Querying and Updating Probabilistic Information in XML

Serge Abiteboul Pierre Senellart

EDBT 2006
March 28th, 2006

Imprecise data

- Many tasks generate imprecise data, with some confidence value:

Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction

Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction
  - Natural Language Processing

Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction
  - Natural Language Processing
  - Data Cleaning

Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction
  - Natural Language Processing
  - Data Cleaning
  - Schema Matching

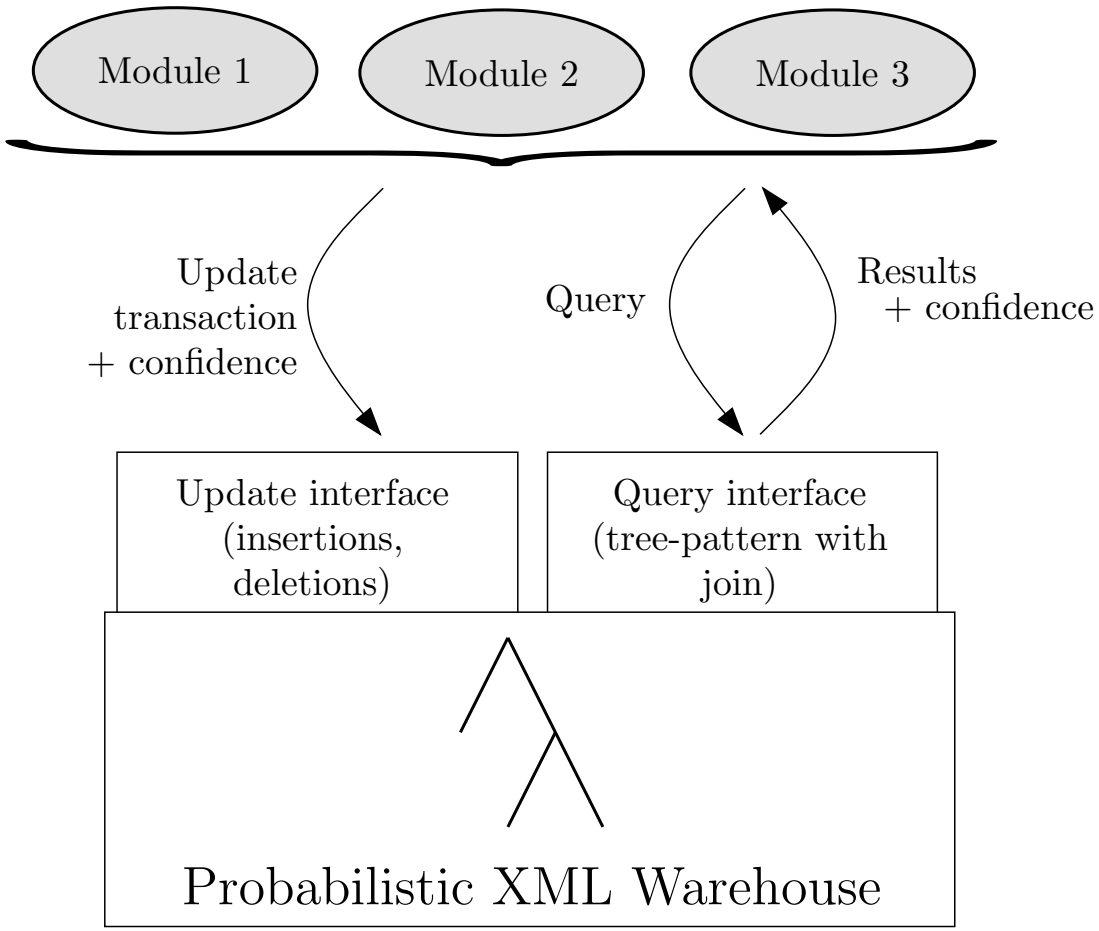
Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction
  - Natural Language Processing
  - Data Cleaning
  - Schema Matching
  - ...

Imprecise data

- Many tasks generate imprecise data, with some confidence value:
  - Information Extraction
  - Natural Language Processing
  - Data Cleaning
  - Schema Matching
  - ...
- Need for a way to manage this imprecision, to work with it throughout an entire complex process.

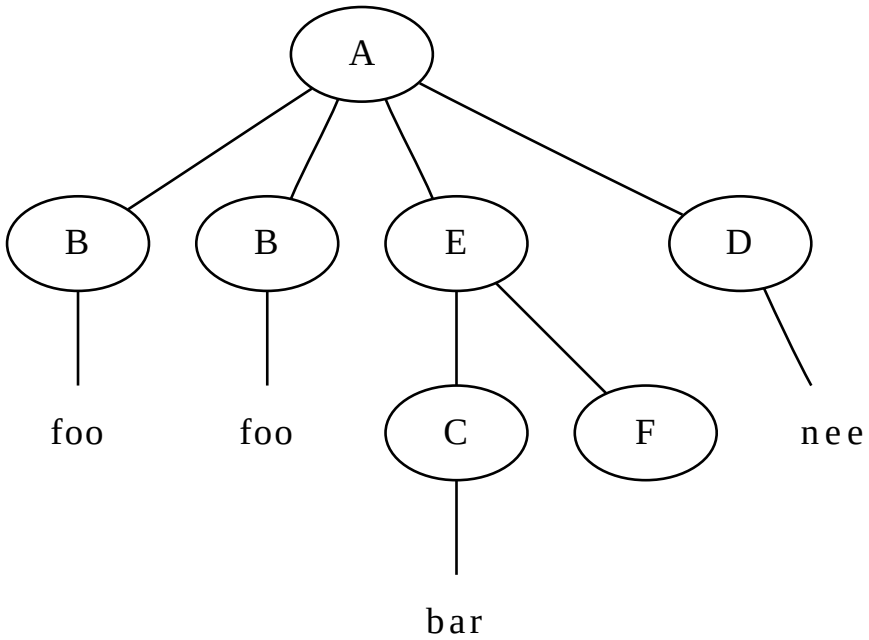
A Probabilistic XML Warehouse



Outline

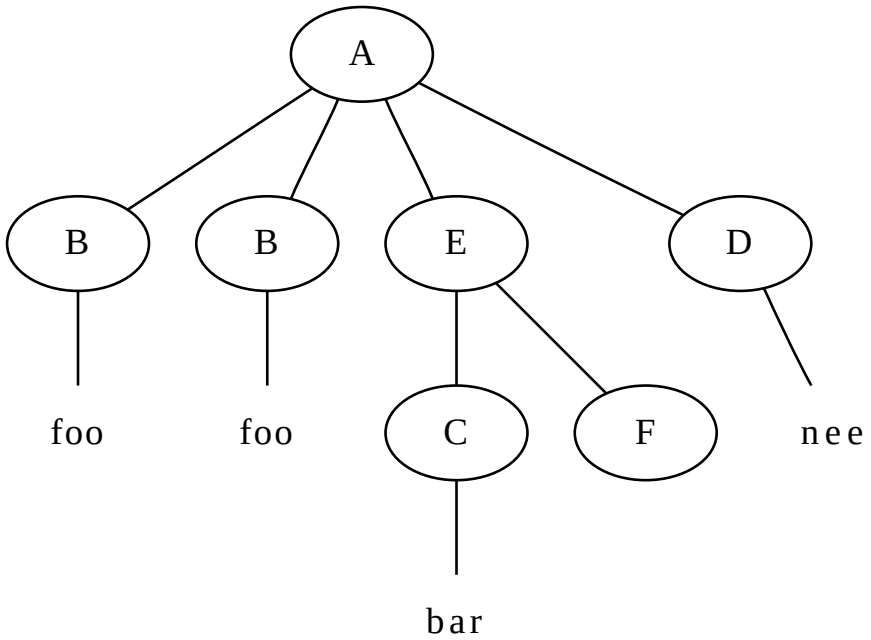
Data Trees

- Finite, unordered, trees.



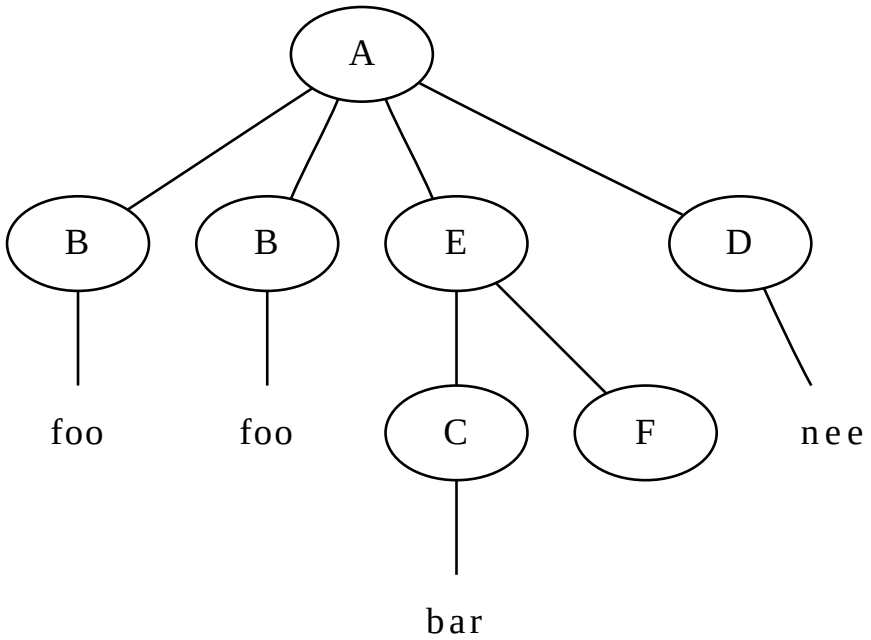
Data Trees

- Finite, unordered, trees.
- No distinction between attribute and element nodes.



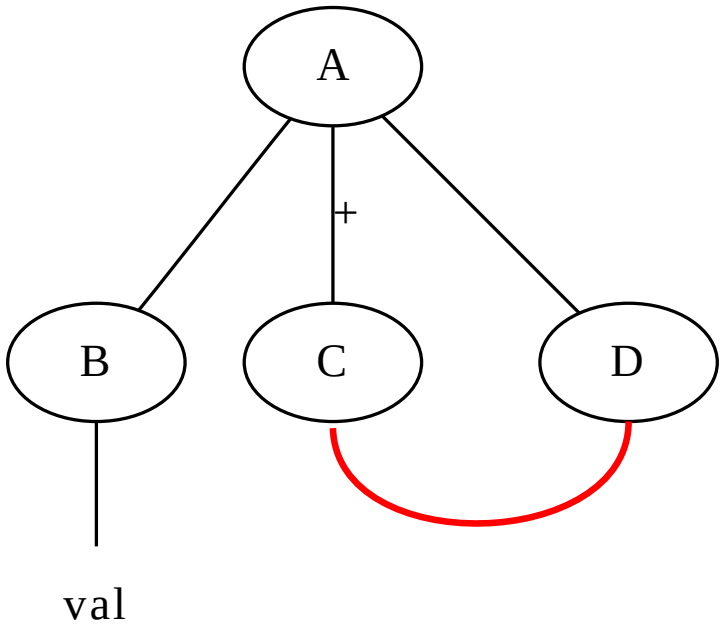
Data Trees

- Finite, unordered, trees.
- No distinction between attribute and element nodes.
- No mixed content.



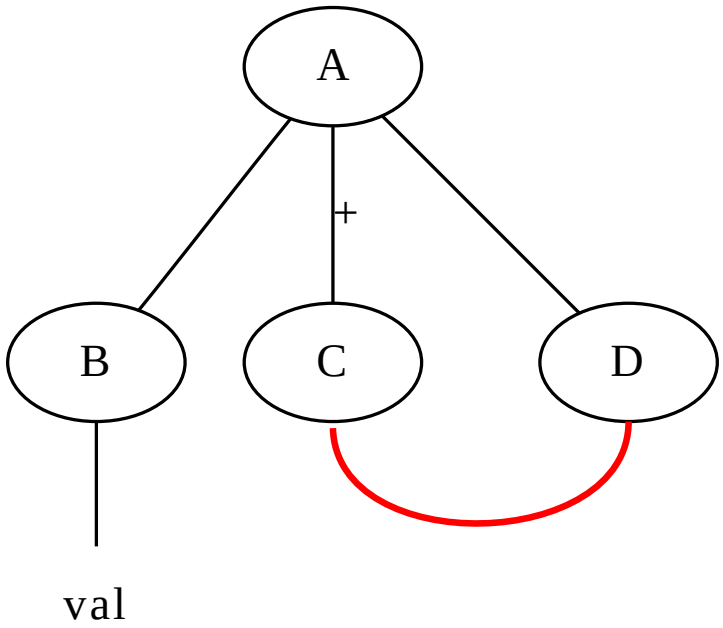
Tree-Pattern With Join Queries

- Queries: Tree-Pattern With Join (TPWJ)
  (standard subset of XQuery)



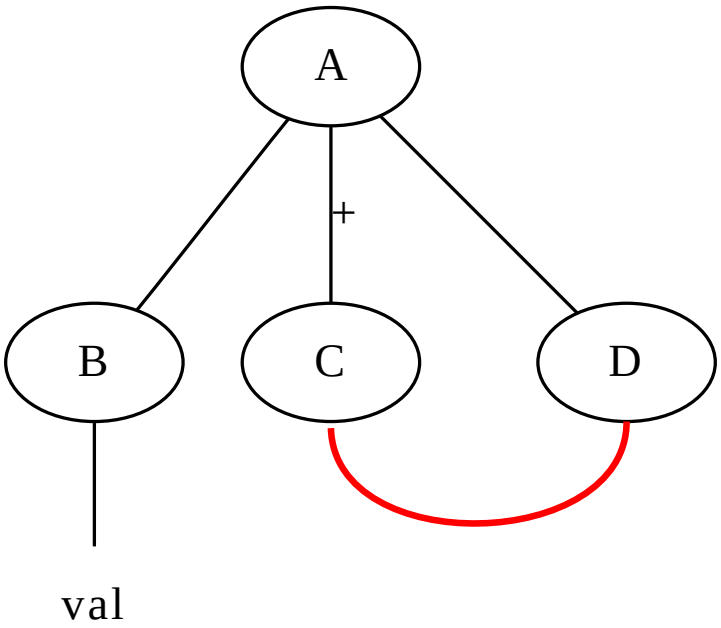
Tree-Pattern With Join Queries

- Queries: Tree-Pattern With Join (TPWJ)
  (standard subset of XQuery)
- Join: by value



Tree-Pattern With Join Queries

- Queries: Tree-Pattern With Join (TPWJ)
  (standard subset of XQuery)
- Join: by value
- Result: minimal subtree containing all the nodes mapped by the query



Update Transactions

- Set of elementary operations:

Update Transactions

- Set of elementary operations:
  - Insertions of subtrees

Update Transactions

- Set of elementary operations:
  - Insertions of subtrees
  - Deletions of subtrees

Update Transactions

- Set of elementary operations:
  - Insertions of subtrees
  - Deletions of subtrees
- Update Transaction: TPWJ query + mapping, stating where to perform operations.

Update Transactions

- Set of elementary operations:
  - Insertions of subtrees
  - Deletions of subtrees
- Update Transaction: TPWJ query + mapping, stating where to perform operations.
- Probabilistic update: update + confidence

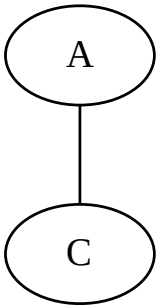
Outline

Possible Worlds Model

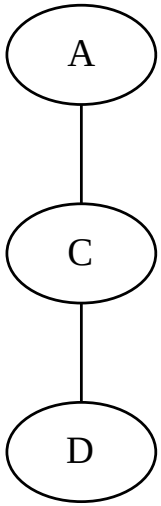
Semantic foundation for probabilistic data: possible worlds model.
Set of tree/probability pairs, one for each possible world.

Possible Worlds Model

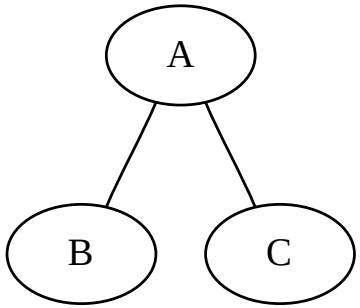
Semantic foundation for probabilistic data: possible worlds model.
Set of tree/probability pairs, one for each possible world.



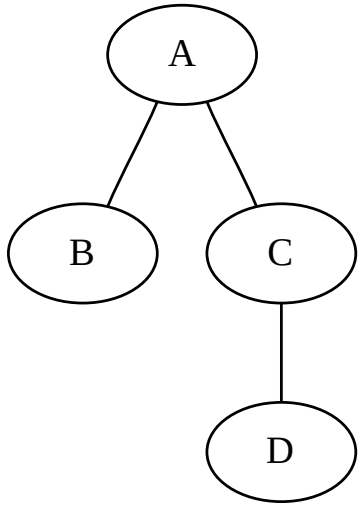
$P = 0.06$



$P = 0.14$



$P = 0.24$



$P = 0.56$

Queries, Updates: Semantic Foundation

Definition

If $T = \{(t_i, p_i)\}$, the result of query Q over the Possible Worlds set T is the normalization of $\{(t, p_i) | t \in Q(t_i)\}$

Queries, Updates: Semantic Foundation

Definition

If $T = \{(t_i, p_i)\}$, the result of query Q over the Possible Worlds set T is the normalization of $\{(t, p_i) | t \in Q(t_i)\}$

Definition

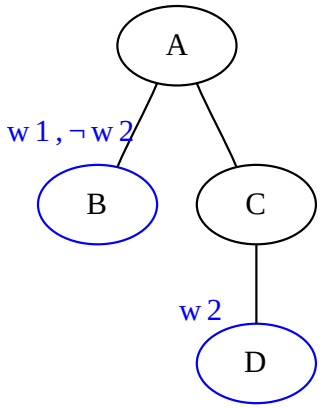
The result of an update t with confidence c on a Possible Worlds set T is the normalization of:

$$\begin{array}{ll} & \{(t, p) \in T \mid t \text{ is not selected by Q}\} \\ \bigcup & \{(\tau(t), p \cdot c) \mid t \text{ is selected by Q}\} \\ \bigcup & \{(t, p \cdot (1 - c)) \mid t \text{ is selected by Q}\} \end{array}$$

Outline

Fuzzy Trees

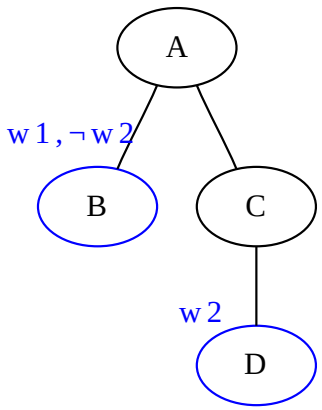
Data tree with event conditions (conjunction of probabilistic events or negations of probabilistic events) assigned to each node.



Event	Proba.
w1	0.8
w2	0.7

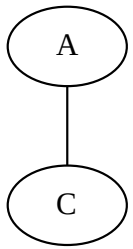
Fuzzy Trees

Data tree with event conditions (conjunction of probabilistic events or negations of probabilistic events) assigned to each node.

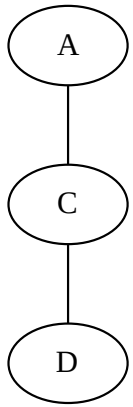


Event	Proba.
w1	0.8
w2	0.7

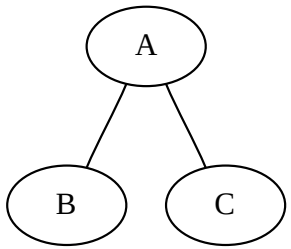
semantics $\longrightarrow$



$P = 0.06$



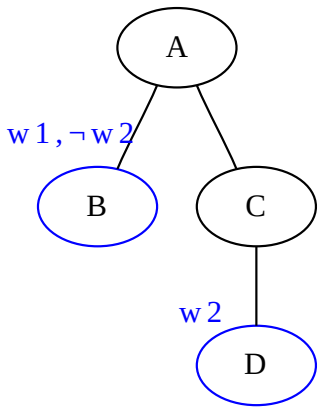
$P = 0.70$



$P = 0.24$

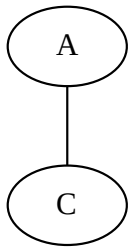
Fuzzy Trees

Data tree with event conditions (conjunction of probabilistic events or negations of probabilistic events) assigned to each node.

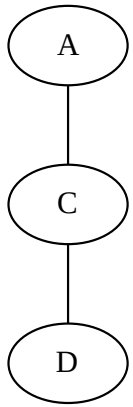


Event	Proba.
w1	0.8
w2	0.7

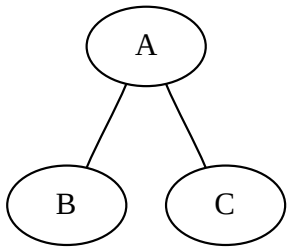
semantics $\longrightarrow$



$P = 0.06$



$P = 0.70$



$P = 0.24$

Theorem

The fuzzy tree model is as expressive as the Possible Worlds model.

Queries on Fuzzy Trees

Definition

Queries on fuzzy trees:

- Query on underlying tree.
- Probabilities: probability of the conjunction of the conditions of nodes of the mapping.

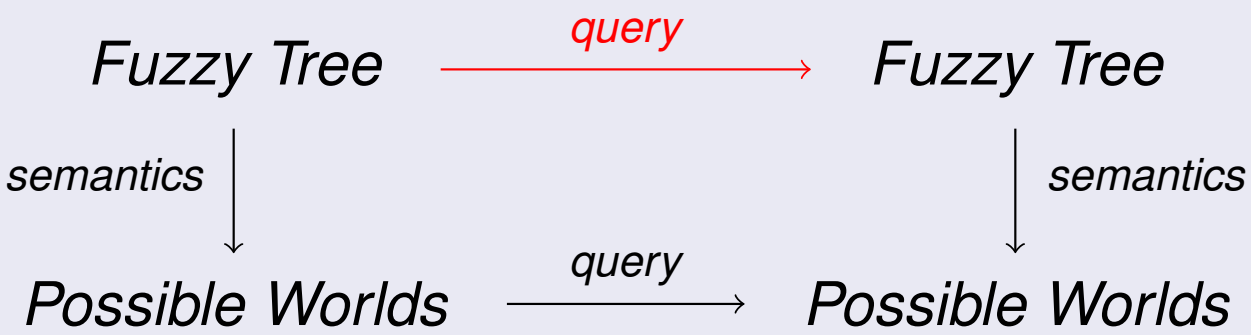
Queries on Fuzzy Trees

Definition

Queries on fuzzy trees:

- Query on underlying tree.
- Probabilities: probability of the conjunction of the conditions of nodes of the mapping.

Theorem



Updates on Fuzzy Trees

- Insertions: no problem. Conditions required for the query to match added to inserted nodes.

Updates on Fuzzy Trees

- Insertions: no problem. Conditions required for the query to match added to inserted nodes.
- Deletions: ok, but more problematic. May yield an exponential growth of the fuzzy tree in case of complex dependencies.

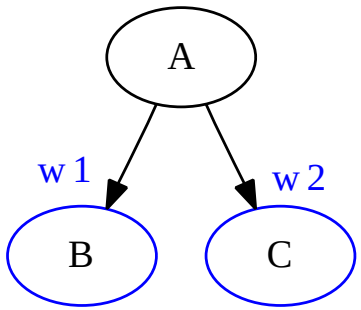
Updates on Fuzzy Trees

- Insertions: no problem. Conditions required for the query to match added to inserted nodes.
- Deletions: ok, but more problematic. May yield an exponential growth of the fuzzy tree in case of complex dependencies.

Theorem

Example: Conditional Replacement

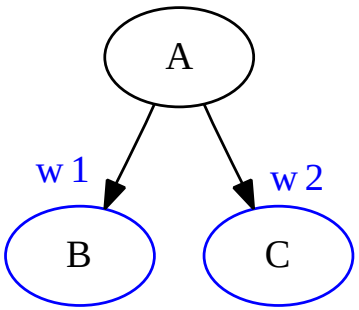
Replacement of *C* by *D* if *B* is present, with confidence 0.9.



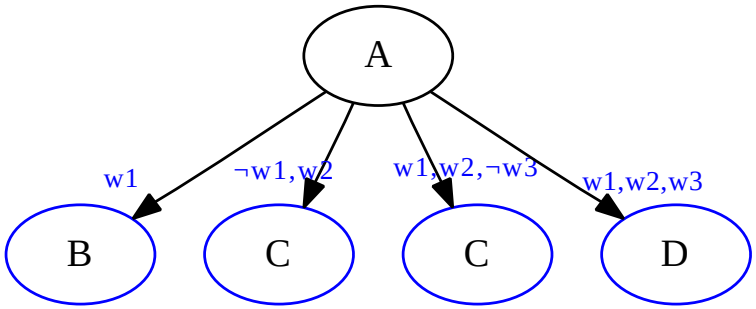
Event	Proba.
w1	0.8
w2	0.7

Example: Conditional Replacement

Replacement of *C* by *D* if *B* is present, with confidence 0.9.



Event	Proba.
w1	0.8
w2	0.7



Event	Proba.
w1	0.8
w2	0.7
w3	0.9

Implementation

- Java-based

Implementation

- Java-based
- File system storage (will look at an XML DB next)

Implementation

- Java-based
- File system storage (will look at an XML DB next)
- Query evaluation: Qizx/open XQuery engine

Implementation

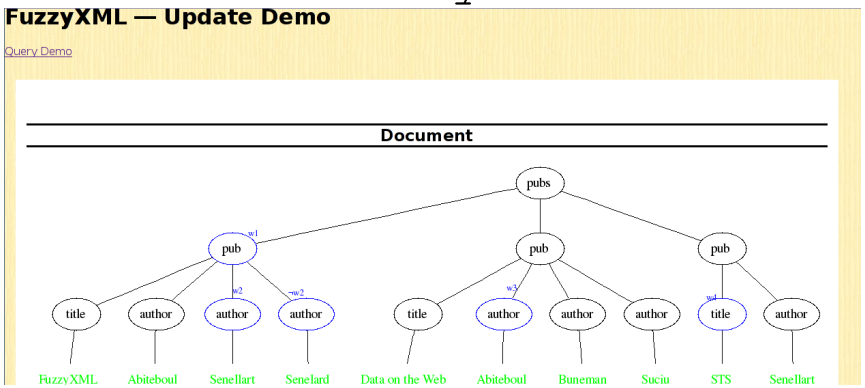
- Java-based
- File system storage (will look at an XML DB next)
- Query evaluation: Qizx/open XQuery engine
- Updates expressed in XUpdate

Implementation

- Java-based
- File system storage (will look at an XML DB next)
- Query evaluation: Qizx/open XQuery engine
- Updates expressed in XUpdate
- Available freely at
  `http://pierre.senellart.com/software/fuzzyxml/`

Implementation

- Java-based
- File system storage (will look at an XML DB next)
- Query evaluation: Qizx/open XQuery engine
- Updates expressed in XUpdate
- Available freely at
  `http://pierre.senellart.com/software/fuzzyxml/`
- cf demo
  `http://pierre.senellart.com/software/fuzzyxml/demo/`



Outline

Summary

- A model for representing probabilistic information for semi-structured data.

Summary

- A model for representing probabilistic information for semi-structured data.
- Sound and complete support for an important subset of XQuery.

Summary

- A model for representing probabilistic information for semi-structured data.
- Sound and complete support for an important subset of XQuery.
- Sound and complete support for XUpdate-based transactions with inserts and deletes.

Summary

- A model for representing probabilistic information for semi-structured data.
- Sound and complete support for an important subset of XQuery.
- Sound and complete support for XUpdate-based transactions with inserts and deletes.
- An implementation based on compilation to XQuery/XUpdate.

Perspectives

- Complexity analysis: query, update, simplification.

Perspectives

- Complexity analysis: query, update, simplification.
- Query optimization.

Perspectives

- Complexity analysis: query, update, simplification.
- Query optimization.
- Fuzzy data simplification.

Perspectives

- Complexity analysis: query, update, simplification.
- Query optimization.
- Fuzzy data simplification.
- Extensions: negation, some limited order.