Aggregating Discrete and Continuous Probabilistic XML

S. Abiteboul,[1] T-H. H. Chan,[2] E. Kharlamov,[1,3] W. Nutt,[3] P. Senellart[4]

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

The University of Hong Kong, November 2009

Outline

1. Probabilistic data management
2. Problem definition:
   How to aggregate Probabilistic XML?
3. Aggregating discrete Probabilistic XML
4. Aggregating continuous Probabilistic XML
5. Further challenges

1. Probabilistic data management
   - Definition
   - Applications
   - Key models

What is a Probabilistic Database?

- PrDB = Set of DBs with probabilities

P = 0.3

P = 0.2

P = 0.5

What is a Probabilistic Database?

- PrDB = Set of DBs with probabilities

P = 0.3

P = 0.2

P = 0.5

- Query answer over PrDB: answer, prob.

What is a Probabilistic Database?

- PrDB = Set of DBs with probabilities

P = 0.3 P = 0.2 P = 0.5

Q Q Q

A A A

- Query answer over PrDB: answer, prob.
  (A, 0.2+0.5)

Where Do the Probabilities Live?

In probabilistic events:

- An item belongs to the database
- A database instance is a correct one
- A tuple is an answer to a query

Where Do the Probabilities Live?

In probabilistic events:

Name	Tel	Prob
Pierre	555 555	0.5
Evgeny	222 333	0.8
Hubert	322 111	0.3

Where Do the Probabilities Live?

In probabilistic events:

- An item belongs to the database
- A database instance is a correct one
- A tuple is an answer to a query

Where Do the Probabilities Live?

In probabilistic events:

Name	Tel
Pierre	555 555
Evgeny	222 333
Hubert	322 111

P = 0.3

Name	Tel
Pierre	555
Evgeny	222
Hubert	322
Werner	999

P = 0.2

Name	Tel
Bob	555
Bill	222
Alex	322

P = 0.5

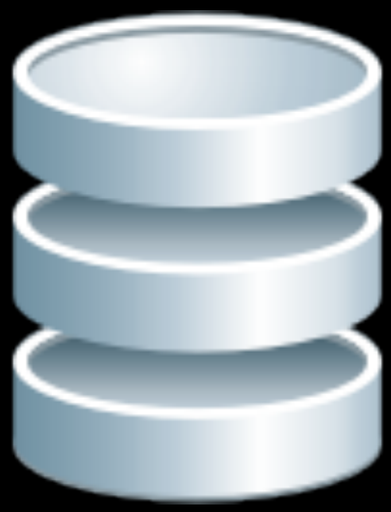
Where Do the Probabilities Live?

In probabilistic events:

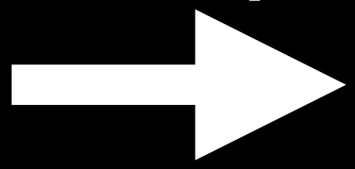
- An item belongs to the database
- A database instance is a correct one
- A tuple is an answer to a query

Where Do the Probabilities Live?

In probabilistic events:



Query



Name	Prob
Pierre	0.5
Evgeny	0.8
Hubert	0.3

Where Do the Probabilities Live?

In probabilistic events:

- An item belongs to the database
- A database instance is a correct one
- A tuple is an answer to a query

- Applications

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

- Relational Probabilistic DBs

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

Name	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

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 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

?

MayBMS

[Antova,Jansen,KochOlteanu:2007]

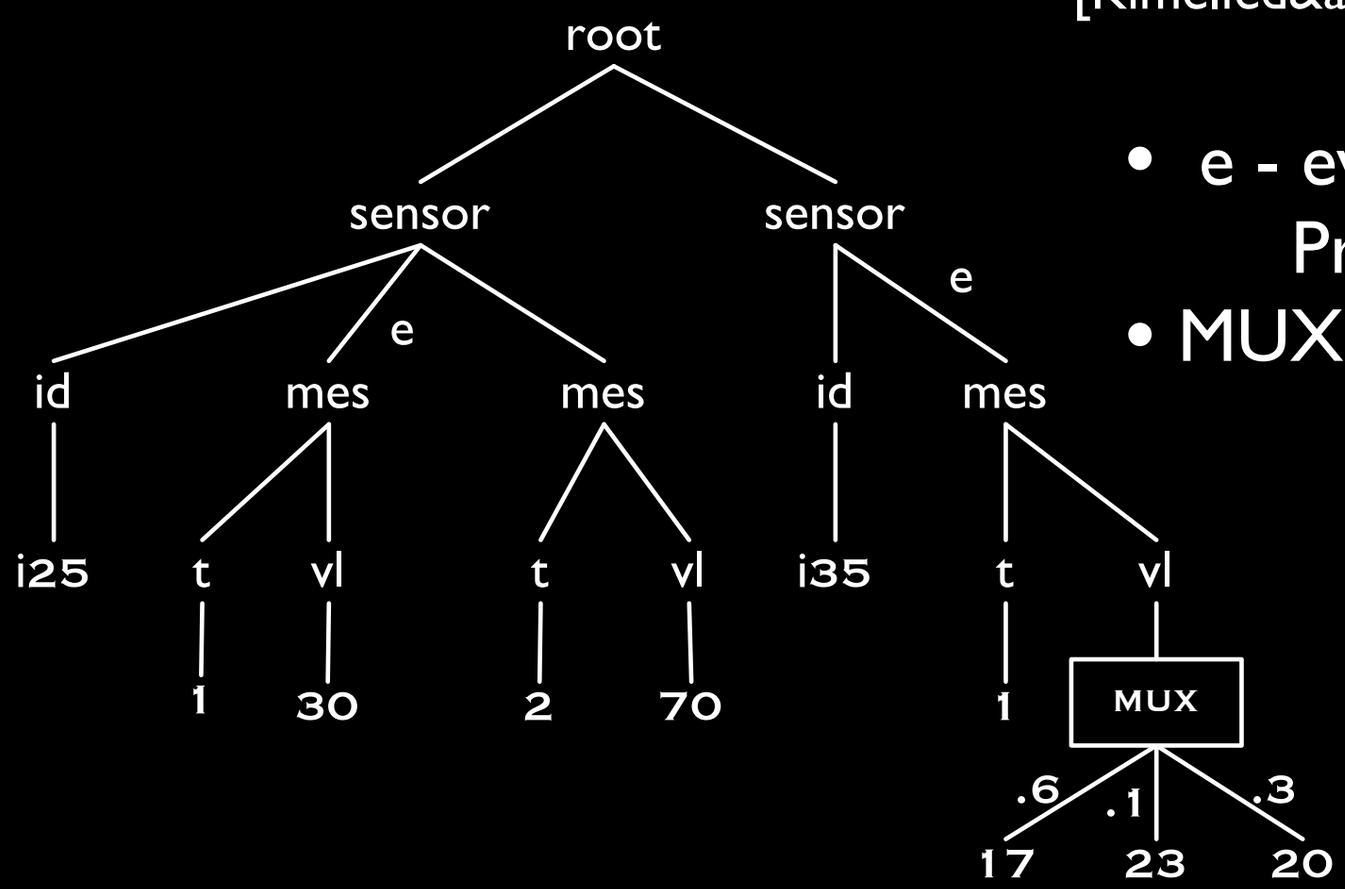
- In a nutshell: Probabilistic C-Tables,
  i.e., tables with (random) variables and
  Boolean conditions on the variables

- 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

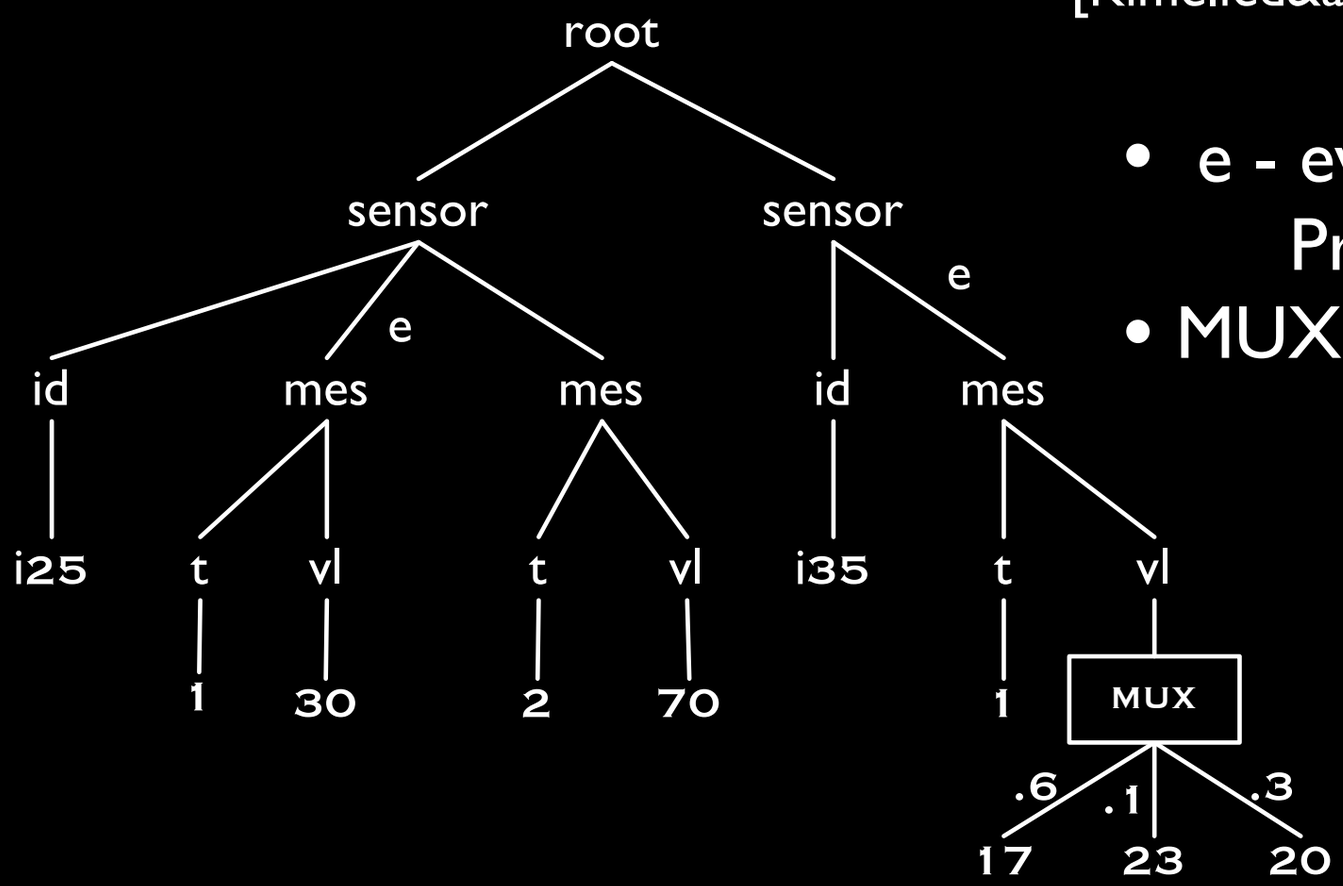
[Kimelfed&at.al.:2007] [Senellart&at.al.:2007]



- 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]



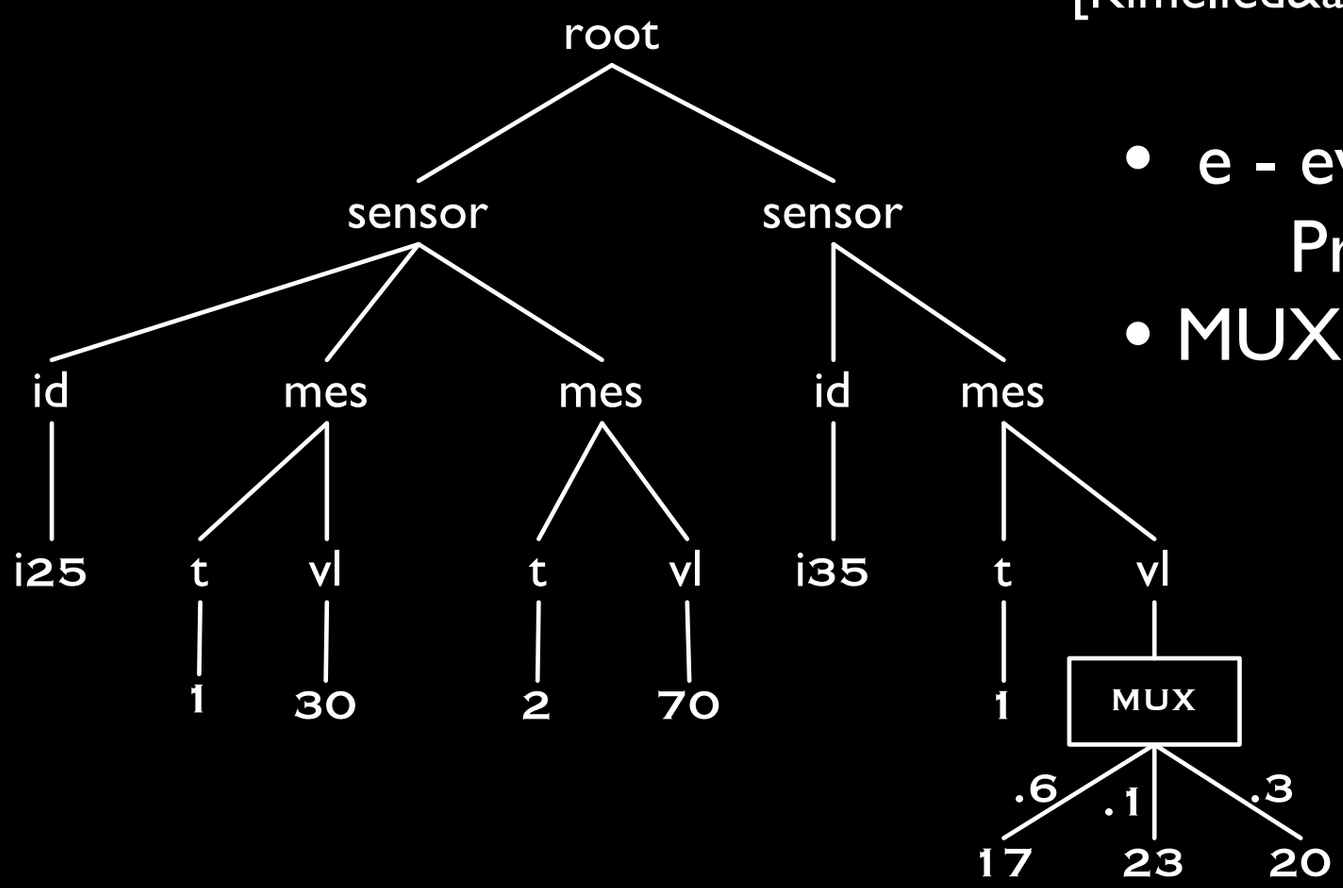
- e - event "sensor is useful"
  Pr(e) = .4
- MUX - mutually exclusive options

Semantics: first world

- e = true (measurement at time one is useful)
- MUX: 23

PXML with Events and Distributional Nodes

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



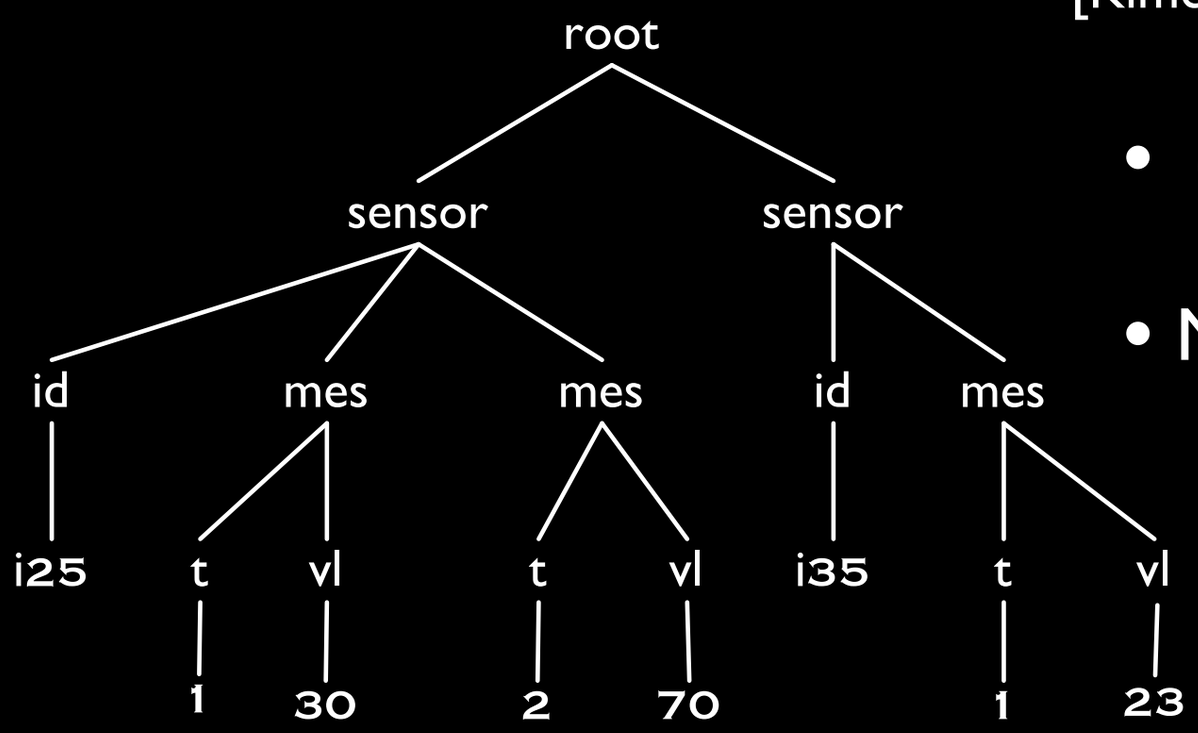
- e - event "sensor is useful"
  Pr(e) = .4
- MUX - mutually exclusive options

Semantics: first world

- e = true (measurement at time one is useful)
- MUX: 23

PXML with Events and Distributional Nodes

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



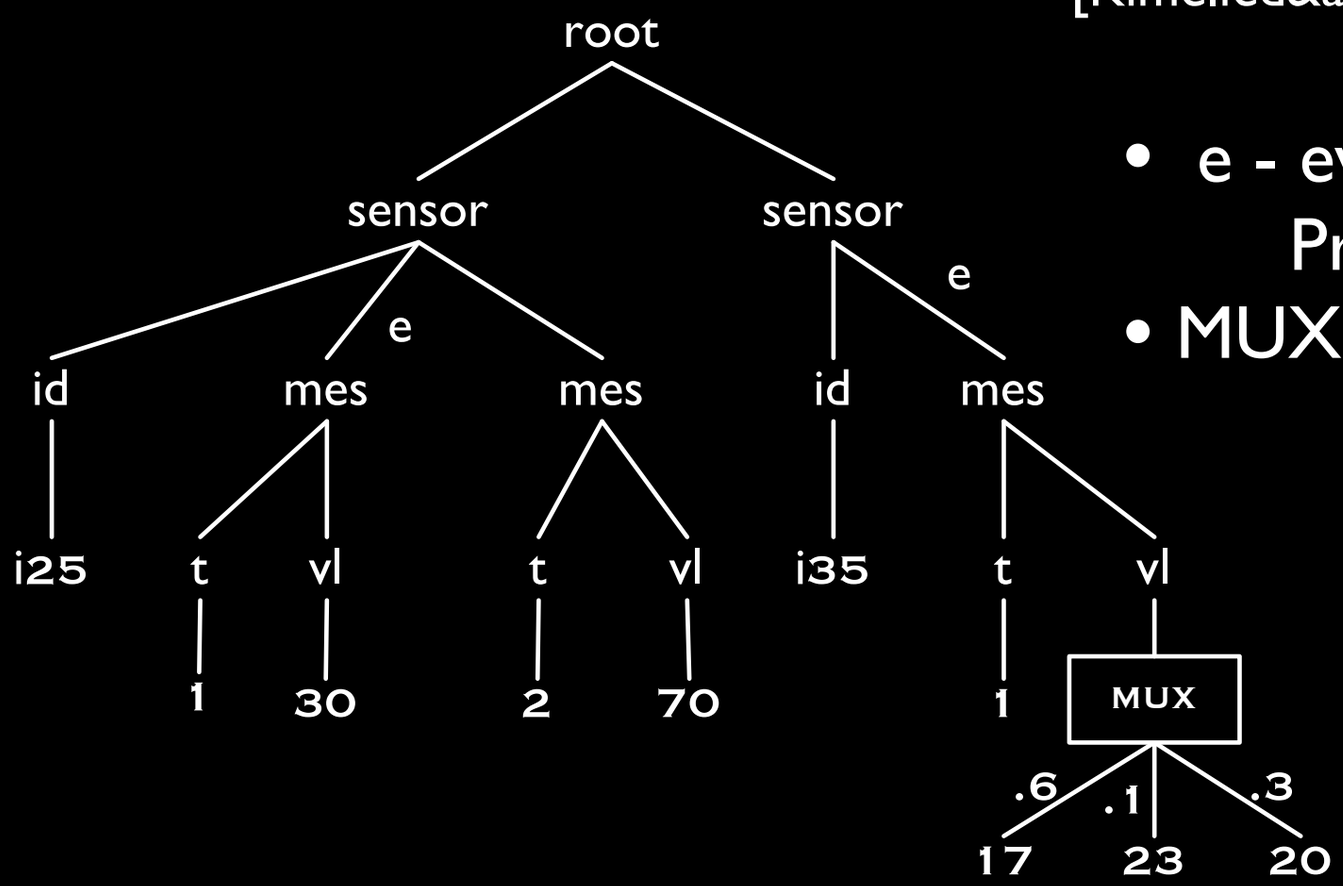
- e - event "sensor is useful"
  Pr(e) = .4
- MUX - mutually exclusive options

Semantics: first world

- e = true (measurement at time one is useful)
- MUX: 23

PXML with Events and Distributional Nodes

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



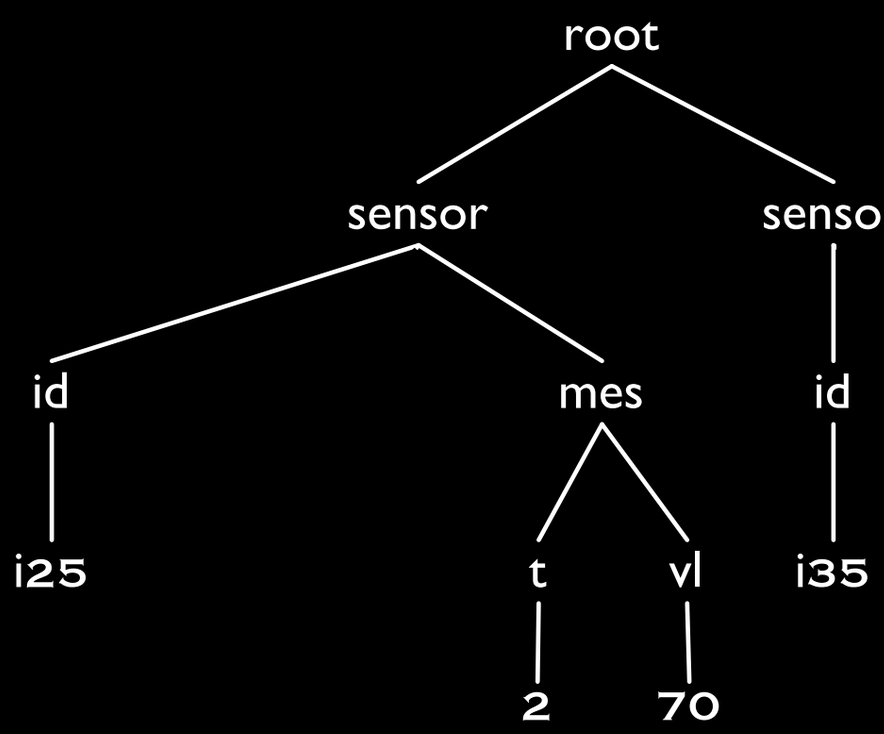
- e - event "sensor is useful"
  Pr(e) = .4
- MUX - mutually exclusive options

Semantics: second world

- e = false (measurement at time one is useless)

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: second world

- e = false (measurement at time one is useless)

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 TP Queries ~ conjunctive, no joins
  - Distributional nodes: PTIME
  - Events: $FP^{\#P}$-complete
- Querying PXML with distributional nodes + NO events + constraints with aggregate functions:
  - PTIME for COUNT and MIN
  - NP-hard for SUM and AVG
- Monte-Carlo simulations for intractable case

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

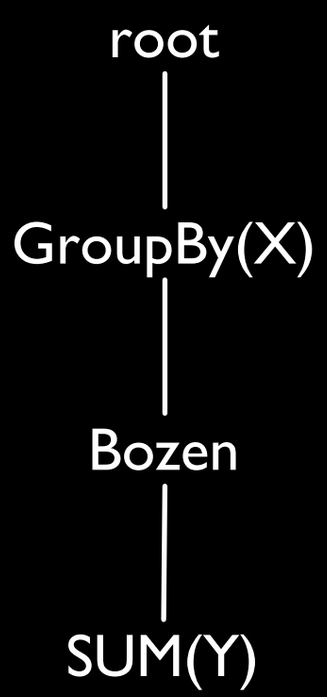
Query Models

a. Single-Path queries - SP

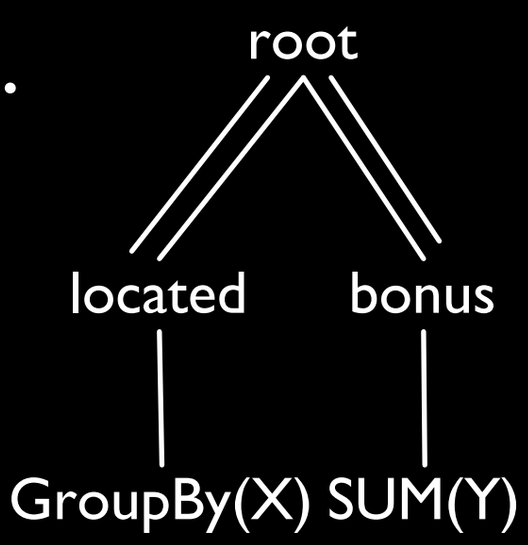
b. Tree-Pattern queries - TP

c. Tree-Pattern queries with Joins - TPJ

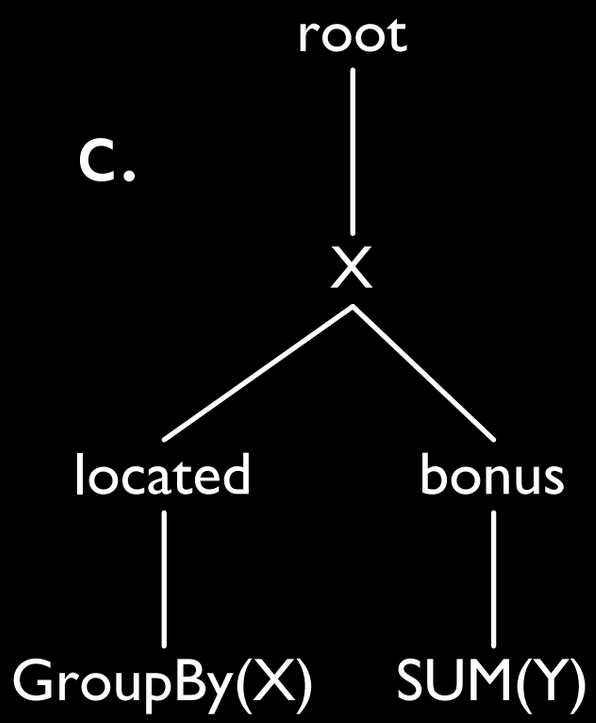
a.



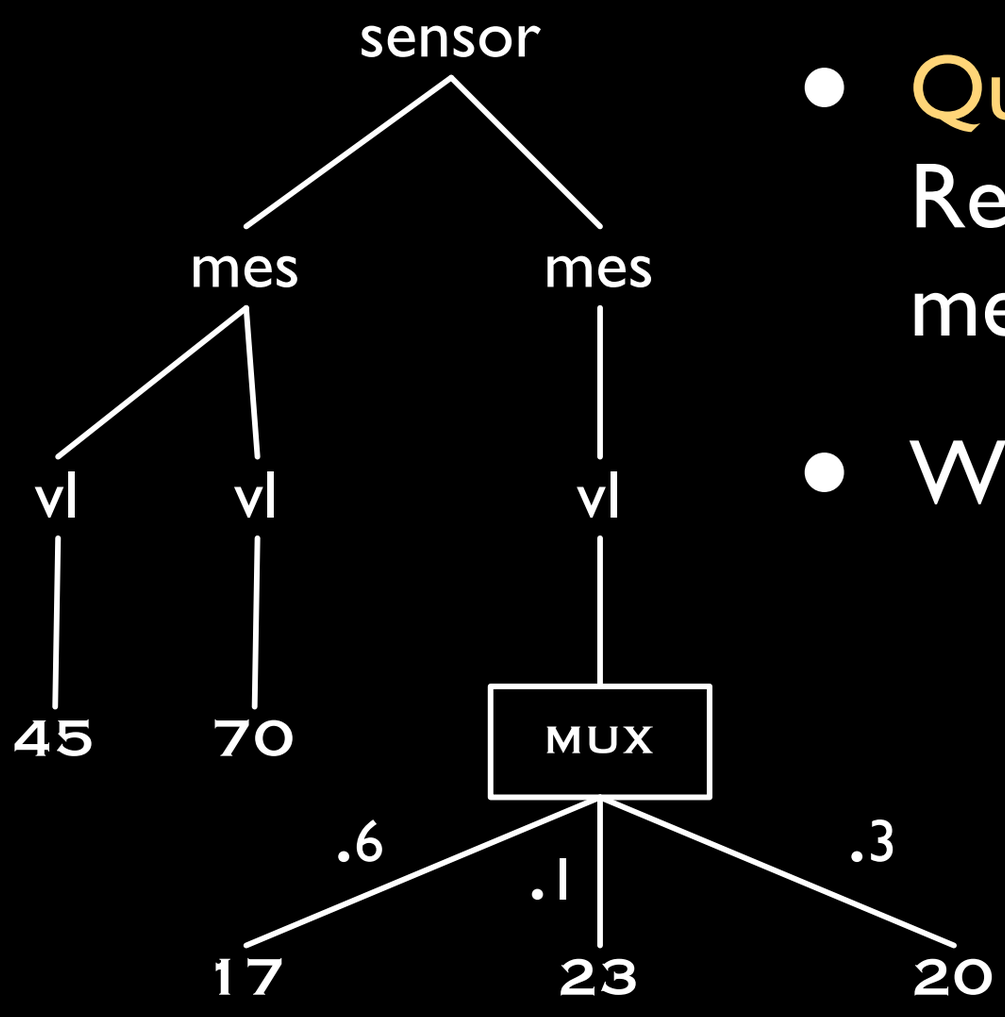
b.



c.



Semantics of AQs



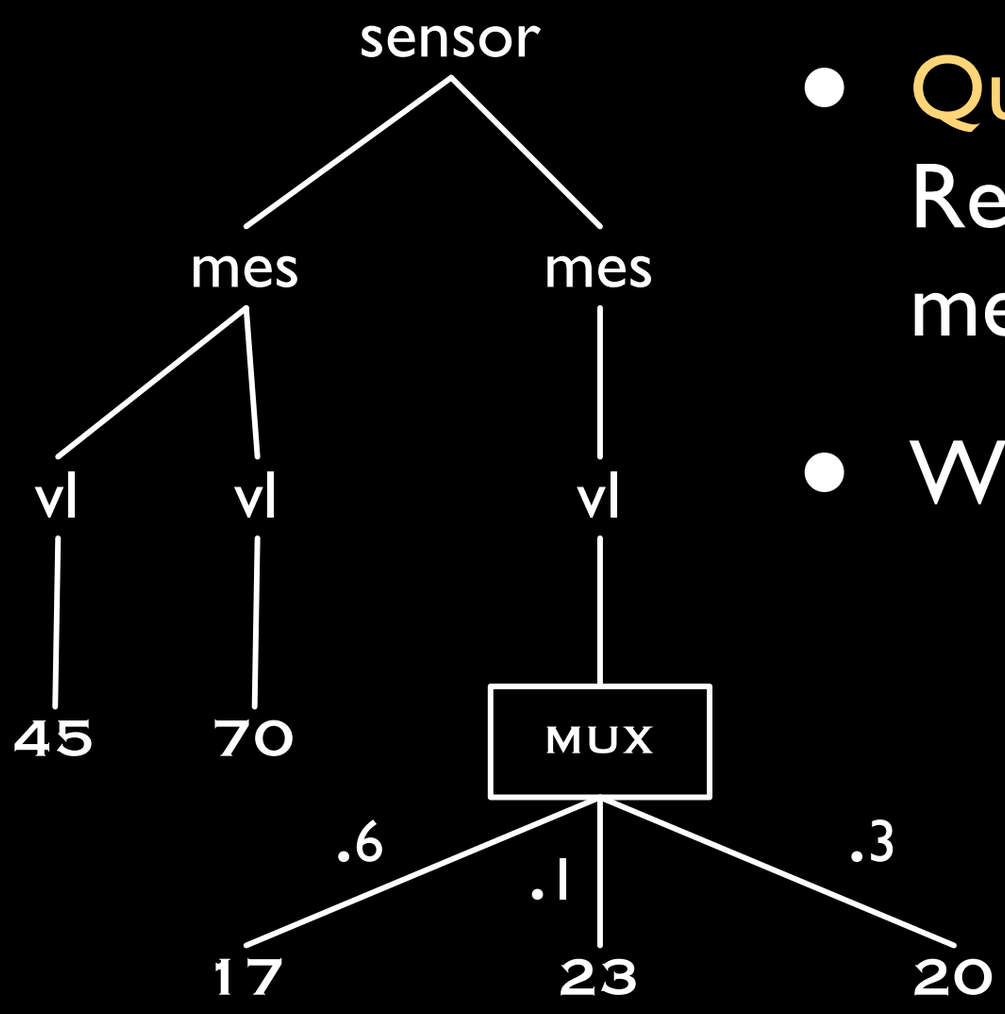
- Query:
  Return the AVG of the measurements
- What should be an answer?

AVG(d17) = 44, Pr(d17)=.6

AVG(d23) = 46, Pr(d23)=.1

AVG(d20) = 45, Pr(d20)=.3

Semantics of AQs



- Query:
  Return the AVG of the measurements

- What should be an answer?

  AVG(d17) = 44, Pr(d17)=.6

  AVG(d23) = 46, Pr(d23)=.1

  AVG(d20) = 45, Pr(d20)=.3

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

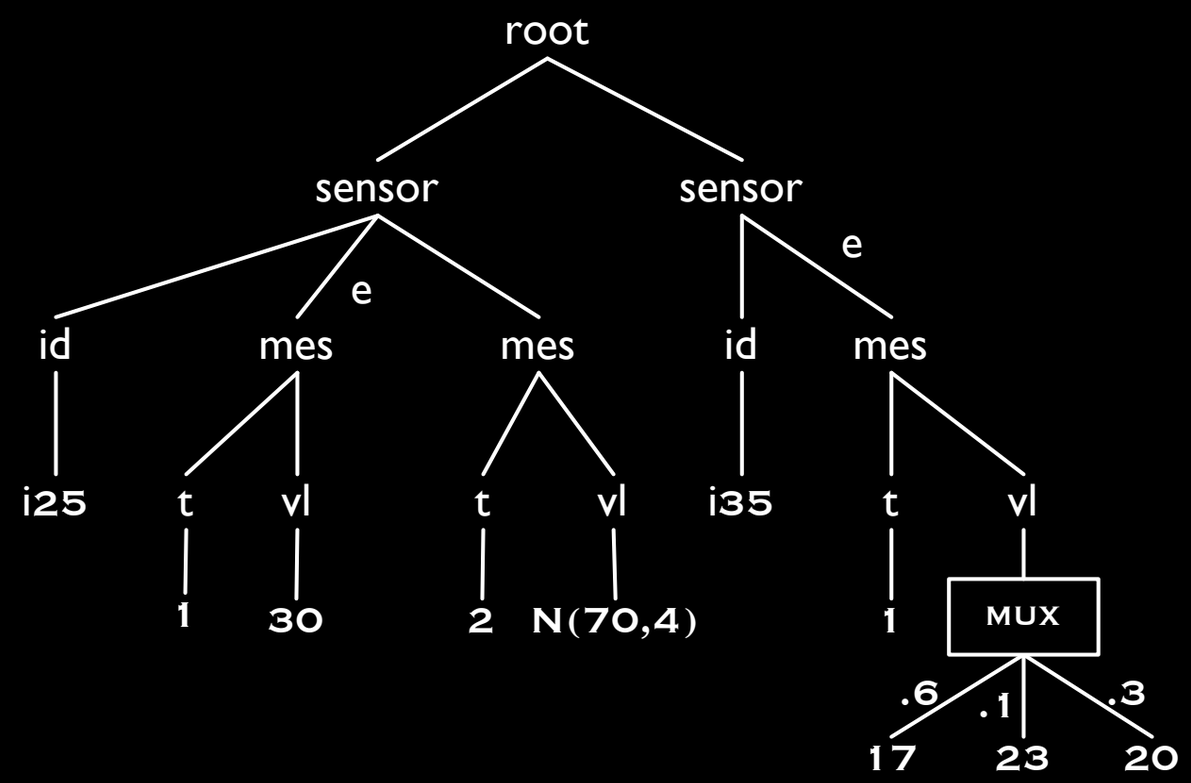
Problems to Investigate for Deterministic PXML

For PXML document D, constant C

- Possible answers:
  decide Pr(Q(D)=C) > 0
- Probability computation:
  compute Pr(Q(D)=C)
- Moment computation:
  compute $E(Q(D)^k)$

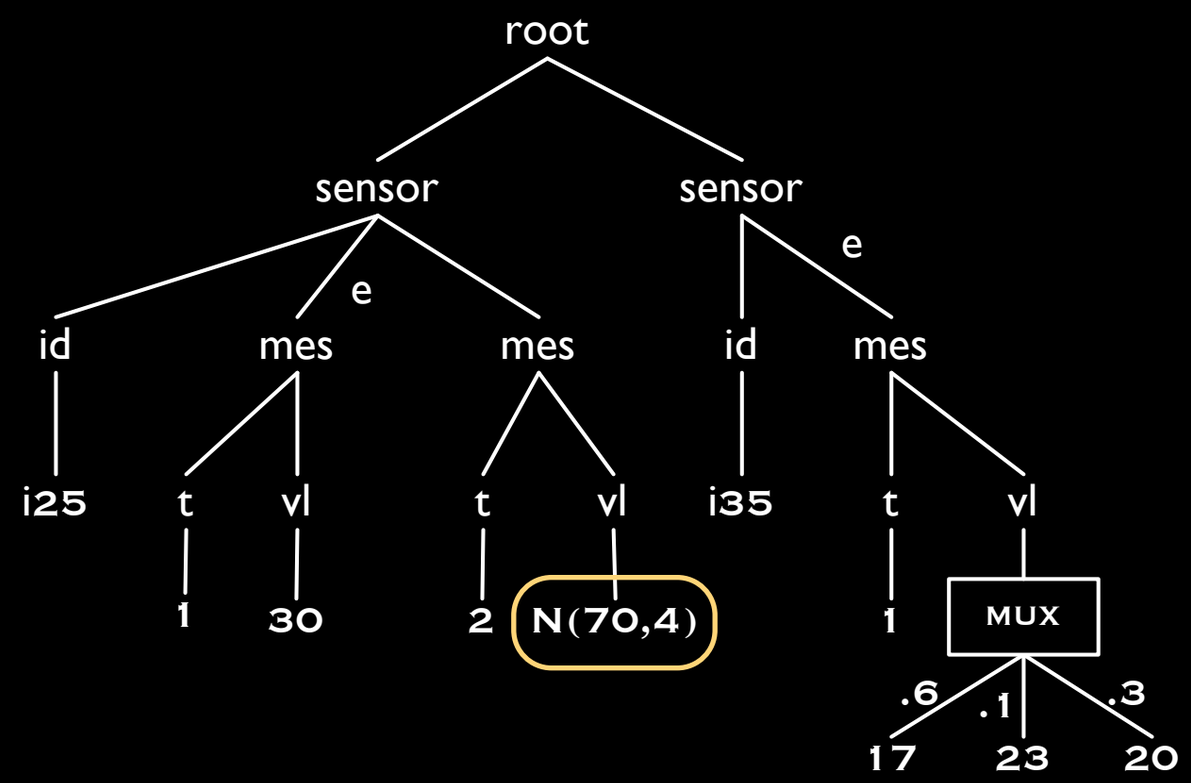
E is "expected value"

Continuous PXML



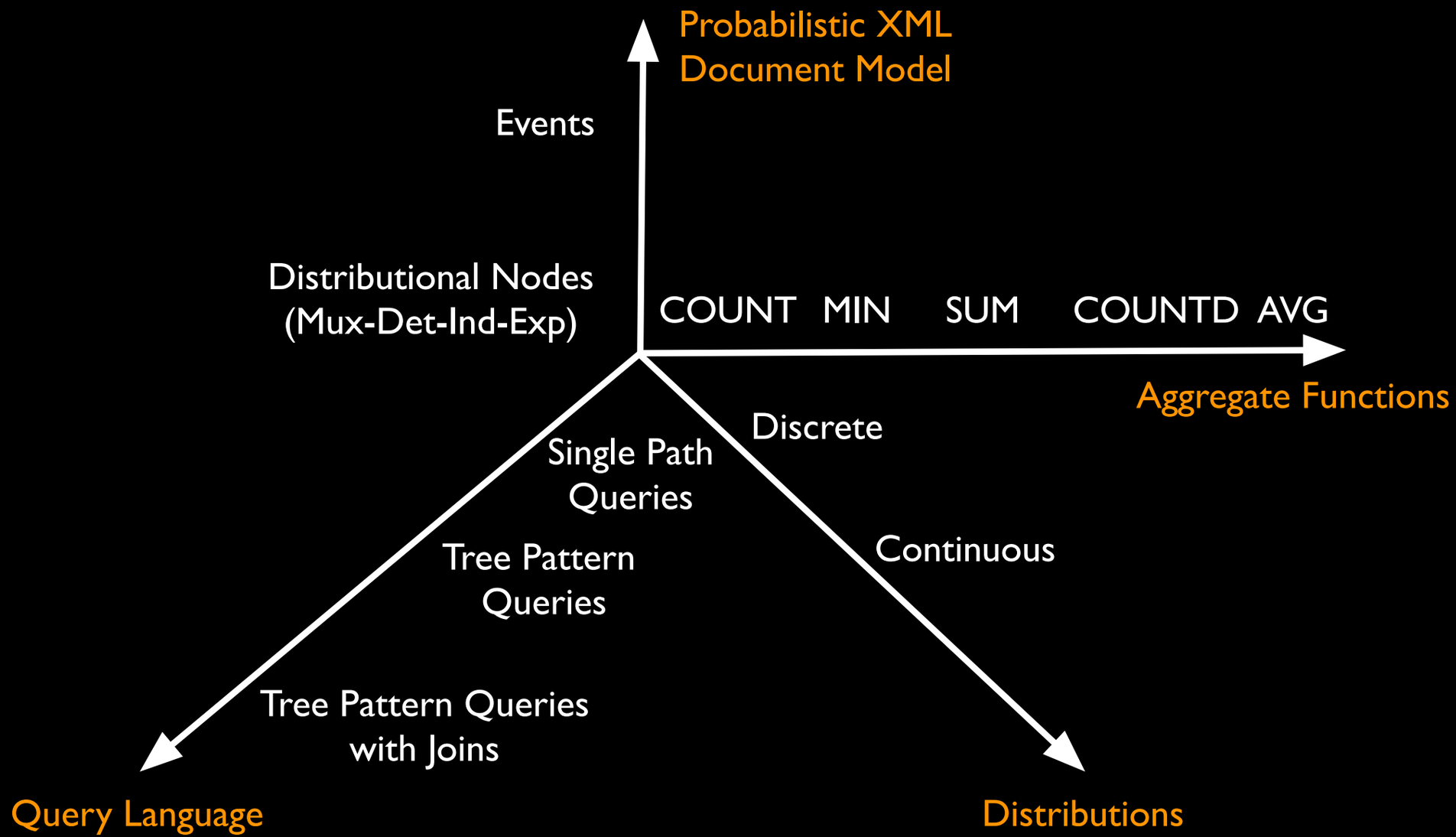
- We want to incorporate continuous distributions in PXML leaves: semantics of CPXML
- We want to aggregate Continuous PXML
- At the moment there is no formal semantics for continuous probabilistic XML models

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

Aggregating PXML-Events

- NP-hardness of possible answers:
  falsifiability for DNF and subset sum problems
- #P-hardness of probability and moments:
  counting # of sat. assignments, # set K-covers, ...
- PTIME moments for COUNT and SUM:
  regrouping sums

Aggregating PXML with Distributional Nodes

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

Aggregates: COUNT, SUM, MIN, COUNTD, AVG

Aggregating PXML with Distributional Nodes

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

Aggregates: COUNT, SUM, MIN, COUNTD, AVG

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 PTIME bottom-up evaluation of COUNT, SUM, MIN, TopK, PARITY, ...

Example of Bottom-up Evaluation

- Divide-and-conquer strategy on composed bags:

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

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

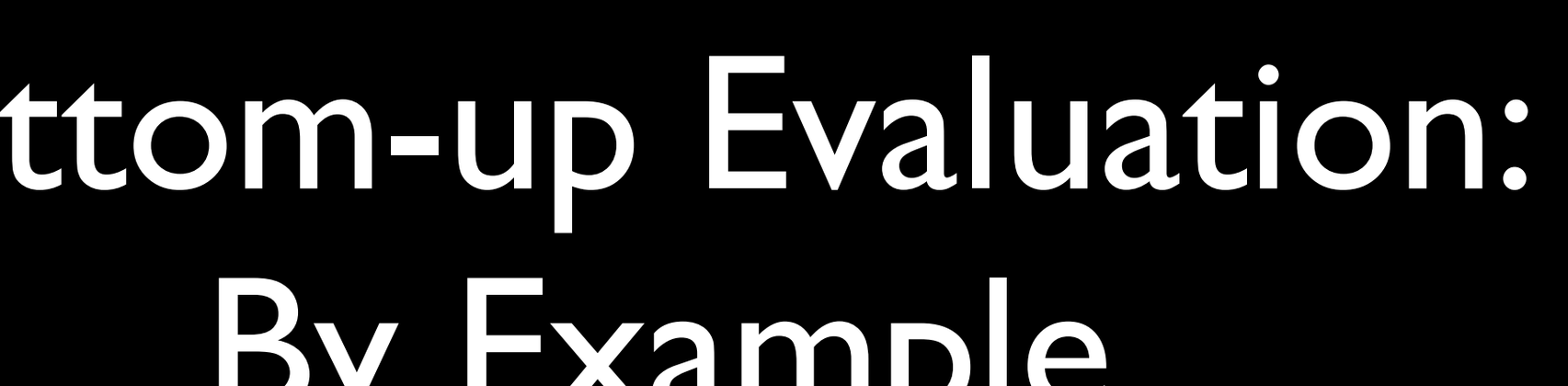
- COUNT, SUM, MIN ✓
- COUNTD, AVG ✗

Bottom-up Evaluation: By Example

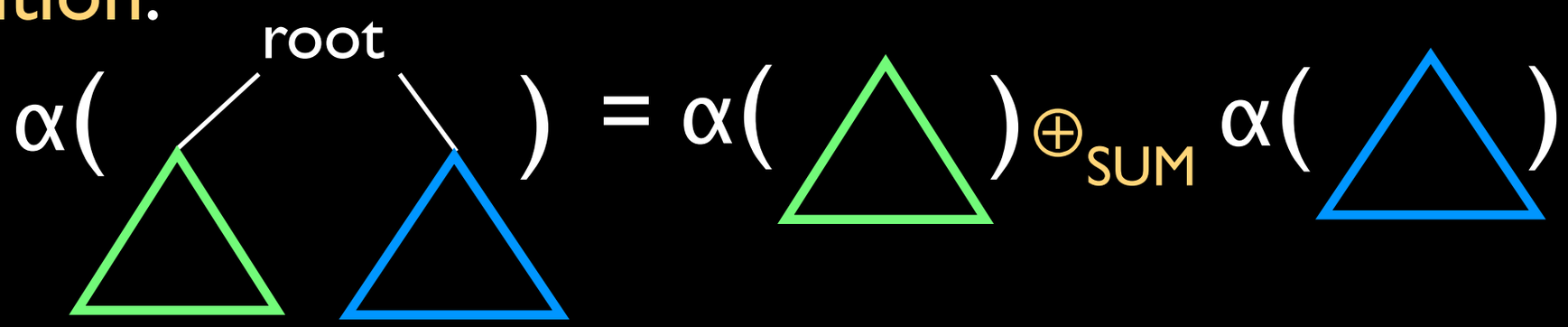
- Divide-and-conquer strategy on composed probability spaces of documents:

α=SUM

Convex Sum:



Convolution:

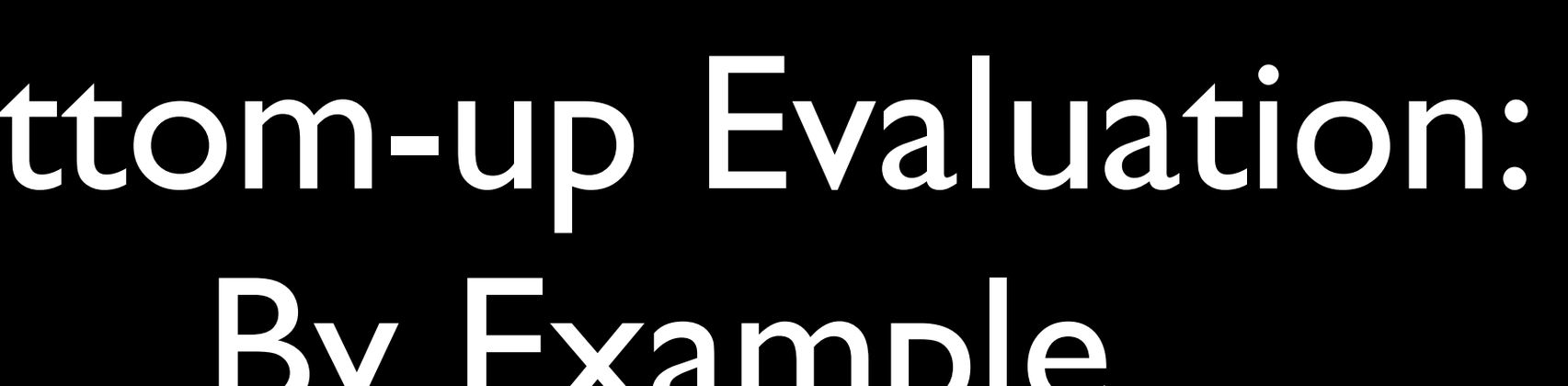


Bottom-up Evaluation: By Example

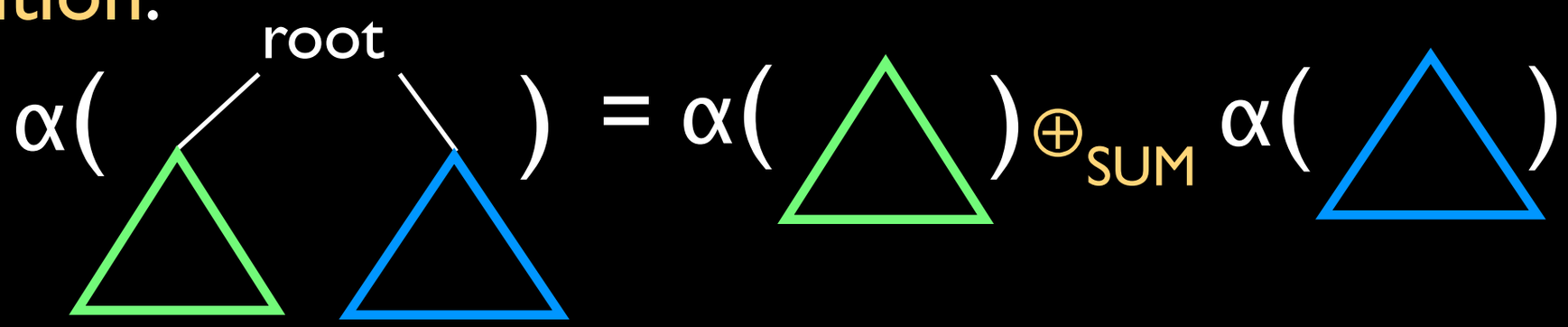
- Divide-and-conquer strategy on composed probability spaces of documents:

α=MIN

Convex Sum:



Convolution:

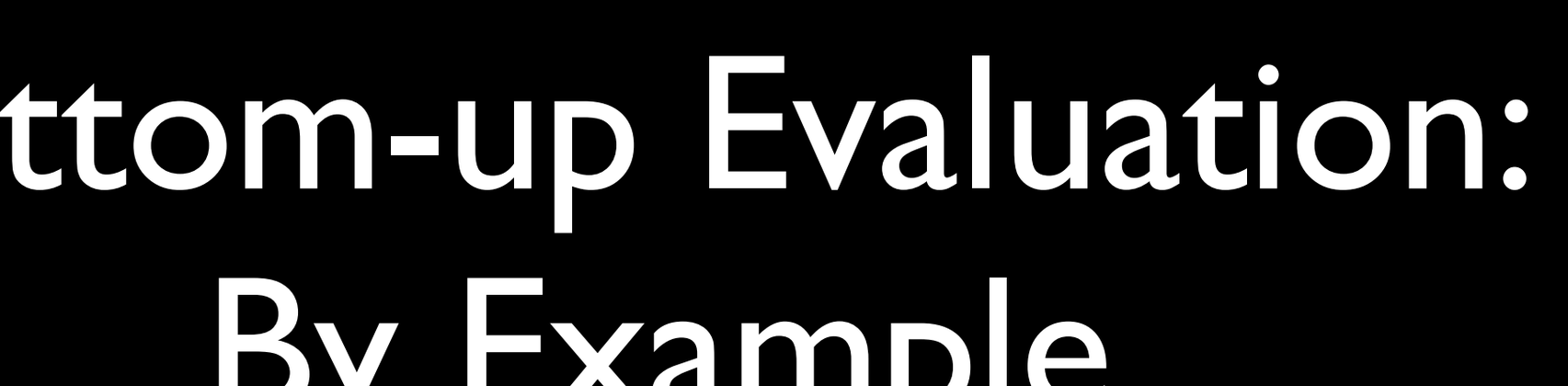


Bottom-up Evaluation: By Example

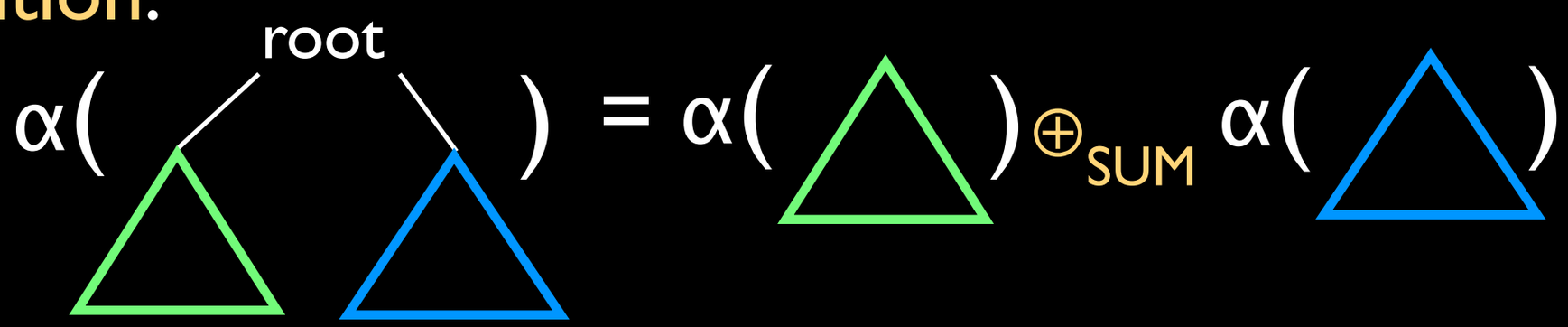
- Divide-and-conquer strategy on composed probability spaces of documents:

α=MIN

Convex Sum:



Convolution:

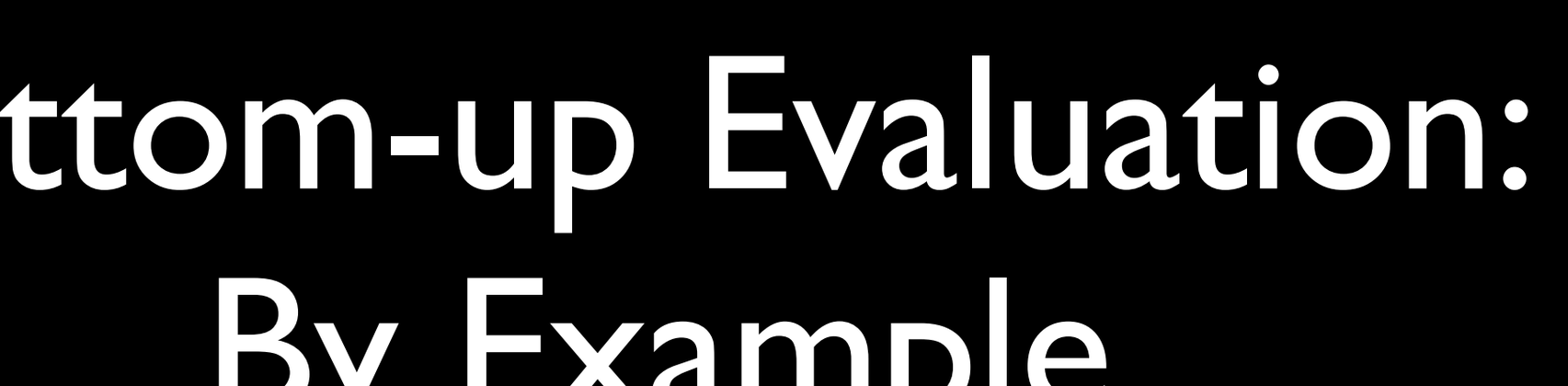


Bottom-up Evaluation: By Example

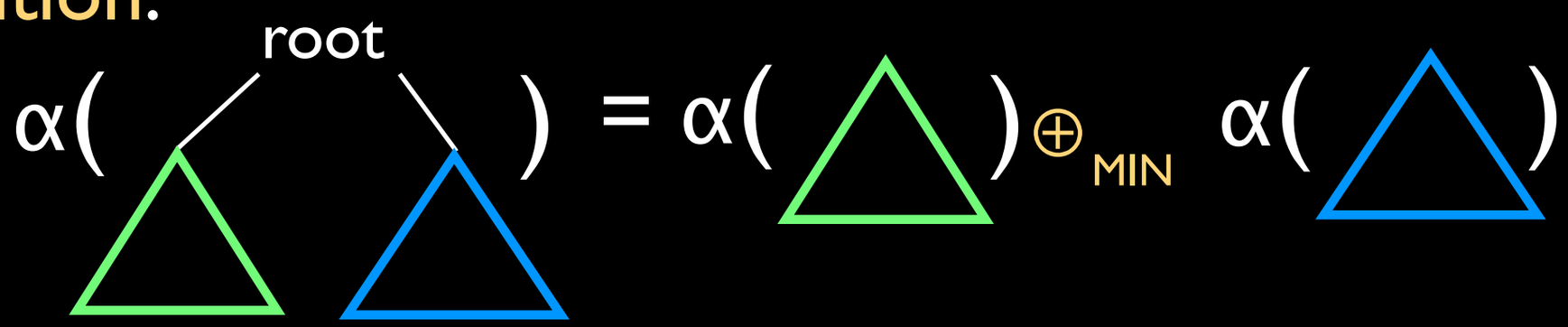
- Divide-and-conquer strategy on composed probability spaces of documents:

α=MIN

Convex Sum:



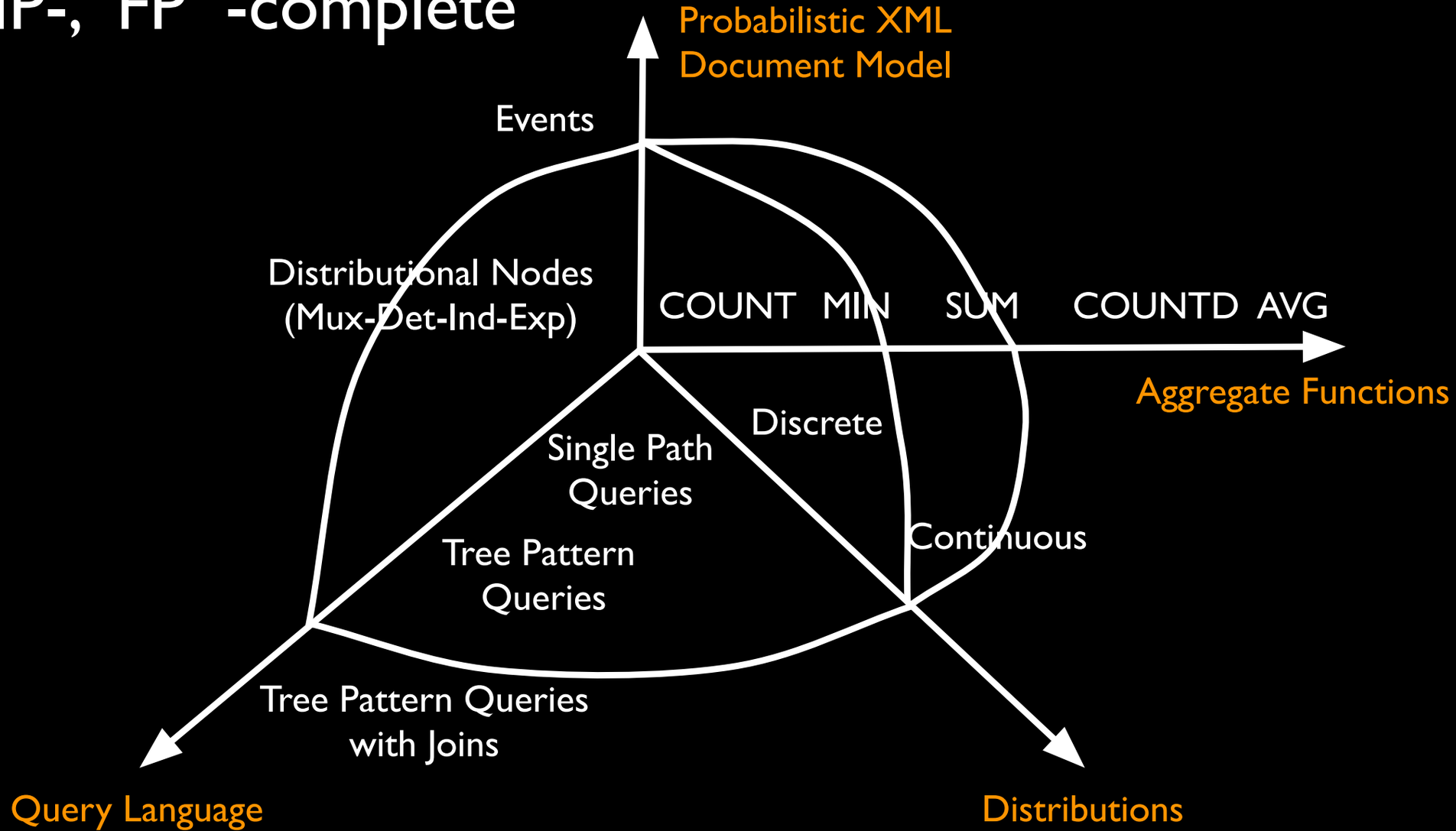
Convolution:



The Problem Space

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

Inside: PTIME



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(AGG(D) = C) - X | > Epsilon
  holds with probability at most Delta

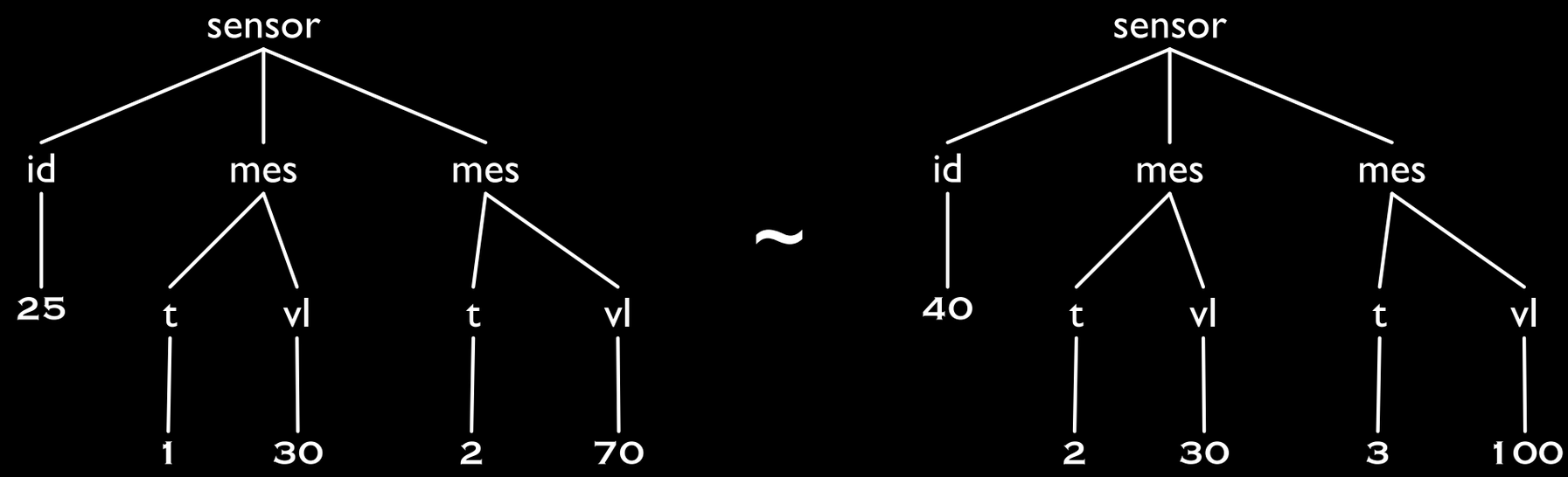
4. Continuous Probabilistic Models

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 probabilities

Probability Measure on Infinite Sets of Trees

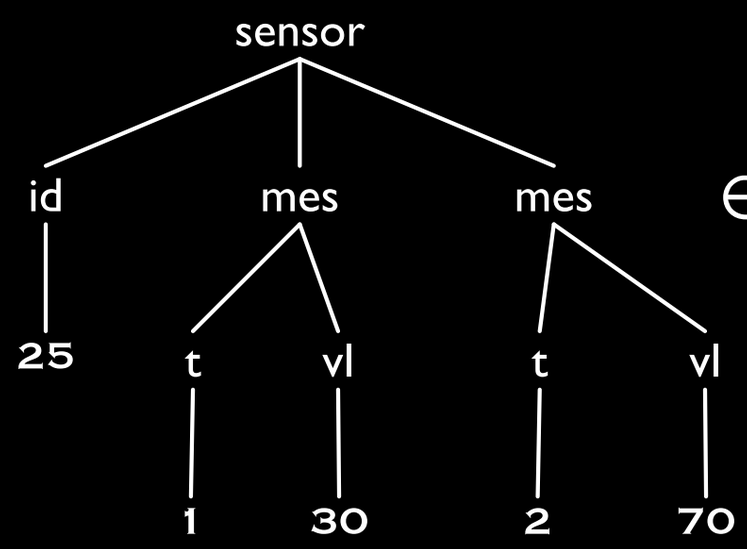
- Defined for trees with leave-labels in Reals
- Exploits the notion of structure equivalence: d~d' if d the same as d' up to renaming of leaves
- [d] is a set of all trees equivalent to d



Probability Measure on Infinite Sets of Trees

- Probability on one equivalence class [d] is defined using some measure M on Borel sets:

If $S \subseteq [d]$ then $Pr(S) := M(leaves(S))$



$\in S \Rightarrow (25, 1, 30, 2, 70) \in leaves(S)$

Probability Measure on Infinite Sets of Trees

- We extend Pr to sets of XML documents that are finite unions of equivalence classes:

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

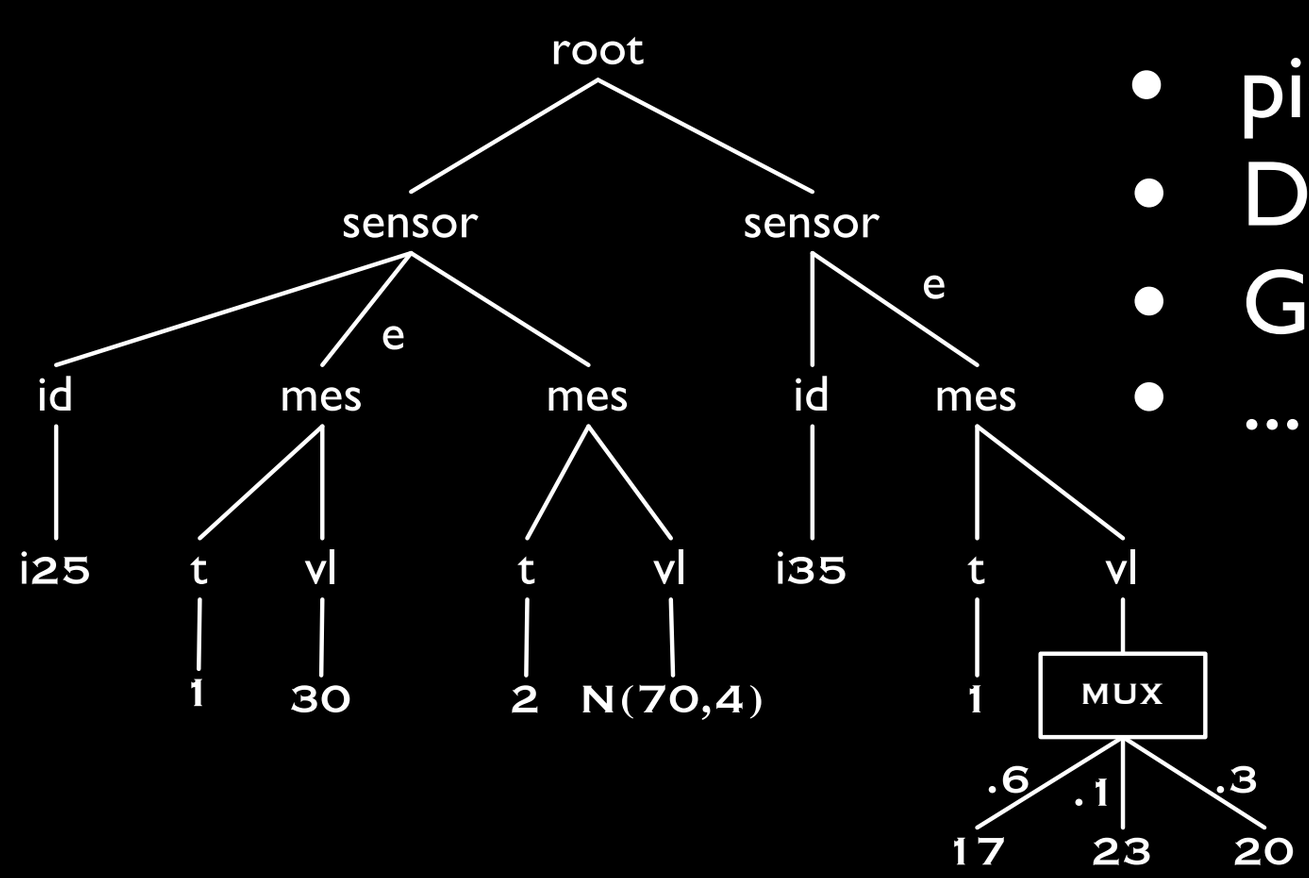
- Let $p_1,\ldots, p_n$ be weights of $[d_i]$, that is $p_1+\ldots+ p_n = 1$ Then

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

$$Pr(S) := p_1 \cdot Pr(S_1)+\ldots+ p_n \cdot Pr(S_n)$$

Continuous PXML Documents

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



- picewise 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 (for SUM, MIN/MAX)
  - Gaussian distributions (for SUM)

Aggregating CPXML

For picewise polynomials:

- Distribution of SP with SUM
  in PTIME in both input and output
- Distribution of SP with MAX and MIN
  in PTIME in data
- All moments of SP with SUM, MIN, MAX are
  in PTIME

5. Further Challenges

PXML vs. Pr RDBs

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

PXML vs. Pr RDBs

Relational Case: BID model [Re&Suciu:2006]	XML Case: distributional nodes
Some SPJR queries: PTIME	TP queries: PTIME
Other queries: #P-hard	non-trivial TPJ queries: #P-hard
Separation: complex conditions	Separation: join 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: not tractable
- Local dependencies: not practical
- Somewhere in between?
  - Why arbitrary dependencies are hard?
  - How to generalize local dependencies, while remaining tractable?

Tractable Extensions

- Arbitrary dependencies: not tractable
- Local dependencies: 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
  - COUT, 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