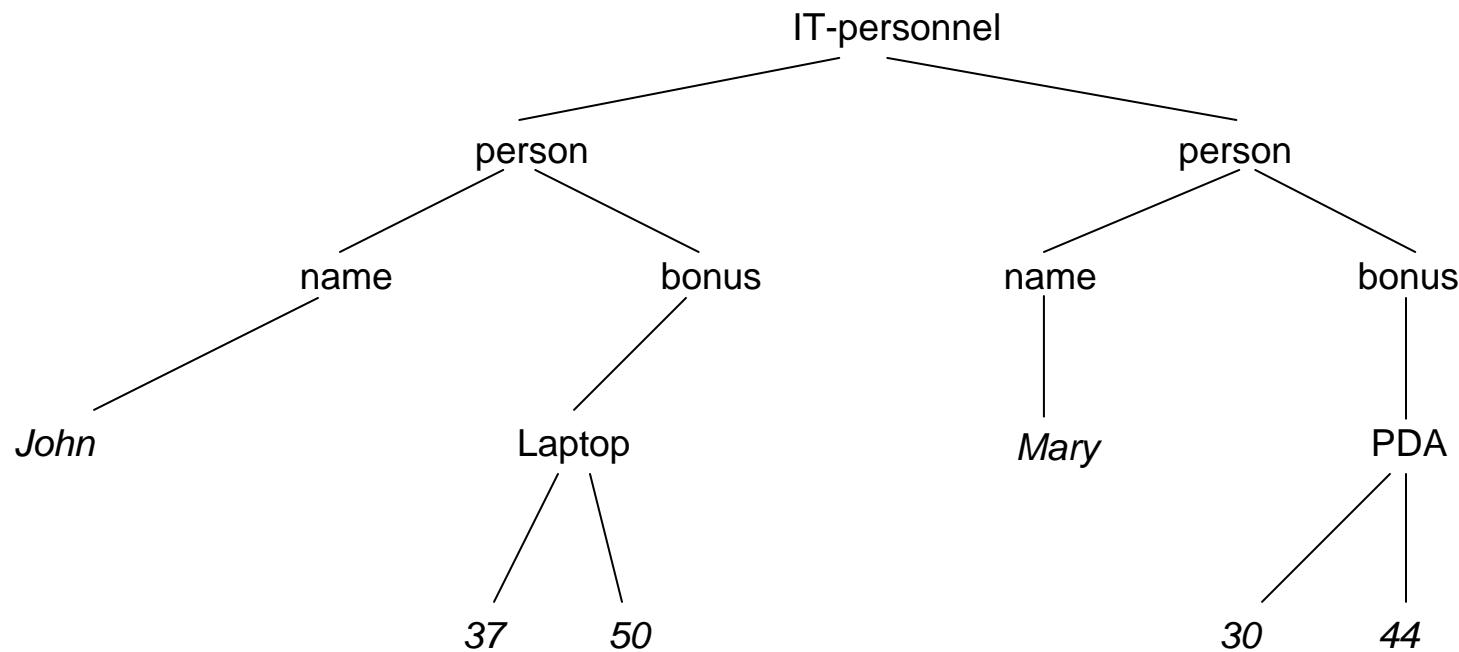
Personnel Data, Instance 2
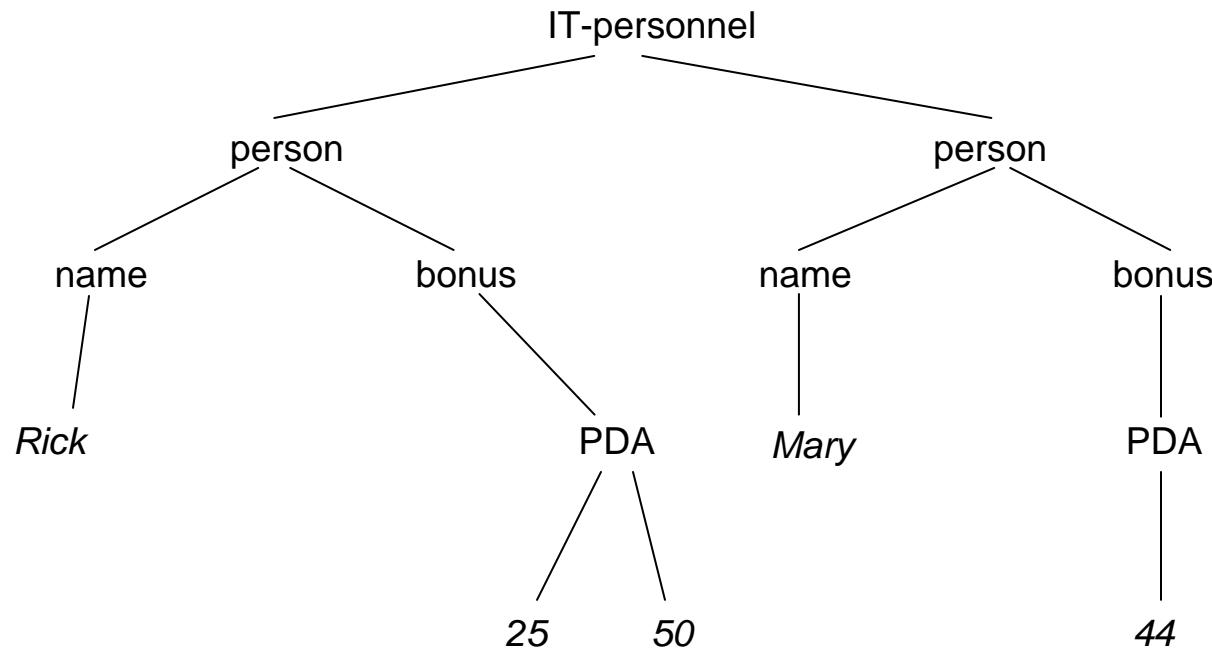
Example: Personnel Queries

"What are the names of the IT personnel?"

"What bonuses were paid for the PDA project?"

"What is the sum of bonuses paid to all employees?"

Personnel DB: Certain/Possible Answers

"What are the names of the IT personnel?"

    Mary: certain   Rick: possible

"What bonuses were paid for the PDA project?"

    44: certain   15: possible

"What is the sum of bonuses paid to all employees?"

    no certain answer   161, 119: possible

$\Rightarrow$ *Aggregate queries depend*
   *on the presence of many data*

If We Had Probabilities …

… we could ask

- What is the probability that the sum of bonuses = 161?

- What are all possible sums of bonuses?
  And what is each one's probability?

- What is the expected value of the sum of bonuses?
  And what the variance?

Distribution of sums of bonuses

Moments

The Problem Space

Events

Probabilistic XML
Document Models

Our focus

Distributional
Nodes
(MUX-DET)

COUNT   SUM   MIN   COUNTD   AVG

Aggregate Function

Single Path Queries

Tree Pattern Queries

Query
Language

Tree Pattern Queries
with Joins
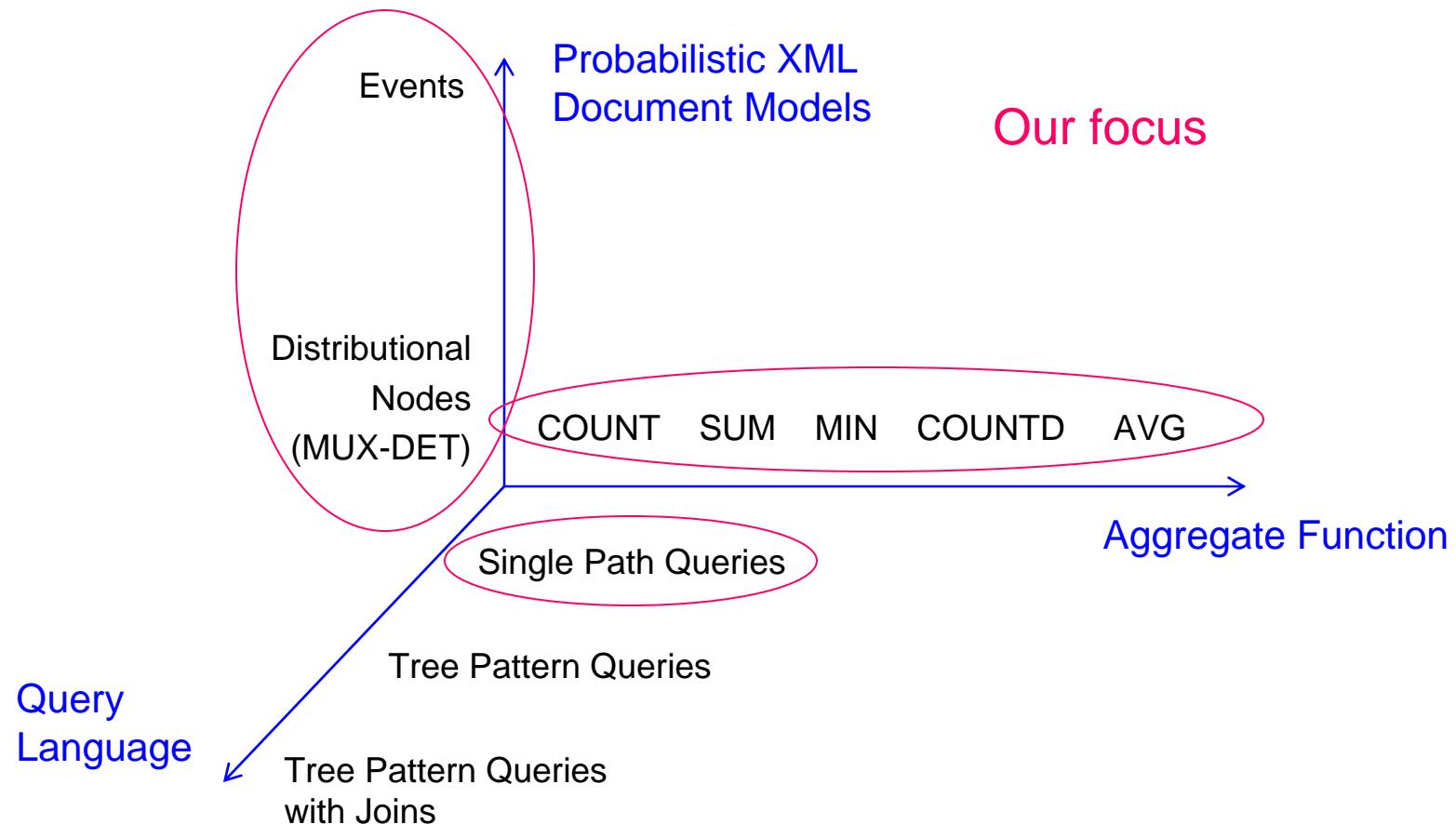
Probabilistic XML: Events   [Abiteboul/Senellart]
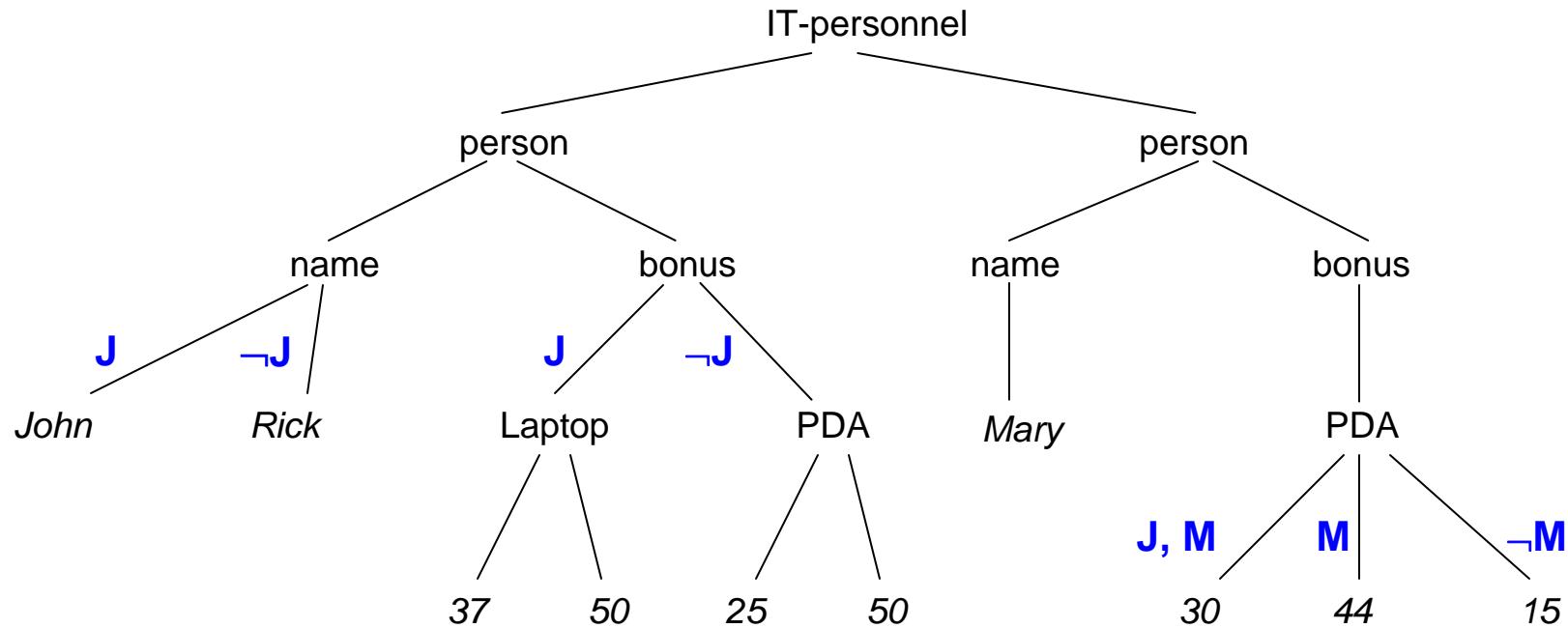
Independent Events

J: John hired for Laptop project

Pr(J) = 0.3

Probabilities of Events
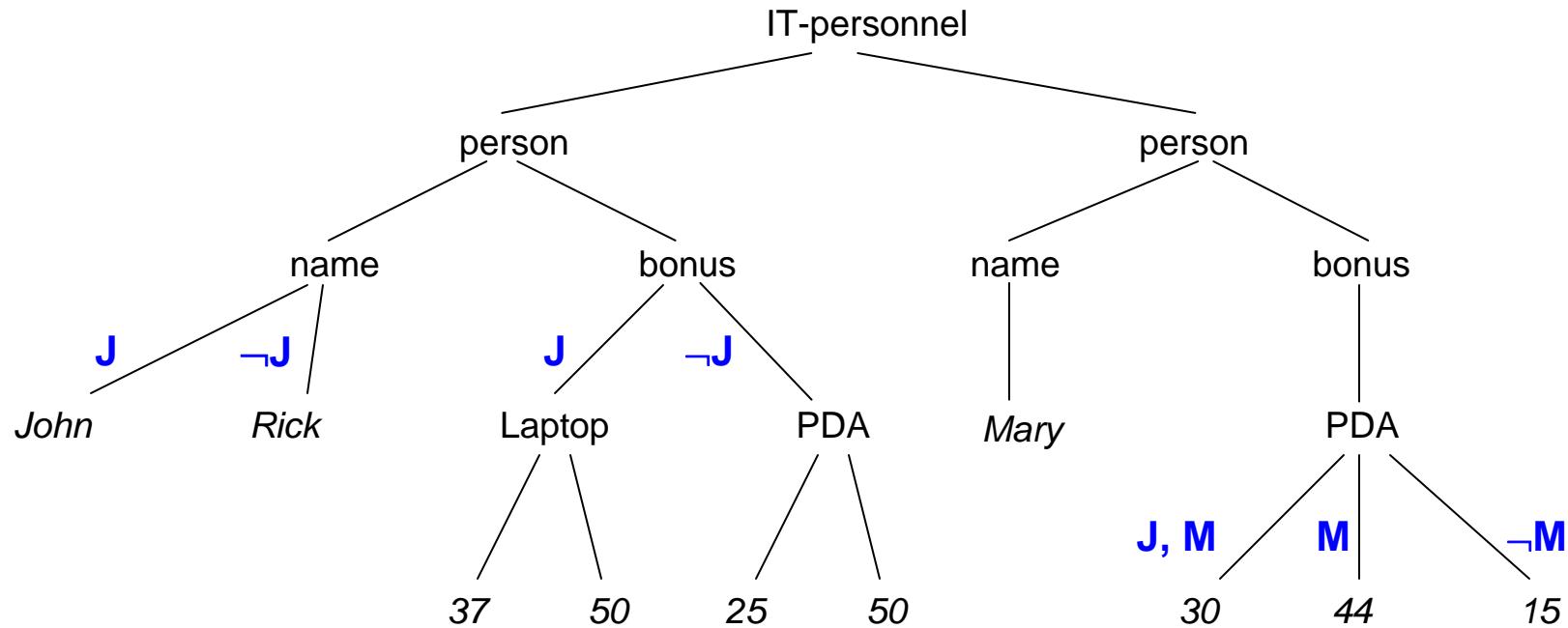
M: Mary worked overtime

Pr(M) = 0.6

"John was hired, Mary worked overtime"

J: John hired for Laptop project   Pr(J) = 0.3

M: Mary worked overtime   Pr(M) = 0.6

"John was hired, Mary worked overtime"
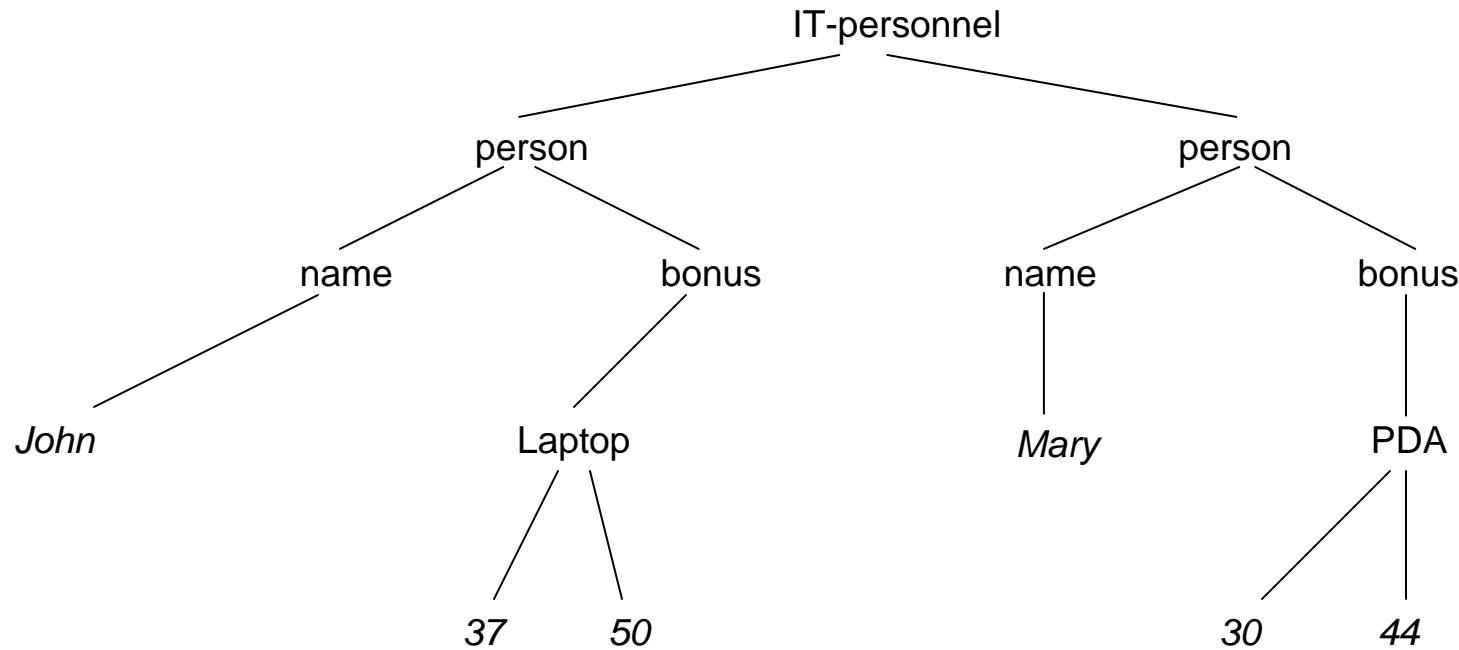
J: John hired for
  Laptop project $Pr(J) = 0.3$

M: Mary worked
  overtime $Pr(M) = 0.6$

$$Pr(d_1) = 0.3 \times 0.6$$

"John wasn't hired, Mary worked overtime"
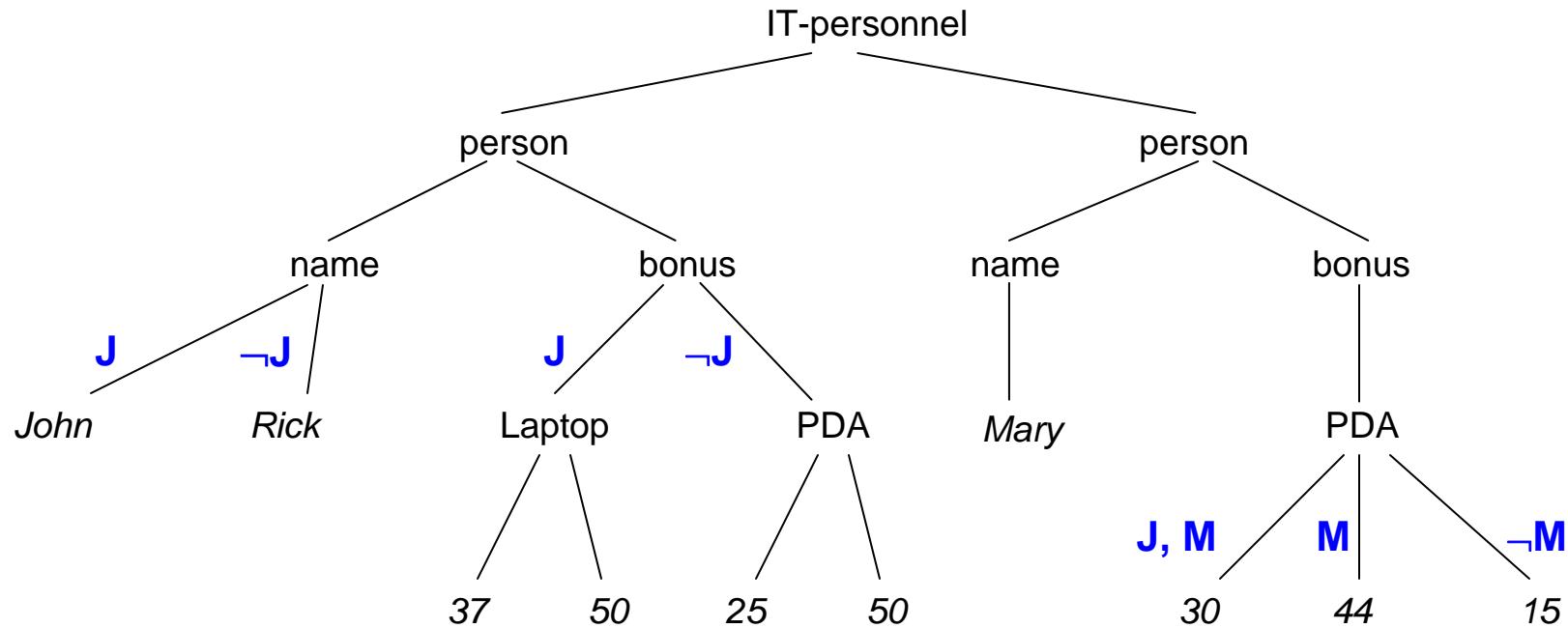
IT-personnel

person   person

name   bonus   name   bonus

J   **¬J**   **J**   **¬J**

John   *Rick*   Laptop   PDA   *Mary*   PDA

J, M   **M**   **¬M**

37   *50*   *25*   *50*   *30*   *44*   *15*

J: John hired for $Pr(J) = 0.3$
    Laptop project

$$Pr(d_2) = 0.7 \times 0.6$$

M: Mary worked $Pr(M) = 0.6$
    overtime

"John wasn't hired, Mary worked overtime"

```
                        IT-personnel
              /                            \
          person                        person
         /      \                      /       \
      name      bonus              name       bonus
       |          \                 |           |
      Rick        PDA              Mary        PDA
                  /  \                          |
                25    50                        44
```

J: John hired for   Pr(J) = 0.3
   Laptop project

 $Pr(d_2) = 0.7 \times 0.6$

M: Mary worked   Pr(M) = 0.6
   overtime

Probabilistic XML: MUX and DET Nodes

[Nierman/ Jagadish, Kimelfeld/ Sagiv]


```
                           IT-personnel
                 /                        \
             person                      person
            /      \                    /      \
        name       bonus            name       bonus
          |          |               |           |
        [MUX]      [MUX]           Mary         PDA
       0.3/ \0.7  0.3/ \0.7                      |
       /     \    /     \                      [MUX]
     John   Rick Laptop  PDA                 0.6/  \0.4
                  / \    / \               [DET]    \
                37  50  25  50             /    \    15
                                          30    44
```

| MUX | Children represent mutually exclusive choices, choices for different mux-nodes are independent |

| DET | Deterministic nodes, children are combined |
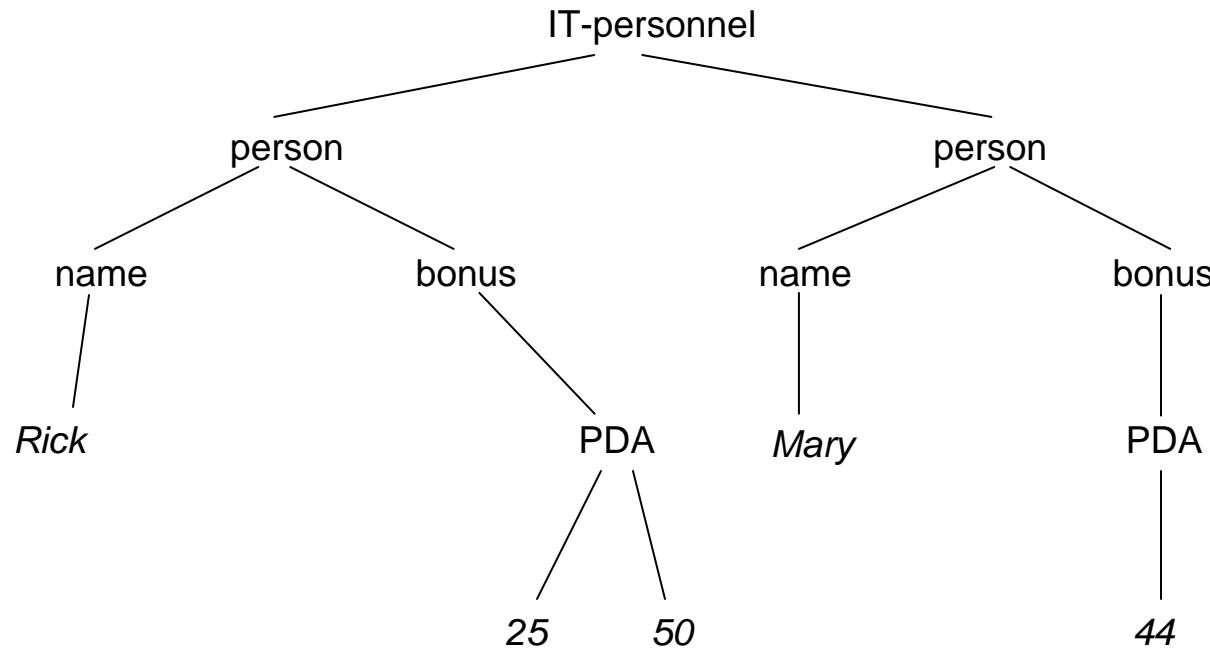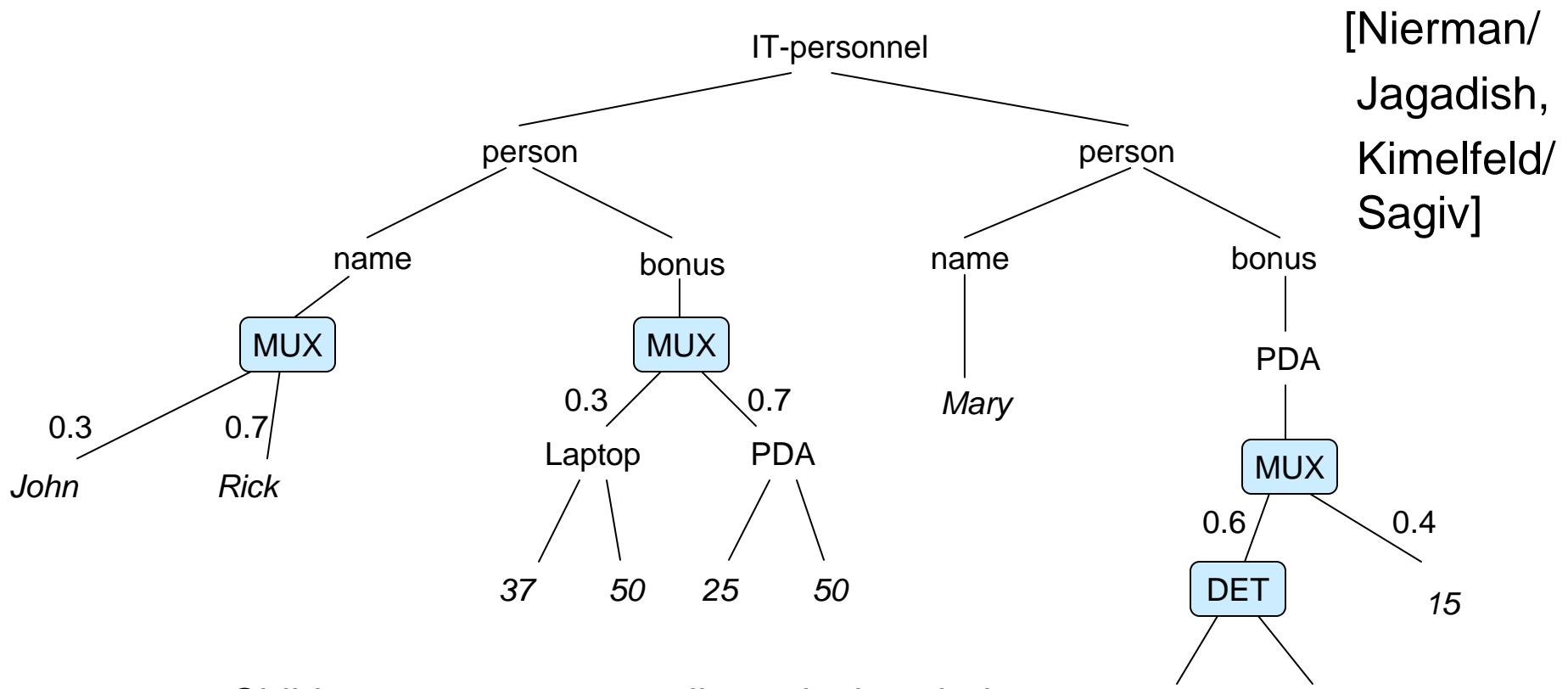
Probabilistic XML: MUX and DET Nodes

$$Pr = 0.3 \times 0.7 \times 0.6$$
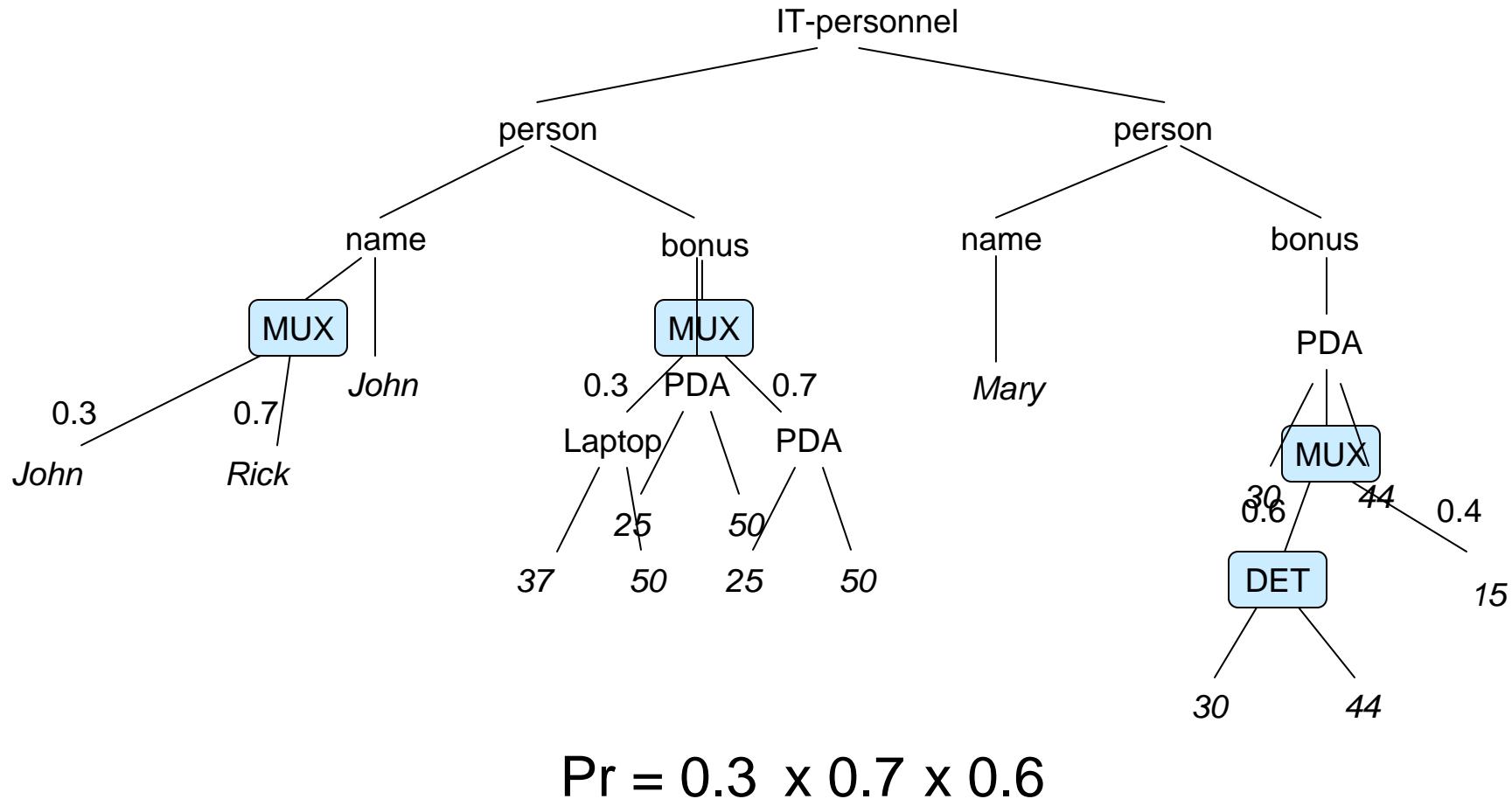
Probabilistic XML (PXML)

- A PXML document D
  - represents (exponentially) many document instances d
  - each with a probability Pr(d)

- PXML document models
  - CIE: long-distance dependencies
  - MUX-DET: only hierarchical dependencies
  - MUX-DET can be expressed by CIE,
    but not (concisely) the other way round

Other models can be reduced to the ones above,
or behave similarly

Aggregate Functions

α: finite bags of values $\rightarrow$ domain

Examples:

- count, countd: finite bags of anything$\rightarrow$ **N**
- sum, avg: finite bags of rational numbers $\rightarrow$ **Q**

Similarly: min, max, parity, top K, ...

Aggregate Queries

$$Q = \alpha(q)$$

Two Layers
- nonaggregate query q(x)
  - returns set of nodes q(d) over instance d

- aggregate function α
  - applied to the labels of nodes in q(d)
  - returns single value

$$\alpha(q(d))$$

Single Path Queries

Simple form of tree pattern queries

Paths of node labels or *,
  connected by "child" and
  "descendant" edges

Return the set of leaf nodes
  reachable from the root
  along such a path

Which bonuses
have been paid?

q_{bonus}

IT-personnel

bonus

*
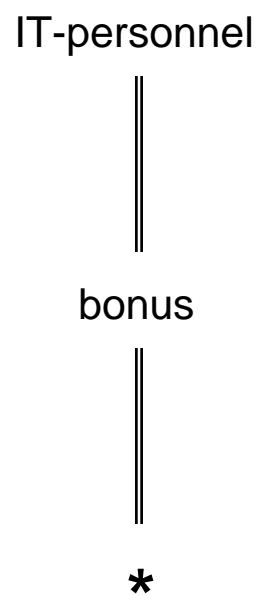
Single Path Aggregate Queries: Examples

- SUM(q_{bonus})

    "What is the sum of all bonuses?"

- MAX(q_{bonus})

    "What is maximal bonus that was paid?"

Answer Distributions

PXML document D, instances $\{d_1, \ldots, d_n\}$

SUM(q_{bonus}) returns exactly one number for every d_i

$\Rightarrow$ SUM(q_{bonus}) is a random variable

$\Rightarrow$ SUM(q_{bonus}) induces a probability distribution over D

$$f(s) = \sum_{\text{SUM}(q_{bonus})(d_i) = s} Pr(d_i),$$

the answer distribution

Notation: SUM(q_{bonus})(D)   or $\alpha(q)(D)$ abstractly

Special Case: Document Aggregation

D with instances $\{d_1, \ldots, dn\}$,

- Applying α to a regular document d_i :

$$\alpha(d_i) := \alpha(\{| \ c \ | \ c \text{ is a value on a leaf of } d_i \ |\})$$

- Applying α to the probabilistic document D:

$$\alpha(D)(c) = \sum_{\alpha(d_i) = c} Pr(d_i)$$

yields again a distribution $\alpha(D)$

Reduction to Document Aggregation

$$\alpha(q)(D) = ?$$

Step 1:   Compute a smaller PXML document

$$D' = q(D)$$

containing only matching paths

Step 2:   Apply α to D'

Depends on document models and simple path queries

Theorem:

$$\alpha(q)(D) = \alpha(D')$$

Applying q_{bonus}

$q_{bonus}($

IT-personnel
- person
  - name
    - J — *John*
    - ¬J — *Rick*
  - bonus
    - J — Laptop
      - *37*
      - *50*
    - ¬J — PDA
      - *25*
      - *50*
- person
  - name
    - *Mary*
  - bonus
    - PDA
      - J, M — *30*
      - M — *44*
      - ¬M — *15*

$)$
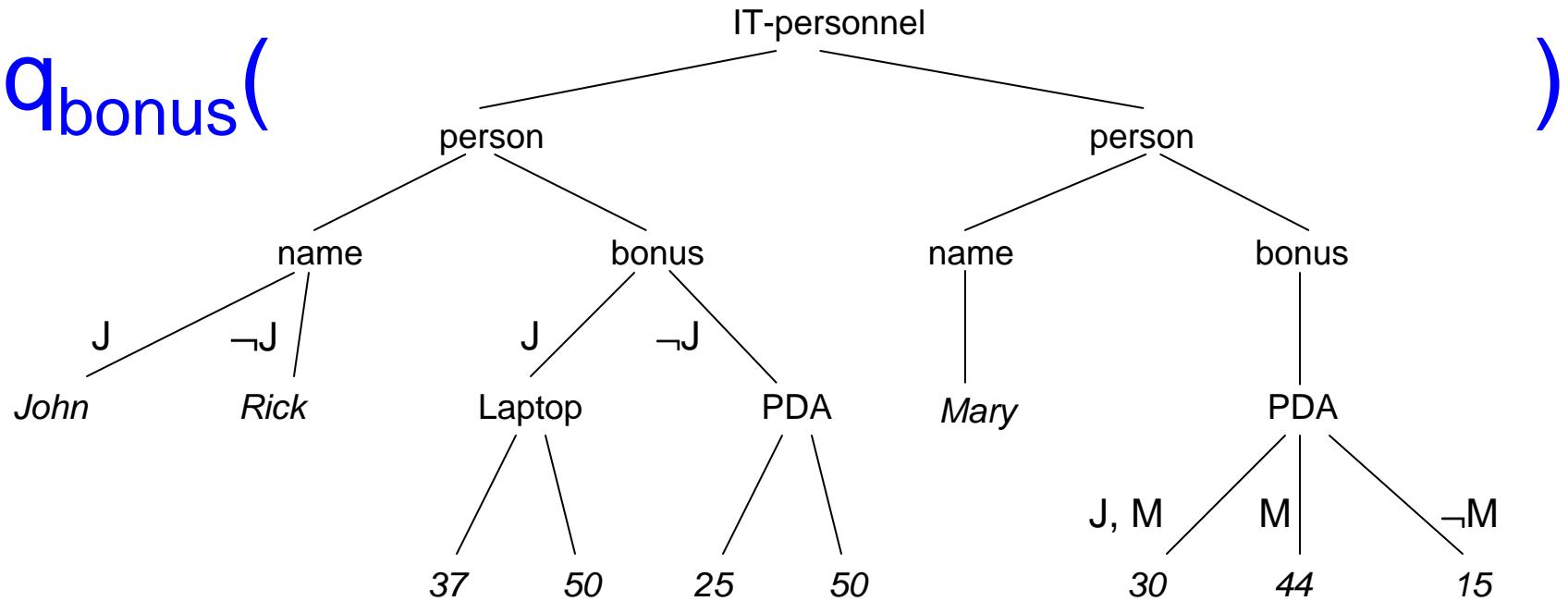
= keep only the paths that match

… analogous for MUX-DET

Evaluating Single Path Queries/2

$q_{bonus}($

$)$

IT-personnel

person

name

MUX

0.3 — John
0.7 — Rick

bonus

MUX

0.3 — Laptop
0.7 — PDA

Laptop: 37, 50
PDA: 25, 50

person

name

Mary

bonus

PDA

MUX

0.6 — DET
0.4 — 15

DET: 30, 44
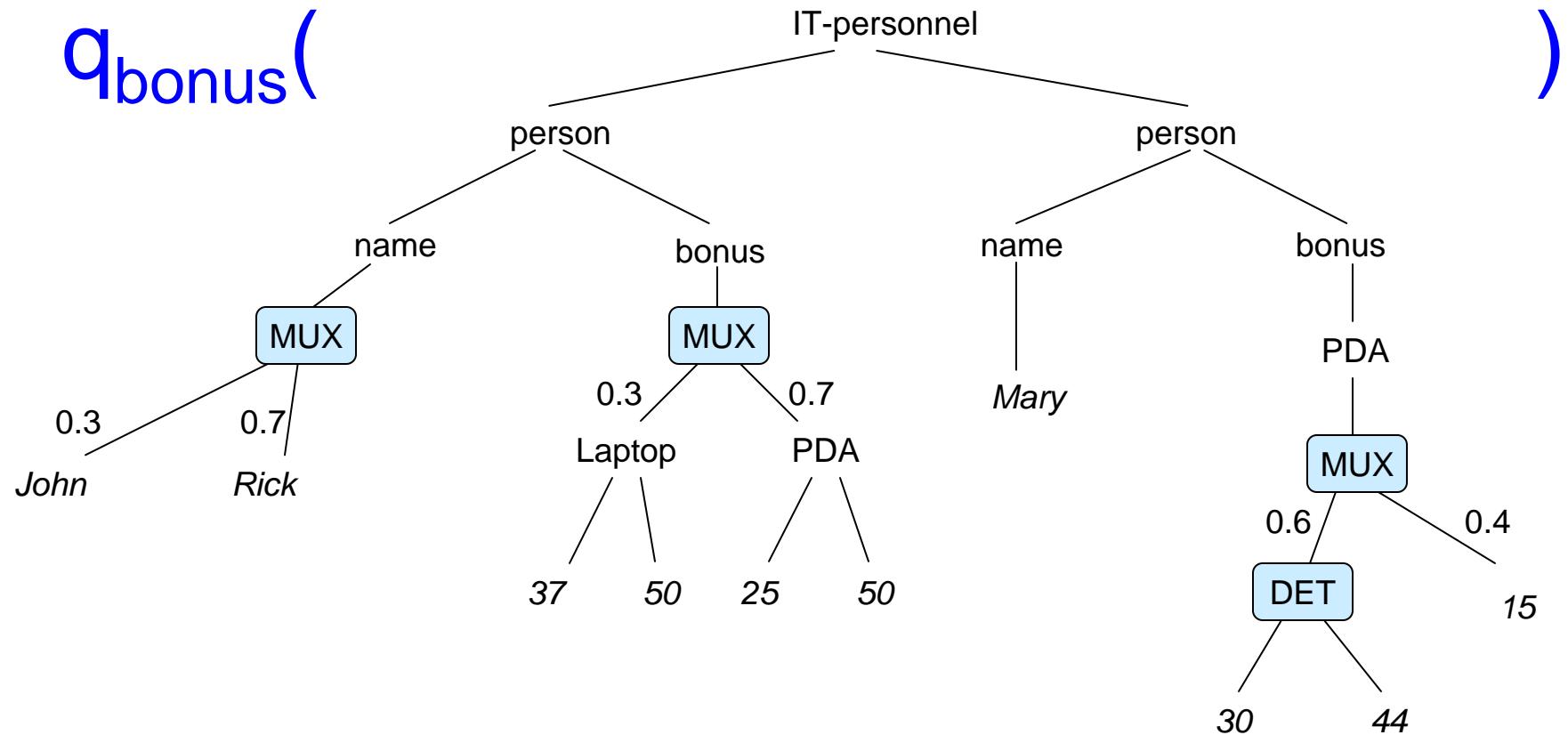
Problems Investigated

PXML document D, constant c

- Possible Value: Decide $\Pr(\alpha(D) = c) > 0$

- Probability Computation: Compute $\Pr(\alpha(D) = c)$

- Moment Computation: Compute $E(\alpha(D)^k)$

E *is "expected value"*

Aggregation over CIE

	COUNT	SUM	MIN	COUNTD	AVG
Possible Value	NP-c	NP-c	NP-c	NP-c	NP-c
Probability Computation	*in $FP^{\#P}$*	*in $FP^{\#P}$*	$FP^{\#P}$-c	$FP^{\#P}$-c	$FP^{\#P}$-c
Moment Computation	P	P	$FP^{\#P}$-c	$FP^{\#P}$-c	$FP^{\#P}$-c

Aggregation over CIE/2

- **Possible Value**: *"Too much propositional logic present"*

- **Probability Computation**: cannot be easier …

- **Moment Computation**:
  - Difficult for MIN, COUNTD, AVG
  - Easy for COUNT and SUM:

  "Moments are sums,
  moments of COUNT and SUM are sums of sums,
  which can be rearranged …"

Aggregation over MUX-DET

	COUNT	SUM	MIN	COUNTD	AVG
Possible Value	P	NP-c	P	NP-c	*In NP*
Probability Computation	P	P in |input| + |distribution|	P	FP$^{\#P}$-c	FP$^{\#P}$-c
Moment Computation	P	P	P	P	P

COUNT, SUM, MIN are Easy ...

… because they allow for divide and conquer evaluation:

$$\text{SUM} \{| a,b,c,d |\} = \text{SUM} \{| a,b |\} + \text{SUM} \{| c,d |\}$$

α is a monoid aggregate function if

$$\alpha(\{| a_1,..., a_n |\}) = \alpha(\{| a_1 |\}) \oplus ... \oplus \alpha(\{| a_n |\})$$

for some commutative monoid $(M, \oplus)$ and all $a_1,..., a_n$ in M

Examples:
- count, sum, min, parity, top K
- countd, avg

Convex Sums and Convolutions

If α is a monoid function, answer distributions
can be computed bottom up, using two operations:

Convex Sum: $\alpha(\quad) = p\ \alpha(\quad) + q\ \alpha(\quad)$

Convolution: $\alpha(\quad) = \alpha(\quad)\ *\ \alpha(\quad)$

depends on the monoid

Convolution of Distributions

$(M, \oplus)$ monoid

$\alpha(D_1)$, $\alpha(D_2)$ distributions of subdocuments

$$((\alpha(D_1) * \alpha(D_2))\,(c) = \sum_{c_1 \oplus c_2 = c} \alpha(D_1)(c_1)\ \alpha(D_2)(c_2)$$

Approximating Query Answers

Over CIE, probability and moment computation can be hard

How good are Monte-Carlo methods?

Classical results (Hoeffding) imply: To achieve

$$| E(\alpha(D)^k) - \text{Estimate} | < \varepsilon \text{ with probability } 1 - \delta$$

at most $O(R^{2k} \varepsilon^{-2} \log 1/\delta)$ samples are needed,

$$\text{where } R = \max |\alpha(d)|.$$

Consequence: Given ε and δ, at most quadratically many samples are needed for $E(\text{COUNTD}(D))$.

Probabilistic Aggregation: Related Work

- Tree pattern queries over MUX-DET
  with HAVING constraints [Cohen/Kimelfeld/Sagiv]

- Conjunctive queries with HAVING constraints over
  relational probabilistic databases [Re/Suciu]

- Work on various special topics in the relational setting
  - probabilistic data streams
  - uncertain schema mappings

Aggregates over PXML: Conclusion

First results of an ongoing project

- Map of the problem space

- Largely complete investigation for single path queries:
  - Intractability for CIE
  - Hierarchical dependencies in MUX-DET can be exploited for monoid aggregation functions

Some results carry over to other models, e.g.,

- Uncertain schema mappings (Dong/Halevy/Yu)

Current work:

- richer query languages, continuous distributions on leaves