

# Updating Probabilistic XML

Evgeny Kharlamov,<sup>1,3</sup> Werner Nutt,<sup>1</sup> Pierre Senellart<sup>2</sup>

<sup>1</sup> Free University of Bozen-Bolzano

<sup>2</sup> Télécom ParisTech

<sup>3</sup> INRIA Saclay – Île-de-France

Updates in XML, Lausanne, March 2010

# Outline

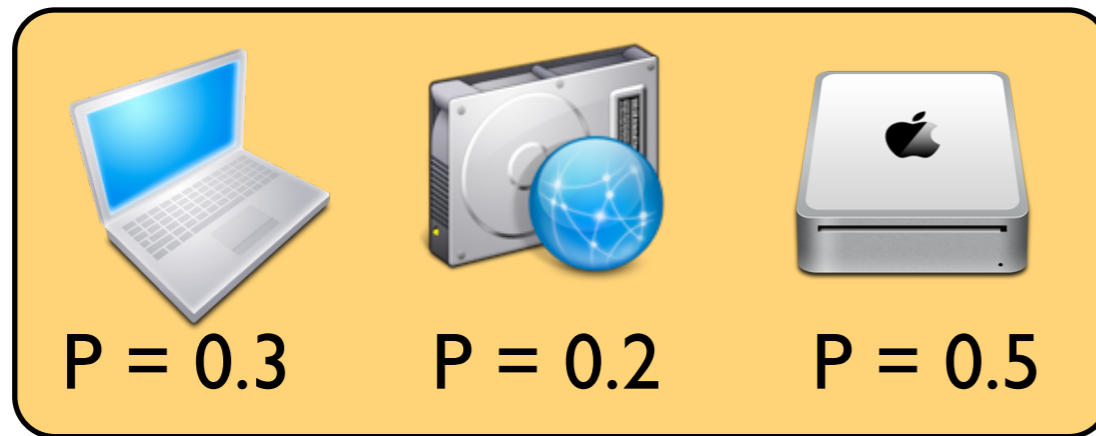
1. Probabilistic data
2. Problem of updates
3. Updating discrete PXML
4. Updating continuous PXML

# Applications of Probabilistic Data

- **Approximate query processing**: ranking, linkage
- **Information extraction**: approximate search for entities (e.g. names) in text
- **Sensor data**: imprecise or missing readings
- ...

# Probabilistic Database

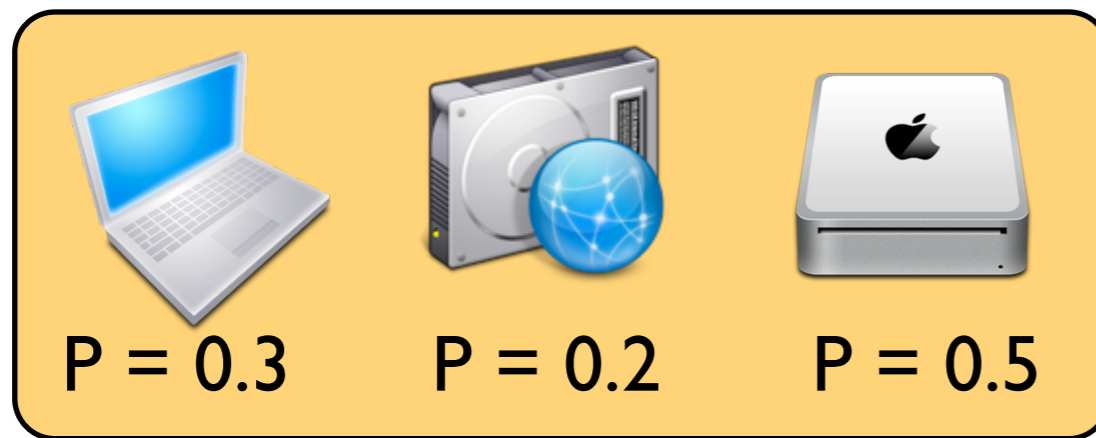
Probabilistic DB:



$P = 0.3$        $P = 0.2$        $P = 0.5$

# Probabilistic Database

Probabilistic DB:



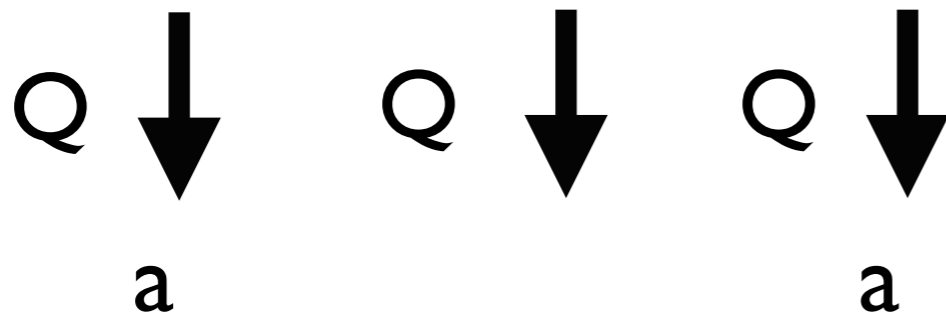
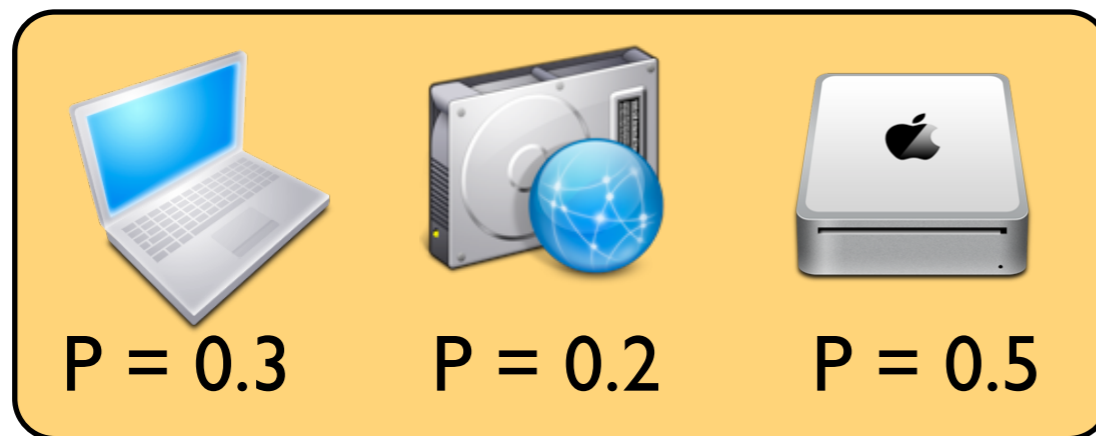
Q ↓  
a

Q ↓

Q ↓  
a

# Probabilistic Database

Probabilistic DB:

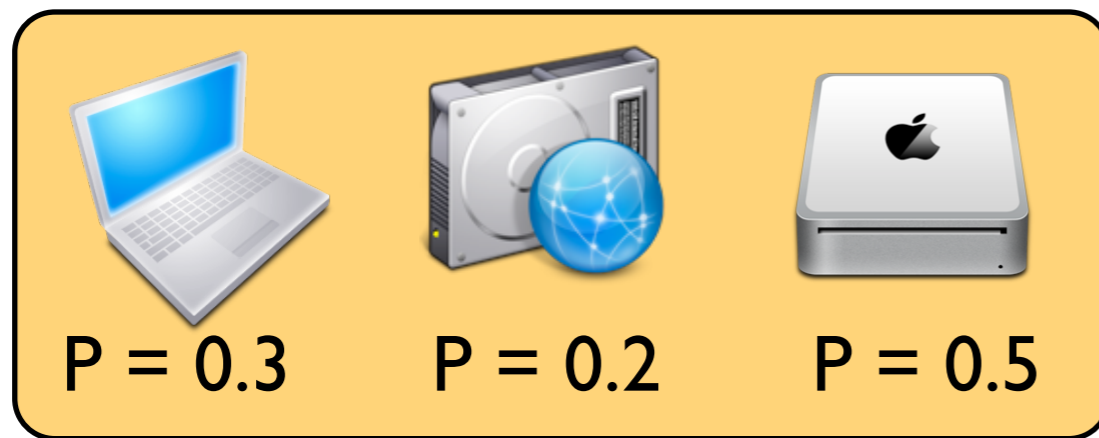


---

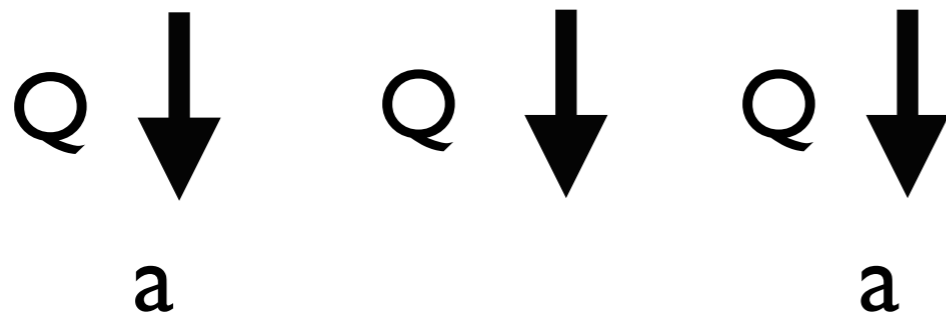
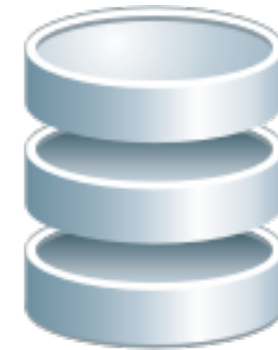
Answer: (a, 0.8)

# Probabilistic Database

Probabilistic DB:



Representation  
of Prob DB:

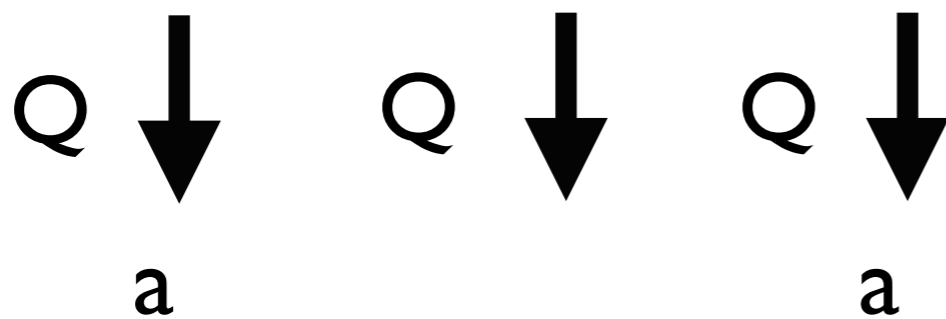
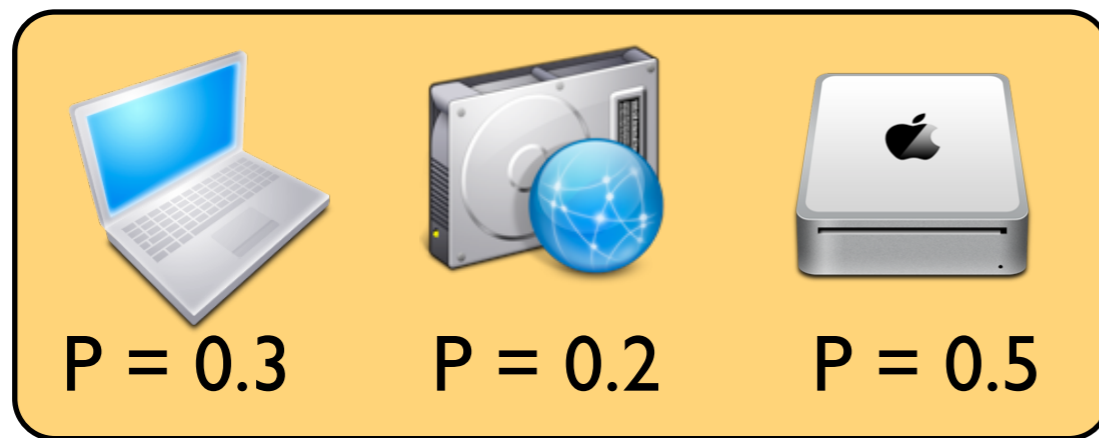


---

Answer: (a, 0.8)

# Probabilistic Database

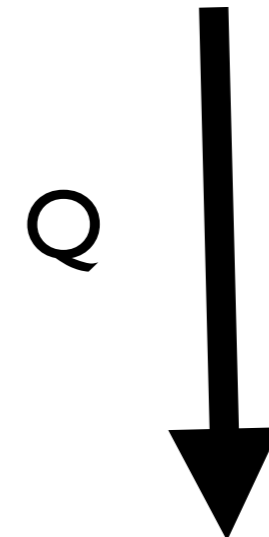
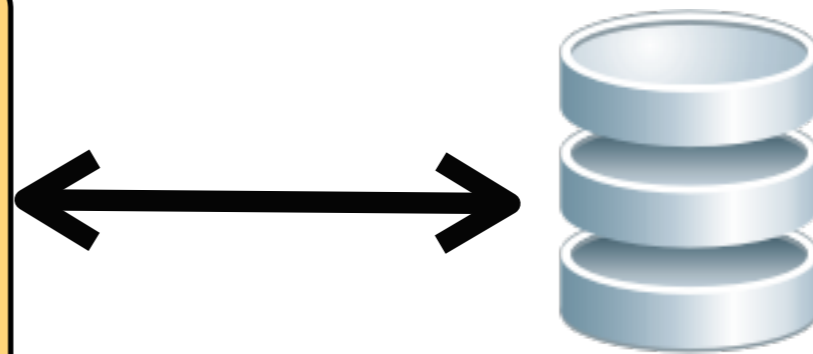
Probabilistic DB:



---

Answer: (a, 0.8)

Representation  
of Prob DB:

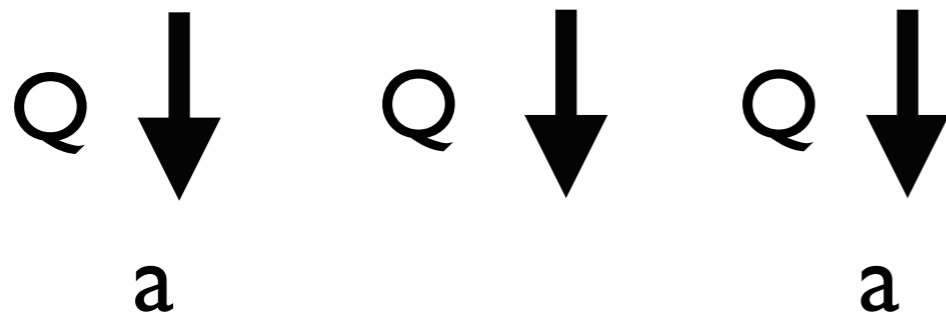
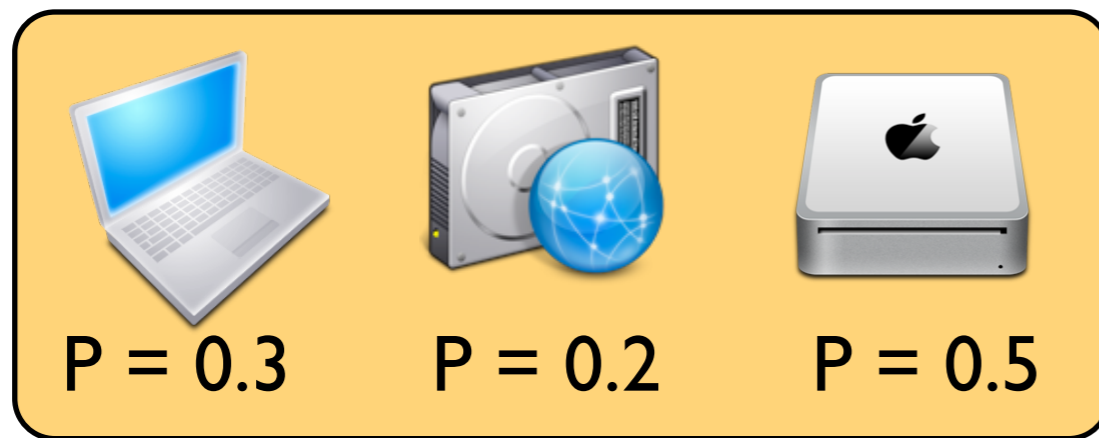


(a, 0.8)



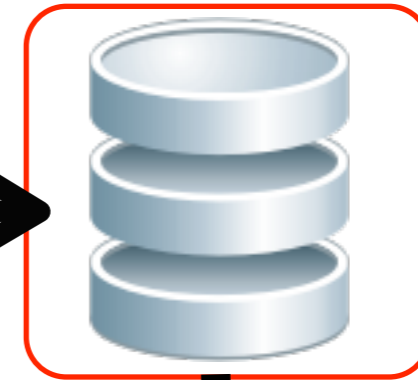
# Probabilistic Database

Probabilistic DB:



Answer: (a, 0.8)

Representation  
of Prob DB:



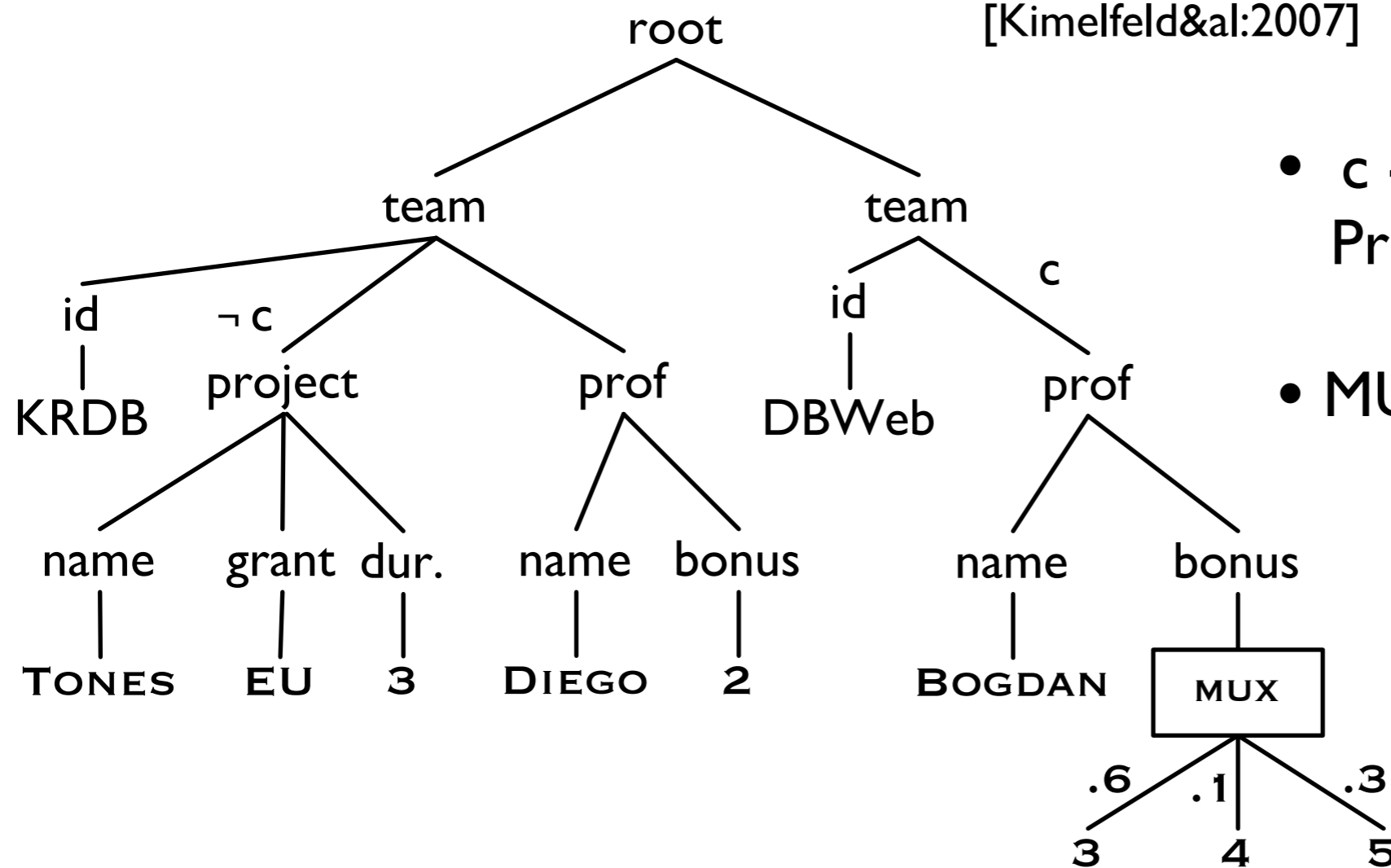
Q

(a, 0.8)

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

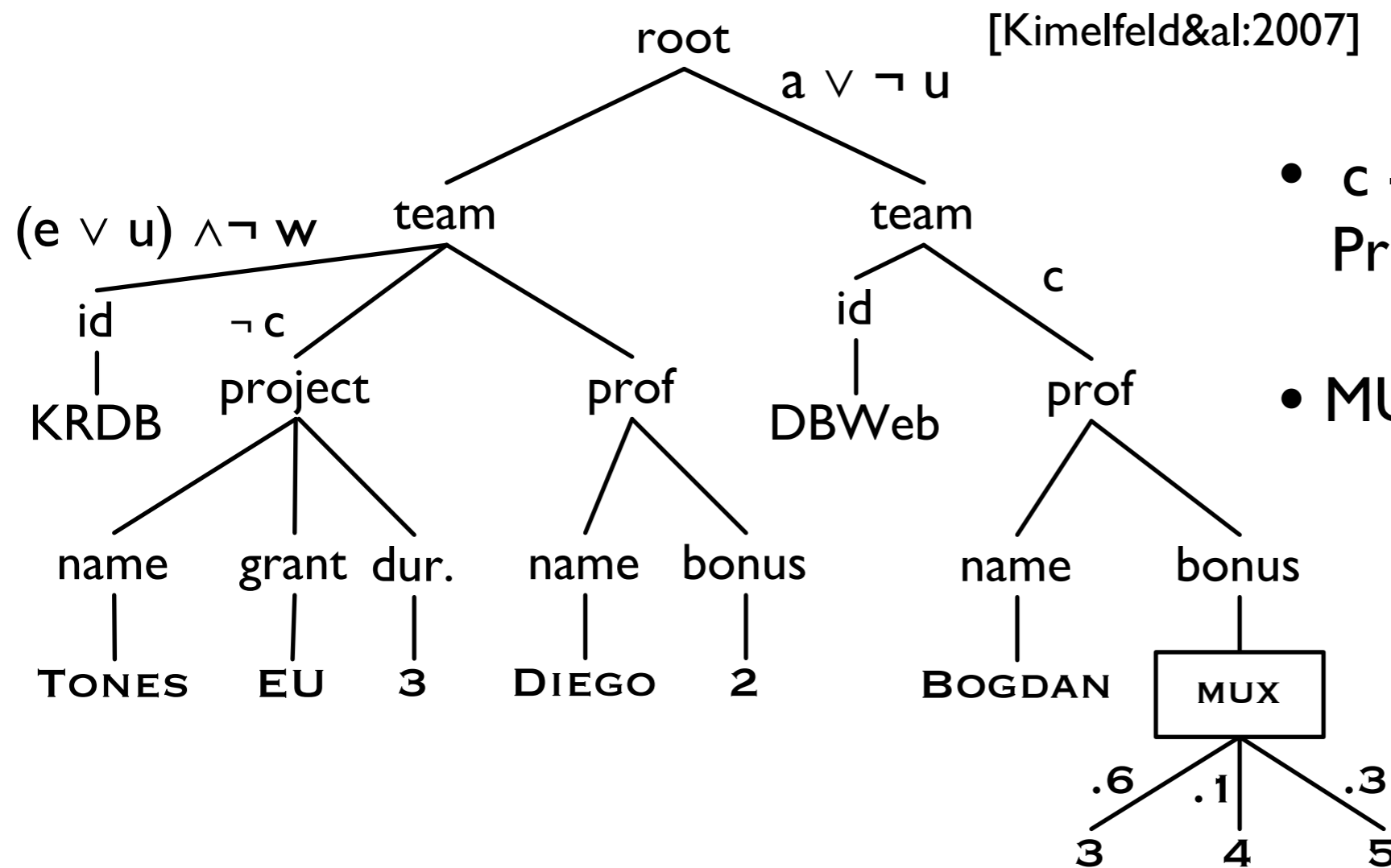
[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

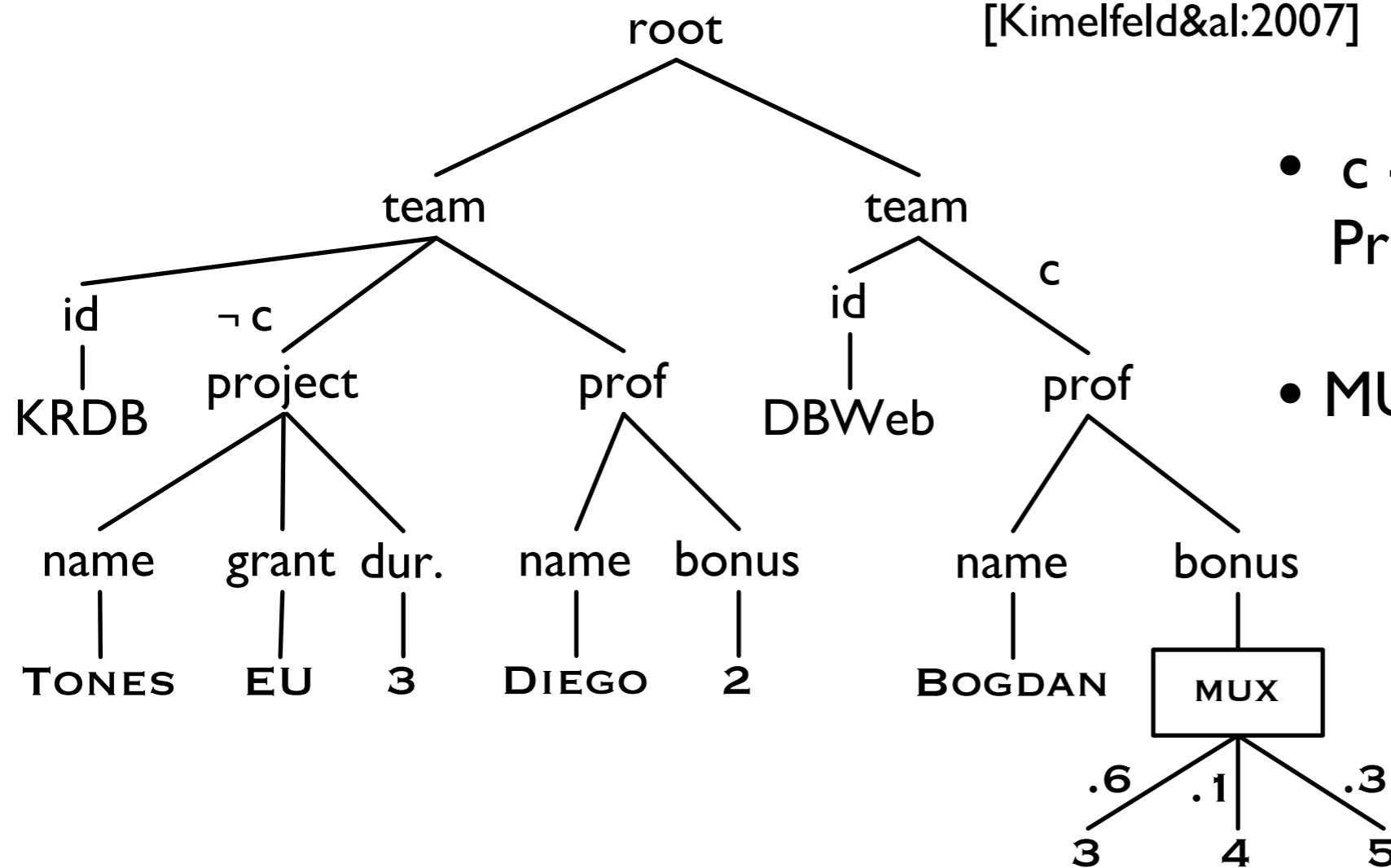


- c - event: “current”  
Pr(c) = .4
- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



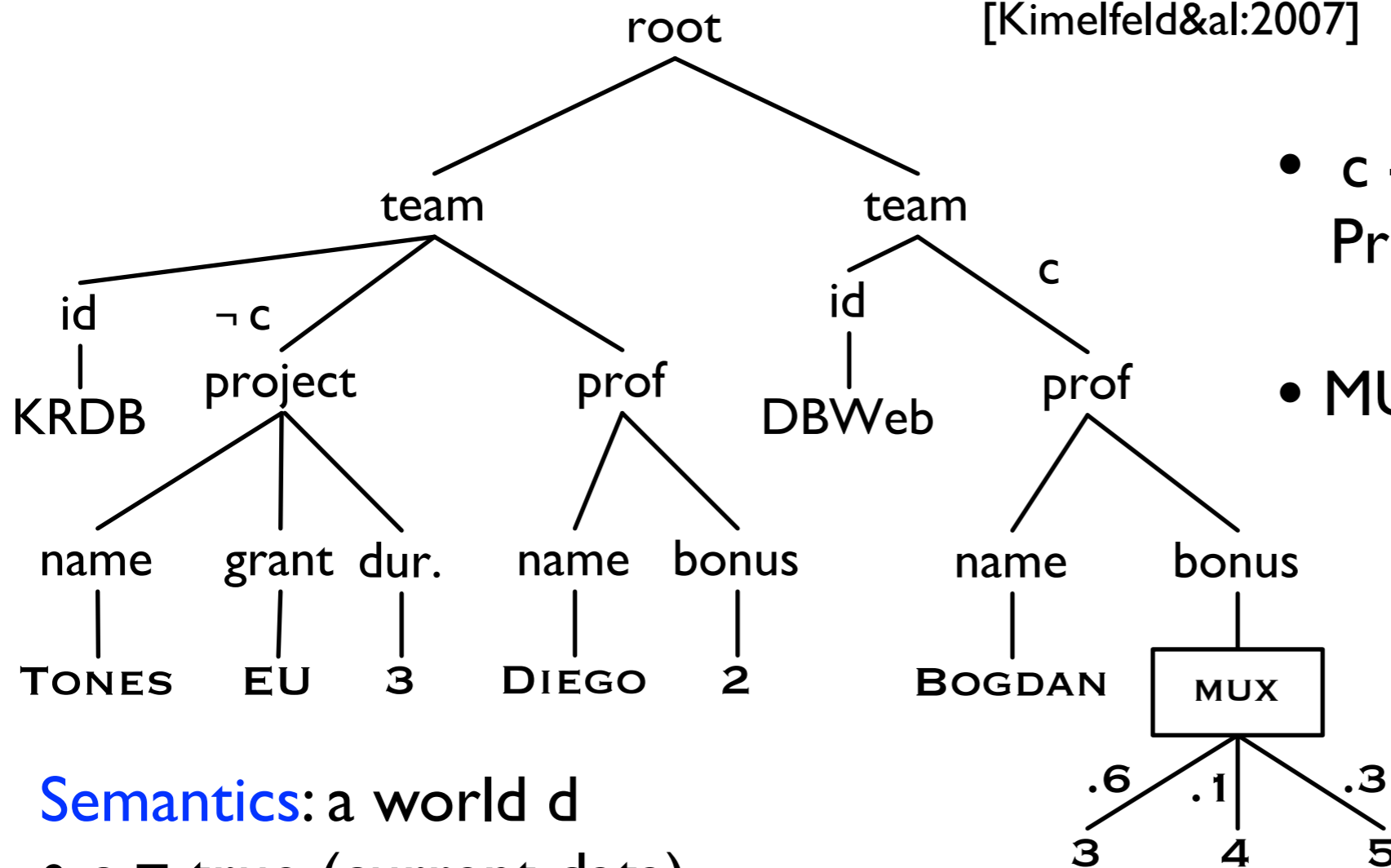
- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
Pr(c) = .4

- MUX - mutually exclusive options

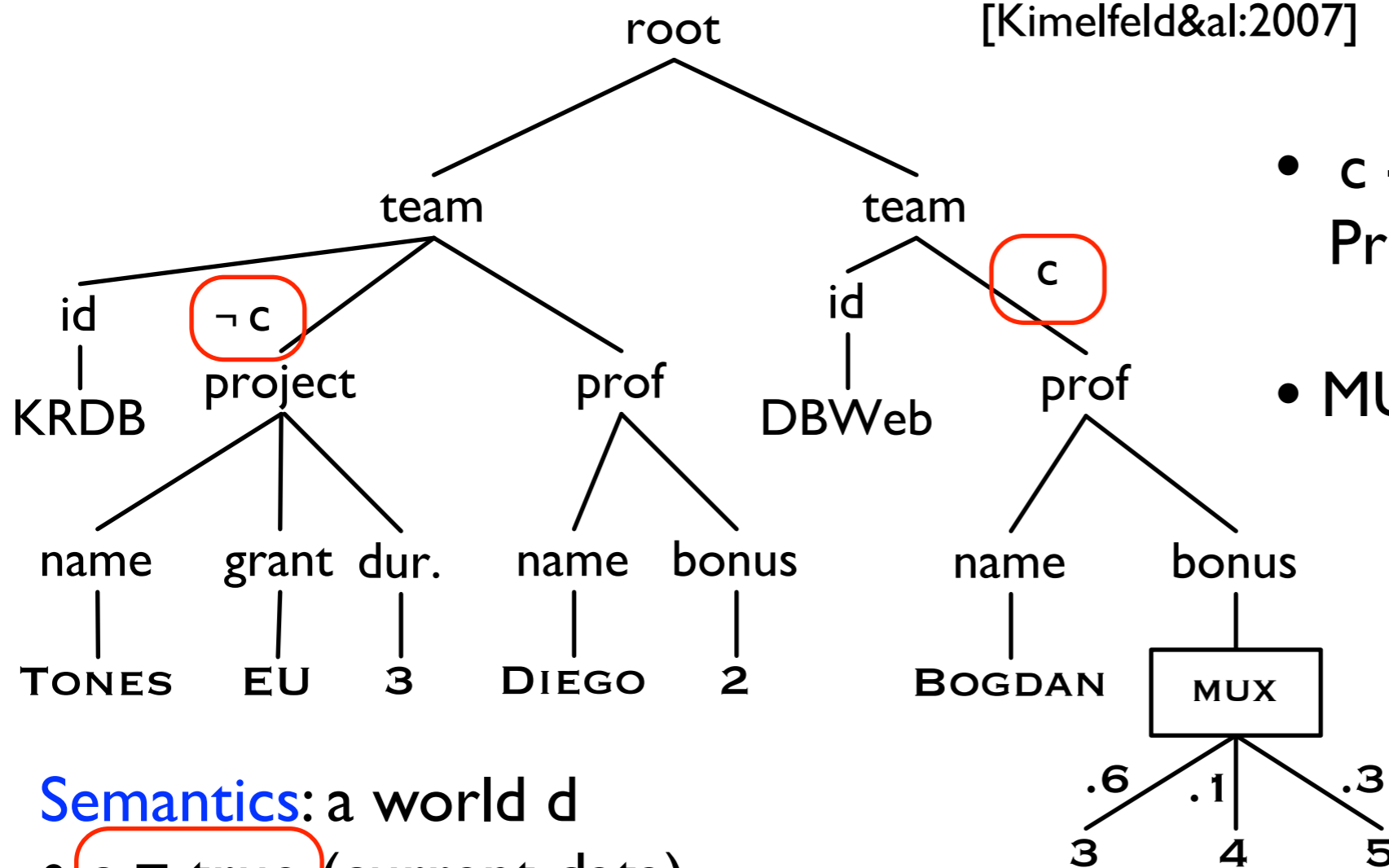
**Semantics:** a world d

- c = true (current data)
- MUX: 4
- Pr(d) = 0.4 x 0.1

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
Pr(c) = .4

- MUX - mutually exclusive options

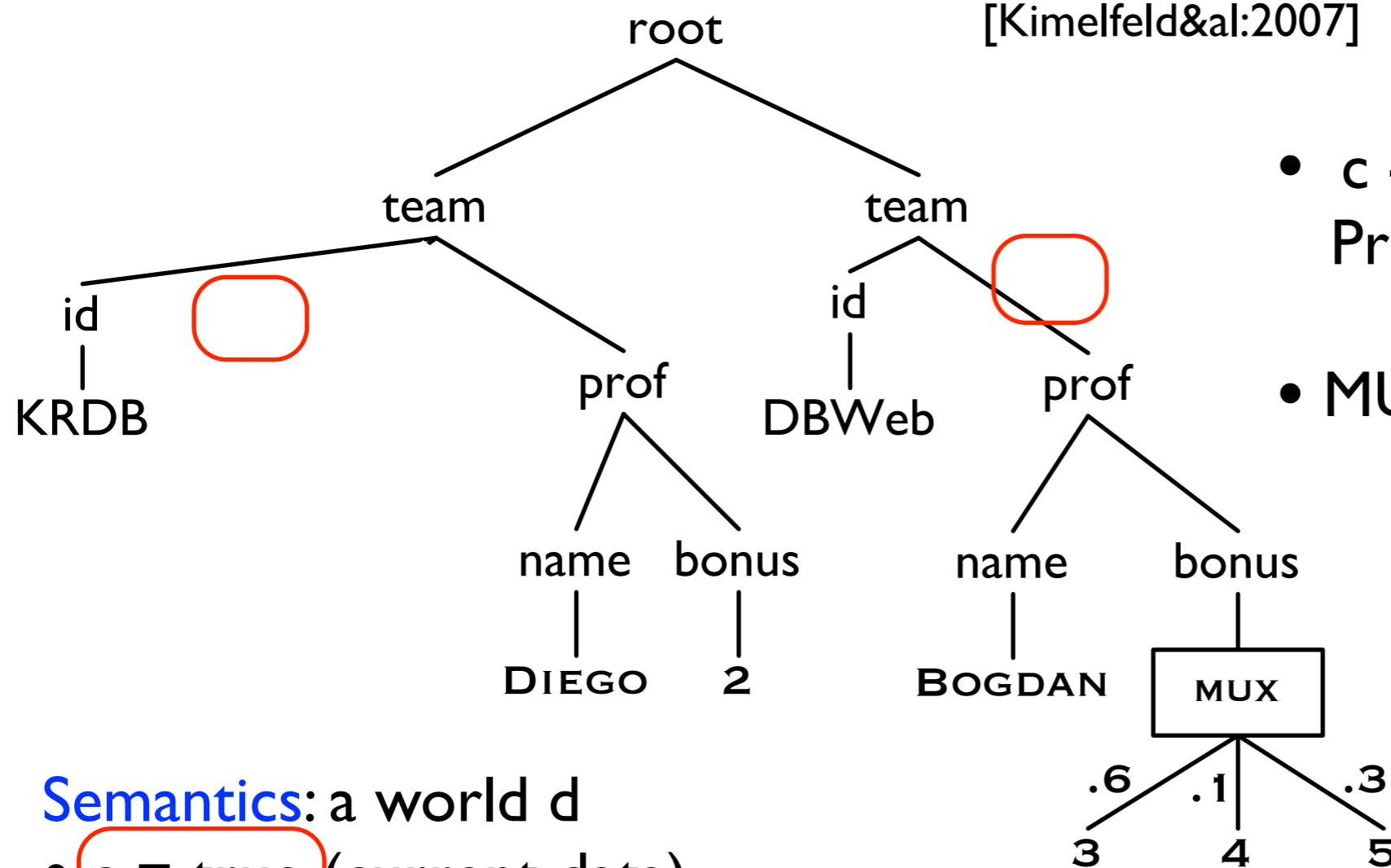
**Semantics:** a world  $d$

- $c = \text{true}$  (current data)
- MUX: 4
- Pr( $d$ ) = 0.4 x 0.1

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

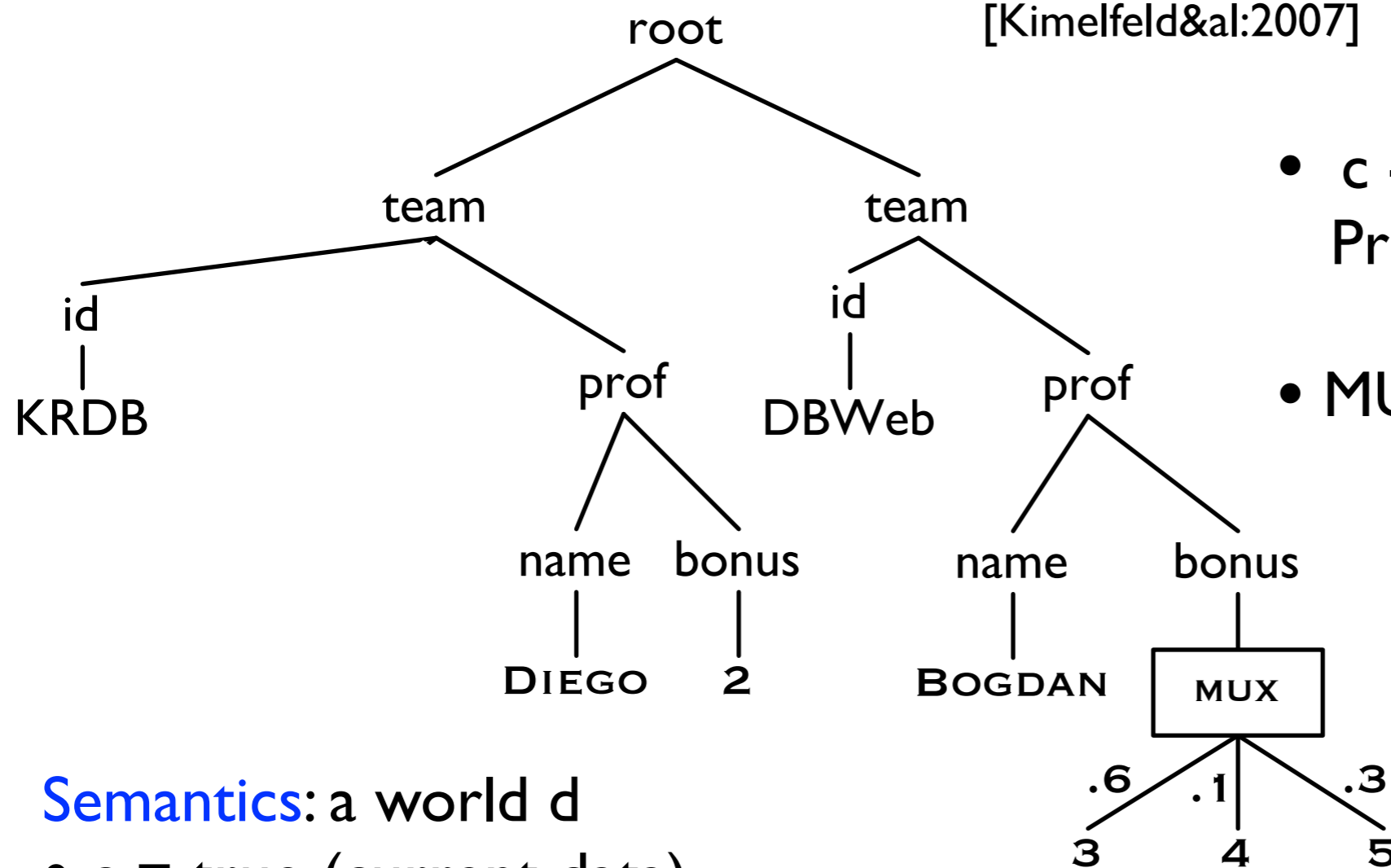
**Semantics:** a world  $d$

- $c = \text{true}$  (current data)
- MUX: 4
- $\Pr(d) = 0.4 \times 0.1$

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

**Semantics:** a world  $d$

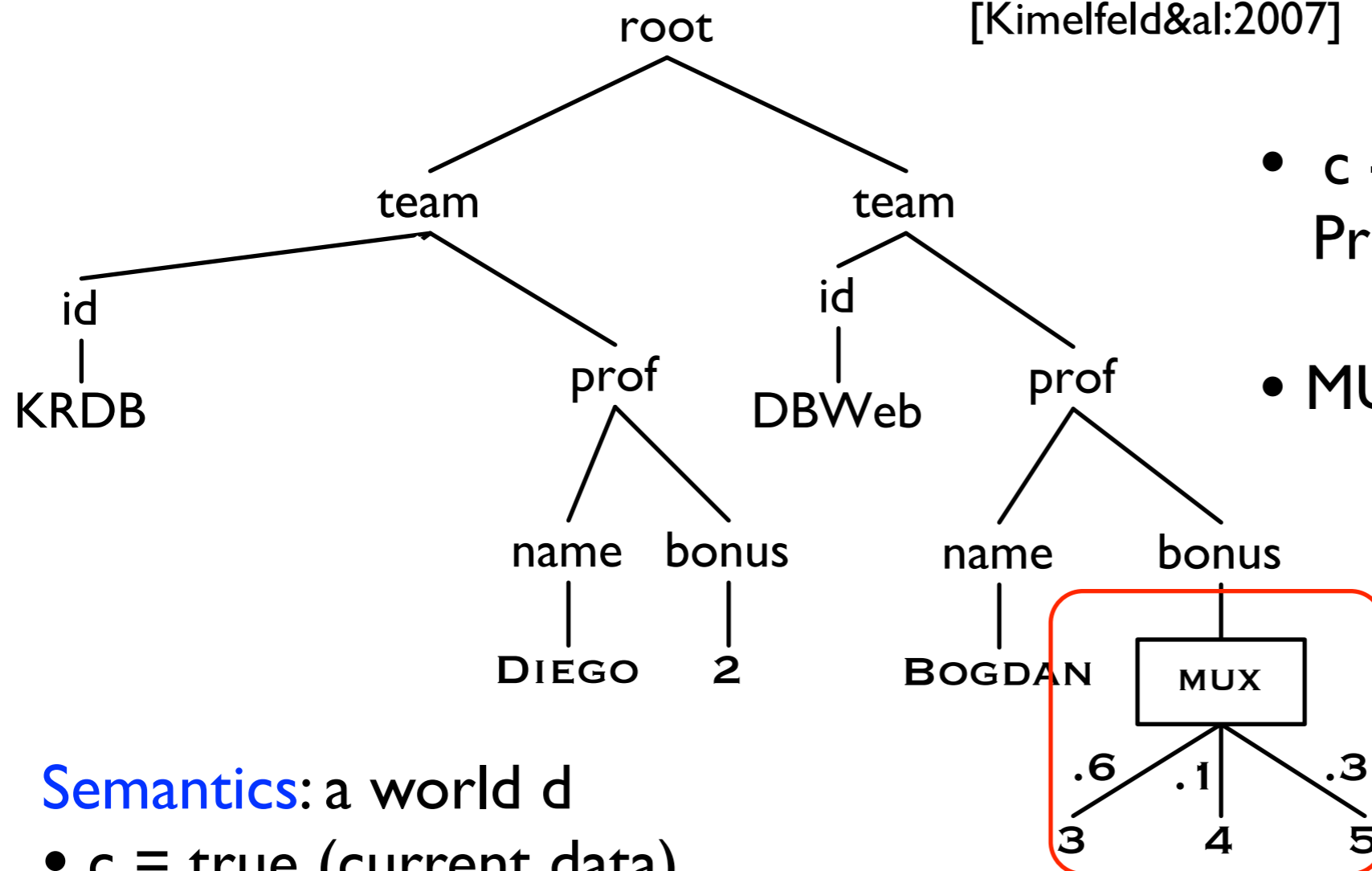
- $c = \text{true}$  (current data)
- MUX: 4
- $\Pr(d) = 0.4 \times 0.1$



# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

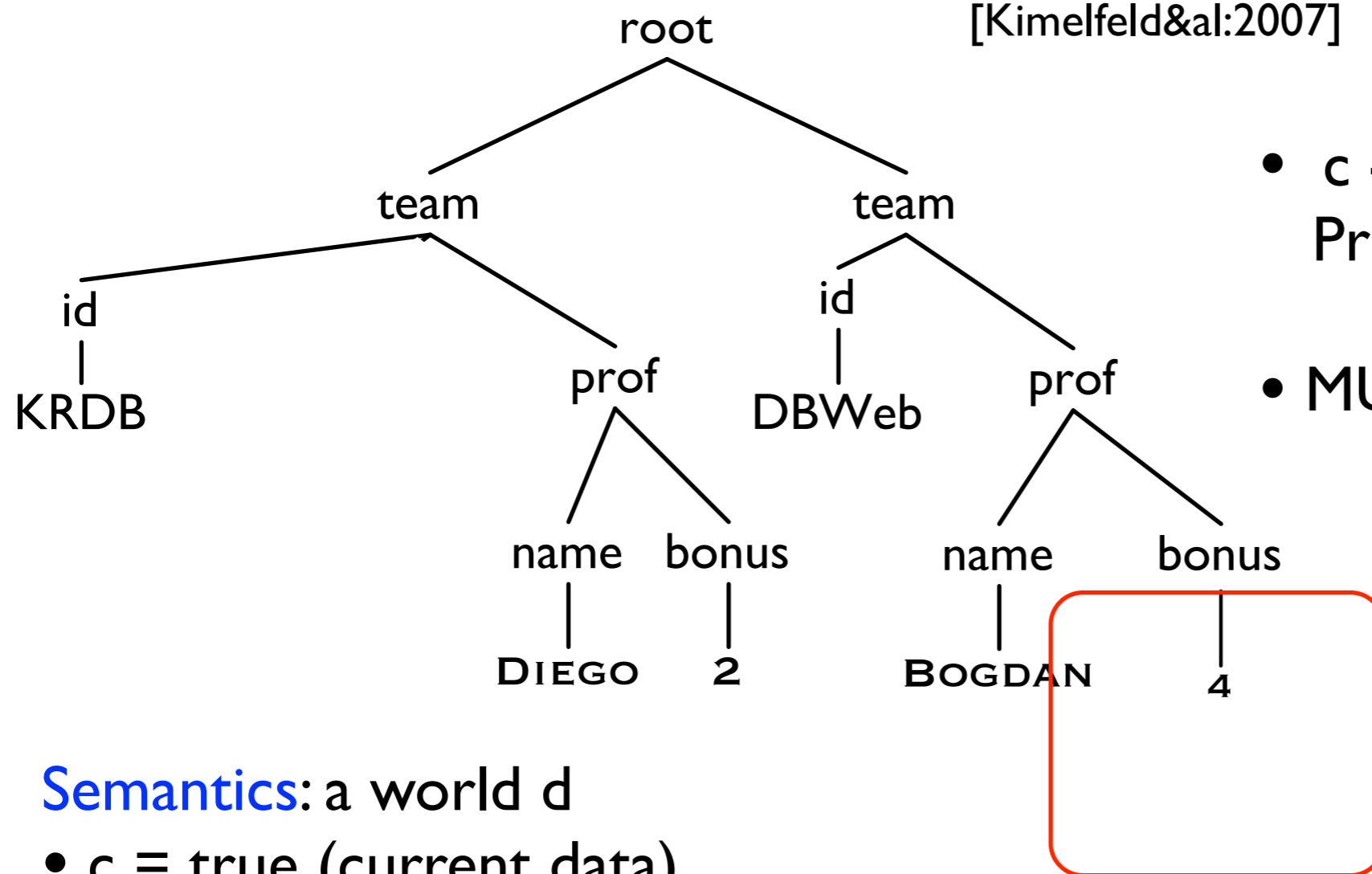
**Semantics:** a world  $d$

- $c = \text{true}$  (current data)
- **MUX: 4**
- $\Pr(d) = 0.4 \times 0.1$

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

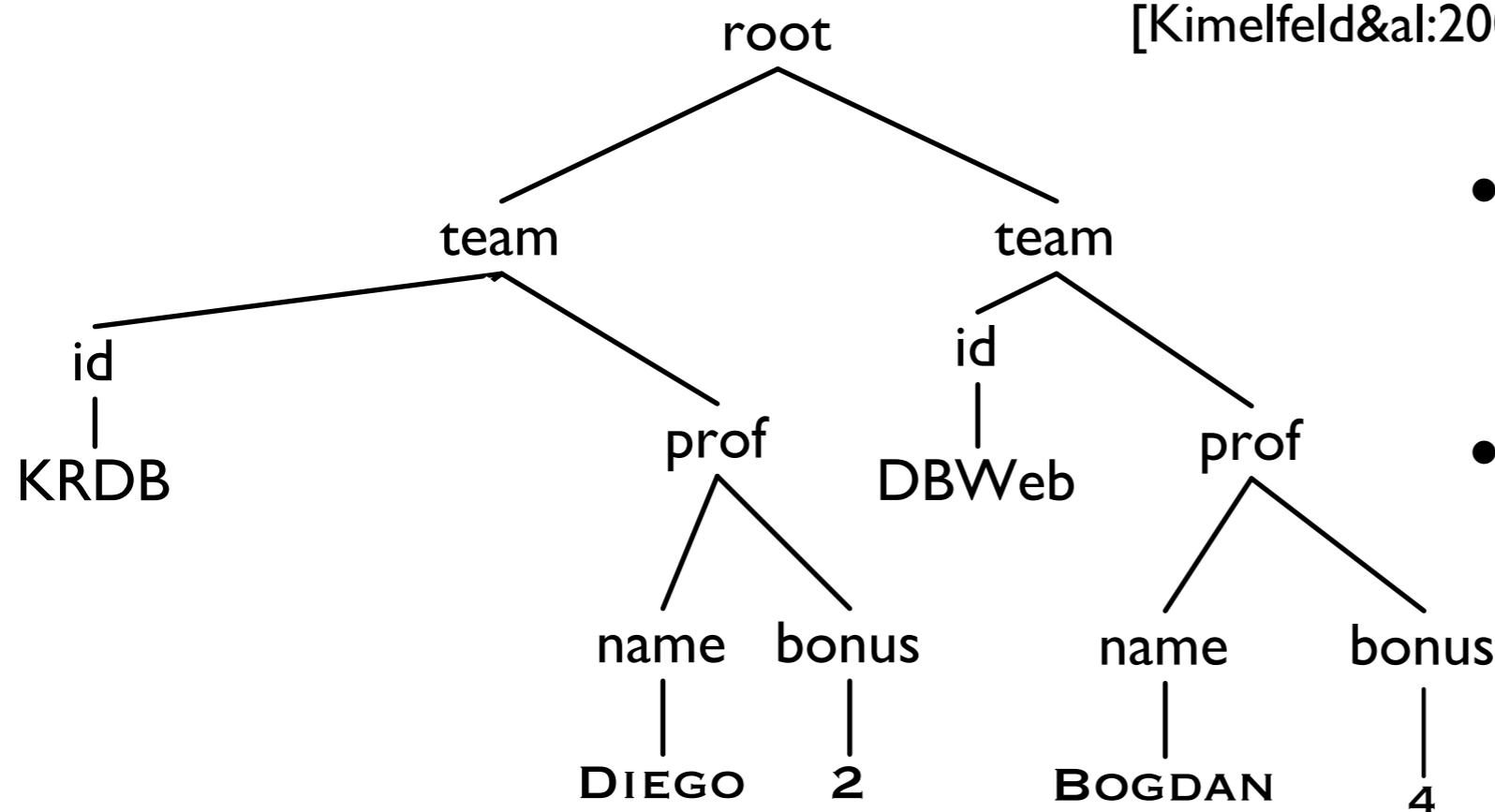
**Semantics:** a world  $d$

- $c = \text{true}$  (current data)
- **MUX: 4**
- $\Pr(d) = 0.4 \times 0.1$

# PXML with Events and Distributional Nodes

[Kimelfeld&al:2007]

[Senellart&al:2007]



- c - event: “current”  
 $\Pr(c) = .4$

- MUX - mutually exclusive options

**Semantics:** a world  $d$

- $c = \text{true}$  (current data)
- MUX: 4
- $\Pr(d) = 0.4 \times 0.1$

# Discrete Probabilistic XML Documents

- Probabilistic XML document  $D$ 
  - represents (exponentially) many documents  $d$
  - each with probability  $\Pr(d)$
- It is achieved by
  - **Events formulas** on edges: over Bool. random vars.  
Capture **long-distance** correlations
  - **Distributional** nodes: Mux, Det.  
Capture **local** (hierarchical) dependancies.

# Discrete Probabilistic XML Documents

- Probabilistic XML document  $D$ 
  - represents (exponentially) many documents  $d$
  - each with probability  $\Pr(d)$
- It is achieved by
  - **Events formulas** on edges: over Bool. random vars.  
Capture **long-di** Special case of event formulas
  - **Distributional** nodes: Mux, Det.  
Capture **local** (hierarchical) dependancies.

# Outline

1. Probabilistic data
2. Problem of updates
3. Updating discrete PXML
4. Updating continuous PXML

# 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

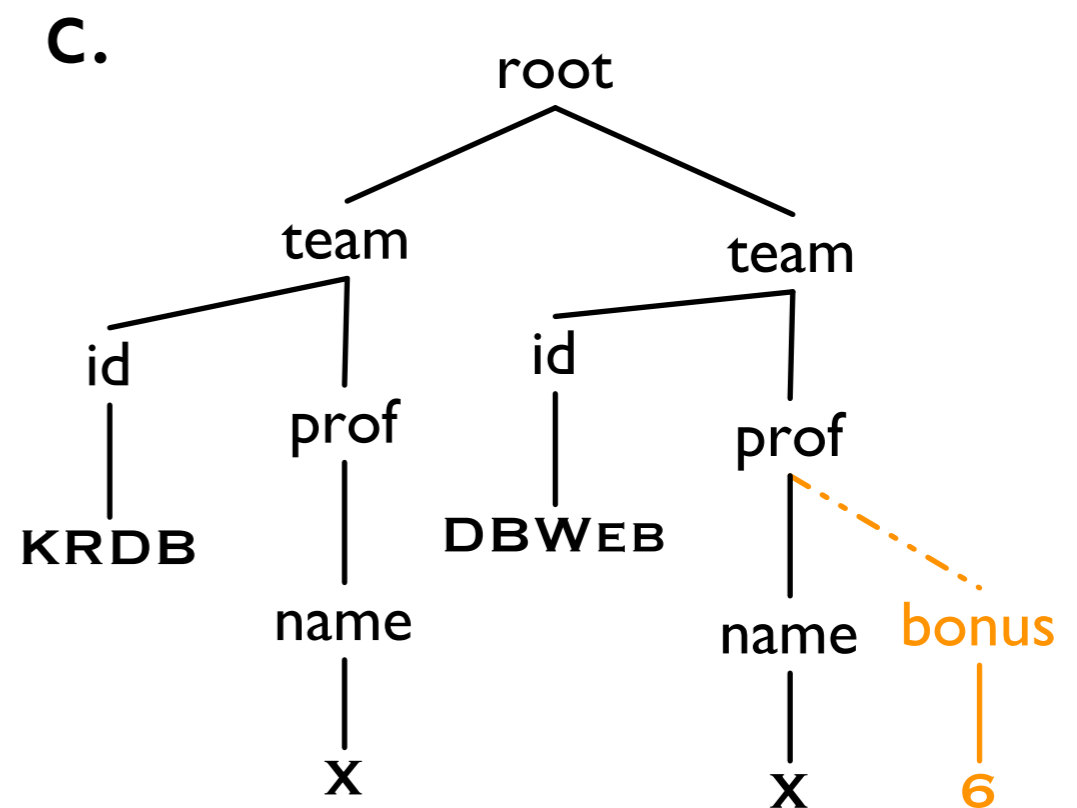
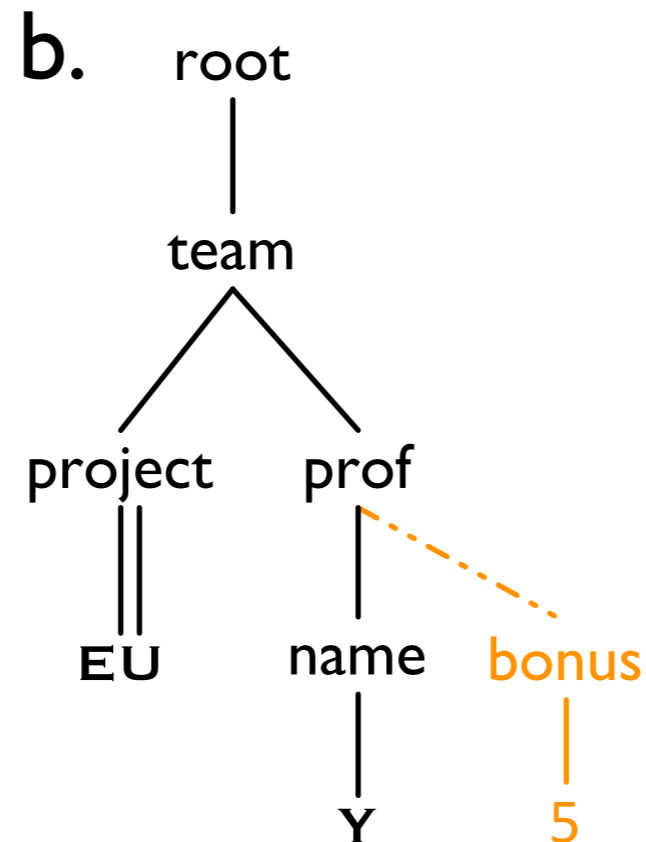
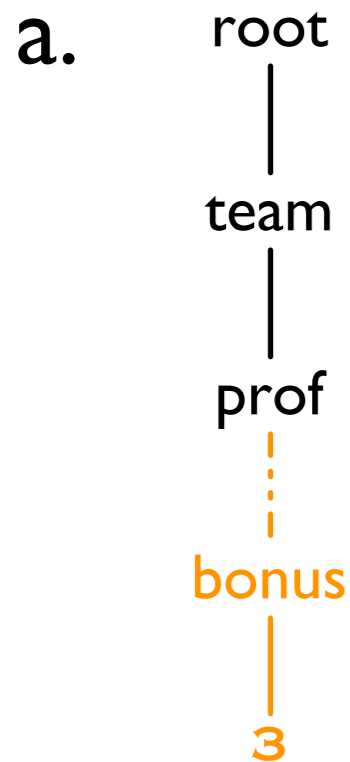
- *XUpdate*, based on XPath
- *XQuery Update Facility*, based on XQuery

# Types of Updates

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

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

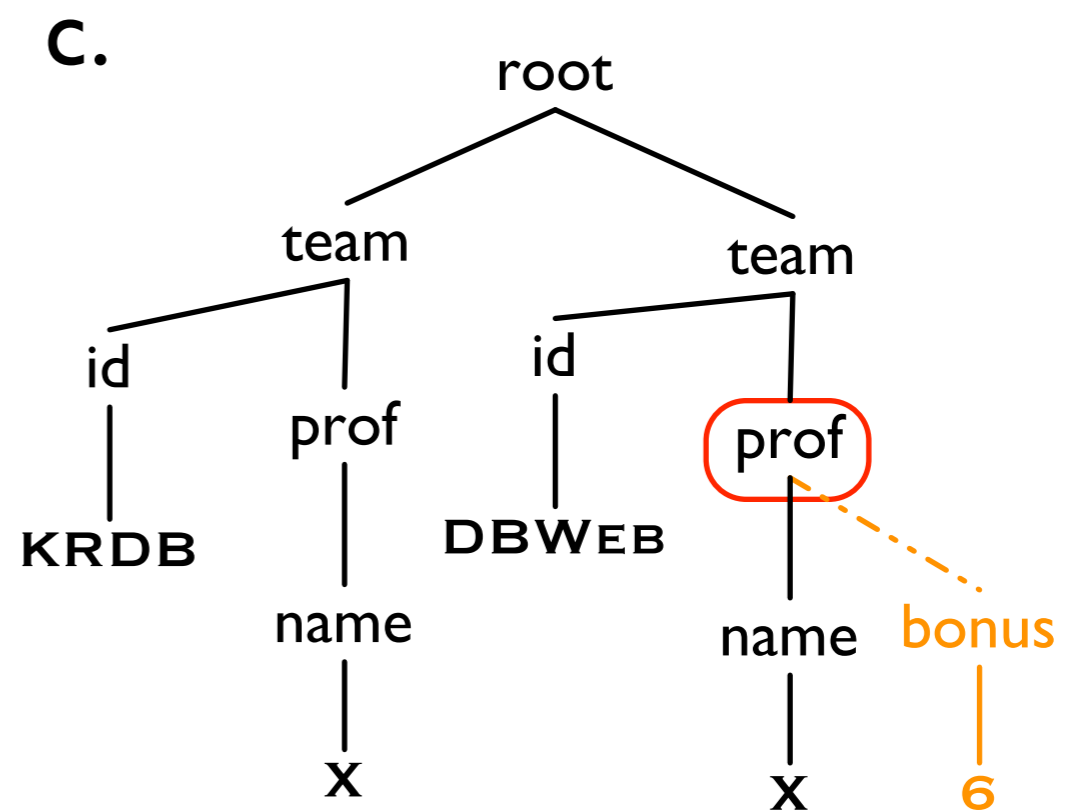
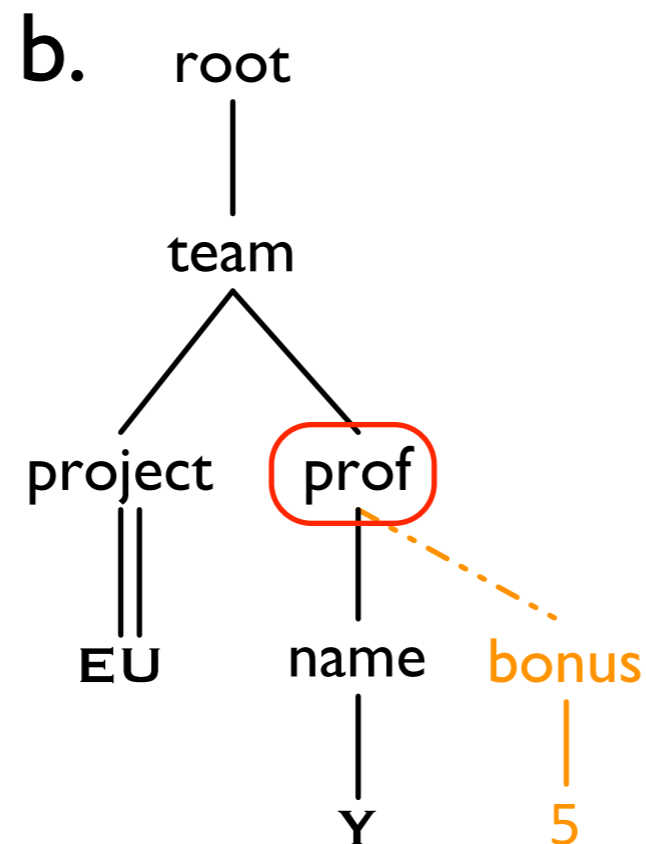
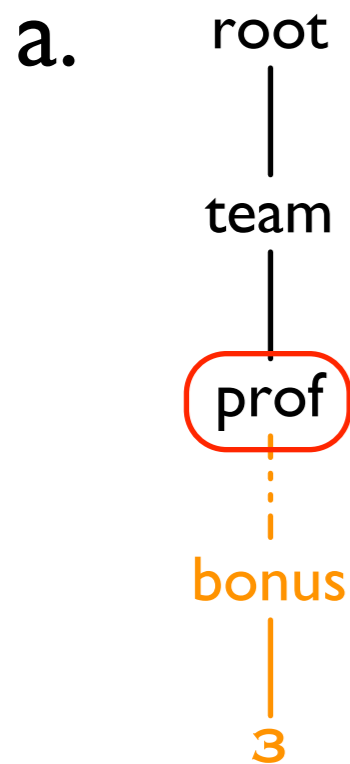


# Types of Updates

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

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ



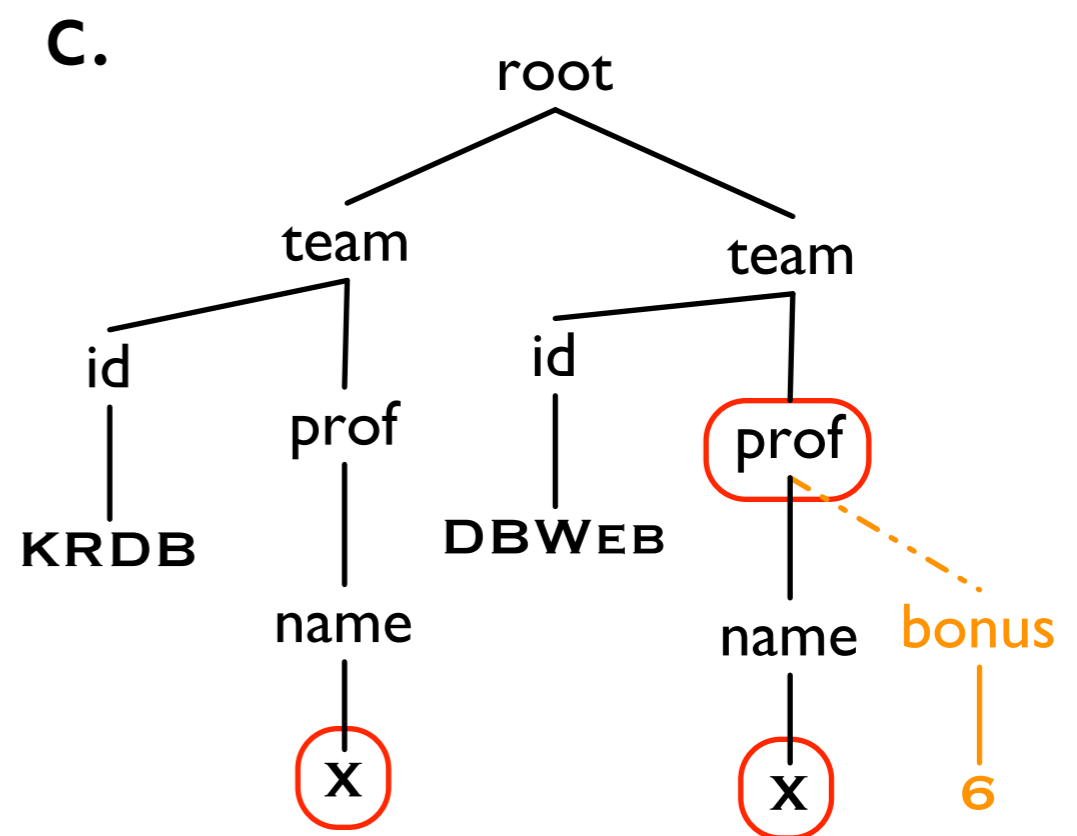
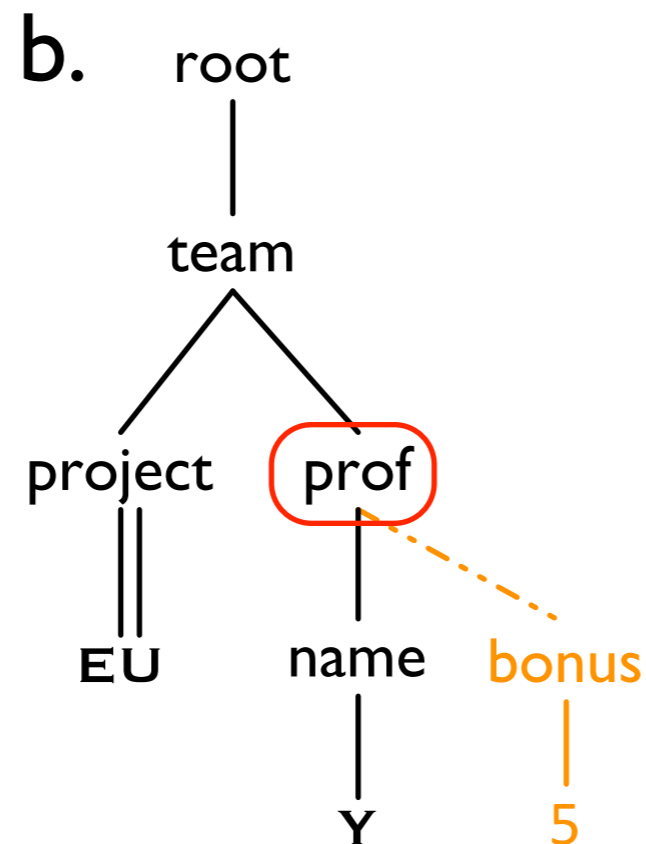
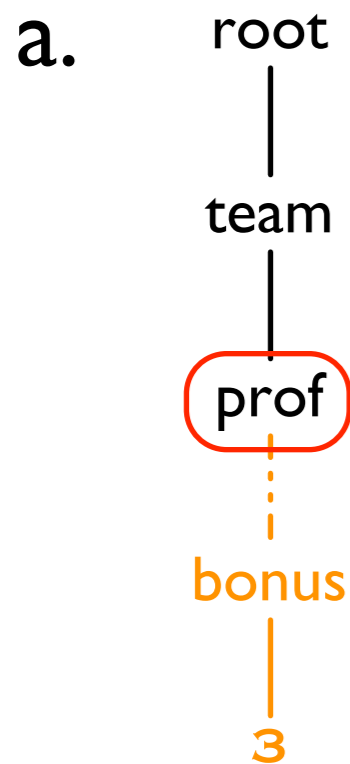


# Types of Updates

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

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

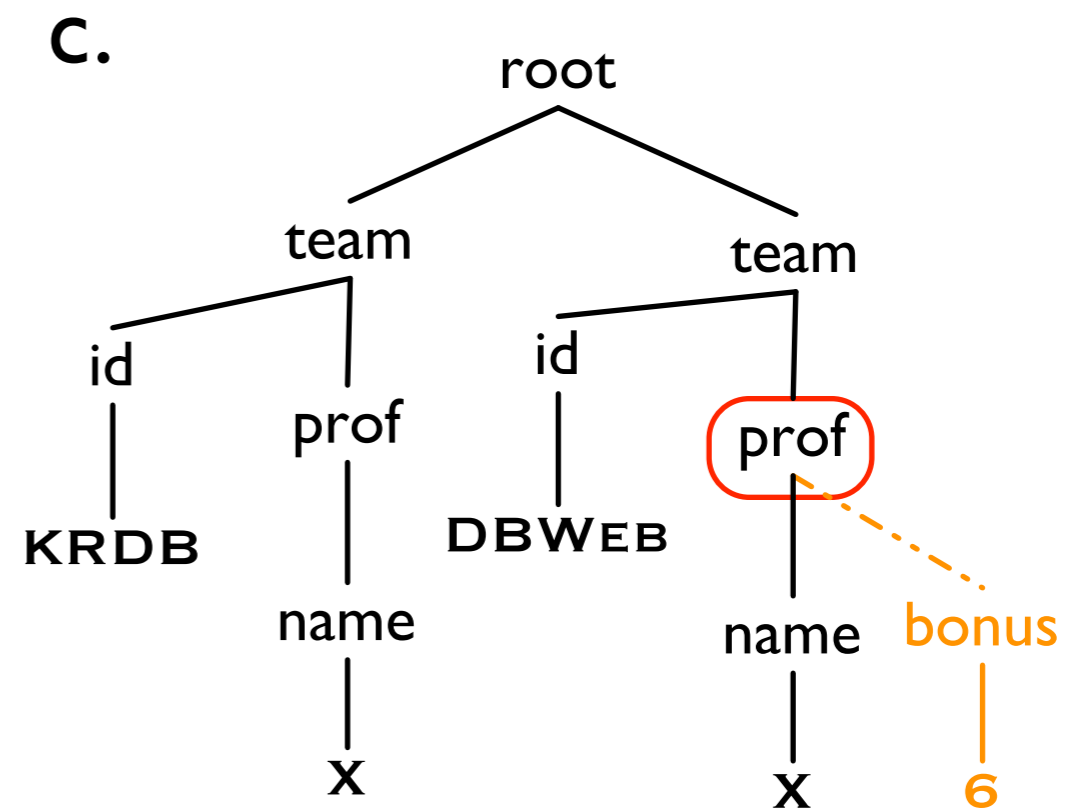
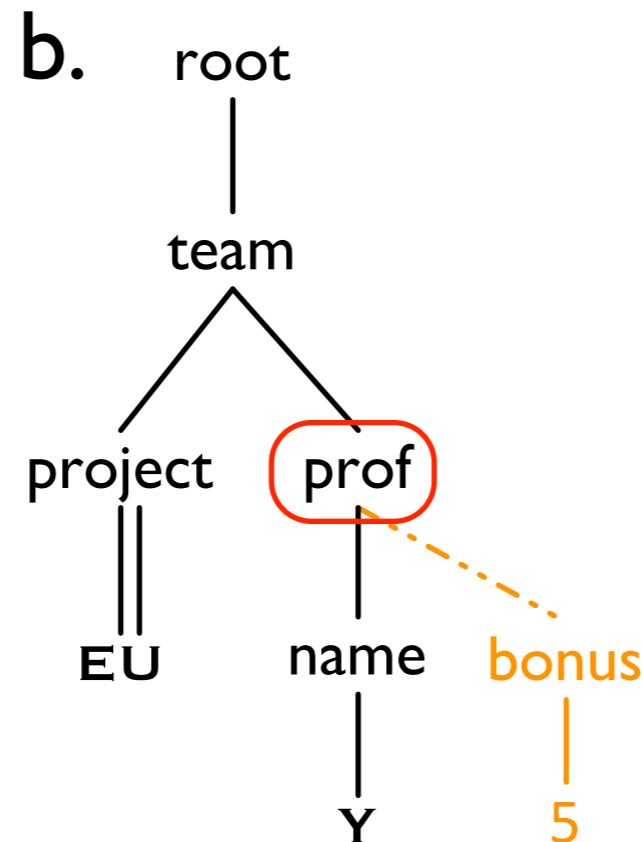
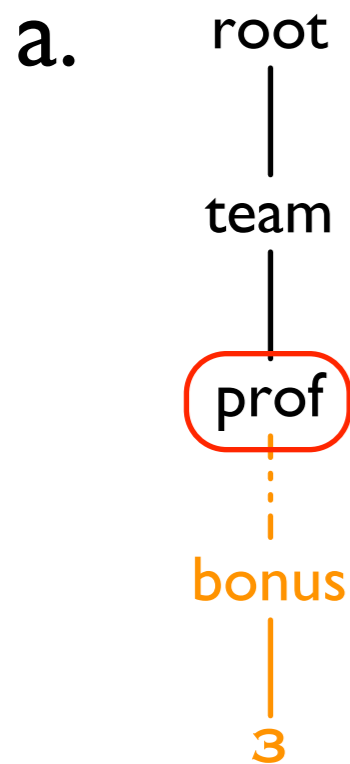


# Types of Updates

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

b. Tree-Pattern updates - TP

c. Tree-Pattern updates with Joins - TPJ

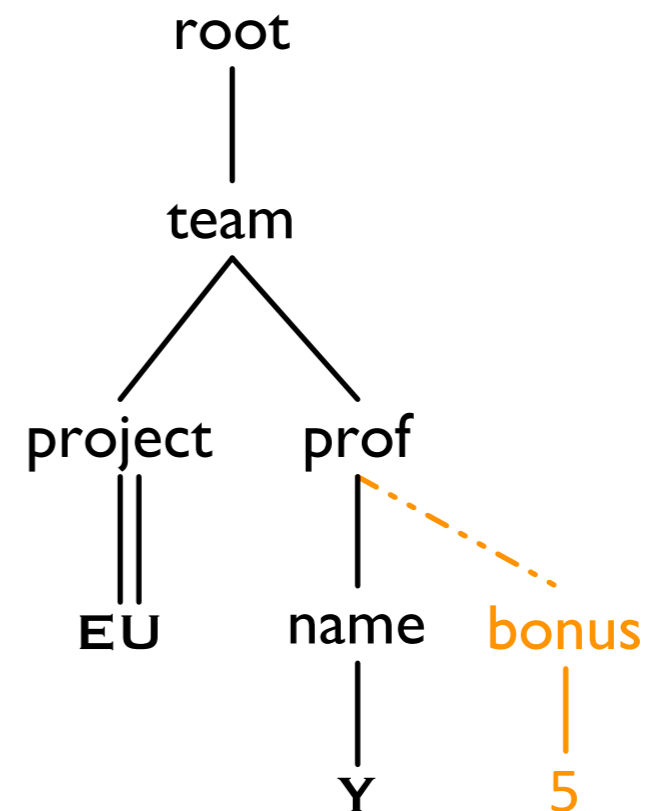


# Semantics of Insertions

- *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

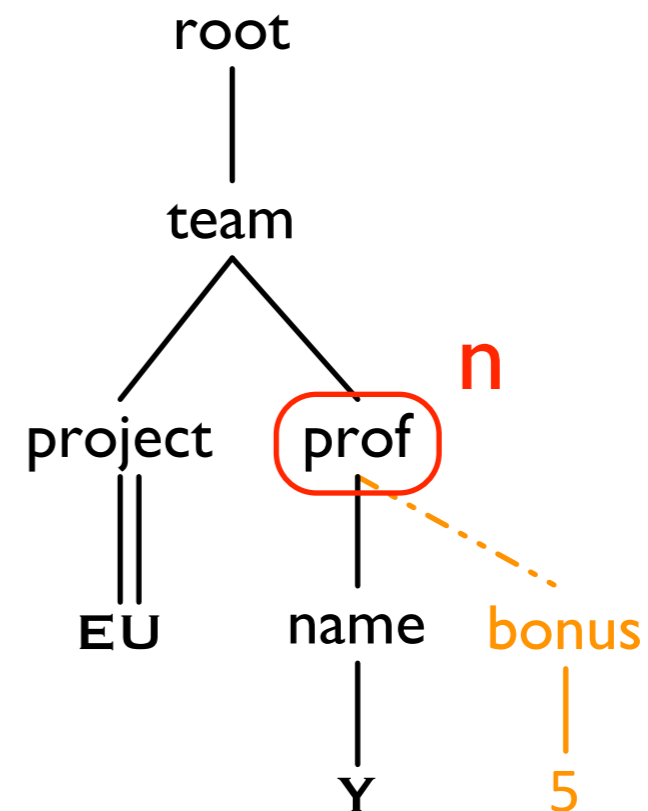
# Semantics of Updates for XML Documents

- **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



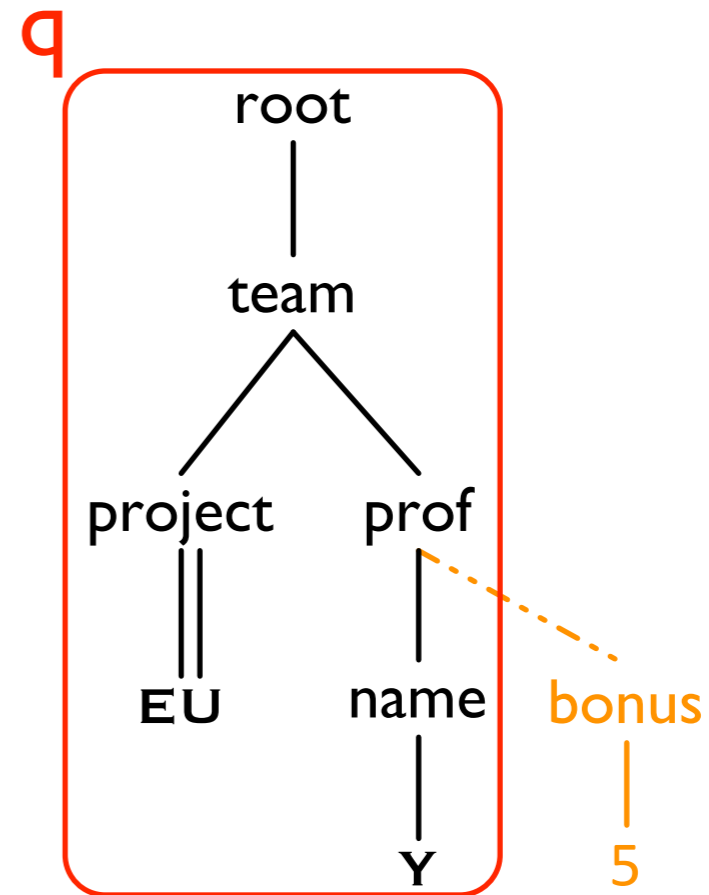
# Semantics of Updates for XML Documents

- **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



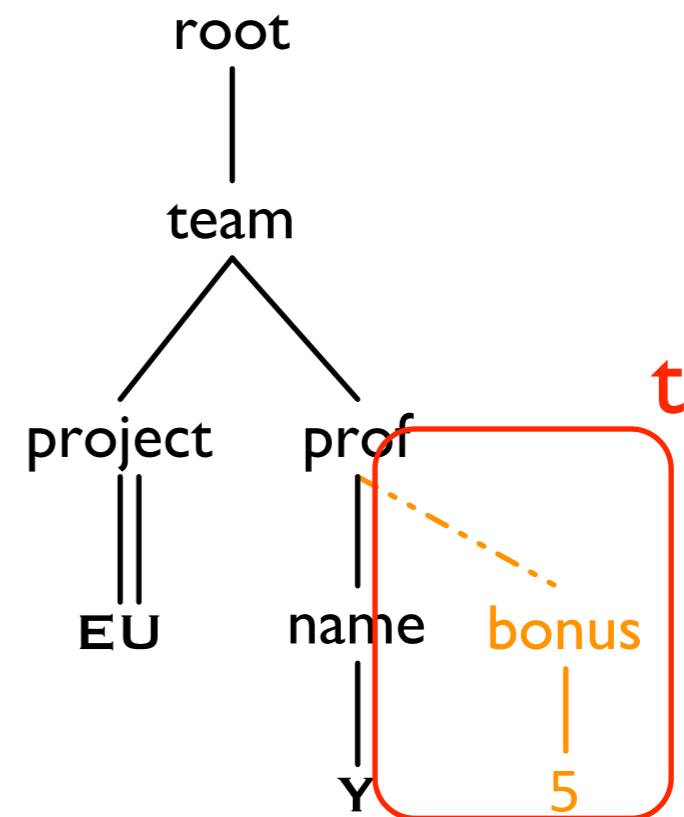
# Semantics of Updates for XML Documents

- **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



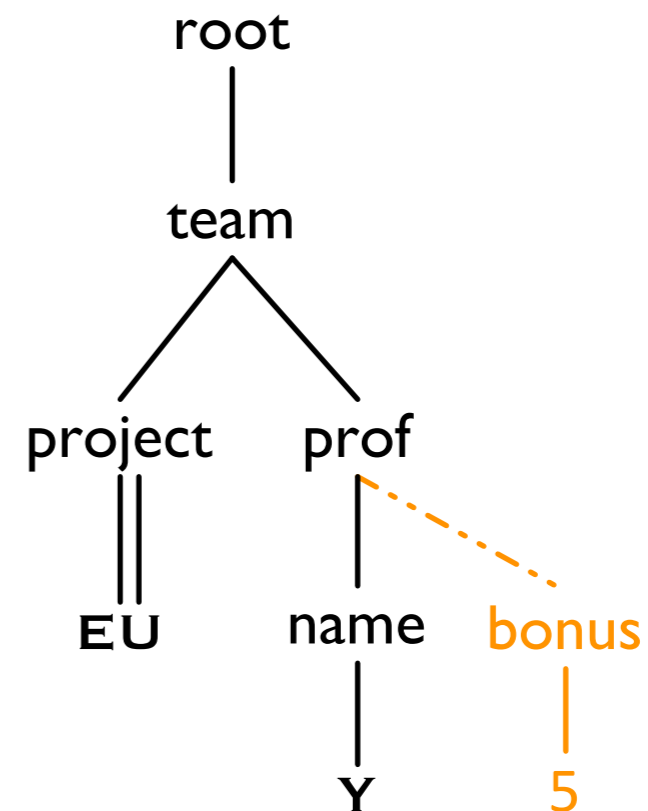
# Semantics of Updates for XML Documents

- **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

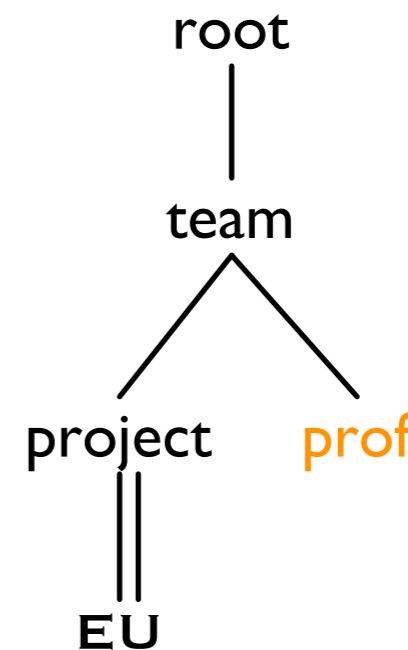




# Deletions

Deletion operation:  $(q, n)$

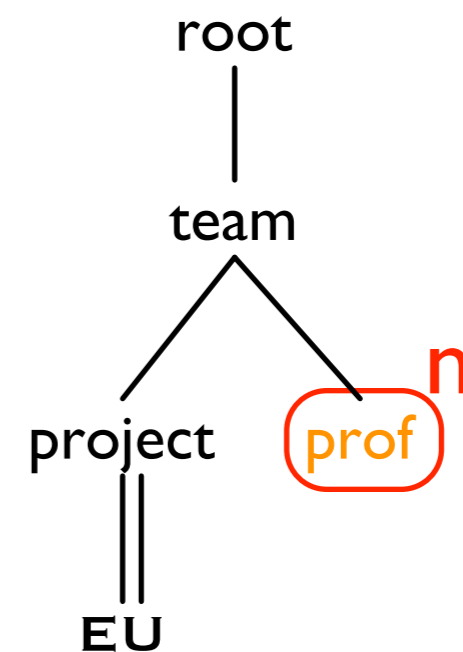
- *Fire a professor if her team is in a EU project*
- For every match of  $n$ , if there is a match of  $q$ , then delete  $n$  and all its descendants
- There is only one semantics for deletions, that is similar to **Only-if** semantics



# Deletions

Deletion operation:  $(q, n)$

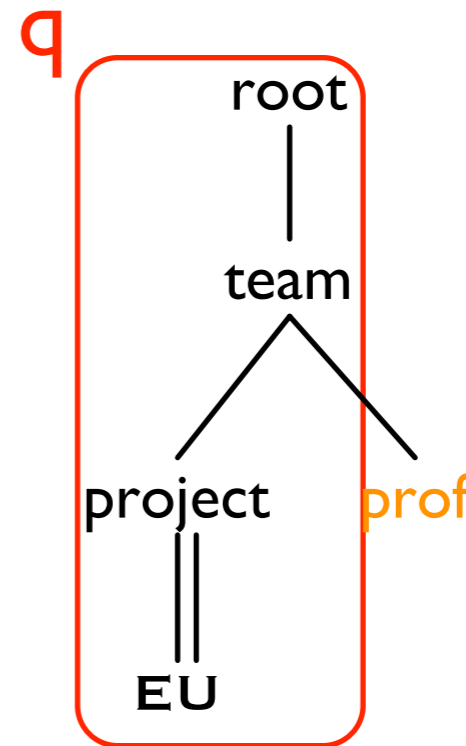
- *Fire a professor if her team is in a EU project*
- For every match of  $n$ , if there is a match of  $q$ , then delete  $n$  and all its descendants
- There is only one semantics for deletions, that is similar to **Only-if** semantics



# Deletions

Deletion operation:  $(q, n)$

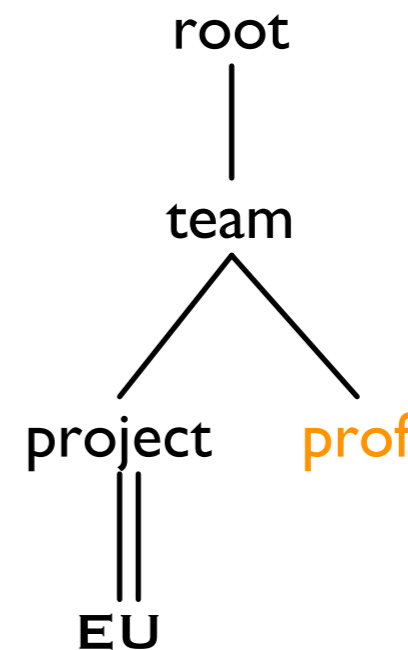
- *Fire a professor if her team is in a EU project*
- For every match of  $n$ , if there is a match of  $q$ , then delete  $n$  and all its descendants
- There is only one semantics for deletions, that is similar to **Only-if** semantics



# Deletions

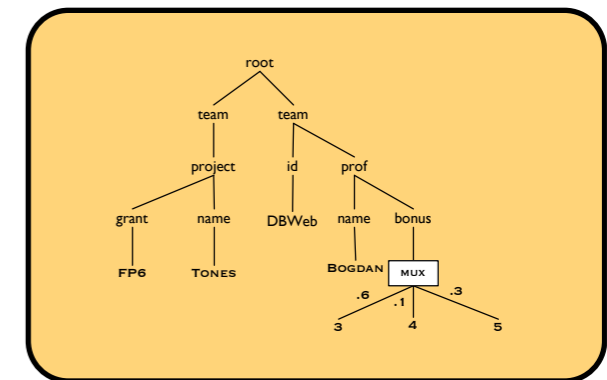
Deletion operation:  $(q, n)$

- *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

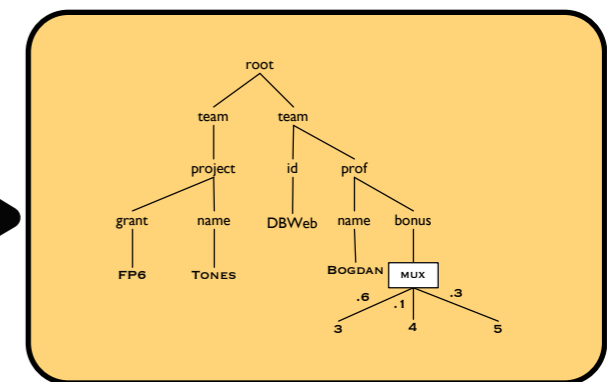
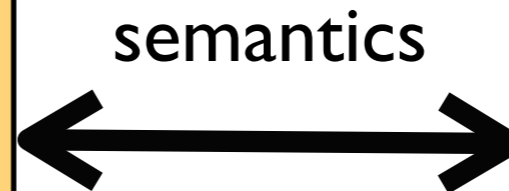
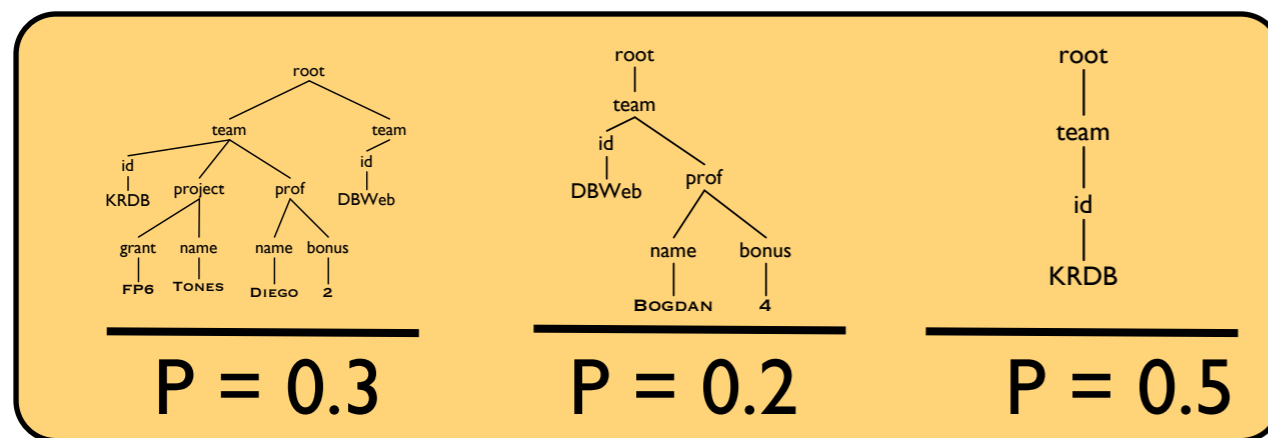
D: PXML doc



# Updating PXML Documents

Probability space of docs

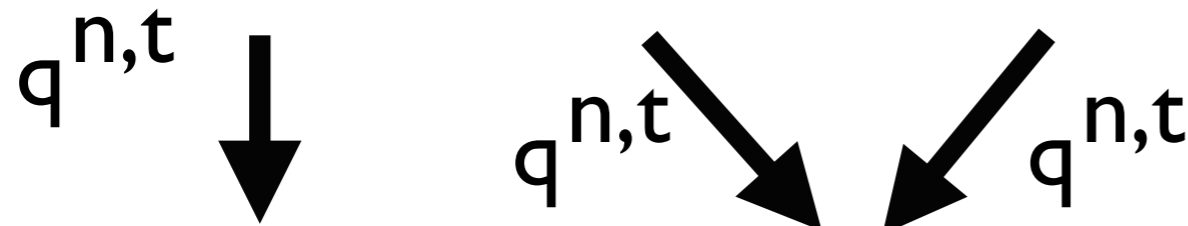
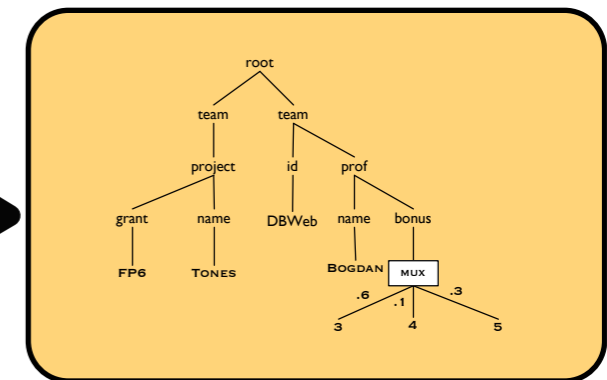
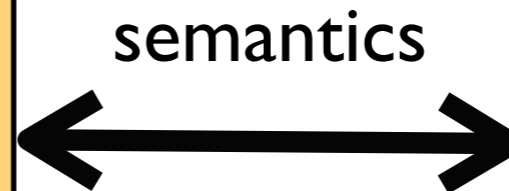
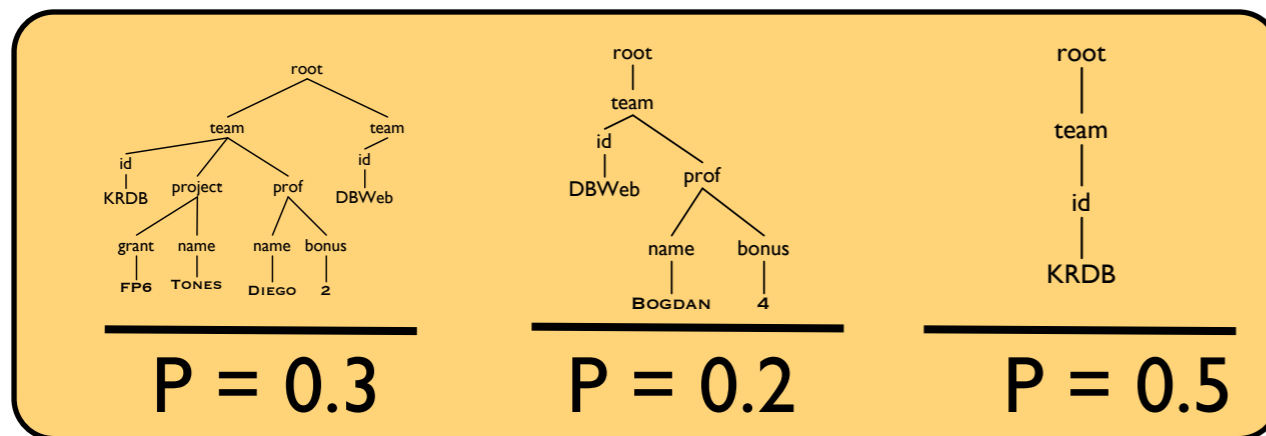
D: PXML doc



# Updating PXML Documents

Probability space of docs

D: PXML doc

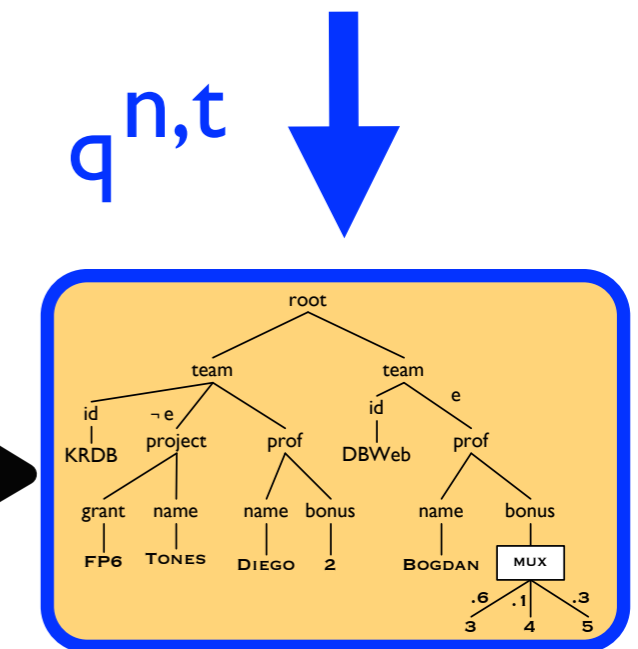
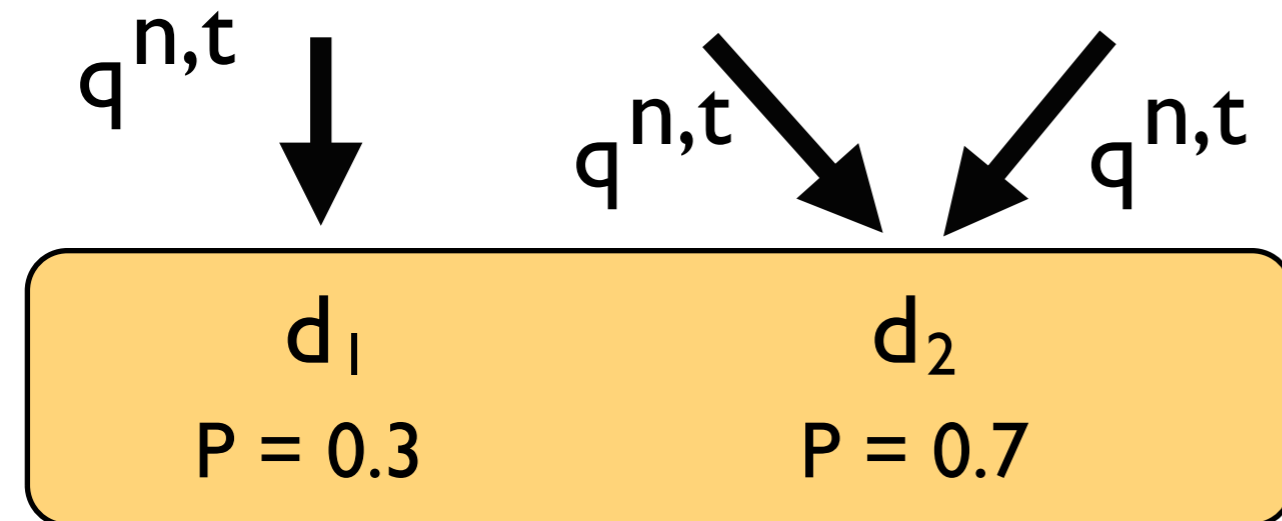
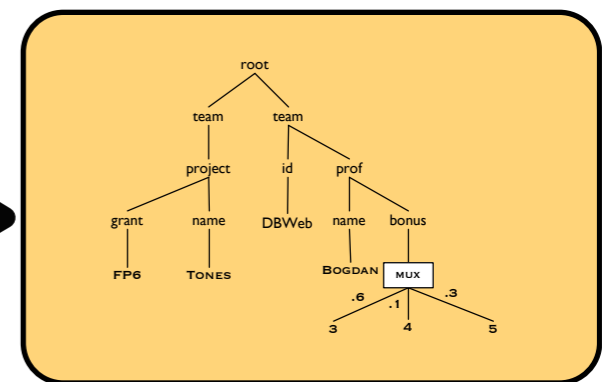
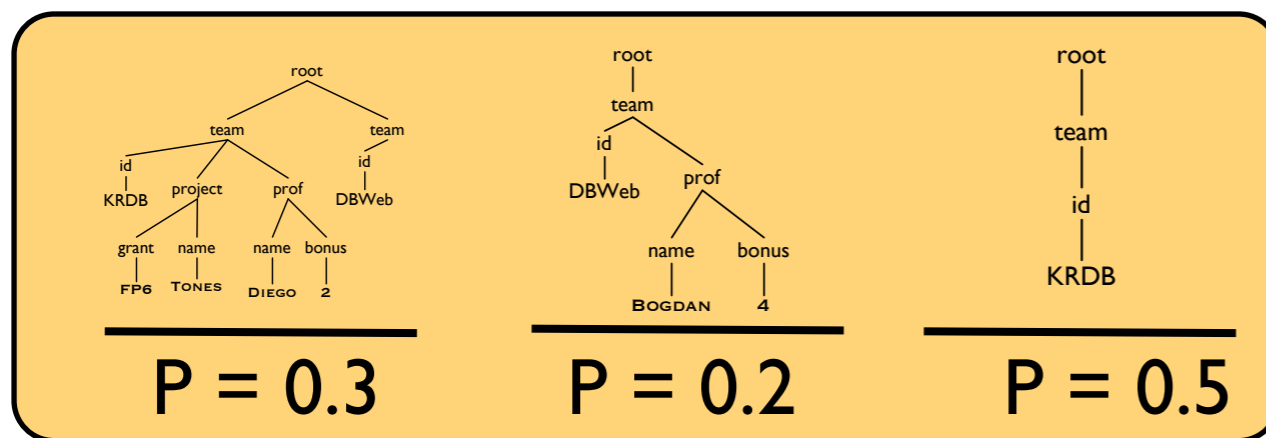


Updated prob. space of docs

# Updating PXML Documents

Probability space of docs

D: PXML doc



Updated prob. space of docs

D<sub>1</sub>: PXML doc



# Problems to Investigate

- We want to study 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?

# Outline

1. Probabilistic data
2. Problem of updates
3. Updating desecrate PXML
4. Updating continuous PXML

# Querying PXML with Tree-Pattern Queries

Queries	Distr. nodes <sup>*</sup>	Event conjunct. <sup>*</sup>	Event formulas
TP	P	#P-complete	
TPJ	#P-complete		

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

**#P functions** - counting counterparts of NP problems.

E.g: counting sat.-assignments for prop. CNF formulas.  
Believed to be harder than NP.

# 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

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

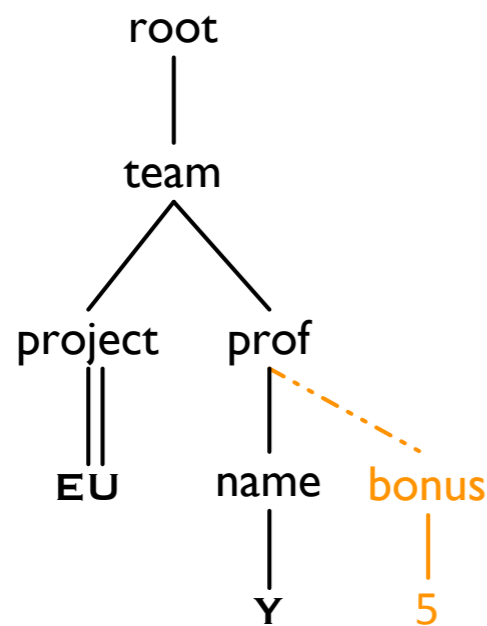
Only-if	Distr. nodes	Event conjunct	Event formulas
RSP	Linear		
SP	$P^*$	#P-hard	Linear
TP	?		P
TPJ	#P-hard		

\* only for queries without descendent edges

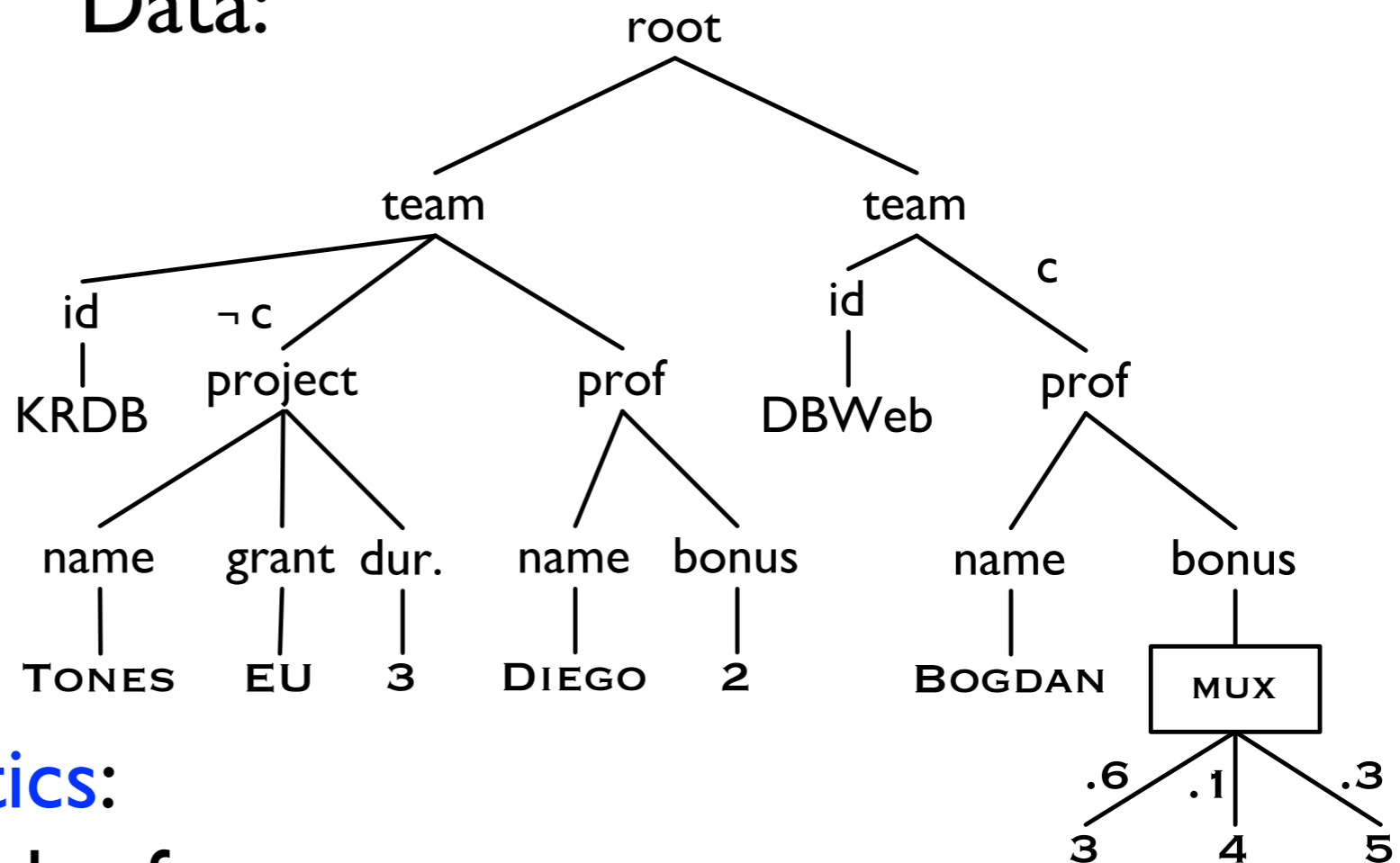
- The same table holds for **deletions**

# Updating PXML: Example

Query:



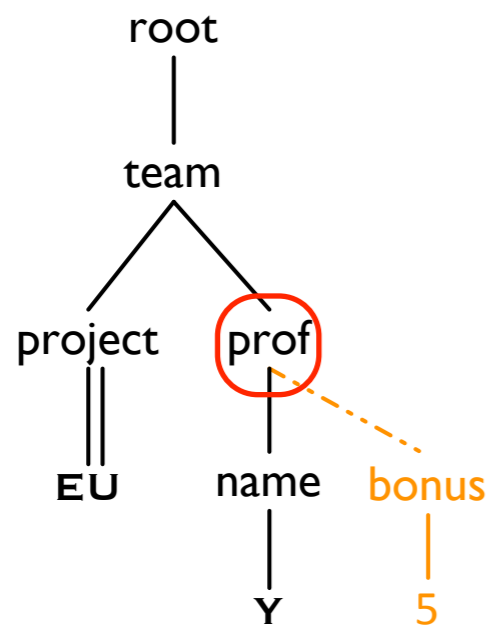
Data:



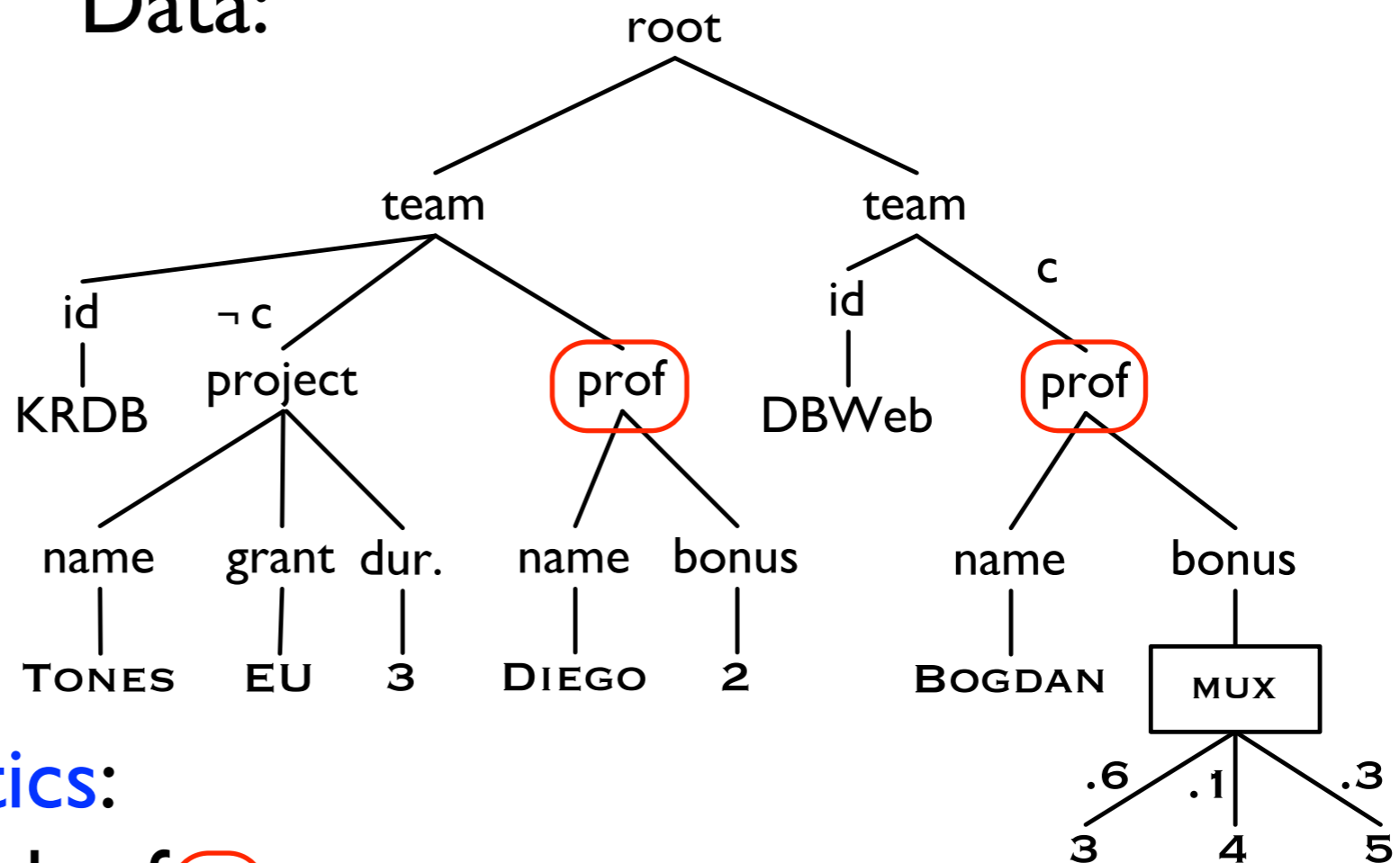
- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

Query:



Data:

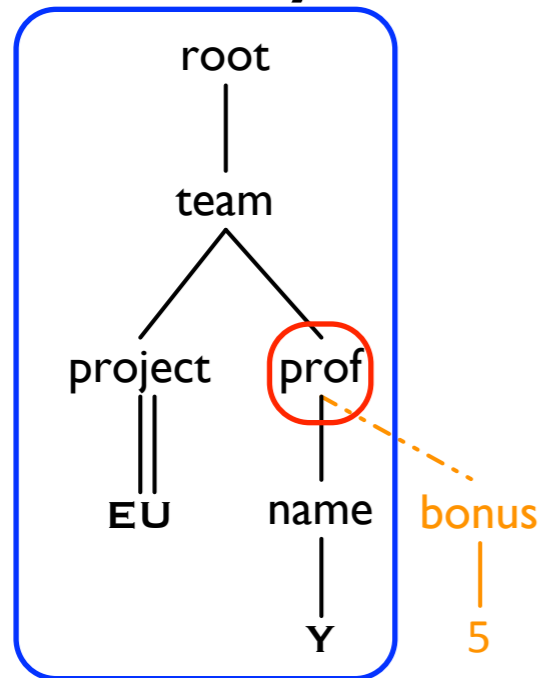


- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

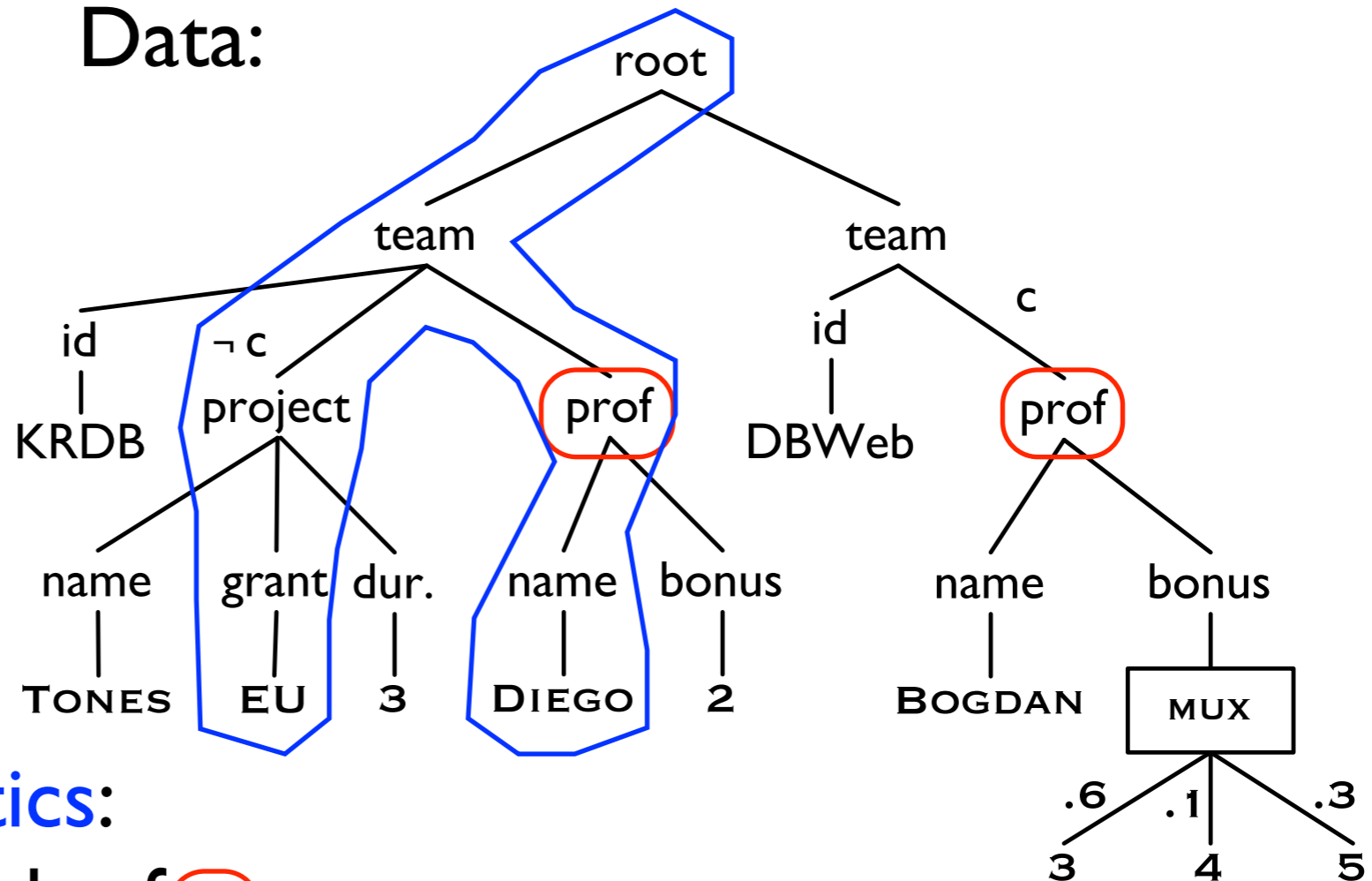


# Updating PXML: Example

Query:



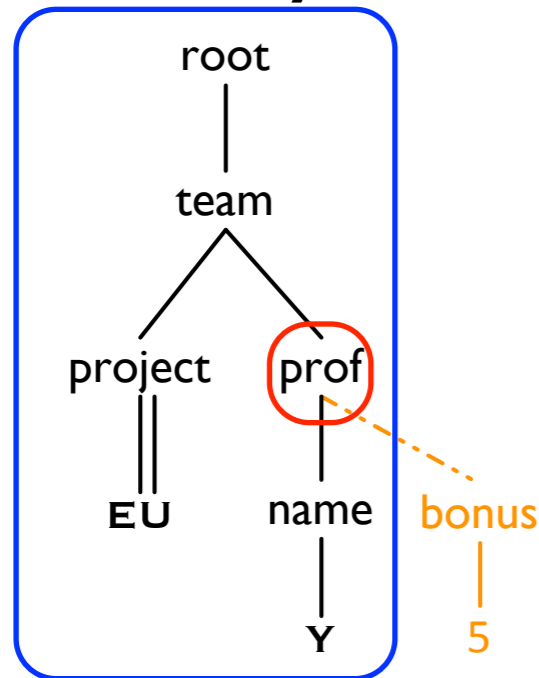
Data:



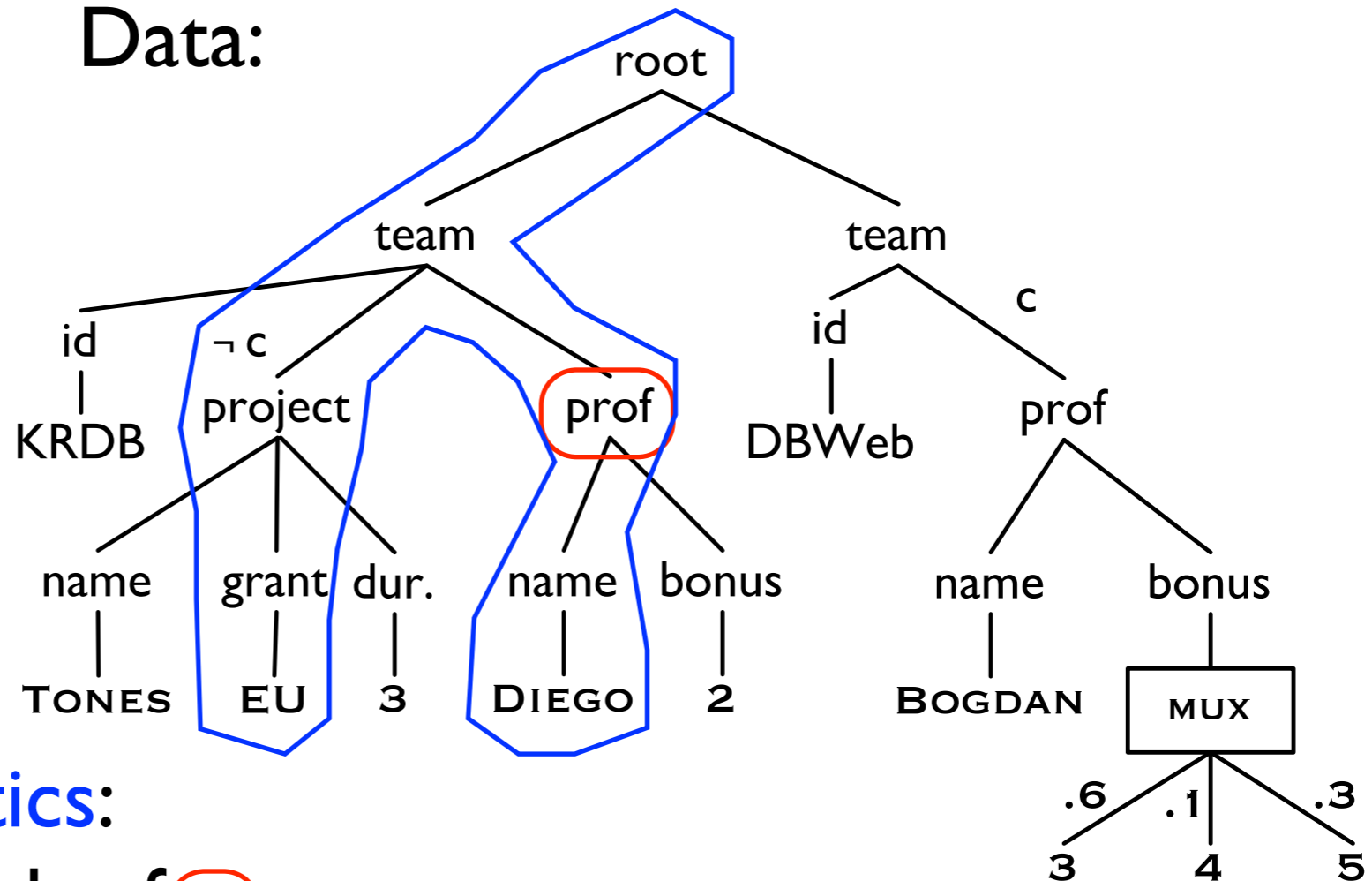
- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

Query:



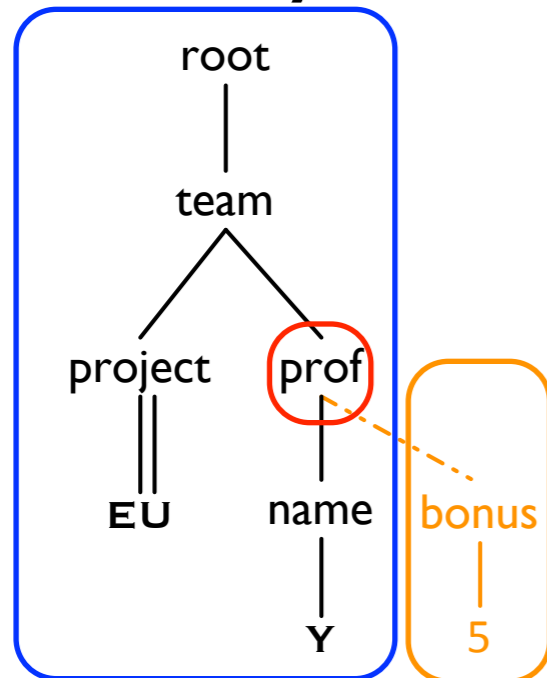
Data:



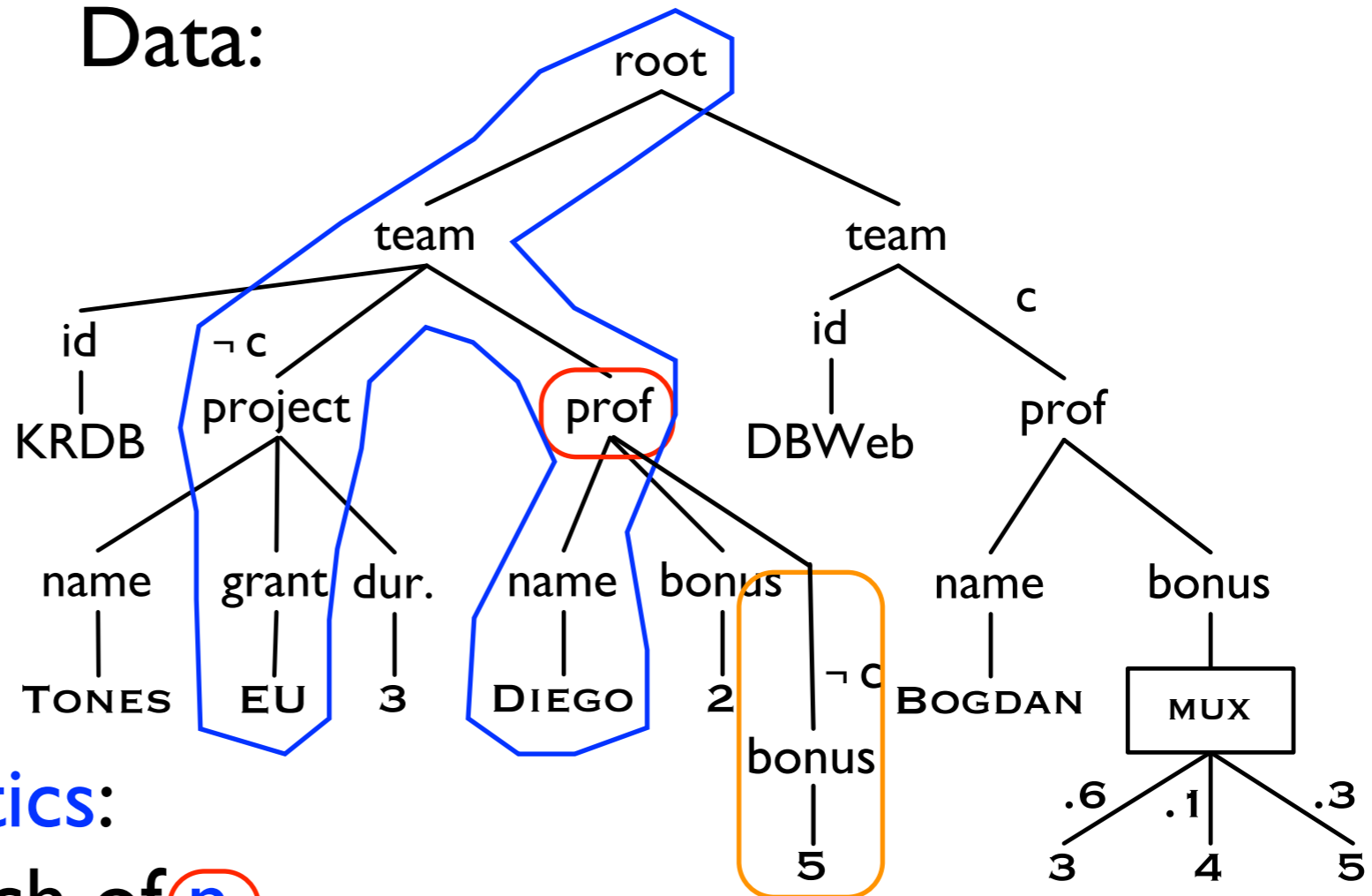
- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

Query:



Data:



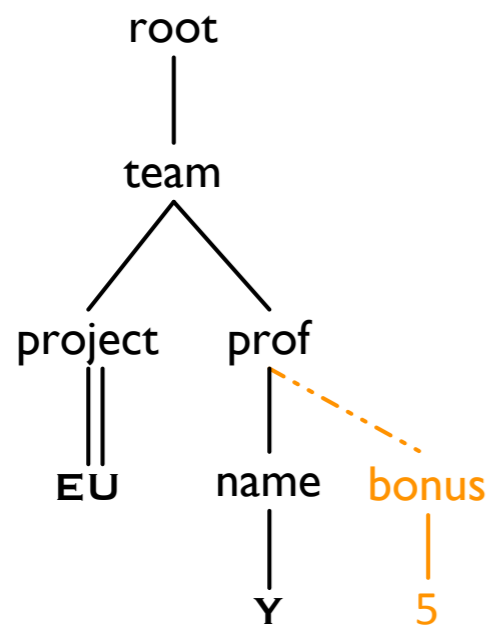
- **Only-if semantics:**

For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**

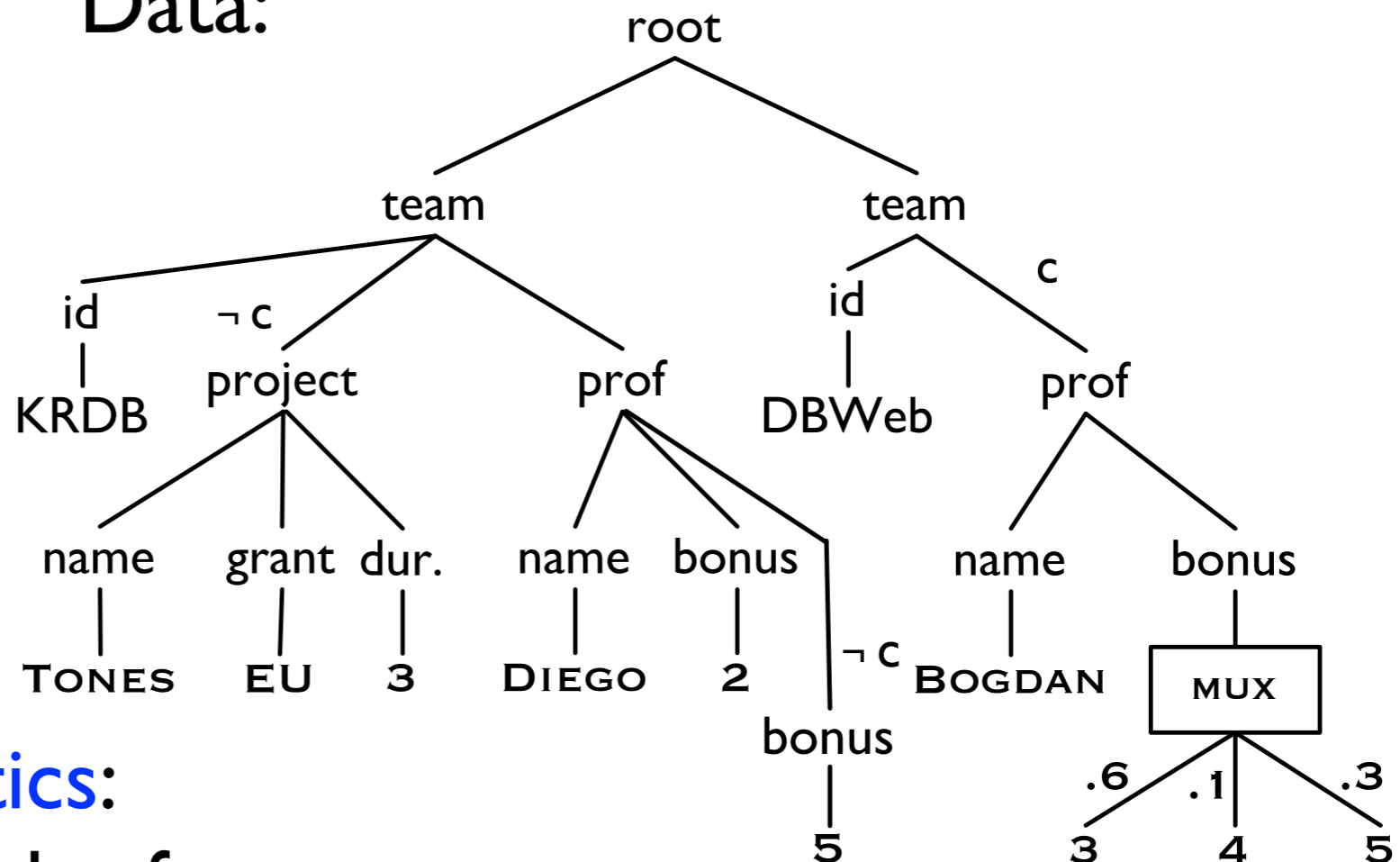
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

Query:



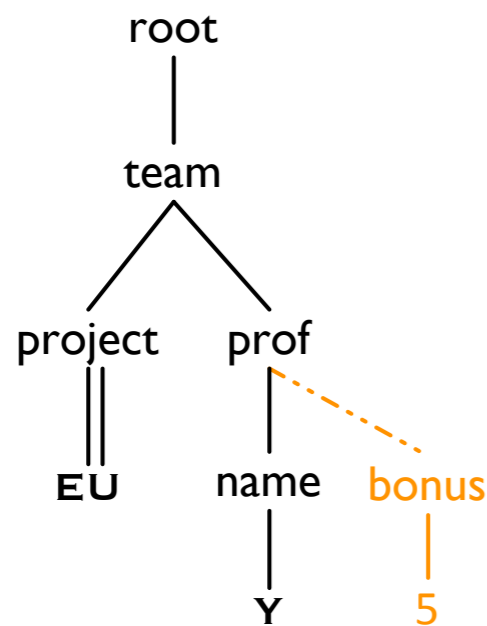
Data:



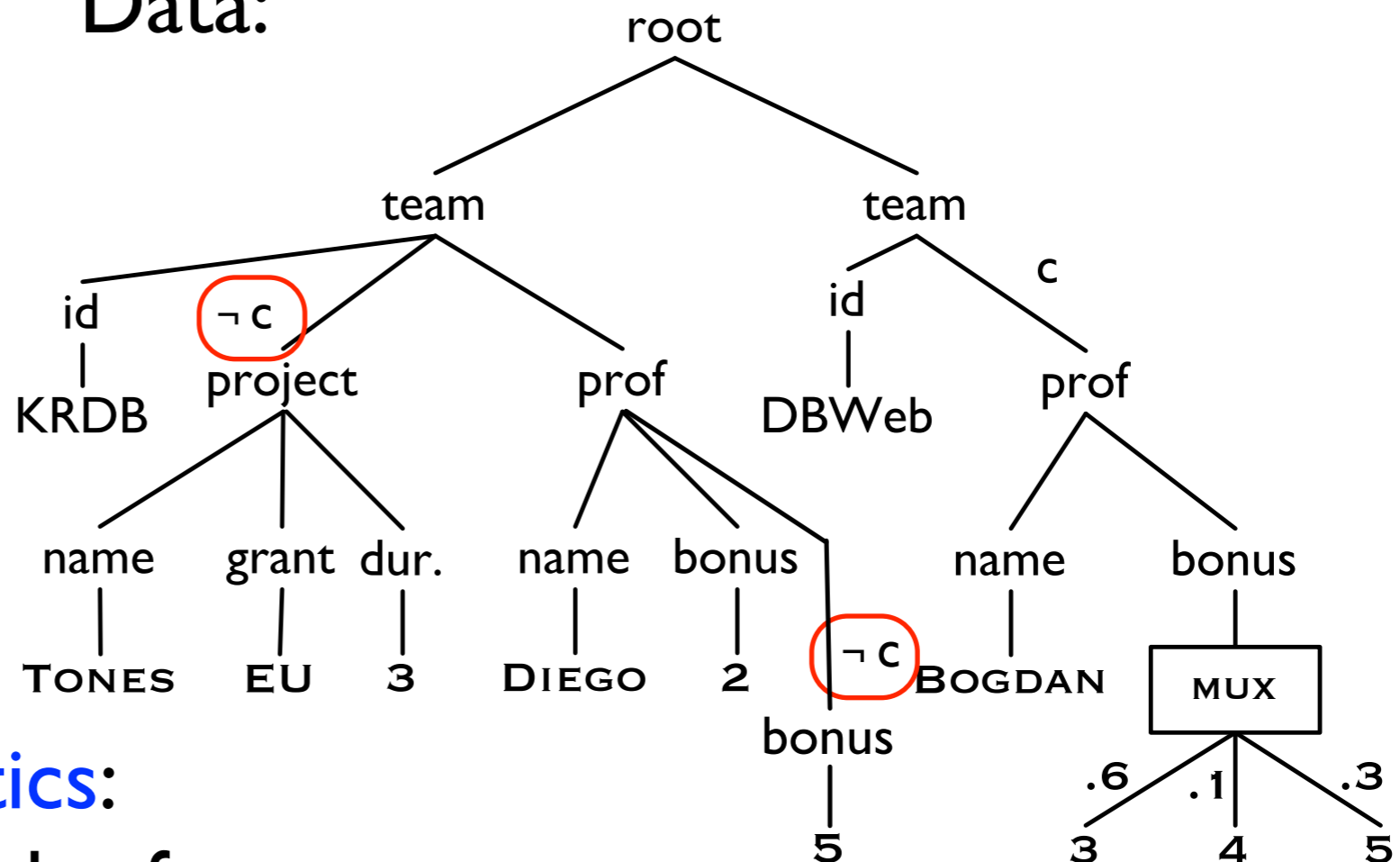
- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

Query:



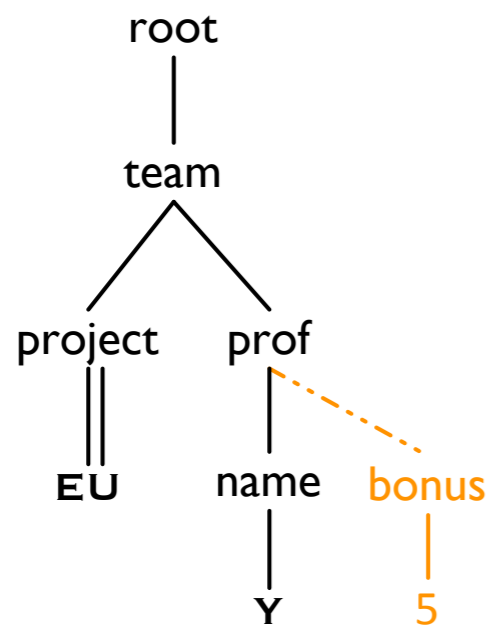
Data:



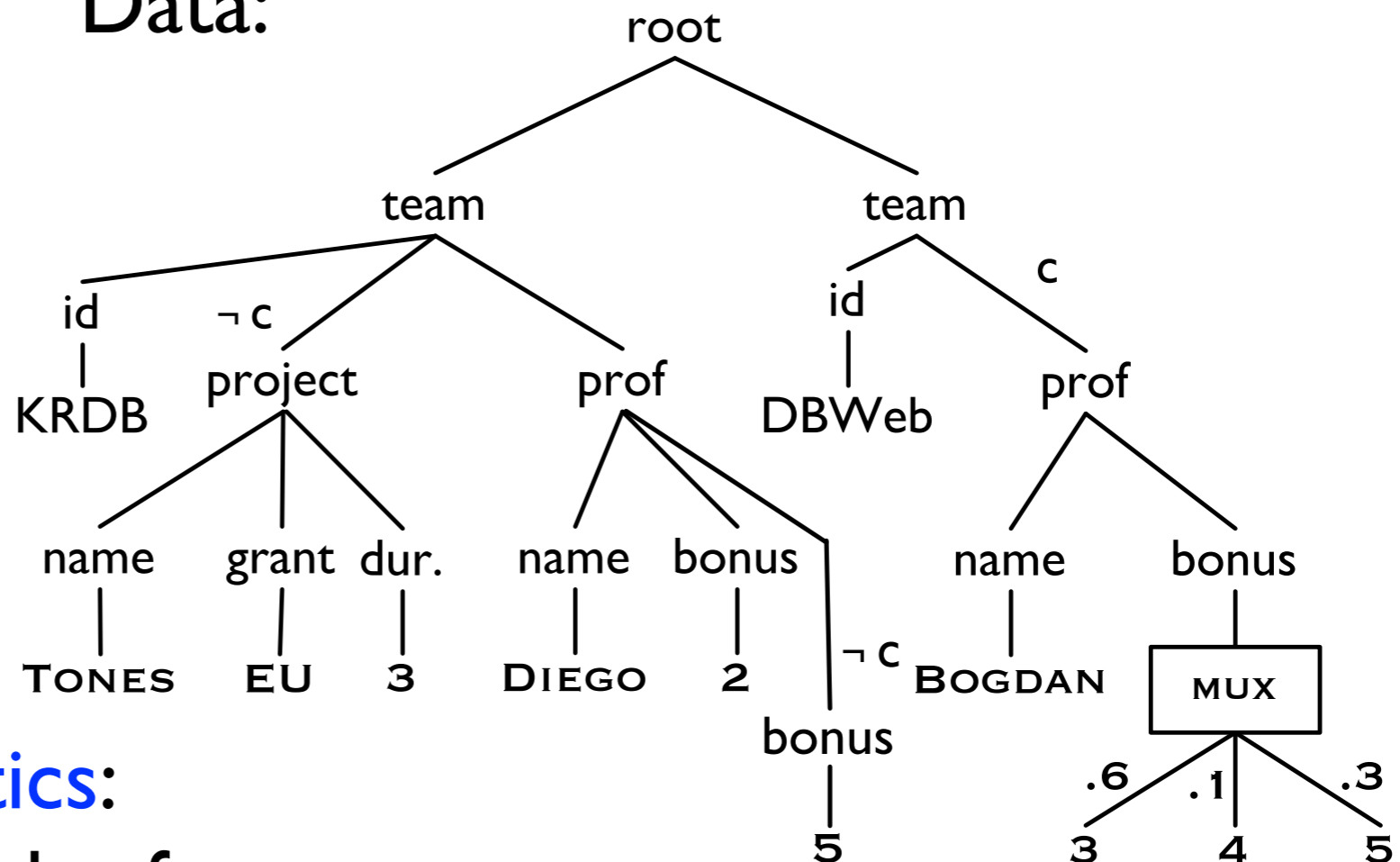
- **Only-if semantics:**  
For every match of **n**,  
if there is a match of **q**,  
then insert **t** under **n**
- in this case only-if and for-all semantics **coincide**

# Updating PXML: Example

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**

# For-all Insertions: Data Complexity

For-all	Distributional nodes	Event conj	Event formulas
RSP	Linear/P <sup>†</sup>		
SP	not in PTIME <sup>*</sup>	Linear/P <sup>†</sup>	
TP	not in PTIME	P	
TPJ	not in PTIME, #P-hard	P <sup>*</sup>	P

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

\* the computation is not in PSPACE, from [Abiteboul&al.:2009]

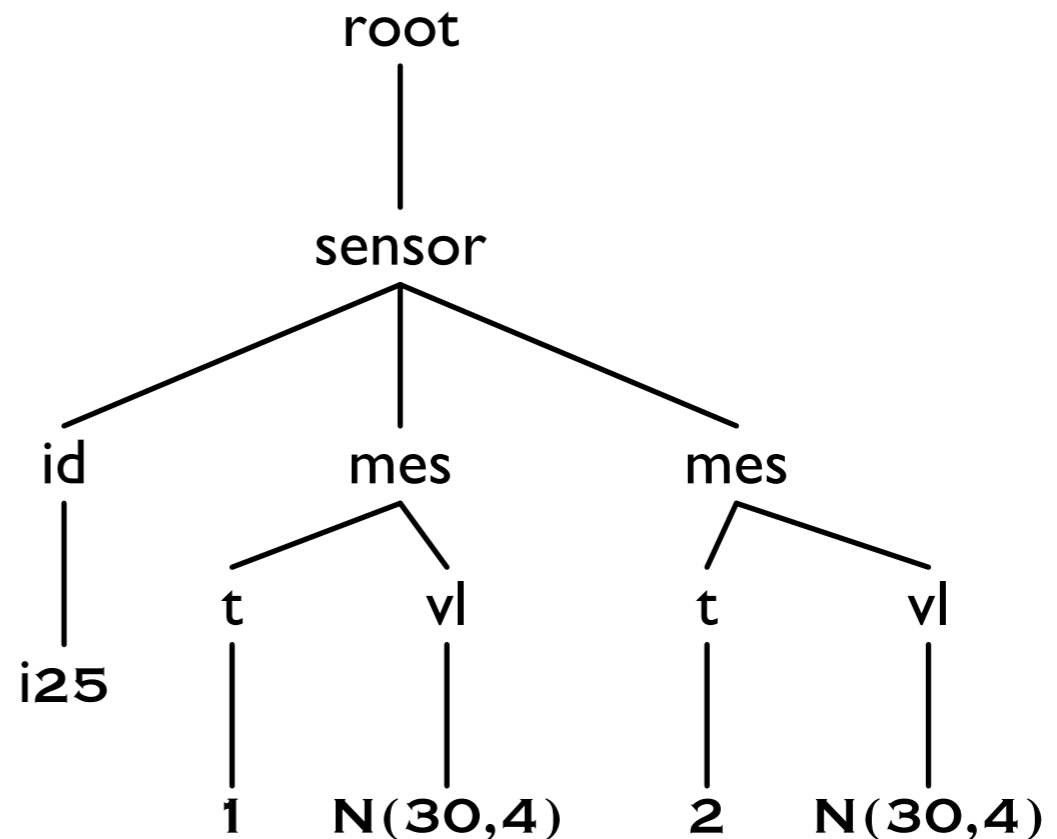
# Outline

1. Probabilistic data
2. Problem of updates
3. Updating discrete PXML
4. Updating continuous PXML



# Continuous PXML

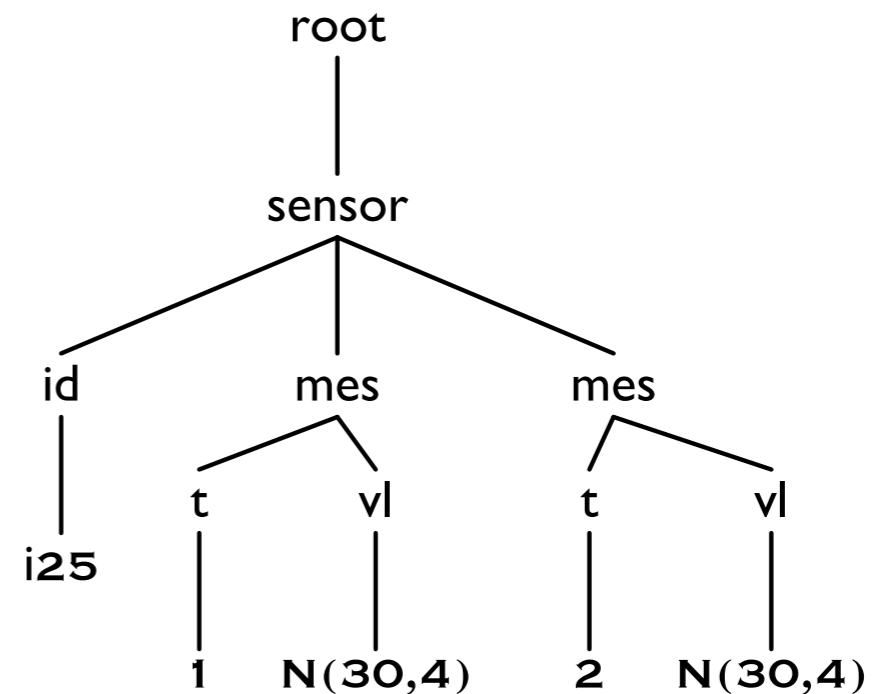
$N(30, 4)$  - Normal distribution



- 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 alert “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

- **Comprehensive picture** of updates' complexity:
  - **Discrete** PXML models with distributional nodes and event formulas
  - RSP, SP, TP and TPJ update operations
- **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

# Madam



- Thank you

# References

- [\[Kimelfeld&al:2007\]](#) - Benny Kimelfeld, Yehoshua Sagiv: Matching Twigs in Probabilistic XML. VLDB 2007: 27-38
- [\[Senellart&al:2007\]](#) - Pierre Senellart, Serge Abiteboul: On the complexity of managing probabilistic XML data. PODS 2007: 283-292