

# Modeling and Aggregating Probabilistic XML

S. Abiteboul,<sup>1</sup> T-H. H. Chan,<sup>2</sup> E. Kharlamov,<sup>3</sup> W. Nutt,<sup>3</sup> P. Senellart<sup>4</sup>

<sup>1</sup>INRIA Saclay – Île-de-France    <sup>3</sup>Free University of Bozen-Bolzano  
<sup>2</sup>The University of Hong Kong    <sup>4</sup>Télécom ParisTech

KRDB Lunch Seminar, November 2009

# Outline

1. Probabilistic data management
2. Problem definition
3. Aggregating discrete Probabilistic XML
4. Aggregating continuous Probabilistic XML
5. Further challenges

# I. Probabilistic data management

- Applications
- Definition
- Some models

# Uncertainty in DBs

[Widom:2007]

- Not a new area - comes from 1970s
- Most initial work was theoretical
- Systems came up recently
- But applications were not ready anyway

# Uncertainty in DBs

[Widom:2007]

- Not a new area - comes from 1970s
- Most initial work was theoretical
- Systems came up recently
- But applications were not ready anyway

Are they now?

# Applications of Probabilistic Data

[Kossmann&Dittrich:2007]

- Approximate query processing
  - Ranking
  - Record linkage

Here: data is deterministic,  
but query answers are probabilistic

# Applications of Probabilistic Data

[Widom:2007]

- **Information extraction**
  - Find & label entities in unstructured text
  - Often probabilistic
- **Information integration**
  - Combine data from multiple sources
  - Inconsistencies

# Applications of Probabilistic Data

[Widom:2007]

- **Scientific experiments**
  - Inexact/incomplete data
  - Many levels of “derived data products”
- **Sensor data management**
  - Approximate readings
  - Missing readings

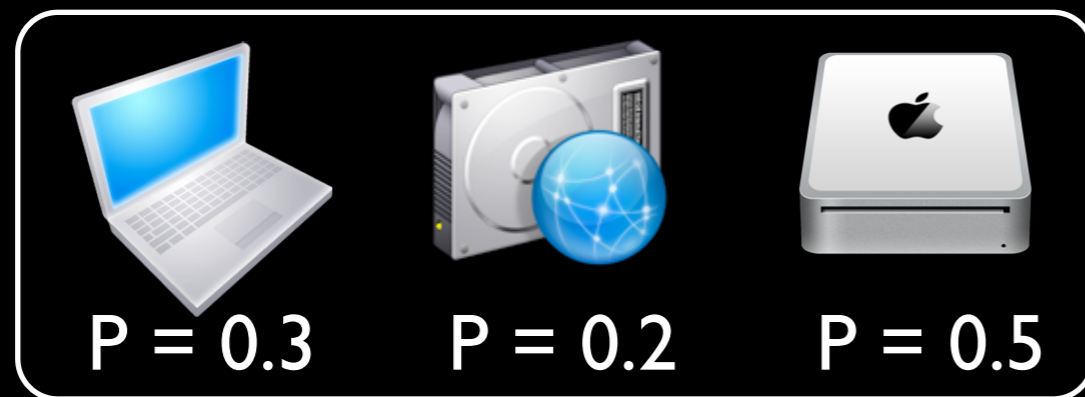
# Applications of Probabilistic Data

[Widom:2007]

- Deduplication (“data cleaning”)
  - Object linkage, entity resolution
  - Often heuristic/probabilistic

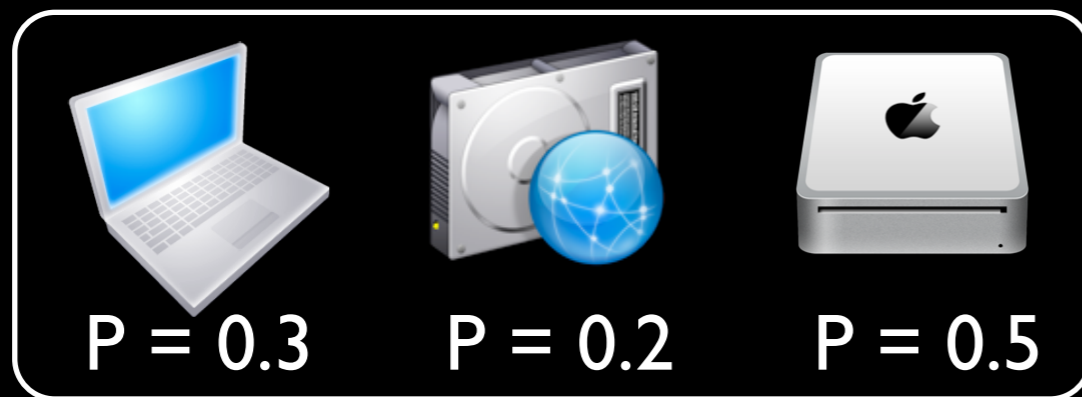
# Probabilistic Database

Probabilistic DB:



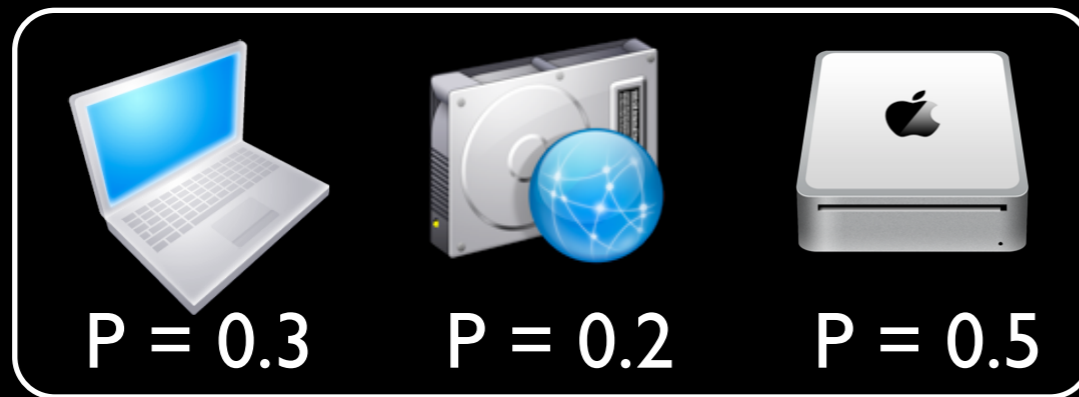
# Probabilistic Database

Probabilistic DB:



# Probabilistic Database

Probabilistic DB:

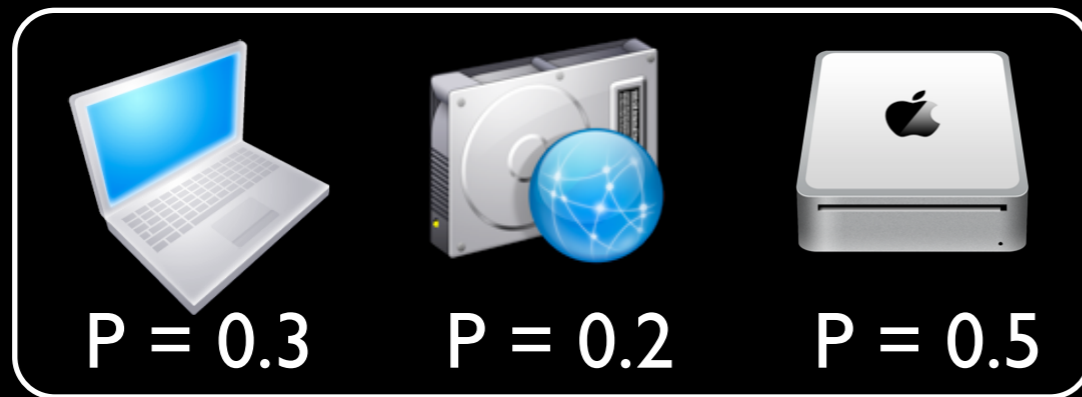


---

Answer: (a, 0.8)

# Probabilistic Database

Probabilistic DB:



Representation  
of Prob DB:



---

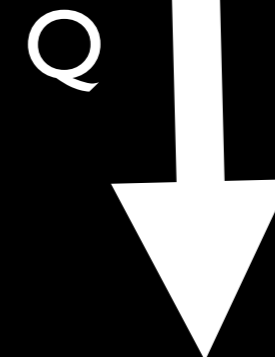
Answer: (a, 0.8)

# Probabilistic Database

Probabilistic DB:



Representation  
of Prob DB:



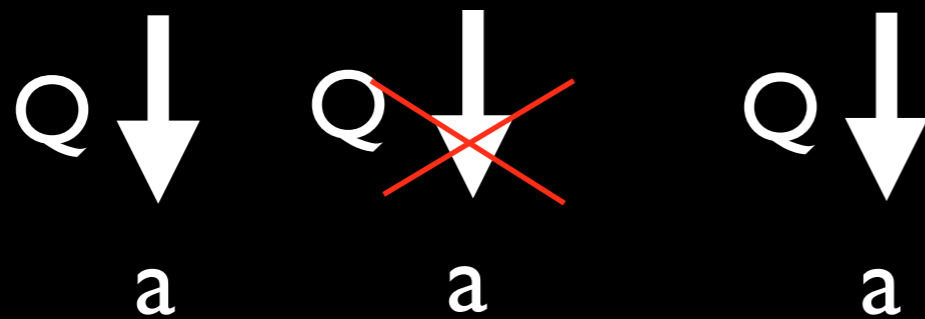
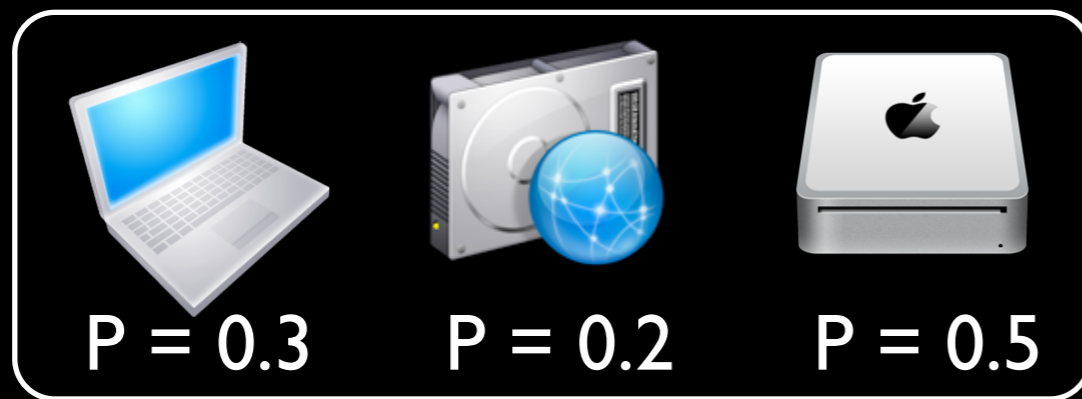
---

Answer: (a, 0.8)

(a, 0.8)

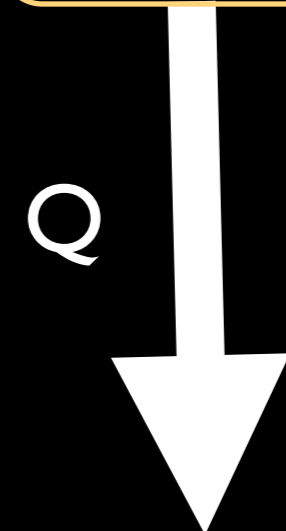
# Probabilistic Database

Probabilistic DB:



Answer: (a, 0.8)

Representation  
of Prob DB:



(a, 0.8)

- Relational Probabilistic DBs

# Uncertainty and Lineage: Trio

[Widom:2007]

First-class interrelated concepts:

- **Data**
- **Uncertainty:**
  - Alternatives
  - maybe
  - confidence
- **Lineage:** Boolean formulas attached to query answers to trace origin of answers

Name	Tel
Hubert 0.2    Bob 0.5	212    433
Pierre	902
Evgeny    Alan	2345

?

# Uncertainty and Lineage: Trio

[Widom:2007]

First-class interrelated concepts:

- **Data**
- **Uncertainty:**
  - Alternatives
  - maybe
  - confidence
- **Lineage:** Boolean formulas attached to query answers to trace origin of answers

Name	Tel
Hubert 0.2    Bob 0.5	212    433
Pierre	902
Evgeny    Alan	2345

?

# Uncertainty and Lineage: Trio

[Widom:2007]

First-class interrelated concepts:

- **Data**
- **Uncertainty:**
  - Alternatives
  - maybe
  - confidence
- **Lineage:** Boolean formulas attached to query answers to trace origin of answers

Name	Tel
Hubert 0.2    Bob 0.5	212    433
Pierre	902
Evgeny    Alan	2345



# Uncertainty and Lineage: Trio

[Widom:2007]

First-class interrelated concepts:

- **Data**
- **Uncertainty:**
  - Alternatives
  - maybe
  - confidence
- **Lineage:** Boolean formulas attached to query answers to trace origin of answers

Name	Tel
Hubert <sup>0.2</sup>    Bob <sup>0.5</sup>	212    433
Pierre	902
Evgeny    Alan	2345

?

# Uncertainty and Lineage: Trio

[Widom:2007]

First-class interrelated concepts:

- **Data**
- **Uncertainty:**
  - Alternatives
  - maybe
  - confidence
- **Lineage:** Boolean formulas attached to query answers to trace origin of answers

Name	Tel
Hubert 0.2    Bob 0.5	212    433
Pierre	902
Evgeny    Alan	2345

?

# Block Independent ProbDBs: MystiQ

[Re&Suciu:2006]

- **Block** = value of the key attribute
- **Independent** tuples across blocs
- **Mutually exclusive** tuples within blocs  
Within a block probabilities sum up to 1

<u>Name</u>	Tel	Age	Prob
Hubert	111	20	0.4
	222	20	0.6
Pierre	111	6	0.1
	333	7	0.3
	444	7	0.6
Evgeny	222	10	0.5
	555	15	0.5

# Block Independent ProbDBs: MystiQ [Re&Suciu:2006]

- **Incomplete** representation system
- **Hierarchical** conjunctive queries and some queries with HAVING are **tractable**.  
Others (#P) hard
- Monte-Carlo **simulations** for intractable cases and TopK

# MayBMS

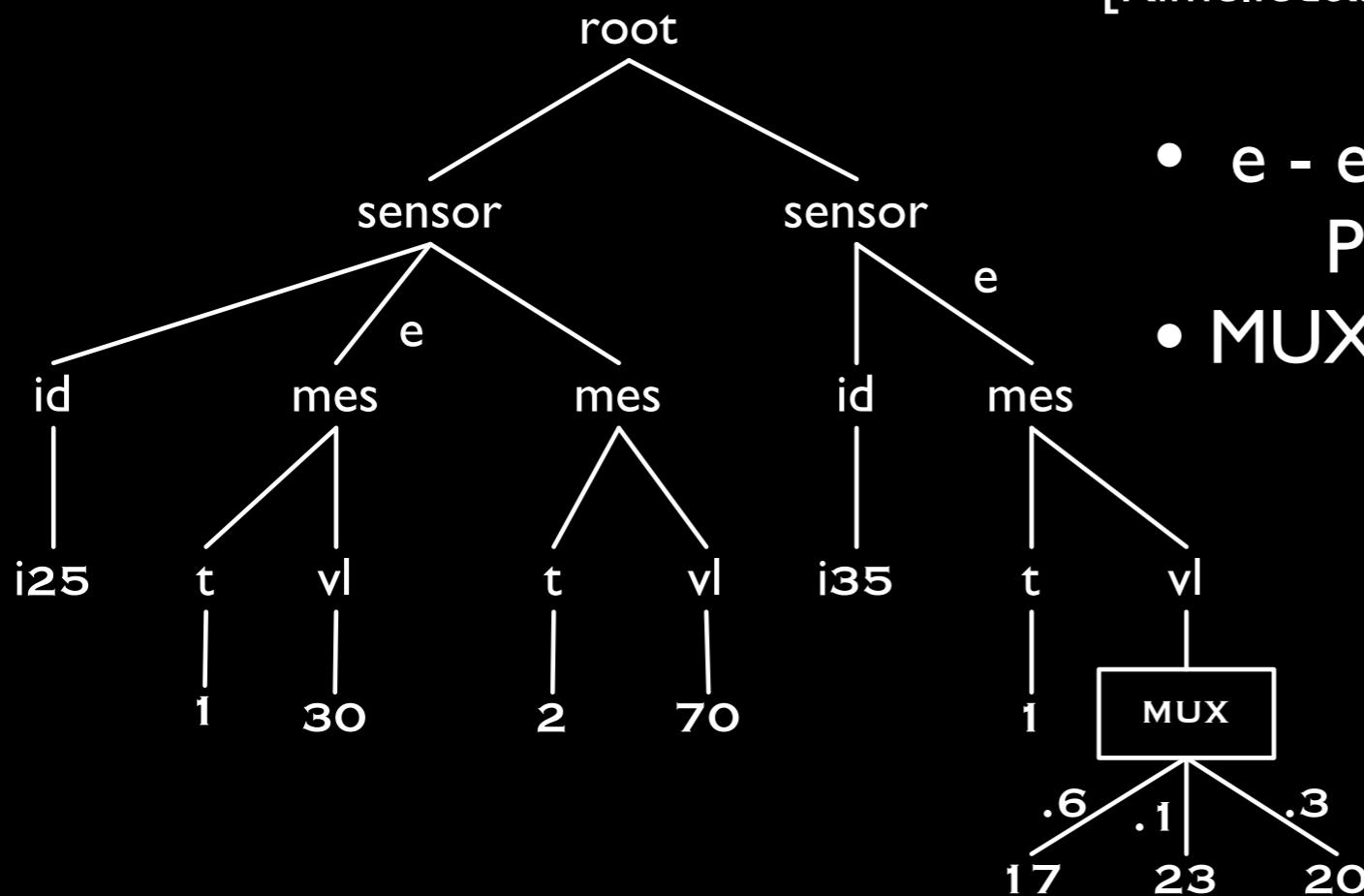
[Antova,Jansen,Koch,Olteanu:2007]

- In a nutshell: Probabilistic **C-Tables**,  
i.e., tables with (random) variables and  
Boolean conditions on the variables
- Realization:  
**U-relations** (vertical data decompositions)  
+ tables of **distributions**
- Complete and closed under **SPJRUD**
- Queries are (#P) **hard**  
unless hierarchical and tuples are independent

- Semi-structured Probabilistic DBs

# PXML with Events and Distributional Nodes

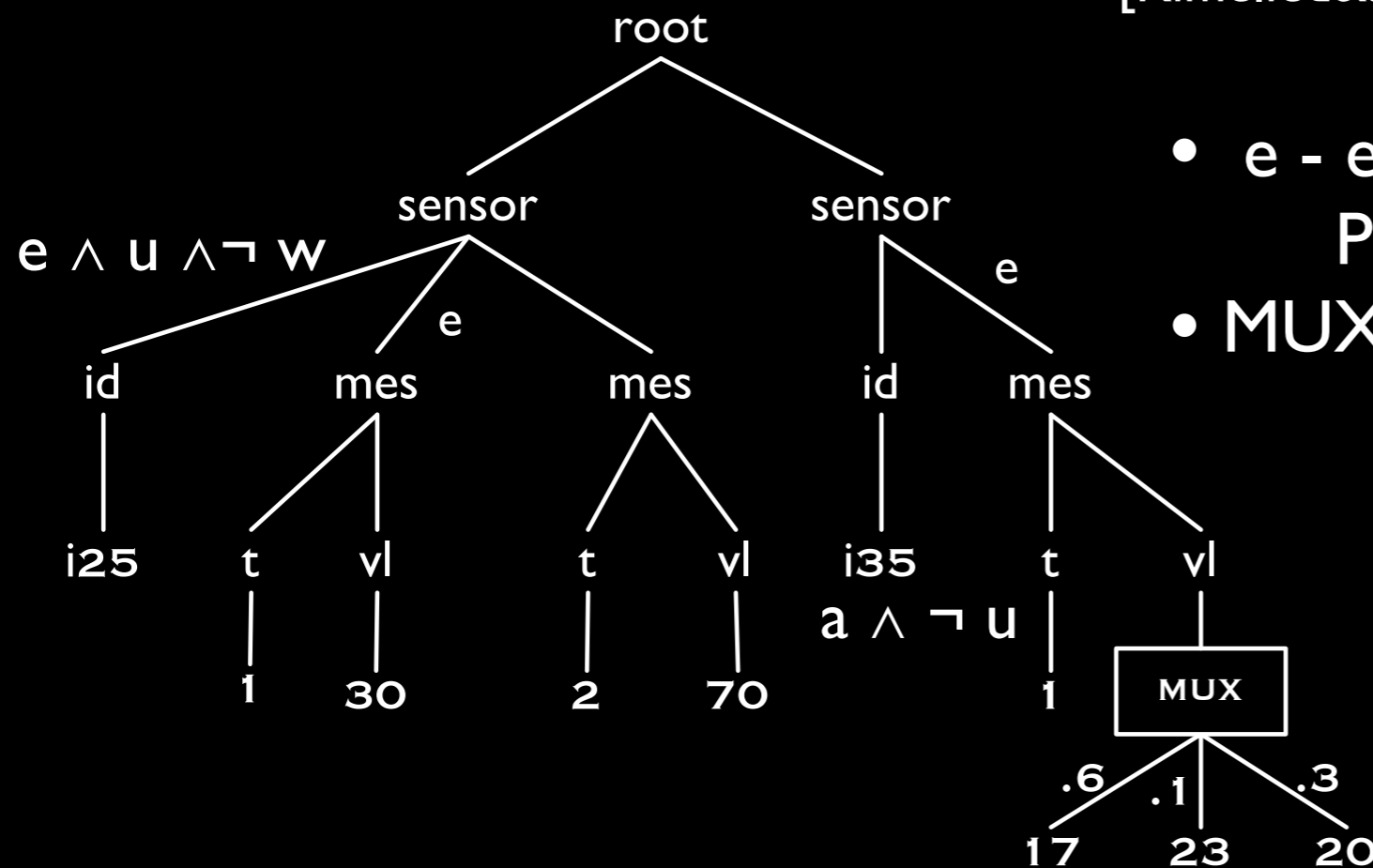
[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



- e - event “sensor is useful”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

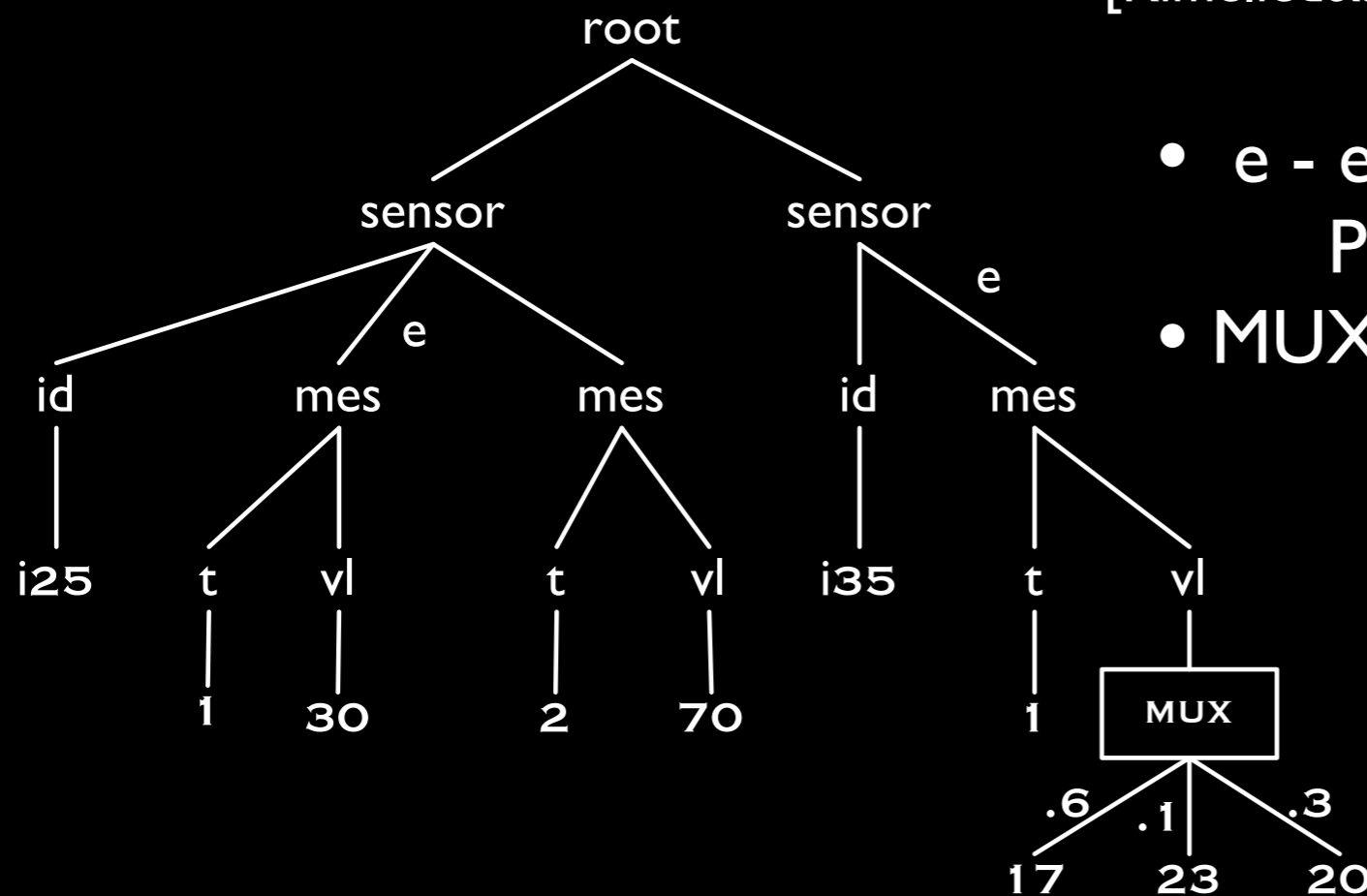
[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



- e - event “sensor is useful”  
Pr(e) = .4
- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

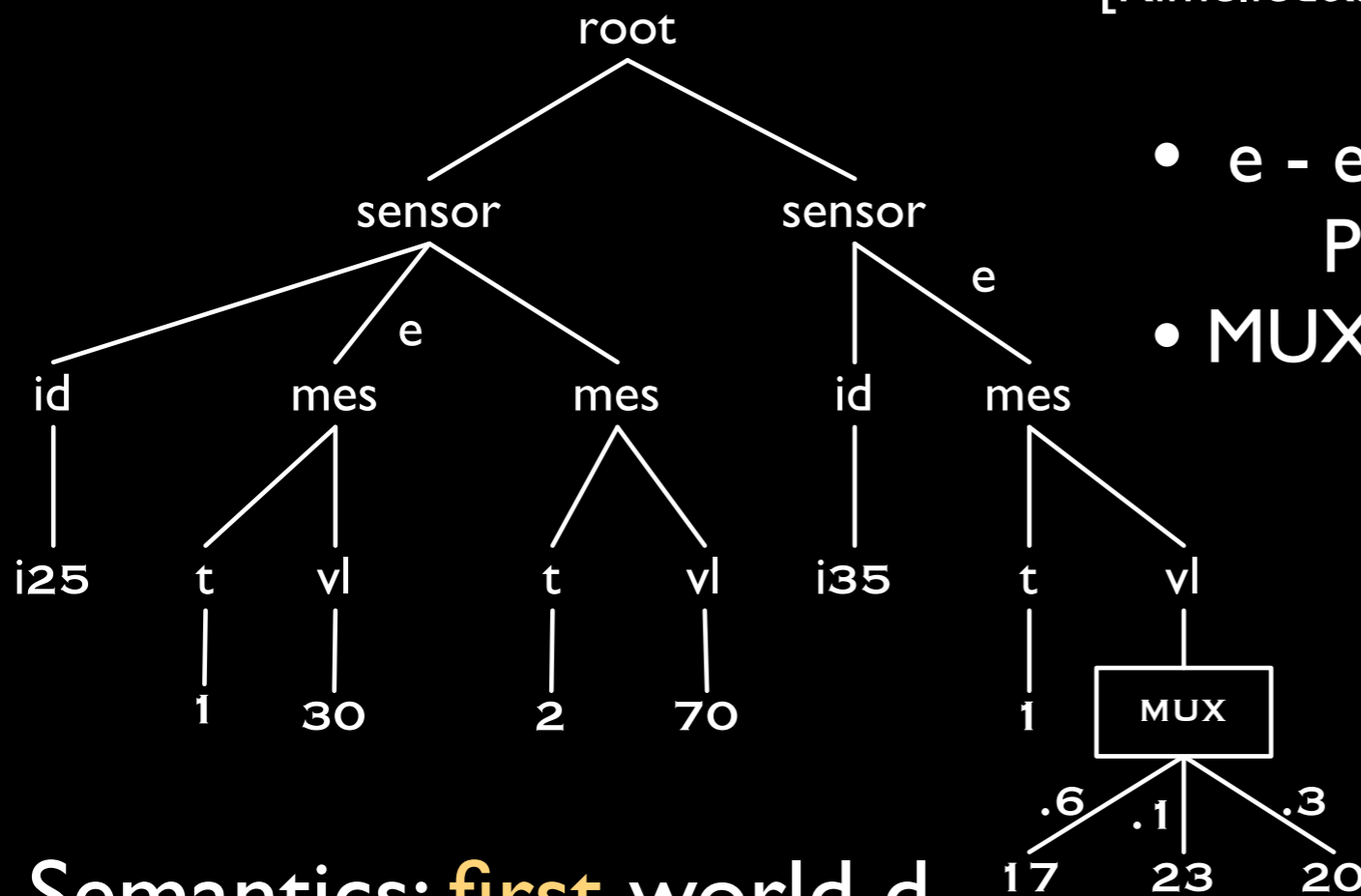
[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



- e - event “sensor is useful”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



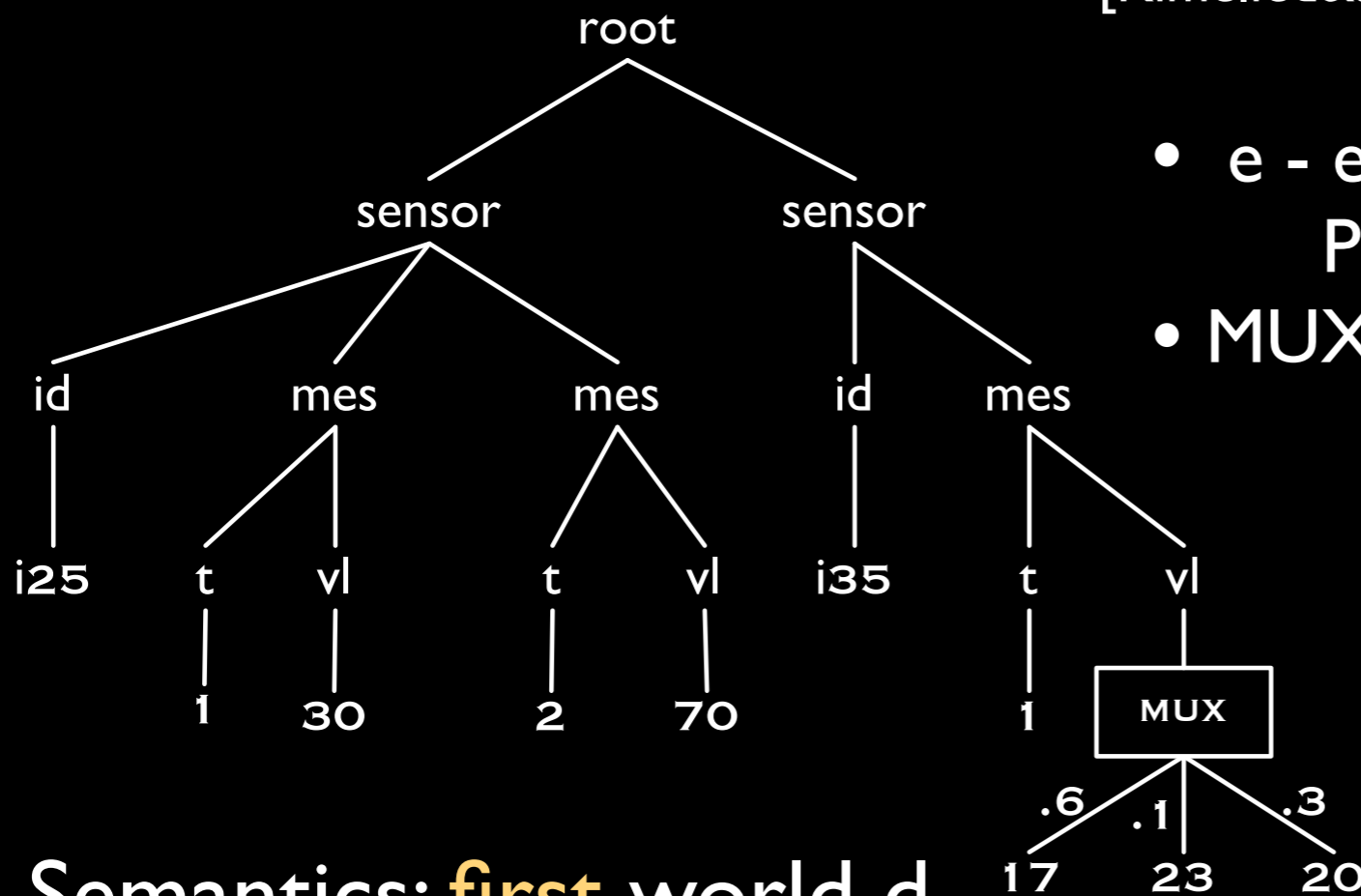
- e - event “sensor is useful”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

Semantics: **first** world d

- e = true (measurement at time one is useful)
- MUX: 23
- $\Pr(d) = 0.4 \times 0.1$

# PXML with Events and Distributional Nodes

[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



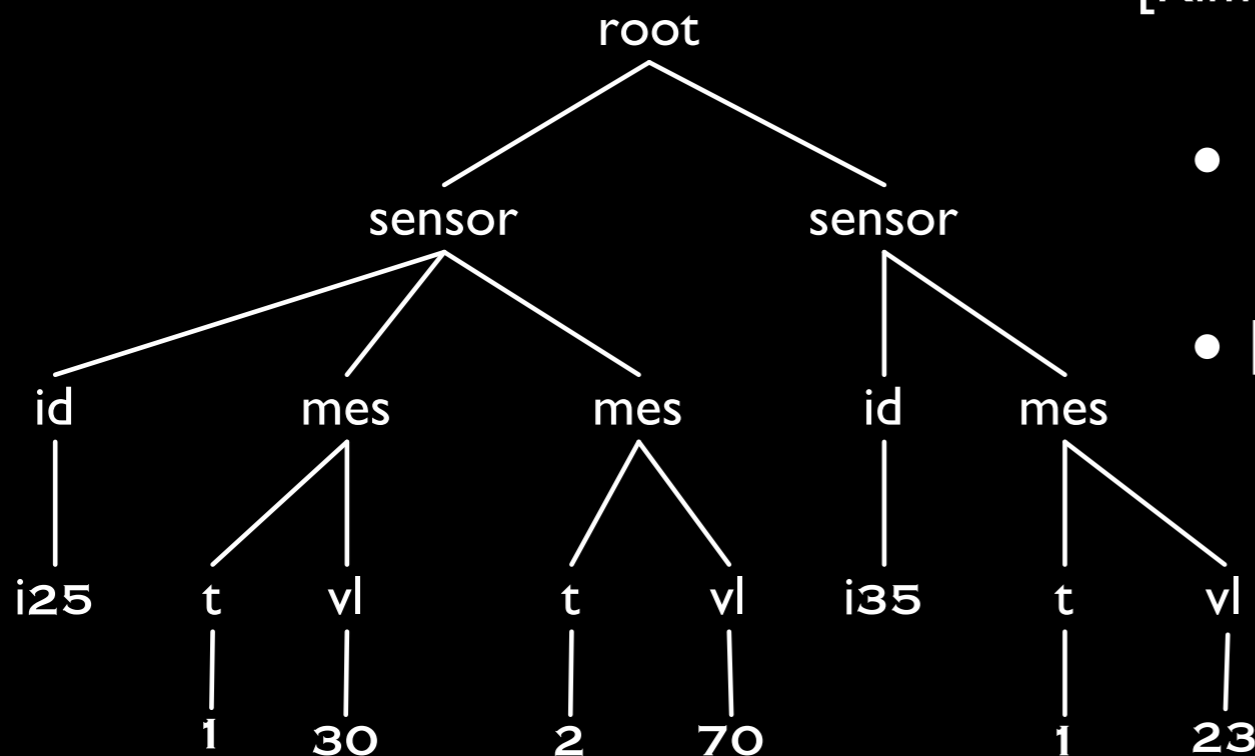
- e - event “sensor is useful”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

Semantics: **first** world d

- e = true (measurement at time one is useful)
- MUX: 23
- $\Pr(d) = 0.4 \times 0.1$

# PXML with Events and Distributional Nodes

[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



- e - event “sensor is useful”  
 $\Pr(e) = .4$
- MUX - mutually exclusive options

Semantics: **first** world d

- e = true (measurement at time one is useful)
- MUX: 23
- $\Pr(d) = 0.4 \times 0.1$

# Discrete Probabilistic XML Documents

- Probabilistic XML document **D**
  - represents (exponentially) many documents **d**
  - each with probability **Pr(d)**
- It is achieved by
  - **Distributional** nodes: Mux, Det, Ind, Exp, that capture local (hierarchical) dependancies
  - **Events** that label edges: Boolean random vars that capture long-distance dependancies

# What is Known? [Kimelfed&at.al.:2007]

[Senellart&at.al.:2007]

- Answering simple XPath queries
  - Distributional nodes: PTIME
  - Events:  $FP^{\#P}$ -complete
- Simple XPath over PXML with distributional nodes + aggregate constraints:
  - PTIME for COUNT and MIN
  - NP-hard for SUM and AVG
- Monte-Carlo simulations for intractable case

# What is Known?

[Kimelfed&at.al.:2007]

[Senellart&at.al.:2007]

- Answering simple XPath queries
  - Distributional nodes: PTIME
  - Events:  $FP^{\#P}$ -complete
- Simple XPath over PXML with distributional nodes + aggregates **constraints**:
  - PTIME for COUNT and MIN
  - NP-hard for SUM and AVG
- Monte-Carlo simulations for intractable case

NO events

# $FP^{\#P}$ - completeness

- $\#P$  - hard problems:  
Counting counterparts of NP-hard problems  
DNF-SAT: is  $\varphi$  satisfiable?  
 $\#DNF-SAT$ : how many sat assignments  $\varphi$  has?
- $FP^{\#P}$  functions:  
functions computable in PTIME  
that have access to  $\#P$ -complete oracle
- $FP^{\#P}$  - complete =  $\#P$  - hard and in  $FP^{\#P}$

## 2. Problems to Investigate

- Aggregate query answering
- Continuous PXML

# Aggregate Queries

- How many sensors were up at time  $t = 1$ ?
- Find the average temperature per sensor

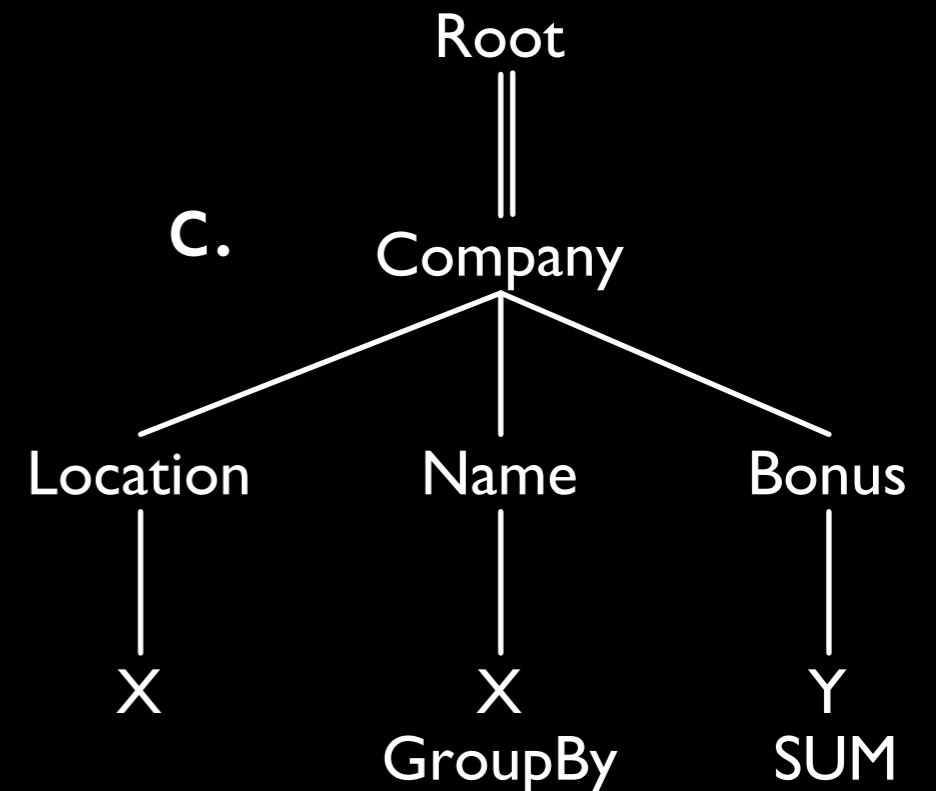
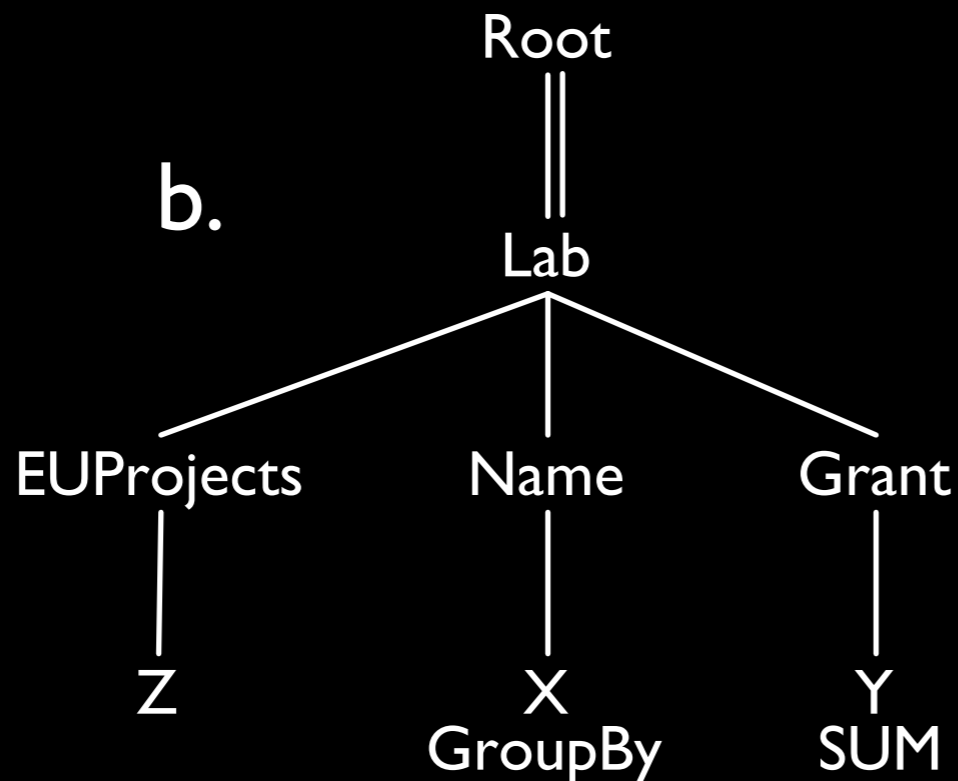
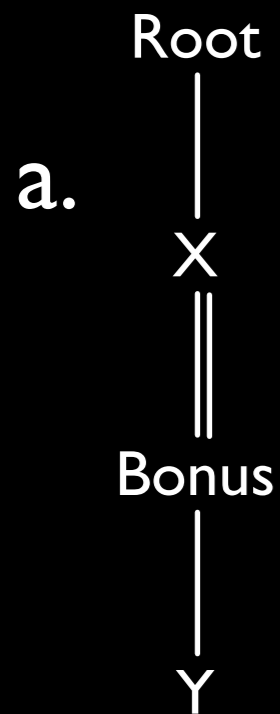
⇒ we want answers queries with **aggregate** functions:  
MIN/MAX, TopK, COUNT, SUM, COUNTD, AVG

# Query Models

a. Single-Path queries - **SP**

b. Tree-Pattern queries - **TP**

c. Tree-Pattern queries with Joins - **TPJ**

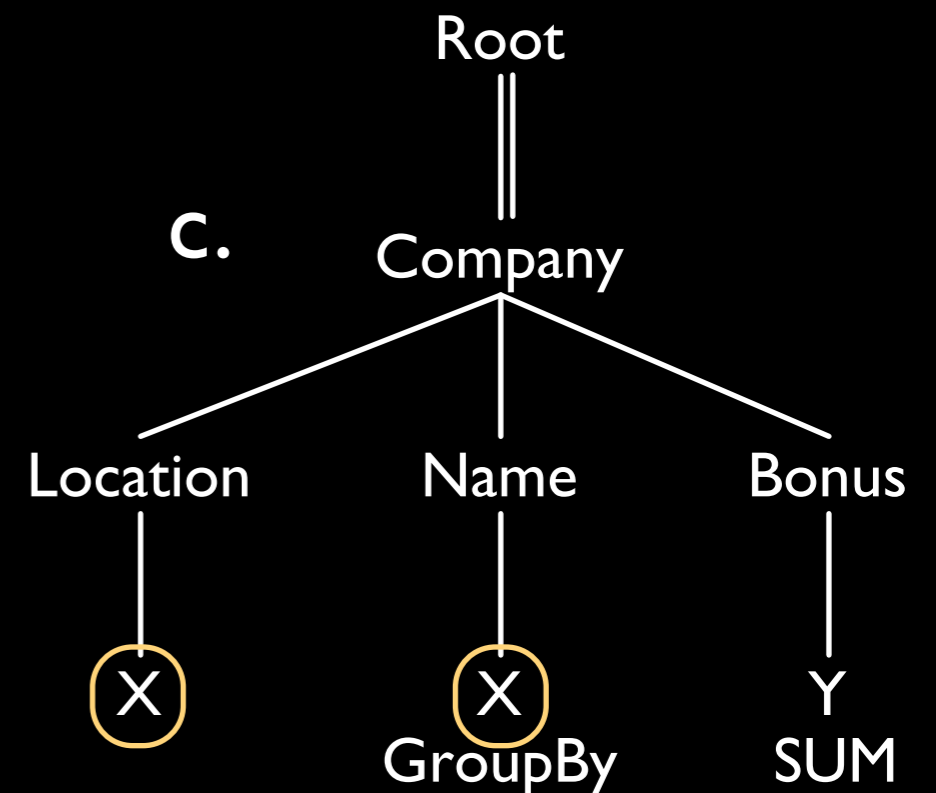
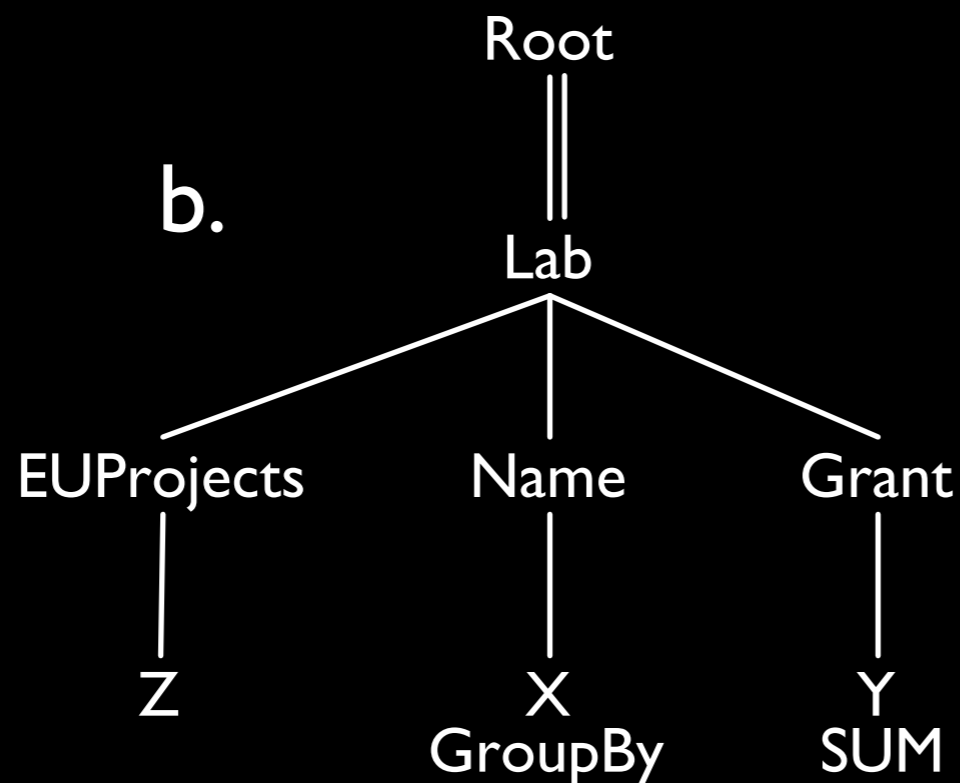
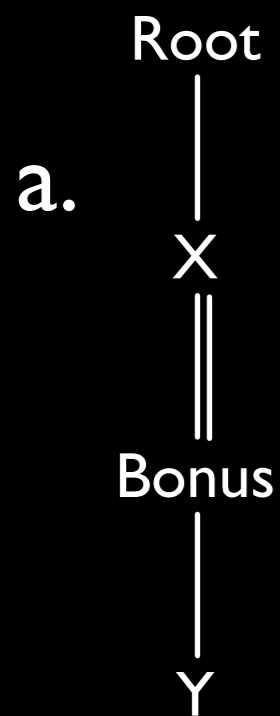


# Query Models

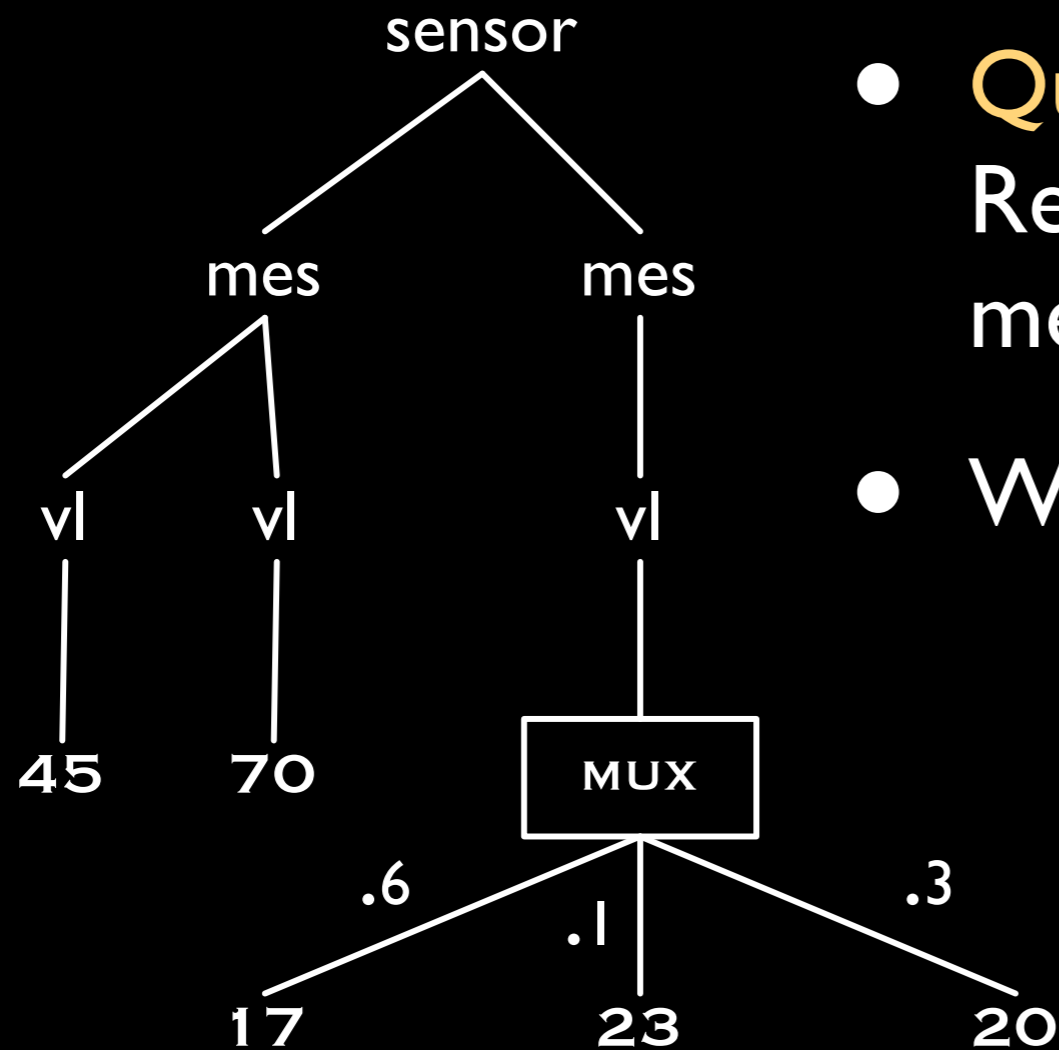
a. Single-Path queries - **SP**

b. Tree-Pattern queries - **TP**

c. Tree-Pattern queries with Joins - **TPJ**



# Semantics of AQs



- **Query:**  
Return the AVG of the measurements

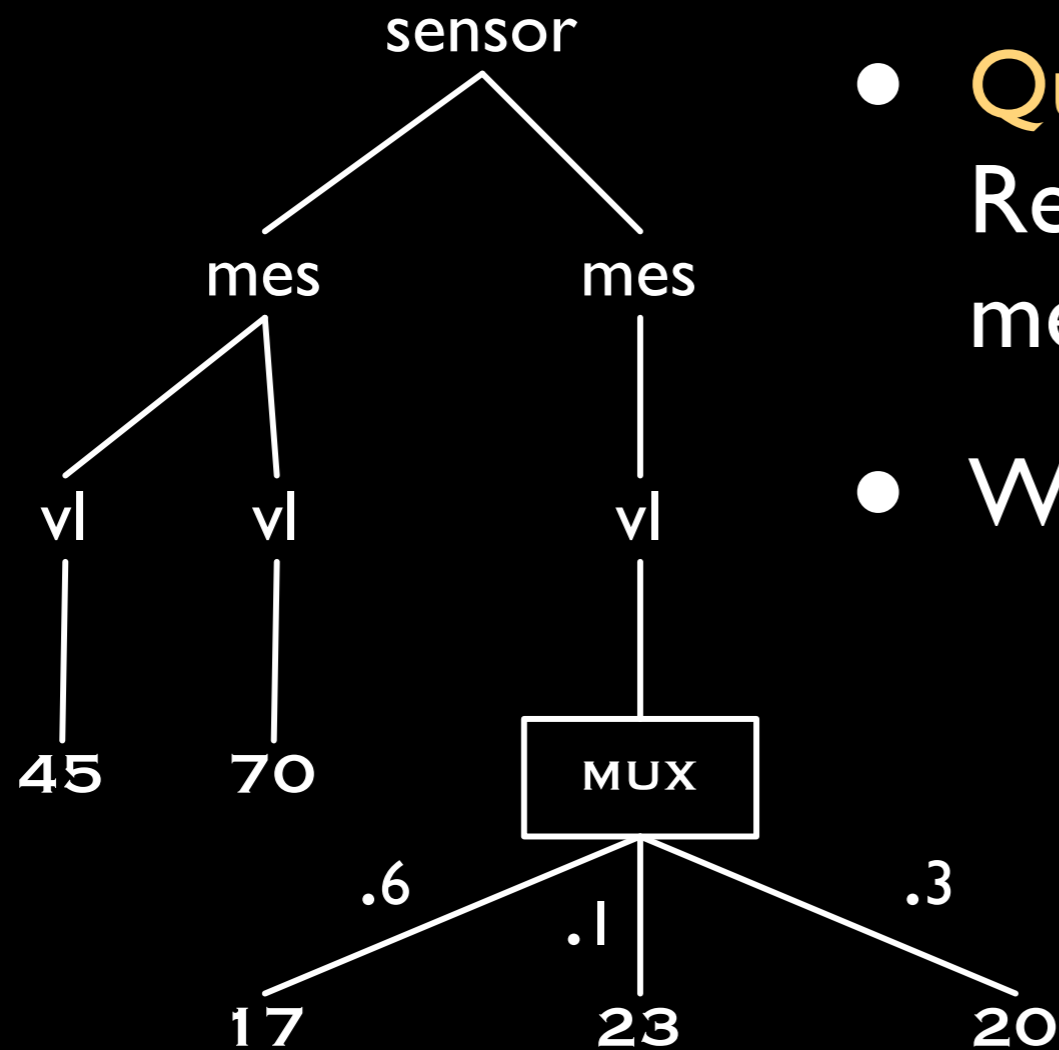
- What should be an **answer**?

$$\text{AVG}(d17) = 44, \text{Pr}(d17) = .6$$

$$\text{AVG}(d23) = 46, \text{Pr}(d23) = .1$$

$$\text{AVG}(d20) = 45, \text{Pr}(d20) = .3$$

# Semantics of AQs



- **Query:**  
Return the AVG of the measurements

- What should be an **answer**?

$$\text{AVG}(d17) = 44, \text{Pr}(d17) = .6$$

$$\text{AVG}(d23) = 46, \text{Pr}(d23) = .1$$

$$\text{AVG}(d20) = 45, \text{Pr}(d20) = .3$$

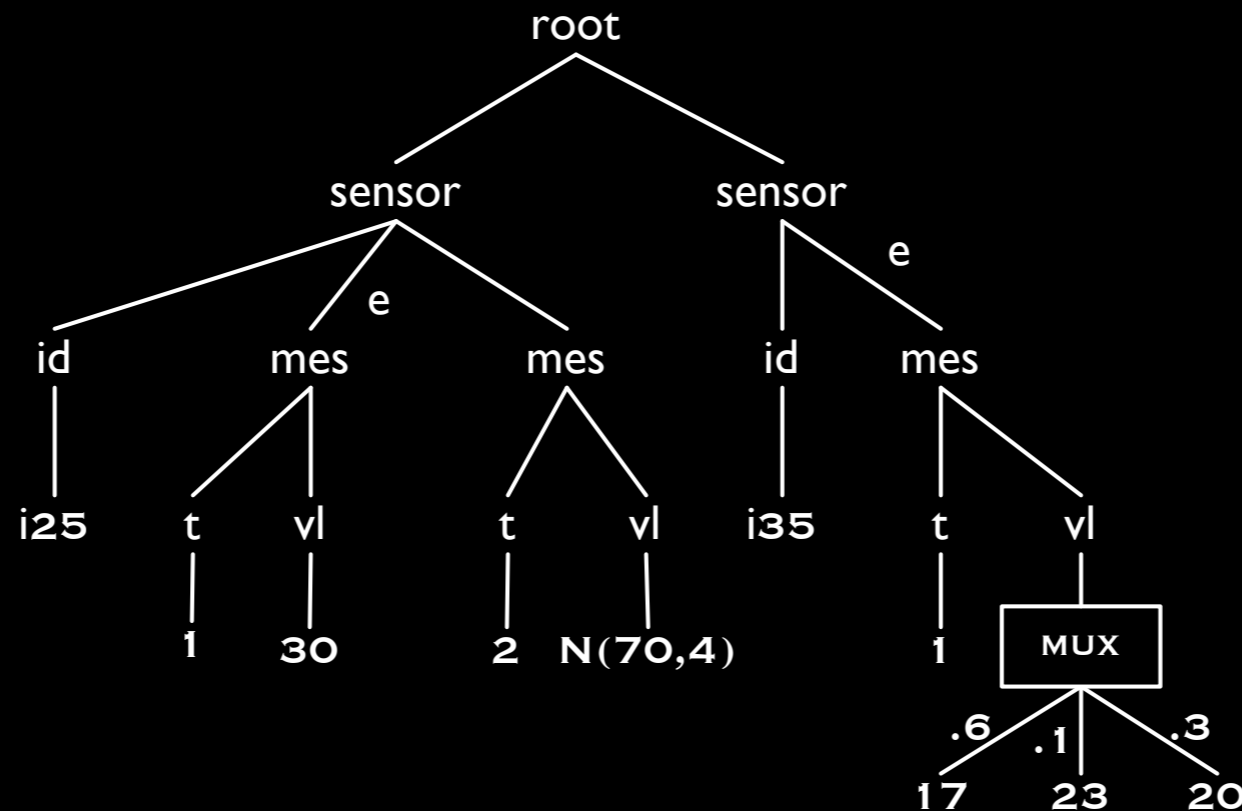
**Distribution** of aggregate values over all documents represented by the PXML document

# Problems to Investigate for Deterministic PXML

For PXML document  $D$ , constant  $C$

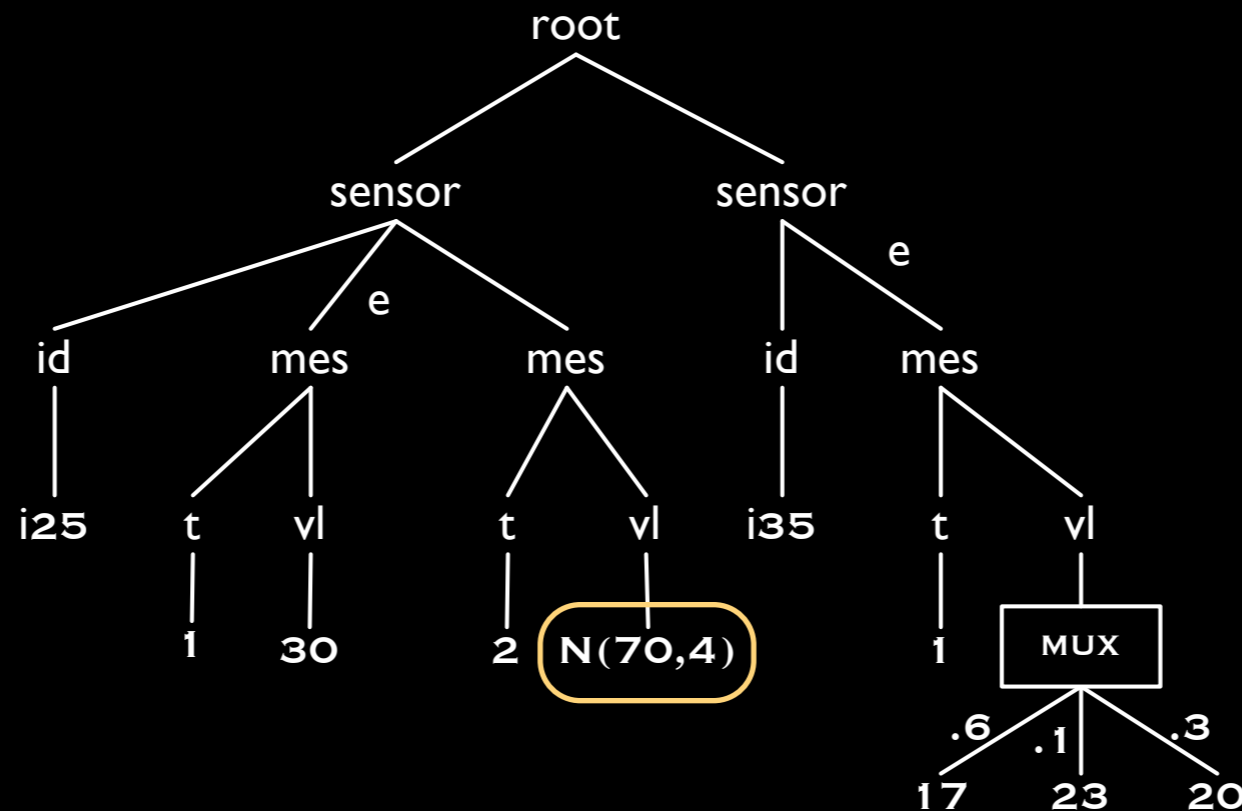
- **Possible answers:**  
decide  $\Pr(Q(D)=C) > 0$
- **Probability computation:**  
compute  $\Pr(Q(D)=C)$
- **Moment computation:**  
compute  $E(Q(D)^k)$        $E$  is “expected value”

# Continuous PXML



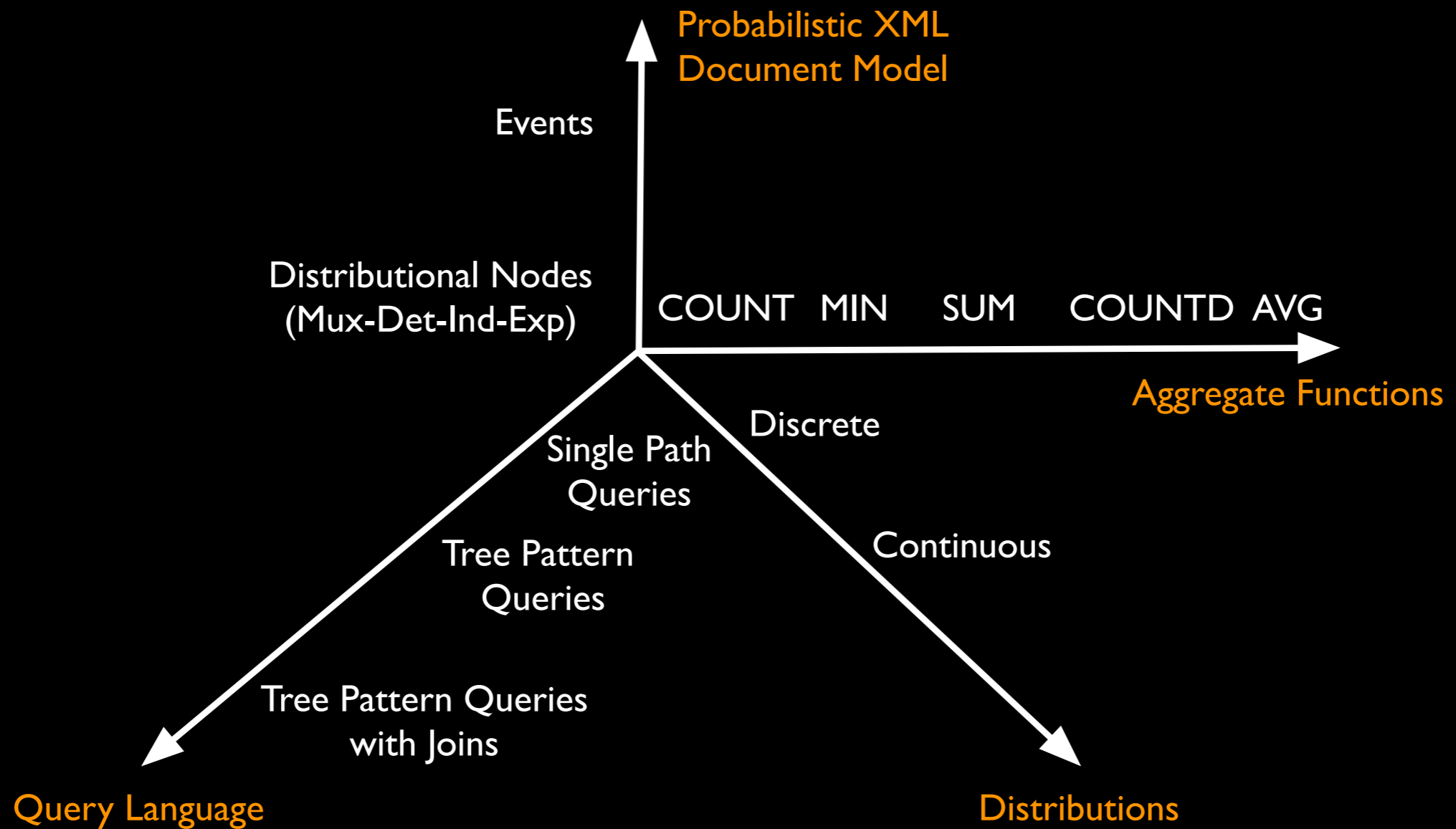
- We want to incorporate continuous distributions in PXML leaves: **semantics** of **CPXML**
- We want to **aggregate** Continuous PXML
- At the moment there is **no** formal **semantics** for continuous probabilistic XML models

# Continuous PXML



- We want to incorporate continuous distributions in PXML leaves: **semantics** of **CPXML**
- We want to **aggregate** Continuous PXML
- At the moment there is **no** formal **semantics** for continuous probabilistic XML models

# The Problem Space



### 3. Our results on Aggregation of Discrete PXML

# Aggregating PXML-Events

	Aggregate Query Language		
Problems	Single Path	Tree Pattern	Tree Pat. Joins
Possible Answers	NP-complete	NP-complete	NP-complete
Probability Computation	$FP^{\#P}$ -complete	$FP^{\#P}$ -complete	$FP^{\#P}$ -complete
Moments Computation	COUNT, SUM: PTIME MIN, COUNTD, AVG: $FP^{\#P}$ -complete	$FP^{\#P}$ -complete	$FP^{\#P}$ -complete

Aggregates: COUNT, SUM, MIN, COUNTD, AVG

Complexity: data-complexity

# Aggregating PXML with Distributional Nodes

	Aggregate Query Language		
Problems	Single Path	Tree Pattern	Tree Pat. Joins
Possible Answers	SUM,AVG, COUNTD: NP-complete		
	COUNT, MIN: PTIME		COUNT, MIN : NP
Probability Computation	SUM,AVG, COUNTD: FP <sup>#P</sup> -complete COUNT, MIN: PTIME		FP <sup>#P</sup> -complete
Probability SUM in  input  + output	PTIME	FP <sup>#P</sup> -complete	FP <sup>#P</sup> -complete
Moments Computation	PTIME	AVG: FP <sup>#P</sup> others: PTIME	FP <sup>#P</sup> -complete

Aggregates: COUNT, SUM, MIN, COUNTD,AVG  
Complexity: data-complexity

# Aggregating PXML with Distributional Nodes

	Aggregate Query Language		
Problems	Single Path	Tree Pattern	Tree Pat. Joins
Possible Answers	SUM,AVG, COUNTD: NP-complete		
	COUNT, MIN: PTIME		COUNT, MIN : NP
Probability Computation	SUM,AVG, COUNTD: FP <sup>#P</sup> -complete COUNT, MIN: PTIME		FP <sup>#P</sup> -complete
Probability SUM in  input  + output	PTIME	FP <sup>#P</sup> -complete	FP <sup>#P</sup> -complete
Moments Computation	PTIME	AVG: FP <sup>#P</sup> others: PTIME	FP <sup>#P</sup> -complete

Aggregates: COUNT, SUM, MIN, COUNTD,AVG  
Complexity: data-complexity

# Tractable Cases

Key components of tractability:

- **Hierarchical** structure of PXML documents imposed by **distributional** nodes
- Some aggregation functions can exploit the hierarchy - **monoid functions**

This allows for **P**TIME **bottom-up** evaluation of COUNT, SUM, MIN, TopK, PARITY, ...

# Monoid Functions

- Divide-and-conquer strategy on composed bags:

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

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

- COUNT, SUM, MIN ✓
- COUNTD, AVG ✗

# Bottom-up Evaluation:

- Divide-and-conquer strategy on composed probability spaces of documents:

$\alpha = \text{SUM}$

Convex Sum:

$$\alpha\left(\begin{array}{c} \text{Mux} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \\ p \quad q \end{array}\right) = p \cdot \alpha(\triangle) + q \cdot \alpha(\triangle)$$

Convolution:

$$\alpha\left(\begin{array}{c} \text{root} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \end{array}\right) = \alpha(\triangle) \oplus_{\text{SUM}} \alpha(\triangle)$$

# Bottom-up Evaluation:

- Divide-and-conquer strategy on composed probability spaces of documents:

$\alpha = \text{MIN}$

Convex Sum:

$$\alpha\left(\begin{array}{c} \text{Mux} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \\ p \quad q \end{array}\right) = p \cdot \alpha(\triangle) + q \cdot \alpha(\triangle)$$

Convolution:

$$\alpha\left(\begin{array}{c} \text{root} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \end{array}\right) = \alpha(\triangle) \oplus_{\text{SUM}} \alpha(\triangle)$$

# Bottom-up Evaluation:

- Divide-and-conquer strategy on composed probability spaces of documents:

$\alpha = \text{MIN}$

Convex Sum:

$$\alpha\left(\begin{array}{c} \text{Mux} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \\ p \quad q \end{array}\right) = p \cdot \alpha(\triangle) \text{MIN} q \cdot \alpha(\triangle)$$

Convolution:

$$\alpha\left(\begin{array}{c} \text{root} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \end{array}\right) = \alpha(\triangle) \oplus_{\text{SUM}} \alpha(\triangle)$$

# Bottom-up Evaluation:

- Divide-and-conquer strategy on composed probability spaces of documents:

$\alpha = \text{MIN}$

Convex Sum:

$$\alpha\left(\begin{array}{c} \text{Mux} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \\ p \quad q \end{array}\right) = p \cdot \alpha(\triangle) \text{MIN} q \cdot \alpha(\triangle)$$

Convolution:

$$\alpha\left(\begin{array}{c} \text{root} \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \end{array}\right) = \alpha(\triangle) \oplus_{\text{MIN}} \alpha(\triangle)$$



# Approximating Query Answers

- Many problems are NP- or FP#P-complete
- There are efficient **Monte-Carlo** approximation techniques for all the problems
- For example: given Epsilon and Delta with polynomially many samples one can compute the estimation  $X$  such that
$$| P(\text{AGG}(D) = C) - X | > \text{Epsilon}$$
holds with probability at most Delta

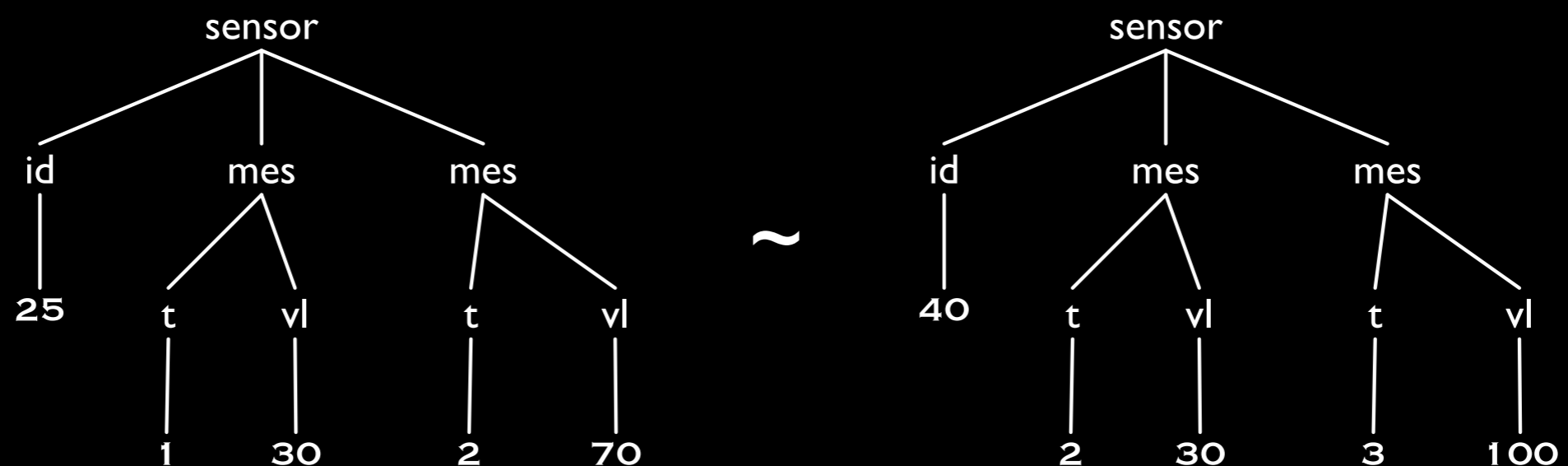
## 4. Continuous Probabilistic Models

# Finite vs Continuous Models

- Finite case:
  - **finite** sets of trees
  - where **every tree** has a non-zero probability
- Continuous case:
  - **infinite** sets of trees
  - where **some** (infinite) **subsets** of trees have non-zero probability measure

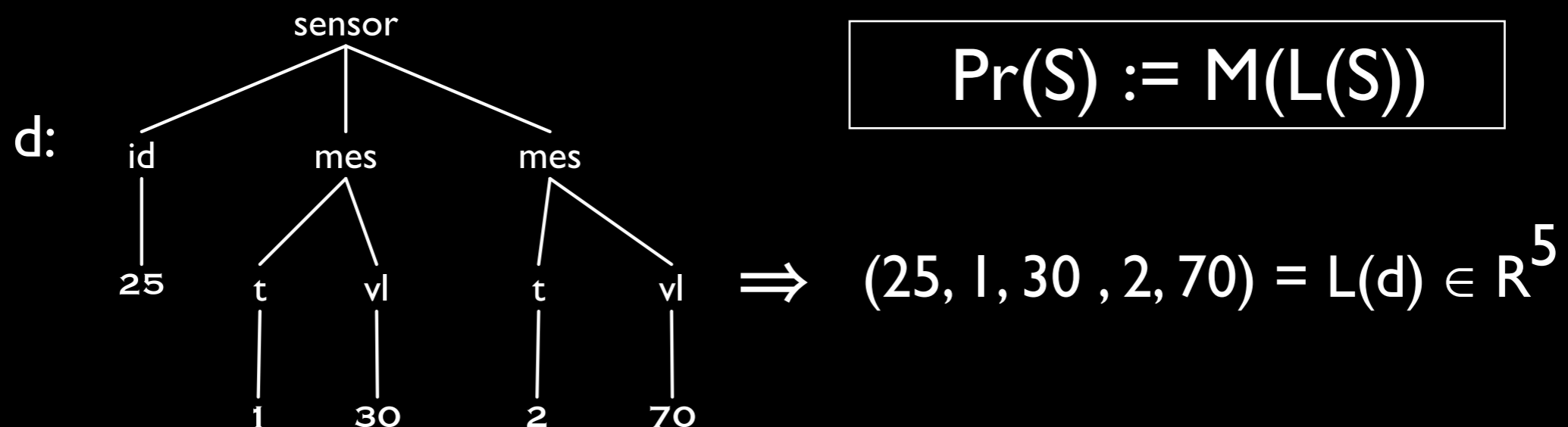
# How to Measure Infinite Sets of Trees?

- Defined for trees with leaf-labels in **Reals**
- Exploits the notion of **structure equivalence**:  
 $d \sim d'$  if  $d$  the same as  $d'$  up to renaming of leaves
- $[d]$  is a set of all trees equivalent to  $d$



# Probability for Structure-Equivalent Trees

- We first define probability for **one** class [d]
- $S \subseteq [d]$  then  $L(S) \subseteq \mathbb{R}^n$   
 = the set of tuple of leave values for S.  
 S is a **Borel** set
- Let M be a measure on **Borel** sets:



# Probability for any Infinite Sets of Trees

- We extend  $\text{Pr}$  to sets of XML documents that are **finite unions** of equivalence classes:

$$G = [d_1] \cup \dots \cup [d_n]$$

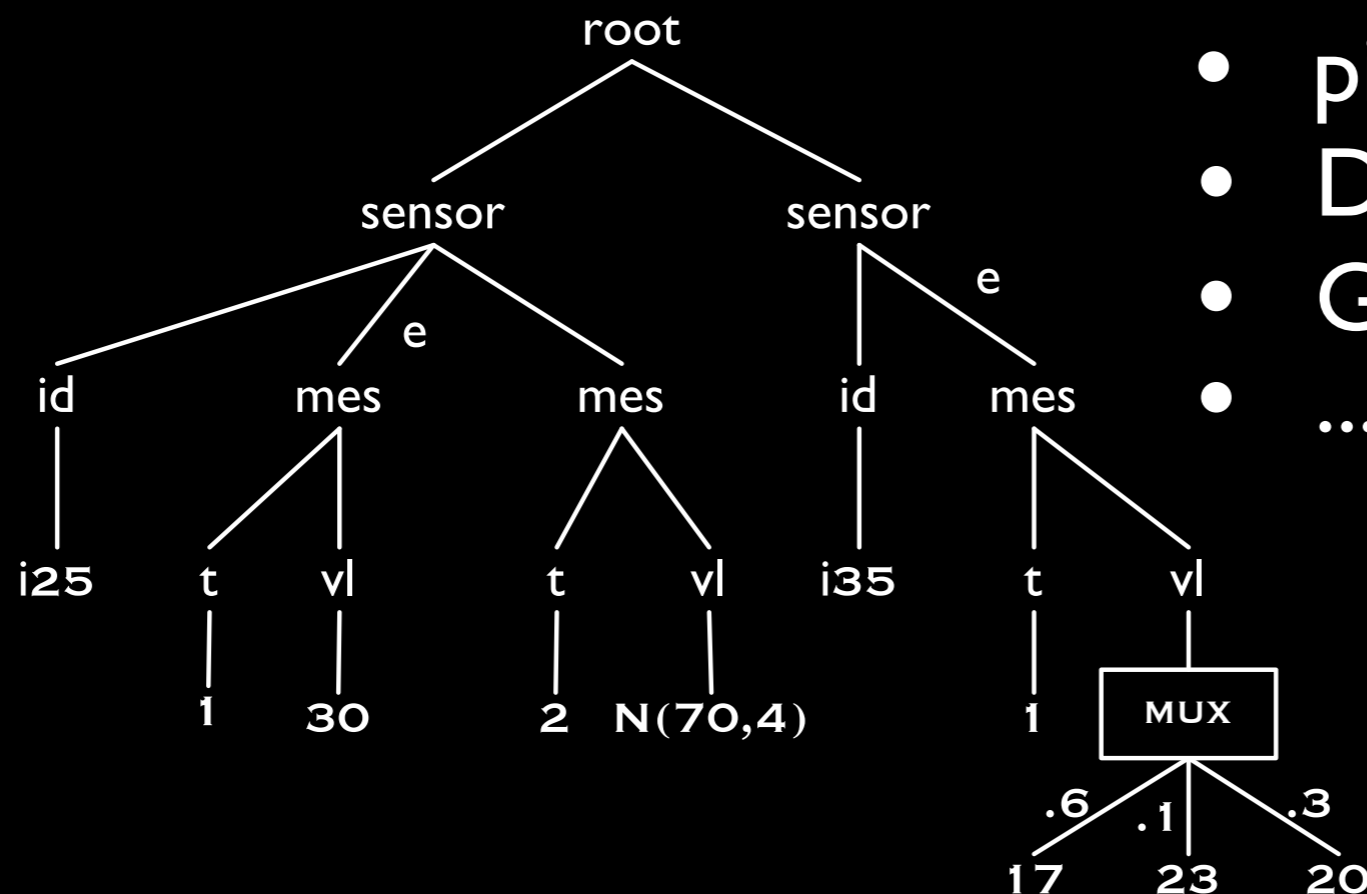
- Let  $p_1, \dots, p_n$  be **weights** of  $[d_i]$ , that is  $p_1 + \dots + p_n = 1$   
Then

If  $S = S_1 \cup \dots \cup S_n$  for  $S_i \subseteq [d_i]$  then

$$\text{Pr}(S) := p_1 \cdot \text{Pr}_1(S_1) + \dots + p_n \cdot \text{Pr}_n(S_n)$$

# Continuous PXML Documents

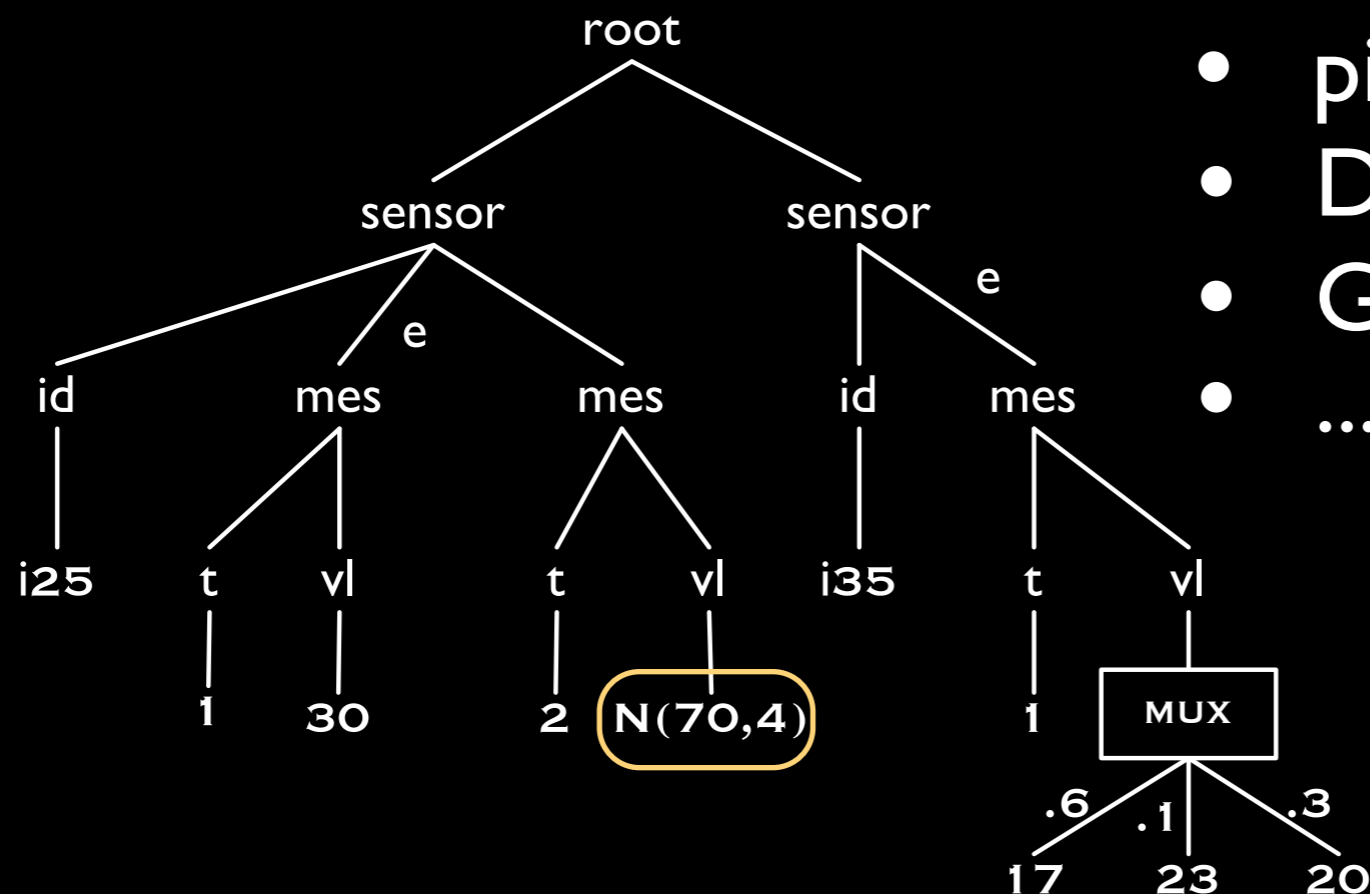
- Extension of discrete PXML with distribution functions stored in the leaves



- piecewise polynomials
- Diracs
- Gaussian
- ...

# Continuous PXML Documents

- Extension of discrete PXML with distribution functions stored in the leaves



- piecewise polynomials
- Diracs
- Gaussian
- ...

# Monoid Aggs. for CPXML with Distr. Nodes.

- How to compute?
  1. Compute aggregation distributions on the leaves
  2. Push distributions bottom-up combining them with **convolutions** and **convex sums**
- It works when distribution on the leaves are closed under convolutions and convex sum
  - piecewise polynomials (SUM, MIN/MAX) **PTIME**
  - Gaussian distributions (SUM) **PTIME**

## 5. Further Challenges

# PXML vs. Pr RDBs

Relational Case: BID model [Re&Suciu:2006]	XML Case: distributional nodes
Some SPJR queries: <b>PTIME</b>	TP queries: <b>PTIME</b>
Other queries: <b>#P-hard</b>	non-trivial TPJ queries: <b>#P-hard</b>
Separation: <b>complex</b> conditions	Separation: <b>join</b> variables

# PXML vs. Pr RDBs

Relational Case: BID model [Re&Suciu:2006]	XML Case: distributional nodes
Some SPJR queries: <b>PTIME</b>	TP queries: <b>PTIME</b>
Other queries: <b>#P-hard</b>	non-trivial TPJ queries: <b>#P-hard</b>
Separation: <b>complex</b> conditions	Separation: <b>join</b> variables

- Why does the XML case seem simpler?
- Is there some insight to be gained from one case to the other?
- Translating XML data and queries to the relational case yields queries with self-joins, a less well-understood setting

# Tractable Extensions

- Arbitrary dependencies (events): **not tractable**
- Local dependencies (MUX): **not practical**
- Somewhere in between?
  - Why arbitrary dependencies are hard?
  - How to generalize local dependencies, while remaining tractable?

# Tractable Extensions

- Arbitrary dependencies (events): **not tractable**
- Local dependencies (MUX): **not practical**
- And can we go further? cf. XML schemas
  - Trees of unbounded depth ?
  - Trees of unbounded width ?
  - Infinite trees?

# Where do Probabilities Come From?!

- Do the numbers assigned as probabilities in PDBMS really **make sense?**
- In some cases, sources of “**good**” probabilities:
  - Statistics
  - Conditional Random Fields
- What about the rest?  
Does it really make sense to model **uncertainty with probabilities?**

# A System That Just Works

- Nothing but **toy systems** exist for PXML
- What should **the system** be based upon:
  - a probabilistic relational DBMS?
  - a native XML DBMS?
- Systems issue: distribution, indexing, etc.
- And need for a **killer application!**  
Probabilistic content warehouse?

# Summing Up

- We got a **comprehensive picture** of aggregation for **discrete PXML**:
  - PXML models with local and global
  - SP, TP, TPJ queries
  - COUNT, SUM, MIN, COUNTD, AVG functions
- We **introduced continuous PXML model** and started studying its aggregation

# Summing Up

Outside: intractable,  
i.e., NP-,  $FP^{\#P}$ -complete

Inside: PTIME

