

# Research on Basic Operations for Query Probabilistic XML Document Based on Path Set

Jianwei Wang

Computer Science and Technology College, Harbin University of Science and Technology, Harbin 150080, China  
Information and Computer Engineering College, Northeast Forestry University, Harbin 150040, China  
Email: jwwang2007@163.com

Zhongxiao Hao

Computer Science and Technology College, Harbin University of Science and Technology, Harbin 150080, China  
Computer Science and Technology School, Harbin Institute of Technology, Harbin 150001, China

**Abstract**—Probabilistic XML tree is a typical probabilistic XML data model for representing uncertainty in the form of probability which can be described the nested probabilistic XML element unit. The path set in probabilistic XML unit tree can be analyzed through parsing it and drawing out the corresponding probabilistic XML unit schema tree. The node probability and the node probability threshold computation are defined first. Then the set operations are defined. The probabilistic XML XQuery sentences are extended to query a node or nodes probability, path probability and to list the possible world set. The instance analysis shows that probabilistic XML data is an effective method representing probability data and the given basic operation can realize the related probability query.

**Index Terms**— probabilistic XML element unit; probabilistic XML data tree; probabilistic XML element tree; query; basic operation

## I. INTRODUCTION

Query optimization technique is an important research problem in any data management system. And how to define a kind of perfect and reasonable algebra is the key factor. Probabilistic XML data is a special extended XML data that can represent the probability information, so to define a new kind of probabilistic XML algebra operator set is also essential for query optimization.

Probabilistic XML algebra mainly has two schemes: extended XML algebra and extended probabilistic relation algebra. The related research contents are described as follows.

The method of extended XML algebra is to add the operations on probabilistic data on the basis of founded XML algebra. XML algebra is a set of XML document operation that follows the given data model according with XML query data model, query language and query algebra in the demand document that is issued by XML query group of W3C. The formal semantics especially, defined as bottom algebra based on navigation way, includes the three aspects: operation object, operation and maturity[1]. From 1999, a series of XML algebra has been proposed, for example TAX(Tree Algebra for XML) in [2], Niagara Algebra in [3] and XAL algebra [4]. The way

of extended XML algebra is applied in [5] where ProTDB system is realized on the basis of TIMBER[2]. The main work is to add the probabilistic XML data management function in query parser and query executor. Because of directly applications of TAX algebra system, the disadvantage is not terse. PEPX system also continues to use this method where did not embody the operation expression and the optimization rules [6].

SP\_algebra is a classical probabilistic XML algebra based on relation algebra [7, 8, 9]. Its operation object is semis\_structured probabilistic object SPO. And the operation symbol involves the normal set operation such as union, intersection and difference, also extends relation algebra operation such as selection, project, cartesian product and join and introduces a new conditionalization operation. The operation procedure firstly transforms the relation table to SPO, secondly applies the SP\_algebra. The shortcoming is the aggregation function and the maturity proof.

New probabilistic XML data query theory also shows that foundation of probabilistic XML algebra is essential. The material content describes as follows.

1) Aggregation operation is an inevitable problem during the course of querying the probabilistic XML data. How to compute aggregation function is studied in [10].

2) Join operation is realized by probabilistic XML data code. Usually probabilistic XML data is enormous, so code and compress has been the facing difficult problem. The compress problem is studied in [11] in order to reduce the node number and does not give material arithmetic procedure. The join operation is studied by ZHANG code in [12].

The third scheme shows a new probabilistic XML data query theory, but does not found the integrated probabilistic XML algebra system. Aggregation operation is an inevitable problem during the course of querying the probabilistic XML data. How to compute aggregation function is studied in [10]. The compress problem is studied in [11] without the essential algorithm procedure. The join operation is studied by ZHANG code in [12]. The XPATH query method is extended to and Twig query in [13]. The tree automata is also applied in

probabilistic XML data tree [14]. And the probabilistic XML data tree auto is introduced in [15]. A new uncertain XML algebra system is found in [16] and is applied to query the uncertain XML data.

These research work studies probabilistic XML algebra operation from different direction. So it also exploits the research direction for foundation native probabilistic XML algebra. The contributions in the paper includes:

1) Probabilistic XML element unit is defined as the basic unit of probabilistic XML document.

2) Probabilistic XML element tree is confirmed as the basic unit of probabilistic XML algebra operation. And the probabilistic XML element schema tree is drawn out by analyzing the path types.

3) The algebra operation symbol, definition and computation procedure is given. And the operation equivalence rules for query optimization are also given.

4) On the basis of functions of XQuery, it can query node probability and node set probability and list the possible world set of probabilistic XML tree.

5) The instance analysis shows that probabilistic XML data is an effective method as the format of probabilistic XML and the related probability query can be realized.

## II. BASIC PROBLEM

**Definition 1** (probability\_valid semis-Structured Information Unit) probability\_valid semis-Structured Information Unit  $\langle probability \rangle \sigma_1, \dots, \sigma_n \langle / probability \rangle$  is a valid probability semis-Structured Information Unit if for each  $\sigma_i \in \{\sigma_1, \dots, \sigma_n\}$  is the form  $\langle probvalue = k \rangle \phi \langle / prob \rangle$

where  $k \in [0,1]$  and  $\phi$  is element value.

Because probabilistic XML data is nested semis-structure XML data, so the probability of the pair  $\langle elementname, elementvalue \rangle$

in possible world set is  $P_i$ , the representing form is described as follows:

```
<elementname EN>
<dist>...
<ind value='pi'>
<dist>
...<poss value='pi'>elementvalue EV i</poss>...
</dist>
...</ind>
</dist>
</elementname EN>
```

This shows that the semantic complete probabilistic XML document fragment at least consists of two parts: an ordinary XML element and a probabilistic XML element unit. And it can be represented as probabilistic XML tree.

**Definition 2** (probabilistic XML data schema) probabilistic XML data schema is defined  $PDTD = (E, A, rule, att, str)$ , where  $E = \{e_1, e_2, \dots, e_n\}$  is a infinite element set,  $e_i (1 \leq i \leq n)$  is the element in the set of probabilistic XML data,  $A = \{a_1, a_2, \dots, a_m\}$  is the attribute set,  $a_j (1 \leq j \leq m)$  is the attribute in the set of

probabilistic XML data,  $rule(E)$  in the set of rules,  $att(\langle e_1, a_{11}, p_{11} \rangle, \dots, \langle e_1, a_{1n}, p_{1n} \rangle, \langle e_2, a_{21}, p_{21} \rangle, \dots, \langle e_2, a_{2n}, p_{2n} \rangle, \dots)$  is the corresponding relation of the element and its value,

where  $\sum_{i=1}^n p_{1i} = 1, \sum_{i=1}^m p_{2i} = 1, \dots$ ,  $str$  is the function from  $E$  to the power set of  $A$ .

**Definition 3** (Probabilistic XML data tree) Probabilistic XML data tree is defined six tuple  $PT = (N, E, r, \phi, V, prob)$ , where  $N$  is the finite set of node  $N = N_{ord} \cup N_{dist} \cup N_{poss}$ , where  $N_{ord}$  is the set of ordinary node,  $N_{dist}$  is the set of distribution node,  $N_{poss}$  is the set of possibility node.  $E \subseteq N \times N$  is the set of edges;  $r$  is the root of  $PT$ ;  $\phi: N_{ord} \rightarrow L$  is the label function of the ordinary node, where  $L$  is the name of element and attribute;  $\forall n \in N_{dist}, \phi(n) = dist$ ;  $\forall n \in N_{poss}, \phi(n) = 'poss'$ .  $V: N_l \rightarrow D$

is the data of leaf node, where  $N_l$  are the leaf nodes  $N_l \subset N_{ord} \subset N_{poss}$  and  $D$  is the data set.  $prob: N_{ord} \cup N_{poss} \rightarrow p$  is the probability value of the given node,  $0 \leq p \leq 1$ , the default value is 1 without assigning the probability value.

From the above definition of probabilistic XML data tree, the probability node is represented by the node dist that the children is the poss nodes i.e. the root node of possible world instance.

**Definition 4** (absolute path set) Absolute path is the node sequence from the root node  $r$  to any leaf node  $V(N_l)(V(N_l) \in D)$  in probabilistic XML tree  $PT$ .  $APATH_{PT}$  denotes the set of all absolute paths and  $|APATH_{PT}|$  denotes the absolute path number of  $APATH_{PT}$ .

**Definition 5** (extended path set) Extended path is a absolute path without any leaf node.  $\forall apath \in APATH_{PT}, epath = delete(last(apath))$ , where  $EPATH_{PT}$  denotes the set of all extended paths in probabilistic XML tree  $PT$  and  $|EPATH_{PT}|$  denotes the extended absolute path number of  $epath$ .

**Definition 6** (path consistent)  $\forall epath \in EPATH_{PT}$ , there is one and only one absolute path  $apath \in APATH_{PT}$ , the equation  $epath = delete(value(ind), value(poss), last(apath))$  comes into existence, then  $epath \sqcap apath$  which shows that the absolute path  $epath$  is consistent with  $apath$ .

**Definition 7** (probabilistic XML unit tree) probabilistic XML unit tree  $PT_{MIN}$  consists of two parts: one and only one absolute path without any distribution nodes and a set of consistent path of extended path.

**Definition 8** (probabilistic XML unit schema tree) probabilistic XML unit schema tree  $MT_{MIN}$  consists of two parts: one and only one absolute path without any distribution nodes and leaf node and a set of consistent path of extended path.

In general, probabilistic XML unit tree and probabilistic XML unit schema tree is shown in figure 1

and figure 2.

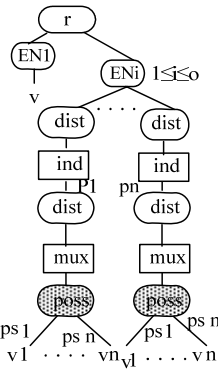


Figure 1. General probabilistic XML unit tree

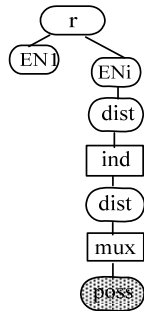


Figure 2. General probabilistic XML unit schema tree

**Example 1** The following probabilistic XML document fragment describes a probabilistic XML element unit. The probabilistic XML unit tree rank is shown in figure 3, where figure 3(a) is probabilistic XML tree T, figure 3(b) is probabilistic XML unit tree rank and figure 3(c) is probabilistic XML unit tree dept.

```

<T>
<teacher>
  <no>0001</no>
  <name>Wang Ming</name>
  <rank>
    <dist>
      <ind value='0.8'>
        <dist>
          <mux>
            <poss prob='0.6'> ass </poss>
            <poss prob='0.4'> ins </poss>
          </mux>
        </dist>
      </ind>
    </dist>
  </rank>
  .....
</teacher>
</T>

```

Here is a fragment of a probabilistic XML Schema DTD for such data.

```

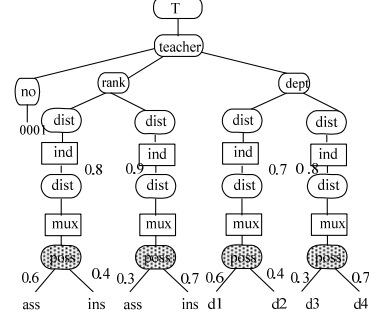
<!ELEMENT T (teacher)>
<!ELEMENT teacher (no, name, rank,dept)>
<!ELEMENT no (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT rank (dist+)>
<!ELEMENT dept (dist+)>

```

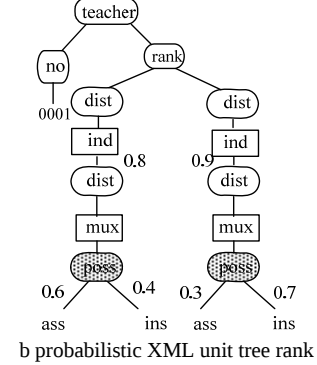
```

<!ELEMENT dist (ind | mux)>
<!ELEMENT ind (dist)>
<ATTLIST ind value CDATA #REQUIRED>
<!ELEMENT mux (poss+)>
<!ELEMENT poss (#PCDATA)>
<ATTLIST poss CDATA #REQUIRED>

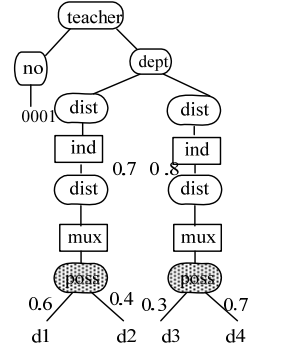
```



a probabilistic XML tree T



b probabilistic XML unit tree rank



c probabilistic XML unit tree dept

Figure 3. Teacher probabilistic XML unit tree

### III. ALGEBRA OPERATION RELATED TO A PROBABILISTIC XML UNIT TREE

#### A. Leaf Node Probability Project Operation

Suppose  $pt$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, then project operation result is the leaf node and its according probability, denoted  $\Pi_{PDTD}(pt)$ , as follows:

$$\Pi_{PDTD}(pt) = \{ \langle v(leaf)_i, p_i \rangle \mid \sum_{i=1}^n p_i = 1 \}$$

The computation procedure mainly consists of three steps. The first is to get the absolute path set  $APATH_{pt}$  through parsing the probabilistic XML unit tree  $pt$ . The second is to compute the absolute path probability by the probability chain principle. The third is to get the triple  $\langle leaf, probability \rangle$ .

```

algorithm project(  $pt$  )
input: absolute path set  $APATH_{pt}$  of  $pt$ 
output: <leaf,probability>
begin
  for  $i=1$  to  $|APATH_{pt}|$ 
    leaf[i]=last(  $APATH_{pt}[i]$  ); prob[i]=1;
    for  $j=1$  to length(  $APATH_{pt}[i]$  )
      if node[j]=ind then
        prob[i]=prob[i]×value(node[j]);
      elseif node[j]=poss then
        prob[i]=prob[i]×value(node[j]);
    return([leaf,prob]);
  end.

```

Theorem 1 Algorithm project () for the leaf node and its according probability in a probabilistic XML unit tree is correct and complete.

Example 2 Compute the leaf node and its probability in Figure 1.

The result is shown below:

$$\Pi_{PDTD}(pt) = \{ \langle 0001, 1 \rangle, \langle ass, 0.48 \rangle, \langle ins, 0.32 \rangle, \langle ass, 0.27 \rangle, \langle ins, 0.63 \rangle \}$$

#### B. Probability Threshold Interception Operation

Suppose  $pt$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, then probability threshold interception operation result is the leaf node and its according probability than given probability threshold, denoted  $S_{\alpha}(pt)$ , as follows:

$$S_{\alpha}(pt) = \{ \langle v(leaf)_i, p_i \rangle \mid p_i > \alpha \}$$

The computation procedure mainly consists of three steps. The first is to get the absolute path set  $APATH_{pt}$  through parsing the probabilistic XML unit tree  $pt$ . The second is to compute the absolute path probability by the probability chain principle. The third is to get the triple <leaf,probability> where the leaf node and its according probability than given probability threshold.

Example 3 compute the leaf node and its probability that the probability threshold is 0.4 in Figure 1(b).

$$S_{0.4}(pt) = \{ \langle ass, 0.48 \rangle, \langle ins, 0.63 \rangle \}$$

#### C. Generalization

The generalization procedure of given probabilistic XML unit tree is enumerating the element and its according probability in the possible world set.

```

algorithm generalization (  $pt$  )
input: absolute path set  $APATH_{pt}$  of  $pt$ 
output: <  $APATH_{pt}$ ,  $p$  >
begin
  list  $CPATH_{pt}$ ;
   $PPATH_{pt} = APATH_{pt} - CPATH_{pt}$ ;
  list  $EPATH_{pt}$ 
  for  $i=1$  to  $|PPATH_{pt}|$ 
     $P[i] = com\_prob(PPATH_{pt}[i])$ ;

```

```

 $PATH_{pt}[i] = delete(PPATH_{pt}[i], dist, ind, mux, poss)$ ;

```

```

 $A[i] = (P[i], PATH_{pt}[i]) \cup CPATH_{pt}$ ;

```

```

return(A);

```

end.

Theorem 2 Algorithm generalization () for enumerating the element and its according probability in the possible world set of probabilistic XML unit tree is correct and complete.

Example 4 Enumerate the element in possible world set of probabilistic XML unit tree in Figure 3(b). The result is shown in figure 4.

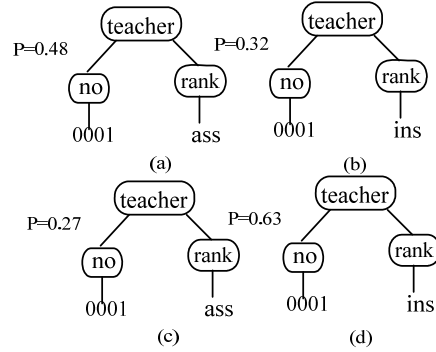


Figure 4. Element in possible world set of probabilistic XML unit tree

### IV. SET OPERATION

#### A. Union

Suppose  $pt_1$  and  $pt_2$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD. The union result of  $pt_1$  and  $pt_2$ , denoted  $pt_1 \cup pt_2$ , as follows:

$$pt = pt_1 \cup pt_2 = \begin{cases} APATH_{pt_1} \cup APATH_{pt_2} & r_{pt_1} = r_{pt_2} \\ APATH_{pt_1} \cup APATH_{pt_2} & r_{pt_1} \neq r_{pt_2} \end{cases}$$

Where  $r_{pt}$  is the added root of  $pt$ .

algorithm union (  $pt_1, pt_2$  )

input: absolute path set of  $pt_1$   $APATH_{pt_1}$ , absolute path

set of  $pt_2$   $APATH_{pt_2}$

output: absolute path set of  $pt$   $APATH_{pt}$

begin

```

 $r_{pt1} = first\_node(APATH_{pt_1})$ ;

```

```

%get the root of  $r_{pt_1}$ 

```

```

 $r_{pt2} = first\_node(APATH_{pt_2})$ ;

```

```

%get the root of  $r_{pt_2}$ 

```

```

if  $r_{pt1} = r_{pt2}$  then

```

```

   $APATH_{pt} = APATH_{pt_1} \cup APATH_{pt_2}$ ;

```

```

else

```

```

   $\overline{APATH_{pt_1}} = concat(r_{pt}, APATH_{pt_1})$ ;

```

```

  %add the new node as the root

```

```

   $\overline{APATH_{pt_2}} = concat(r_{pt}, APATH_{pt_2})$ ;

```

```

  %add the new node as the root

```

```

   $APATH_{pt} = \overline{APATH_{pt_1}} \cup \overline{APATH_{pt_2}}$ ;

```

```

return(  $APATH_{pt}$  )

```

end.

Theorem 3 Algorithm union () for merging the two

given probabilistic XML unit tree is correct and complete.

### B. Intersection

Suppose  $pt_1$  and  $pt_2$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, The intersection result of  $pt_1$  and  $pt_2$ , denoted  $pt = pt_1 \cap pt_2$ , as follows:

$$APATH_{pt} = \begin{cases} APATH_{pt_1} \cap APATH_{pt_2} = \{capath\}_{pt_1} \cap \{capath\}_{pt_2} & r_{pt_1} = r_{pt_2} \\ \phi & r_{pt_1} \neq r_{pt_2} \end{cases}$$

algorithm intersection ( $pt_1, pt_2$ )

input: absolute path set of  $pt_1$   $APATH_{pt_1}$ ,

absolute path set of  $pt_2$   $APATH_{pt_2}$

output: absolute path set of  $pt$   $APATH_{pt}$

begin

$r_{pt1} = \text{first\_node}(APATH_{pt_1});$

%get the root of  $r_{pt_1}$

$r_{pt2} = \text{first\_node}(APATH_{pt_2});$

%get the root of  $r_{pt_2}$

if  $r_{pt1} = r_{pt2}$  then

$APATH_{pt} = APATH_{pt_1} \cap APATH_{pt_2};$   
 $= \{capath\}_{pt_1} \cap \{capath\}_{pt_2};$

else

$APATH_{pt} = \phi;$

return( $APATH_{pt}$ )

end.

Theorem 4 Algorithm intersection () for computing the difference of the two given probabilistic XML unit tree is correct and complete.

### C. Difference

Suppose  $pt_1$  and  $pt_2$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, The intersection result of  $pt_1$  and  $pt_2$ , denoted  $pt = pt_1 - pt_2$ , as follows:

$pt = pt_1 - pt_2 = \{apath \mid apath \in APATH_{pt_1} \text{ and } apath \notin APATH_{pt_2}\}$

algorithm difference ( $pt_1, pt_2$ )

input: absolute path set of  $pt_1$   $APATH_{pt_1}$ ,

absolute path set of  $pt_2$   $APATH_{pt_2}$

output: absolute path set of  $pt$   $APATH_{pt}$

begin

$r_{pt1} = \text{first\_node}(APATH_{pt_1});$

%get the root of  $r_{pt_1}$

$r_{pt2} = \text{first\_node}(APATH_{pt_2});$

%get the root of  $r_{pt_2}$

if  $r_{pt1} = r_{pt2}$  then

$APATH_{pt} = APATH_{pt_1} - APATH_{pt_1} \cap APATH_{pt_2};$

else

$APATH_{pt} = APATH_{pt_1};$

return( $APATH_{pt}$ )

end.

Theorem 5 Algorithm difference () for computing the difference of the two given probabilistic XML unit tree is correct and complete.

Example 4 Compute the difference of probabilistic XML unit trees in figure 3(b) and (c). The result is shown in figure 5.

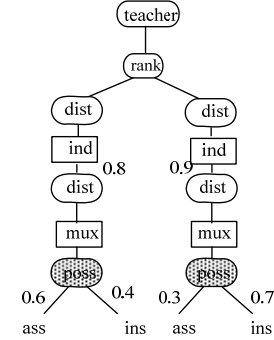


Figure 5. Difference result of probabilistic XML unit trees

### D. Cartesian product

Suppose  $pt_1$  and  $pt_2$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, The intersection result of  $pt_1$  and  $pt_2$ , denoted  $pt_1 \times pt_2$ , as follows:

$pt = pt_1 \times pt_2$

Example 5 Compute the Cartesian product of probabilistic XML unit trees in figure 3(b) and (c). The result is shown in figure 6.

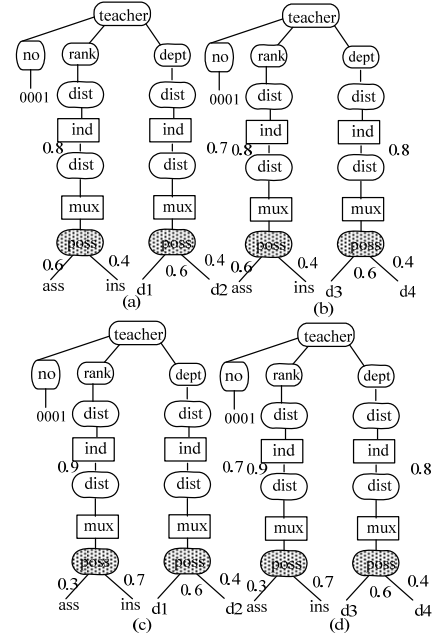


Figure 6. Product tree set of rank and dept

## V. JOIN OPERATION

Join operation of two probabilistic XML unit tree is the unavoidable operation. Suppose  $pt_1$  and  $pt_2$  be a probabilistic XML unit tree conforming to probabilistic XML unit schema PDTD, The join result of  $pt_1$  and  $pt_2$ , denoted  $pt_1 \bowtie_v pt_2$ , as follows:

```

 $pt = pt_1 \triangleright \triangleleft_v pt_2$ 
algorithm join (  $pt_1, pt_2$  )
input: absolute path set of  $pt_1$   $APATH_{pt_1}$ ,
      absolute path set of  $pt_2$   $APATH_{pt_2}$ 
output: absolute path set of  $pt$   $APATH_{pt}$ 
begin
  for i=1 to  $|APATH_{pt_1}|$ 
    {  $APATH_{pt_1}[i]$ =delete(root);
      %delete the root node
      EM1[i]=root(  $APATH_{pt_1}[i]$  ),
      EV1[i]=leaf(  $APATH_{pt_1}[i]$  );}
  %compute the triple<elementname,elementvalue>
  for i=1 to  $|APATH_{pt_2}|$ 
    {  $APATH_{pt_2}[i]$ =delete(root);
      %delete the root node
      EM2[i]=root(  $APATH_{pt_2}[i]$  ),
      EV2[i]=leaf(  $APATH_{pt_2}[i]$  );}
  for i=1 to |EM1|{
    for j=1 to |EM2|{
      if EM1[i]= EM2[j] then {
        if EV1[i]= EV2[j] then{
          locate EM1[i] in  $APATH_{pt_1}[i]$ ;
          locate EM2[j] in  $APATH_{pt_2}[j]$ ;
          apath[i]=  $APATH_{pt_1}[i]$  concat
                      $APATH_{pt_2}[j]$ ;
          %join operation of two absolute
          path %with same element
          root(  $pt_2$  )=EM1[i];
          %  $pt_2$  become the subtree of
           $pt_1$ , the %root of  $pt_2$  the common node
          prefix=  $APATH_{pt_2}[i]$ 
          %the absolute path in EM1become
          the %prefix absolute path of  $pt_2$  } } } }
  for i=1 to  $|APATH_{pt_2}|$  {
     $APATH_{pt_2}[i]$ =prefix /  $APATH_{pt_2}[i]$ ; }
  %compute the new absolute path of subtree
  in %  $pt_2$ 
   $APATH_{pt} = APATH_{pt_1} \cup APATH_{pt_2}$ ;
  return(  $APATH_{pt}$  )
end.

```

Theorem 6 Algorithm join () for computing the difference of the two given probabilistic XML unit tree is correct and complete.

## VI. PROBABILISTIC XML QUERY EXPRESSION REALIZATION

XQuery is the standard language for querying XML documents using structural and content restrictions. It is also flexible enough to query a broad spectrum of XML information sources, including both databases and documents.

In order to query the probabilistic XML document, the expression of XQuery can be extended. The application of basic operation is introduced, like the following ones: Example 6 If  $n0=0001$  in T.xml, how to query the probability with the condition rank="ass". The extended XQuery sentence is shown below.

```

FOR $a IN document("T.xml")//T/rank
  $b IN $a/n0
  WHERE $a/rank ="ass" and $b /n0=0001 RETURN
<rank>($a,prob )</rank>

```

the corresponding basic exoression is shown below:

$\Pi_{PDTD}(pt)$

the node sequence of its path :

T/teacher/dist/ind[0.8]/dist/mux/poss[0.6]/'ass'

T/teacher/dist/ind[0.9]/dist/mux/poss[0.3]/'ass'

the probability computation formula is shown below:

$$\begin{aligned}
 P(rank, rank = 'ass')_2 &= \prod_{k=1}^i p(node_k) \\
 &= p(node_1) \times p(node_2) \times \dots \times p(node_i) \\
 &= 0.8 \times 0.6 = 0.48
 \end{aligned}$$

$$\begin{aligned}
 P(rank, rank = 'ass')_2 &= \prod_{k=1}^i p(node_k) \\
 &= p(node_1) \times p(node_2) \times \dots \times p(node_i) \\
 &= 0.9 \times 0.3 = 0.27
 \end{aligned}$$

the result:

$$\Pi_{PDTD}(pt | rank = 'ass') = \{0.48, 0.27\}$$

Example 7 If  $n0=0001$  in T.xml, how to query the probability with the condition rank="ass" and dept="d1" or "d2". The extended XQuery sentence is :

```

FOR $a IN document("T.xml")//T/rank and dept
  $b IN $a/n0
  WHERE $a/rank ="ass" and $b /n0=0001 and
    $c/dept ="d1" or $c/dept ="d2" RETURN
<rank>($a, $c, prob )</rank>

```

the corresponding basic expression is shown below:

$$\Pi_{PDTD}(pt_1 \cup pt_2) = \Pi_{PDTD1}(pt_1) \cup \Pi_{PDTD2}(pt_2)$$

$\Pi_{PDTD1}(pt_1)$  is firstly computed .

Example 8 simplify probabilistic XML document T.xml. The simplification of probabilistic XML document T.xml is to enumerate the elements of possible world set in probabilistic XML data tree. The extended XQuery sentence is :

```

FOR $a IN document("T.xml")//T
  WHERE $b /n0 and $a/rank and $c/dept RETURN
<prob>p</prob>
<no>$a </no>
<rank>$b</rank>
<deptk>$c</dept>

```

The corresponding basic expression is shown below:

$$pws_{rank} = \{ \langle rank_i, p_i \rangle | 0 \leq p_i \leq 1 \}$$

$$pws_{dept} = \{ \langle dept_i, p_i \rangle | 0 \leq p_i \leq 1 \}$$

$$t = pws_{rank} \times pws_{dept} = pws_T(rank \times dept)$$

The possible world set of dept is shown in figure 7. The possible world set of T is shown in figure 8.

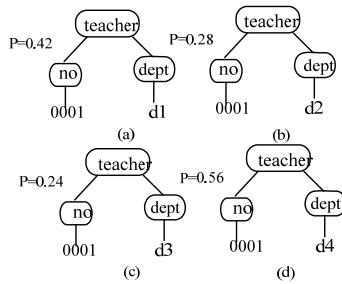


Figure 7. Possible world set of dept

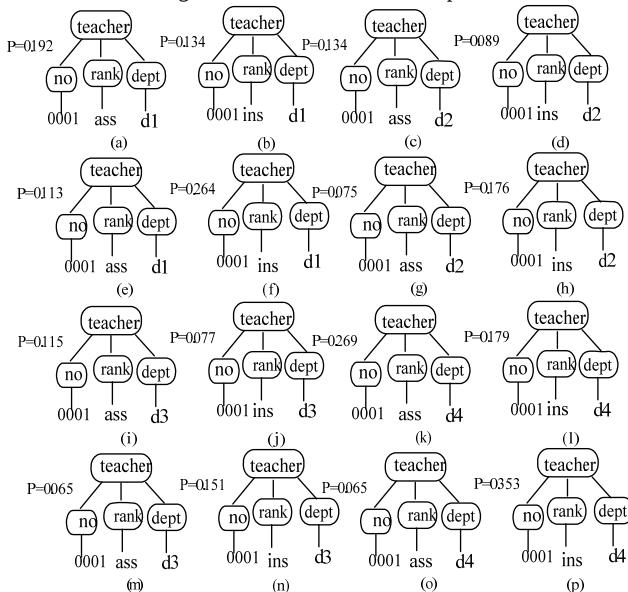


Figure 8. Possible world set of T

## VII. CONCLUSION

Probabilistic data can be represented as probabilistic XML element unit in the direction of XML data and also be represented as probabilistic XML unit tree. So the basic operation set on probabilistic XML unit tree is defined. Because probabilistic XML data can be described nested probabilistic XML element unit, the basic operation set provides the precondition and basis. The instance analysis shows that the basic operation set of probabilistic XML unit tree is correct and effective.

Through the validity and maturation of query response, the query results of probabilistic XML document is involved with probability. Listing all query results can enhance the complexity of query algorithm, so how to accept or reject the query results will be the continued research content.

## ACKNOWLEDGMENT

The authors would like to thank Hei longjiang National Science Grant No.F200925, for funding this research. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

- [1] <http://www.w3.org/TR/>
- [2] H.V. Jagadish, L.V.S. Lakshmanan, D. Srivastava and K.Thompson. TAX: A Tree Algebra for XML[C]. In Proc.

- DBPL Conf. pp. 149-164. Rome, Italy, 2001
- [3] L. Galanis, et al., Following the paths of XML Data: An algebraic framework for XML query evaluation[R]. Technical report, University of Wisconsin, 2001
- [4] F. Frasca, Geert-Jan Houben, and C. Pau. XAL: an Algebra for XML Query Optimization[C]. ACM International Conference Proceeding Series; Vol. 18 , Proceedings of the 13th Australasian database conference - Volume 5,2002,p: 49 - 56
- [5] Niernman and H. V. Jagadish. ProTDB: Probabilistic data in XML[C]. Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002
- [6] Te Li, Qihong Shao, and Yi Chen. PEPX: A Query-Friendly Probabilistic XML Database[C]. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management. CIKM '06, ACM Press, 2006: 848-849
- [7] Wenzhong Zhao, Alex Dekhtyar, and Judy Goldsmith.A Framework for Management of Semistructured Probabilistic Data[J].Journal of Intelligent Information Systems.2005,25(3): 293-332
- [8] Alex Dekhtyar,Krol Kevin Mathias, Praveen Gutti. Structured Queries for Semistructured Probabilistic Data[C].TDM'2006
- [9] Alex Dekhtyar, Judy Goldsmith, Sean R. Hawkes. Semistructured Probabilistic Databases[C]. In Proc. Statistical and Scienti Database Management Systems.2001,p:36-45
- [10] Hubert Chan, Evgeny Kharlamov, Werner Nutt, Pierre Senellar. Aggregate queries for discrete and continuous probabilistic XML[C] , Proceedings of the 13th International Conference on Database Theory,Mar. 2010:50-61
- [11] Irma Veldman, Ander Keijzer, Maurice Keulen. Compression of Probabilistic XML Documents[C], Proceedings of the 3rd International Conference on Scalable Uncertainty Management, Aug. 2009: 255-267
- [12] Yawen Li, Guoren Wang, Junchang Xin, Ende Zhang, Zeling Qiu. Holistically Twig Matching in Probabilistic XML[C] , Proceedings of the 2009 IEEE International Conference on Data Engineering ,Mar. 2009 :1649-1656
- [13] Lijun Chang, Jeffrey Xu Yu, Lu Qin. Query ranking in probabilistic XML data[C], Proceedings of the 12th International Conference on Extending Database Technology, Mar. 2009 : 156-167
- [14] Sara Cohen, Benny Kimelfeld, Yehoshua Sagiv. Running tree automata on probabilistic XML[C], Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems ,Jun. 2009 : 227-236
- [15] Ma, Z.M., Liu, Jian; Yan, Li.Fuzzy data modeling and algebraic operations in XML[J].International Journal of Intelligent Systems.v 25, n 9, September 2010 ,v (25), n (9), p: 925-947
- [16] Stelios Paparizos and H.V. Jagadish. The Importance of Algebra for XML Query Processing[C], EDBT 2006 Workshops, LNCS 4254,2006,pp:126-135

**Jianwei Wang**, born in 1973, received her bachelor degree in the School of Computer Science and Technology of Harbin University of Science and Technology in 1996. Now she is associate professor in the College of Information and Computer Engineering of Northeast Forestry University. Her main research interests include uncertain data management, signal processing ,pattern recognition.

**Zhongxiao Hao**, born in 1940. He received his bachelor degree in the Department of Mathematics from Heilongjiang University in 1964. Now he is professor and Ph. D. supervisor in the Department of Computer Science and Technology, Harbin Institute of Technology and Harbin University of Science and Technology. His main research interests include relational database, null database, acyclic database, active database and temporal database.