IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 13, 2019, accepted February 5, 2019, date of publication February 8, 2019, date of current version March 4, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2898228*

# An Algebra for Fuzzy Spatiotemporal Data in XML

**LUYI BAI AND LIN ZHU**
School of Computer and Communication Engineering, Northeastern University (Qinhuangdao), Qinhuangdao 066004, China

Corresponding author: Luyi Bai (baily@neuq.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402087, in part by the Natural Science Foundation of Hebei Province under Grant F2019501030, and in part by the Fundamental Research Funds for the Central Universities under Grant N172304026.

**ABSTRACT** A formal algebra is essential for applying standard database-style query optimization to XML queries. We develop such algebra for manipulating fuzzy spatiotemporal XML data. We also introduce several formal characterizations of fuzzy spatiotemporal algebraic equivalences and study how XQuery expressions can be transformed into algebraic expressions. It shows that the algebra can lay a firm foundation for managing fuzzy spatiotemporal XML data.

**INDEX TERMS** Algebra, fuzzy spatiotemporal data, XML.

## I. INTRODUCTION

The majority of spatiotemporal applications [1], [2] require efficiently managing (fuzzy) spatiotemporal data both from academia and industry [3]–[6]. The flexibility of a semi-structured fuzzy spatiotemporal data model [7], coming from general data model in XML [8], [9], spatiotemporal data model in XML [10], and fuzzy data model in XML [11], is the core of managing fuzzy spatiotemporal data. On the other hand, since XML suggests itself as medium for integrating and exchanging data from different sources, it becomes a natural model choice. Since fuzzy theory [3], [12] and fuzzy mathematical programs [13] have been proposed, studies on fuzzy spatiotemporal data have only recently started and still merit further attention.

As evidenced by the successful relational technology [14], a formal algebra is absolutely essential for applying standard database-style query optimization to XML queries. Due to its significance, researches on this issue have been extensively proposed [15]–[18]. However, they do not provide a formal algebra that can support the fuzzy spatiotemporal XML queries, although their efforts deal with one of the aspects of fuzzy spatiotemporal elements. As a result, we need a valid formal algebra for the XML queries that can serve as a well understood and order-sensitive intermediate representation.

Accordingly, the motivation of the paper is trying to develop algebra for manipulating fuzzy spatiotemporal XML data. The paper makes the following main contributions:

- We present a general algebraic framework for supporting fuzzy spatiotemporal XML queries, containing set operations and other useful operations.
- Several formal characterizations of the fuzzy spatiotemporal algebraic equivalences are investigated, which lay a firm foundation for optimizing fuzzy spatiotemporal queries in XML.
- We study how XQuery expressions can be transformed into algebraic expressions by using algebra to capture fuzzy spatiotemporal queries expressed in XQuery.

The rest of the paper is organized as follows. Section II presents the related work. Representation of fuzzy spatiotemporal data in XML is introduced in Section III. After investigating several algebraic operations to apply fuzzy spatiotemporal XML queries in Section IV, we study algebraic equivalences in Section V. In Section VI, we present how XQuery expressions can be transformed into algebraic expressions. Experiments are evaluated in Section VII. Section VIII gives a comparative study and Section IX concludes the paper.

## II. RELATED WORK

Concerning on spatiotemporal data in XML, there are efforts to combine spatial and temporal properties into one framework, and analyze spatiotemporal data in XML. Huang *et al.* [10] propose an approach to represent and query spatiotemporal data in XML. Liu and Wan [19] propose a feature-based spatiotemporal data model and use the Native XML Database to store the spatiotemporal data. The work of Franceschet *et al.* [20] describe a translation algorithm that maps spatiotemporal conceptual schemas into XML schemas, and propose a framework that allows one to validate XML documents containing spatiotemporal information with respect to spatiotemporal conceptual schemas. Furthermore, concerning on fuzzy spatiotemporal data in XML,

The associate editor coordinating the review of this manuscript and approving it for publication was waleed Alsabhan.

Bai *et al.* [21] propose a fuzzy spatiotemporal XML data model, but not support algebraic operations.

The algebraic approaches have been proven to be an effective way for queries in XML database systems [22]. With the emergence of XQuery designed by the W3C as a standard query language for XML, efforts are made to apply algebra to XML queries [16], [23]–[25]. Fernandez *et al.* [24] propose an algebra for XML query using regular-expression types, which has been submitted to the W3C XML Query Working Group. Buratti and Montesi [16] propose a novel algebra for XML based on a simple data model in which trees and forests are the counterpart of the relational tuples and relations. The work of Che and Sojitrawala [23] presents a new algebra called DUMAX designed for XML and XML queries, which is introduced to help fuse node-based features and tree-based features (both are essential for XML) and to achieve accelerated execution of XML queries in large XML databases. Wang *et al.* [25] present a general approach for supporting order-sensitive XQuery-to-SQL translation that works irrespective of the chosen XML-to-relational data mapping and the selected order-encoding method. Unfortunately, the proposed algebraic approaches do not support (fuzzy) spatiotemporal data, although there are other XML algebraic approaches, such as probabilistic XML algebra [26] and fuzzy XML algebra [8]. Nevertheless, although the above researches do not straight forwardly deal with fuzzy spatiotemporal data in XML, their efforts play a fundamental role in algebraic operations on fuzzy spatiotemporal data in XML.

## III. REPRESENTATION OF FUZZY SPATIOTEMPORAL DATA IN XML

The basic data structure of fuzzy spatiotemporal XML is the data tree. In order to manage fuzzy spatiotemporal XML data, fuzzy spatiotemporal XML data tree should be employed. We start by introducing some simple concepts [21].

*Definition 1:* Let $V$ be a finite set of vertices, $E \in V \times V$ be a set of edges, and $\ell$: $E \to \Gamma$ be a mapping from edges to a set $\Gamma$ of strings called labels. The triple $G = (V, E, \ell)$ is an edge-labeled directed data tree.

Definition 1 is defined for edge-labeled directed data tree, and inspired by the tree patterns of XML but ignores a number of XML features such as data types, ordering, and the distinction between elements and attributes. Because ordering is one of the most import elements for managing fuzzy spatiotemporal XML data, partial order is introduced to describe the ordering in XML.

*Definition 2:* A relation $R$ is a partial order (denoted as "$\leq$") on a set $S$ if it has the following properties:

- Reflexivity: $\forall \alpha \in S, (\alpha, \alpha) \in R$.
- Antisymmetry: if $(\alpha, \beta) \in R$ and $(\beta, \alpha) \in R$, then $\alpha = \beta$.
- Transitivity: if $(\alpha, \beta) \in R$ and $(\beta, \gamma) \in R$, then $(\alpha, \gamma) \in R$.

On the basis of the definitions above, we turn to develop the fuzzy spatiotemporal XML data model. After defining fuzzy spatiotemporal XML data and fuzzy spatiotemporal XML data tree, correspondences between them are investigated. In addition, structure of the fuzzy spatiotemporal XML data model is formally defined. According to the nature of spatiotemporal data, we firstly give the definition of fuzzy spatiotemporal XML data as followings.

*Definition 3:* (Fuzzy spatiotemporal XML data). Fuzzy spatiotemporal data *FSP* is a 5-tuple, *FSP* = (*OID*, *ATTR*, *FP*, *FM*, *FT*), where

- *OID* is the changing history of the spatiotemporal data.
- *ATTR* is fuzzy attributes of spatiotemporal data.
- *FP* is fuzzy position of spatiotemporal data.
- *FM* is fuzzy motion of spatiotemporal data.
- *FT* is fuzzy time of spatiotemporal data.

Since XML data are structured, XML can represent fuzzy information naturally. In the case of XML document, we may have membership degrees associated with elements and possibility distributions associated with attribute values of elements. In succession, we investigate how to represent fuzzy XML spatiotemporal data in the XML document modified and extended by [8].

*Definition 4 (Fuzzy Spatiotemporal XML Data Tree):* For the fuzzy XML spatiotemporal document, we have $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$, including:

- $V = \{V_1, V_2, \ldots, V_n\}$ is a finite set of vertices.
- $\psi \subset \{(V_i, V_j) | V_i, V_j \in V\}$, $(V, \psi)$ is a directed tree.
- $T \in \psi \subset \{(V_i, V_j) | V_i, V_j \in V\}$ specifies time of $V_j$.
- $\varpi$ is the nesting depth of $V$ in the spatiotemporal data tree of the document. The $\varpi$ of root node is 1, and the $\varpi$ of each following level adds 1.
- $\wp$ preserves the order information in the fuzzy spatiotemporal XML data tree. It can be generated by counting word numbers from root of the fuzzy data tree until the start and end of the element, respectively. Here, we use preorder traversal.
- $\tau$ is a set of labels.
- For the node $v \in V$ and the label $\nabla \in \tau$, $\xi(v, \nabla)$ specifies $v$ exists with label $\nabla$.
- $\delta \in \tau$ is a mapping from the element node $v \in V$ to membership degree functions. It defines the possibility of the element node exists.
- $\pi \in \tau$ is a mapping from the attribute node $v \in V$ to possibility distribution functions. It defines the possibility of the attribute node exists.

*Definition 5:* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ and $f = (V', \psi', T', \varpi', \wp', \tau', \xi', \delta', \pi')$ are two fuzzy spatiotemporal XML data trees. $f$ is a subtree of $F$, written $f \propto F$, when

- $V' \subseteq V$, $\psi' = \psi \cap V' \times V'$.
- if $(V_i, V_j) \in \psi$ and $V_i \in V'$, then $V_j \in V'$.
- $T' \subseteq T$.
- if $V_i \in V$, $V_i' \in V'$, and $\varpi(V_i) = max\{.\varpi_1, \varpi_2, \ldots, \varpi_n\}$, $\varpi(V_i') = max\{\varpi_1', \varpi_2', \ldots \varpi_n'\}$, then $\varpi(V_i') \leq \varpi(V_i)$.
- $\wp'$, $\xi'$ and $\tau'$ indicate the restriction of $\wp$, $\xi$ and $\tau$ to the nodes in $V'$, respectively.
- $\delta' \subseteq \delta$ and $\pi' \subseteq \pi$.

| | | |
|---|---|---|
| <cloud density $T_s$ = "$t_3$" $T_e$ = "$t_6$"><br><Dist type = "disjunctive"><br><Val Poss = 0.85>thick</Val><br><Val Poss = 0.75>thin</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_4$" $T_e$ = "$t_8$"><br><Dist type = "disjunctive"><br><Val Poss = 0.65>thick</Val><br><Val Poss = 0.85>moderate</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_3$" $T_e$ = "$t_8$"><br><Dist type = "disjunctive"><br><Val Poss = 0.85>thick</Val><br><Val Poss = 0.85>moderate</Val><br><Val Poss = 0.75>thin</Val><br><Dist></cloud density><br></cloud density> |
| A | B | A ∪ B |

**FIGURE 1.** Fuzzy union operation on two fuzzy spatiotemporal XML fragments.

According to the above definitions, a fuzzy spatiotemporal XML data tree can be composed of multiple fuzzy spatiotemporal XML data subtrees, and multiple fuzzy spatiotemporal XML data trees can comprise a fuzzy spatiotemporal XML data forest.

*Theorem 1:* Given any set of fuzzy spatiotemporal XML data trees $F = \{f_1, f_2, \ldots, f_n\}$, there exists a corresponding fuzzy spatiotemporal XML data forest matching with $F$.

*Theorem 2:* $\lambda$ is a fuzzy partial order relationship on a fuzzy spatiotemporal XML data tree $F$, there exists a fuzzy transitivity relationship $\lambda'$ that satisfies

- $\lambda \in \lambda'$.
- $\lambda'$ is a partial order relationship stems from $\lambda$, which keeps the reflexivity and antisymmetry of $\lambda$.

*Proof:* Firstly, we construct $\eta = \{\vartheta \mid \lambda \subseteq \vartheta\}$. It is obvious that $(\eta, \subseteq)$ is a partial order. Assuming $\vartheta = \bigcup_{i \in I} \vartheta_i$, and $\{\vartheta_i\}_{i\in I} \subseteq \eta$ are an ordering subtrees of $F$. For $\forall i \in I$, there is $\lambda \subseteq \vartheta_i$. According to the definition of partial order, $\vartheta_i(f, f) = 1$. For each $f \in F$, we have $\vartheta(f, f) = \bigvee_{i\in I} \vartheta_i(f, f) = 1$. Thus $\vartheta$ has reflexivity.

Secondly, for every $\vartheta_i \subseteq \vartheta_j$, where $i, j \in I$, $\forall x, y \in F$, $F$ has antisymmetry, denoted as $\varpi$. Then we have $\vartheta(x, y)\varpi\vartheta(y, x) = [\bigvee_{i\in I} \vartheta_i(x, y)]\varpi[\bigvee_{j\in I} \vartheta_j(y, x)] = \bigvee_{i,j\in I}[\vartheta_i(x, y)\varpi\vartheta_j(y, x)]$. On the other hand, $\vartheta_i(x, y)\varpi\vartheta_j(y, x) \leq \vartheta_j(x, y)\varpi\, \vartheta_j(y, x) = 0$. Accordingly, $\bigvee_{i,j\in I}[\vartheta_i(x, y)\varpi\vartheta_j(y, x)] = 0$. Thus $\vartheta(x, y)\varpi\vartheta(y, x) = 0$, $\vartheta$ has antisymmetry $\varpi$.

Finally, we turn to the proof of transitivity. for every $\vartheta_i \subseteq \vartheta_j$, where $i, j \in I$, $\forall x, y, z \in F$, $F$ has transitivity $\Im$. Then $\vartheta(x, y) \Im \vartheta(y, z) = [\bigvee_{i\in I} \vartheta_i(x, y)]\Im[\bigvee_{j\in I} \vartheta_j(y, z)] = \bigvee_{i,j\in I}[\vartheta_i(x, y)\Im\vartheta_j(y, z)]$ $\vartheta_i(x, y) \Im \vartheta_j(y, z) \leq \vartheta_j(x, y) \Im \vartheta_j(y, z) \leq \vartheta_j(x, z) \leq \vartheta(x, z)$. Thus $\vartheta(x, y) \Im \vartheta(y, z) \leq \vartheta(x, z)$, $\vartheta$ has transitivity $\Im$.

Accordingly, there exists maximal element $\lambda'$ in $\eta$ that satisfies $\lambda \in \lambda'$ $\lambda'$ is a partial order relationship that stems from $\lambda$, keeping the reflexivity and antisymmetry of $\lambda$.

As XML document is a labeled ordered rooted tree, we regard a fuzzy spatiotemporal data as a structured tree for metadata. As a result, operations between fuzzy spatiotemporal data are actually operations between trees. Consequently, fuzzy spatiotemporal XML data trees should be considered as isomorphic. □

*Definition 6:* Let fuzzy spatiotemporal XML data trees $f_1 = (V_1, \psi_1, T_1, \varpi_1, \wp_1, \tau_1, \xi_1, \delta_1, \pi_1)$ and $f_2 = (V_2, \psi_2, T_2, \varpi_2, \wp_2, \tau_2, \xi_2, \delta_2, \pi_2)$ be the subtrees of $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$. Then $f_1$ and $f_2$ are isomorphic (recorded $f_1 \cong f_2$), when

- $V_1 \cup V_2 \subseteq V$, $\psi_1 \cup \psi_2 \subseteq \psi$, $T_1 \cup T_2 \subseteq T$, $\tau_1 \cup \tau_2 \subseteq \tau$.
- $\varpi_1 = \varpi_2 \leq \varpi$.
- There is a one-to-one mapping, $\Im\xi$: $\xi_1 \rightarrow \xi_2$, which makes $\forall \Im\xi(\xi_1) = \xi_2$.

*Theorem 3:* Fuzzy data tree $F$ and its subtree $f$ are isomorphic.

The proof of this theorem follows the analysis of fuzzy spatiotemporal XML data tree and the corresponding definitions. It is quite straightforward.

## IV. ALGEBRAIC OPERATORS

This section presents issues on how to formally design fuzzy spatiotemporal XML algebra. All operators take collections of fuzzy spatiotemporal XML data trees as input and produce a collection of fuzzy spatiotemporal XML data trees as output.

### A. SET OPERATIONS

In this subsection, we will provide four standard fuzzy set operations on fuzzy spatiotemporal XML data, which are fuzzy union (∪), fuzzy intersection (∩), fuzzy difference (−), and Cartesian product (∞).$\omega(a)$ and $\varphi(a)$ are used to represent the pattern and instance on $a$ (where $a$ is an algebraic expression).

*Definition 7 (Fuzzy Union):* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ and $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees. When $f_i$ and $f_j$ are isomorphic, the fuzzy union (∪) can be defined as follows:

- $\omega(f_i \cup f_j) = (V_{i\cup j}, \psi_{i\cup j}, T_{i\cup j}, \varpi_{i\cup j}, \wp_{i\cup j}, \tau_{i\cup j}, \xi_{i\cup j}, \delta_{i\cup j}, \pi_{i\cup j})$.
- $\varphi(f_i \cup f_j) \in \{e \mid e \in \varphi(f_i) \text{ or } e \in \varphi(f_j)\}$.

Fig. 1 shows a fuzzy union operation on two fuzzy spatiotemporal XML data trees $A$ and $B$, where $T_s$ and $T_e$ represent the starting and ending time points, respectively. According to Definition 7, the fuzzy spatiotemporal XML data trees $A$ and $B$ are isomorphic. On the basis of fuzzy set theory, we have $\delta_r = \text{Max}(\delta_i, \delta_j)$. Here, $\delta_i$ and $\delta_j$ are membership degrees of $A$ and $B$; $\delta_r$ is the membership degree of fuzzy union result; Max $(\delta_i, \delta_j)$ returns the maximize value of $\delta_i$ and $\delta_j$. By default, $\delta_r = 1$. In Fig. 1, the time interval $[t_3, t_8]$ in the fuzzy union result $A \cup B$ comes from the union of time interval $[t_3, t_6]$ and $[t_4, t_8]$. On the other hand, the "Poss" value of "thick" 0.85 comes from Max (0.85 ∪0.65) = 0.85; the "Poss" value of "moderate" 0.85 comes

| | | |
|---|---|---|
| <cloud density $T_s$ = "$t_3$" $T_e$ = "$t_6$"><br><Dist type = "disjunctive"><br><Val Poss = 0.85>thick</Val><br><Val Poss = 0.75>thin</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_4$" $T_e$ = "$t_8$"><br><Dist type = "disjunctive"><br><Val Poss = 0.65>thick</Val><br><Val Poss = 0.85>moderate</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_4$" $T_e$ = "$t_6$"><br><Dist type = "disjunctive"><br><Val Poss = 0.65>thick</Val><br><Dist><br></cloud density> |
| A | B | A ∩ B |

**FIGURE 2.** Fuzzy intersection operation on two fuzzy spatiotemporal XML fragments.

| | | |
|---|---|---|
| <cloud density $T_s$ = "$t_3$" $T_e$ = "$t_6$"><br><Dist type = "disjunctive"><br><Val Poss = 0.85>thick</Val><br><Val Poss = 0.75>thin</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_4$" $T_e$ = "$t_8$"><br><Dist type = "disjunctive"><br><Val Poss = 0.65>thick</Val><br><Val Poss = 0.85>moderate</Val><br><Dist><br></cloud density> | <cloud density $T_s$ = "$t_3$" $T_e$ = "$t_4$"><br><Dist type = "disjunctive"><br><Val Poss = 0.35>thick</Val><br><Val Poss = 0.75>thin</Val><br><Dist><br></cloud density> |
| A | B | A – B |

**FIGURE 3.** Fuzzy difference operation on two fuzzy spatiotemporal XML fragments.

from Max (0 ∪ 0.85) = 0.85; the "Poss" value of "thin" 0.75 comes from Max (0.75 ∪ 0) = 0.75.

*Definition 8 (Fuzzy Intersection):* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ and $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees. When $f_i$ and $f_j$ are isomorphic, the fuzzy intersection (∩) can be defined as follows:

- $\omega(f_i \cap f_j) = (V_{i\cap j}, \psi_{i\cap j}, T_{i\cap j}, \varpi_{i\cap j}, \wp_{i\cap j}, \tau_{i\cap j}, \xi_{i\cap j}, \delta_{i\cap j}, \pi_{i\cap j})$.
- $\varphi(f_i \cap f_j) \in \{e|e \in \varphi(f_i) \text{ and } e \in \varphi(f_j)\}$.

Fig. 2 represents a fuzzy intersection operation on two fuzzy spatiotemporal XML data trees *A* and *B* based on the membership degrees and possibility distribution. According to fuzzy set theory, we have $\delta_r = \text{Min}(\delta_i, \delta_j)$. Here, $\delta_i$ and $\delta_j$ are membership degrees of *A* and *B*; $\delta_r$ is the membership degree of fuzzy intersection result; Min $(\delta_i, \delta_j)$ returns the minimize value of $\delta_i$ and $\delta_j$. The fuzzy spatiotemporal object does not exist and can be omitted if $\delta_r = 0$. In Fig. 2, the time interval $[t_4, t_6]$ in the fuzzy intersection result $A \cap B$ comes from the intersection of time interval $[t_3, t_6]$ and $[t_4, t_8]$. On the other hand, the "Poss" value of "thick" 0.65 comes from Min (0.85 ∩ 0.65) = 0.65; the "Poss" value of "moderate" 0 comes from Min (0 ∩ 0.85) = 0; the "Poss" value of "thin" 0.75 comes from Min (0.75 ∩ 0) = 0. As a result, "moderate" and "thin" attributes are omitted in the fuzzy intersection result since their membership degrees are 0.

*Definition 9 (Fuzzy Difference):* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ and $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees. When $f_i$ and $f_j$ are isomorphic, the fuzzy difference (−) can be defined as follows:

- $\omega(f_i - f_j) = (V_{i-j}, \psi_{i-j}, T_{i-j}, \varpi_{i-j}, \wp_{i-j}, \tau_{i-j}, \xi_{i-j}, \delta_{i-j}, \pi_{i-j})$.
- $\varphi(f_i - f_j) \in \{e|e \in \varphi(f_i) \text{ and } e \notin \varphi(f_j)\}$.

Fig. 3 shows a fuzzy difference operation on fuzzy spatiotemporal XML data trees *A* and *B* based on the membership degrees and possibility distribution. According to fuzzy set theory, we have $\delta_r = \text{Min}(\delta_i, \delta'_j)$, where $\delta'_j = 1 - \delta_j$. Here, $\delta_i$ and $\delta_j$ are membership degrees of *A* and *B*; $\delta'_j$ is the complement of $\delta_j$; $\delta_r$ is the membership degree of fuzzy difference result. In Fig. 3, the time interval $[t_3, t_4]$ in the fuzzy difference result $A - B$ comes from the difference between time interval $[t_3, t_6]$ and $[t_4, t_8]$. On the other hand, the "Poss" value of "thick" 0.35 comes from Min (0.85 ∩(1 − 0.65)) = 0.35; the "Poss" value of "moderate" 0 comes from Min (0 ∩ (1 − 0.85)) = 0; the "Poss" value of "thin" 0.75 comes from Min (0.75 ∩ (1 − 0)) = 0.75. As a result, "moderate" attribute is omitted in the fuzzy difference result since its membership degrees is 0.

*Definition 10 (Cartesian Product):* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ and $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees. When $f_i$ and $f_j$ are isomorphic, the Cartesian product (∞) can be defined as follows:

- $\omega(f_i \infty f_j) = (V_{i\infty j}, \psi_{i\infty j}, T_{i\infty j}, \varpi_{i\infty j}, \wp_{i\infty j}, \tau_{i\infty j}, \xi_{i\infty j}, \delta_{i\infty j}, \pi_{i\infty j})$.
- $\varphi(f_i \infty f_j) \in \{(o_1, o_2)|o_1 \in \varphi(f_i) \text{ and } o_2 \in \varphi(f_j)\}$.

The Cartesian product operation takes two fuzzy spatiotemporal trees $f_i$ and $f_j$ as input and produces output fuzzy spatiotemporal trees corresponding to the "Bjuxtaposition" of every pair of trees from $f_i$ to $f_j$. According to fuzzy set theory, we have $\delta r = Min(\delta_i, \delta_j)$. Here, $\delta_i$ and $\delta_j$ are membership degrees of *A* and *B*; $\delta_r$ is the membership degree of Cartesian product result.

*Theorem 4:* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ and $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees, then we have:

- $f_i \cup f_j$ are fuzzy spatiotemporal XML data trees.
- $f_i \cap f_j$ are fuzzy spatiotemporal XML data trees.
- $f_i - f_j$ are fuzzy spatiotemporal XML data trees.
- $f_i \infty f_j$ are fuzzy spatiotemporal XML data trees.

*Definition 11:* Suppose $f_1 = (V_1, \psi_1, T_1, \varpi_1, \wp_1, \tau_1, \xi_1, \delta_1, \pi_1), f_2 = (V_2, \psi_2, T_2, \varpi_2, \wp_2, \tau_2, \xi_2, \delta_2, \pi_2), \ldots, f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$ are fuzzy spatiotemporal XML data trees. $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is the reconstruction of $f_1, f_2, \ldots, f_i$, if there exists a rule $\Im : T \rightarrow T'$ to meet that $F$ comes from $f_1, f_2, \ldots, f_i$ on the basis of Theorem 4.

### B. OTHER USEFUL OPERATIONS

The fuzzy selection (F-Selection) operator $\sigma$ filters the fuzzy spatiotemporal data trees using a special predicate that can be any combination of logical operators and simple qualifications. It accepts a set of fuzzy spatiotemporal XML data trees $F$ as input, applies a given condition to each element node and attribute node, and returns an entire set of the selecting trees, which is not only the content of required result, but also the structure of objective trees.

*Definition 12 (F-Selection):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, if there is a predicate $\rho$, then we have the definition of F-Selection:

$$\sigma_\rho(F) = \{a | a \Leftarrow F \cap con(\rho, a)\}.$$

Here, the predicate $\rho$ and the variable $a$ are bound to a set of fuzzy spatiotemporal XML data trees, and function *con* $(\rho, a)$ is used for extracting the corresponding elements and attributes meeting the selecting condition.

*Theorem 5:* Suppose fuzzy spatiotemporal XML data trees $f_i (i = 0, 1, 2, \ldots)$ are results from the F-Selection on a fuzzy spatiotemporal XML data tree $F$, then we have $f_i \cong F$.

*Proof:* As Definition 12 shown, it does not introduce new fuzzy spatiotemporal objects after F-Selection operation. Thus $f_i \propto F$, and then we have $f_i \cong F$ according to Theorem 3. □

The fuzzy projection (F-Projection) operator $\pi$ may be regarded as eliminating element nodes or attribute nodes other than specified in the fuzzy spatiotemporal XML data trees. In the substructure resulting from nodes elimination, we would expect the partial hierarchical relationships between surviving nodes that existed in the input trees to be preserved.

*Definition 13 (F-Projection):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, if there is a fuzzy projection function $\kappa$, then we have the definition of F-Projection:

$$\pi_\kappa(F) = \{\kappa(a) | a \in F \wedge \kappa(a) \in F.$$

*Theorem 6:* Suppose $\pi_\kappa(F)$ and $\pi'_\kappa(F)$ are results from the F-Projection on a fuzzy spatiotemporal XML data tree $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$. If $a \wp a'$, then we have $\kappa(a) \wp \kappa(a')$.

*Proof:* As Definition 13 shown, the fuzzy projection operator is regarded as eliminating element nodes or attribute nodes, and preserves the same partial hierarchical relationships in the substructure resulting from F-Projection operation as before. Because $\wp$ preserves the order information in the fuzzy spatiotemporal XML data tree. Then we have $\kappa$ $(a) \wp \kappa$ $(a')$ if $a \wp a'$, where $a, a' \in F$. □

The fuzzy join (F-Join) operator selects fuzzy spatiotemporal XML data trees meeting the stated predicate, and construct new fuzzy spatiotemporal XML data trees. It is the F-Selection from the Cartesian product of the fuzzy spatiotemporal XML data trees.

*Definition 14 (F-Join):* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$, $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees, and $f_1 \cong f_2$. If there is a predicate $\rho$, then we have the definition of F-Join ($\theta$):

- $\omega(f_i \theta_\rho f_j) = \omega_\rho(f_i \infty f_j)$.
- $\varphi(f_i \theta_\rho f_j) \subseteq \varphi_\rho(f_i \infty f_j)$.

*Theorem 7:* Suppose $f_i = (V_i, \psi_i, T_i, \varpi_i, \wp_i, \tau_i, \xi_i, \delta_i, \pi_i)$, $f_j = (V_j, \psi_j, T_j, \varpi_j, \wp_j, \tau_j, \xi_j, \delta_j, \pi_j)$ are fuzzy spatiotemporal XML data trees. then we have $(f_i \theta f_j) \cong f_i \cong f_j$.

*Proof:* According to the nature of fuzzy join, $(f_i \theta f_j) = \sigma$ $(f_i \infty f_j)$. By Theorem 4, $f_i \infty f_j$ are fuzzy spatiotemporal XML data trees. Furthermore, by Theorem 5, $\sigma$ $(f_i \infty f_j) \cong f_i \cong f_j$. As a result, $(f_i \theta f_j) \cong f_i \cong f_j$. □

Fuzzy grouping (F-Grouping) operator $\Gamma$ can be used to represent subtree of fuzzy spatiotemporal XML data tree that aggregate or summarize information from a group of similar element nodes or attribute nodes.

*Definition 15 (F-Grouping):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, and *gro* $(a, \rho)$ is a grouping function over the variable $a$ bounding to a set of fuzzy spatiotemporal XML data trees under predicate $\rho$. Then the F-Grouping $\Gamma$ can be defined as:

$$\Gamma_\rho(F) = \{a | a \Leftarrow F \cap gro(a, \rho)\}.$$

Fuzzy ordering (F-Ordering) operator $\Theta$ can be used to reorder the required element nodes or attribute nodes. The order function *ord* $(a, \rho)$ is a function over the variable $a$ bounding to a set of fuzzy spatiotemporal XML data trees under predicate $\rho$, which may be a scalar function such as *avg*. The default ordering is by the order of the first element in the original set of fuzzy spatiotemporal XML data tree.

*Definition 16 (F-Ordering):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, and *gro* $(a, \rho) = \{a | a \in F \wedge \zeta(a, \wp, \beta) \Leftarrow \zeta(a, \wp, \alpha)\}$ is a grouping function over the variable $a$ under predicate $\rho$, where $\zeta : \alpha \rightarrow \beta$. The F-Ordering $\Theta$ can be defined as:

$$\Theta_\rho(F) = \{a | a \Leftarrow F \cap ord(a, \rho)\}.$$

The Bind operator extracts fuzzy spatiotemporal data from an input tree according to a given filter, and produces a structure that contains the variable bindings resulting from the pattern matching.

*Definition 17 (Bind):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, and *bind* $(a, \rho) = \{a | a \in F \wedge \zeta(a, \rho, \beta) \Leftarrow \zeta(a, \rho, \alpha)\}$ is a binding function over the variable $a$ bounding to a set of fuzzy spatiotemporal XML data trees according to a given filter $\rho$.

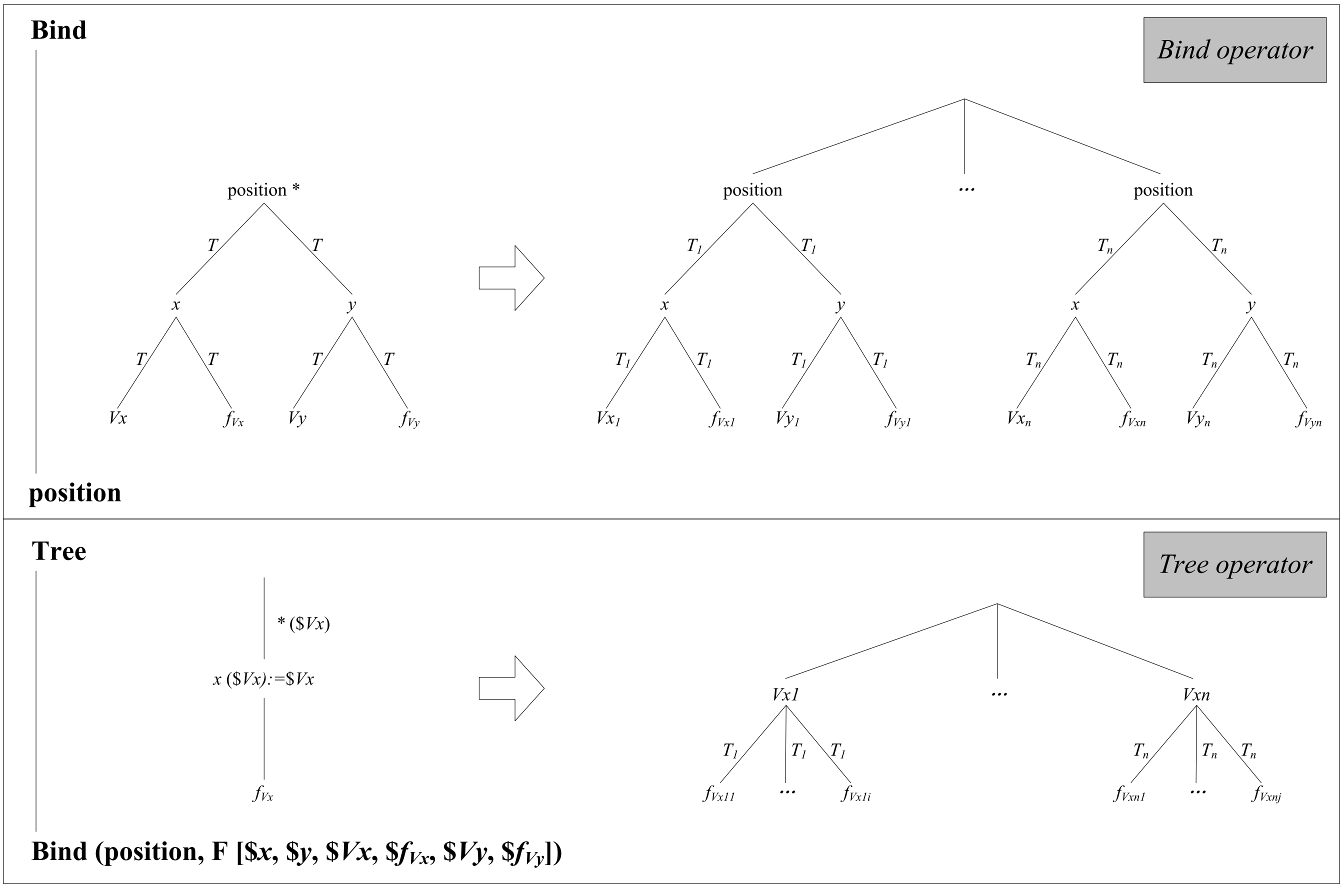



**FIGURE 4.** Bind and Tree operators.

Then the Bind operator can be defined as:

$$Bind(F) = \sum_{bind\ (a,\ \rho)} \{a|a \in F \wedge \zeta(a, \rho, \beta) \Leftarrow \zeta(a, \rho, \alpha)\}.$$

The Tree operator is an inverse operation to the Bind operator, which is applied on Tab structures and returns a collection of trees conforming to some input pattern. It can be used to generate a new nested XML structure.

*Definition 18 (Tree):* Suppose $F = (V, \psi, T, \varpi, \wp, \tau, \xi, \delta, \pi)$ is a fuzzy spatiotemporal XML data tree, and *tree* $(a, \rho)$ is a treeing function over the variable $a$ bounding to a set of fuzzy spatiotemporal XML data trees according to a given filter $\rho$. Then the Tree operator can be defined as:

$$Tree(F) = \sum_{tree\ (a,\ \rho)} \{a|a \in F \wedge tree(a, \rho) \in F\}.$$

In Fig. 4, the Bind operator is applied on the tree representing the XML collection of position, with a filter that binds for each position its $\$x$, $\$y$, $\$Vx$, $\$f_{Vx}$, $\$Vy$, and $\$f_{Vy}$; the Tree operator is applied on the result of the previous Bind, where $F$ [$\$x$, $\$y$, $\$Vx$, $\$f_{Vx}$, $\$Vy$, $\$f_{Vy}$] denotes the corresponding filter. The positions are grouped according to $\$Vx$, with each subtree containing $\$f_{Vx}$ at different temporal intervals.

## V. ALGEBRAIC EQUIVALENCES AND OPTIMIZATION

### A. EQUIVALENCES

Algebraic equivalences indicate that the results replacing the corresponding relations of the two expressions with the same relations are the same. In this section, several algebraic equivalences are presented. Then, strategies of reasonable sequence of operations are proposed.

*Proposition 1:* Let $f_1, f_2$ be fuzzy spatiotemporal XML data trees, $\rho_s$ be the fuzzy selection predicate. Then we have:

$$\sigma_{\rho s}(f_1 \cup f_2) = \sigma_{\rho s}(f_1) \cup \sigma_{\rho s}(f_2).$$
$$\sigma_{\rho s}(f_1 \cap f_2) = \sigma_{\rho s}(f_1) \cap \sigma_{\rho s}(f_2).$$
$$\sigma_{\rho s}(f_1 - f_2) = \sigma_{\rho s}(f_1) - \sigma_{\rho s}(f_2).$$

*Proposition 2:* Let $f_1, f_2$ be fuzzy spatiotemporal XML data trees, $\rho_p$ be the fuzzy projection predicate. Then we have:

$$\pi_{\rho p}(f_1 \cup f_2) = \pi_{\rho p}(f_1) \cup \pi_{\rho p}(f_2).$$
$$\pi_{\rho p}(f_1 \cap f_2) = \pi_{\rho p}(f_1) \cap \pi_{\rho p}(f_2).$$
$$\pi_{\rho p}(f_1 - f_2) = \pi_{\rho p}(f_1) - \pi_{\rho p}(f_2).$$

*Proposition 3:* Let $F$ be a fuzzy spatiotemporal XML data tree, $\rho_{si} (i = 1, 2, \ldots, n)$ and $\rho_{sj}$ $(j = 1, 2, \ldots, n)$ be the fuzzy selection predicates. Then we have:

$$\sigma_{\rho si}(\sigma_{\rho sj}(Bind(F))) = \sigma_{\rho si \wedge \rho sj}(Bind(F)).$$

*Proposition 4:* Let $F$ be a fuzzy spatiotemporal XML data tree, $\rho_{pi}(i = 1, 2, \ldots, n)$ and $\rho_{pj}(j = 1, 2, \ldots, n)$ be the fuzzy projection predicates. If $\{\rho_{pi}\} \subseteq \{\rho_{pj}\}$, then we have:proposition

$$\pi_{\rho pi}(\pi_{\rho pj}(Bind(F))) = \pi_{\rho pi \wedge \rho sj}(Bind(F)) = \pi_{\rho pi}(Bind(F)).$$

*Proposition 5:* Let $F$ be a fuzzy spatiotemporal XML data tree, $\rho_{si}(i = 1, 2, \ldots, n)$ be the fuzzy selection predicate, $\rho_{pj}(j = 1, 2, \ldots, n)$ and $\rho_{pk}(k = 1, 2, \ldots, n)$ be the fuzzy projection predicates.

- If $\rho_{si}$ is the fuzzy selection predicate on $j$, then we have:

$$\sigma_{\rho si}(\pi_{\rho pj}(Bind(F))) = \pi_{\rho pj}(\sigma_{\rho si}(Bind(F))).$$

- If $\rho_{si}$ is the fuzzy selection predicate on $j$ and $k$, then we have:

$$\sigma_{\rho si}(\pi_{\rho pj}(Bind(F))) = \pi_{\rho pj}(\sigma_{\rho si}(\pi_{\rho pk}(Bind(F)))).$$

*Proposition 6:* Let $f_1, f_2$ be fuzzy spatiotemporal XML data trees, $\rho_{si}(i = 1, 2, \ldots, n)$ and $\rho_{sj}$ $(j = 1, 2, \ldots, n)$ be the fuzzy selection predicates.

- If $\rho_s = \rho_{si}$, then we have:

$$\sigma_{\rho s}(Bind(f_1) \infty (Bind(f_2))) = \sigma_{\rho si}(Bind(f_1)) \infty Bind(f_2).$$

- If $\rho_s = \rho_{si} \cap \rho_{sj}$, and $\rho_{si}$ is the fuzzy selection predicate on $i$, then we have:

$$\sigma_{\rho s}(Bind(f_1) \infty (Bind(f_2))) = \sigma_{\rho si}(Bind(f_1)) \infty \sigma_{\rho sj}(Bind(f_2)).$$

- If $\rho_{si}$ is the fuzzy selection predicate on $i$, and $\rho_{sj}$ is the fuzzy selection predicate on $i$ and $j$, then we have:

$$\sigma_{\rho s}(Bind(f_1) \infty (Bind(f_2))) = \sigma_{\rho sj}(\sigma_{\rho si}(Bind(f_1))) \infty Bind(f_2).$$

*Proposition 7:* Let $f_1, f_2$ be fuzzy spatiotemporal XML data trees, $\rho_{pi}(i = 1, 2, \ldots, n)$ and $\rho_{pj}(j = 1, 2, \ldots, n)$ be the fuzzy projection predicates, then we have:

$$\pi_{\rho pi, \rho pj}(Bind(f_1) \infty (Bind(f_2))) = \pi_{\rho pi, \rho pj} \times (Bind(f_1) \infty \pi_{\rho pi, \rho pj}(Bind(f_2)).$$

*Proposition 8:* Let $f_1, f_2$ be fuzzy spatiotemporal XML data trees, $\rho_s$ be the fuzzy selection predicate, $\rho_j$ be the fuzzy join restriction.

- If $\forall a \in f_1, b \in f_2, \rho_s(a \vee b) = \rho_s(a)$, then we have:

$$\sigma_{\rho s}(Bind(f_1)\theta_{\rho j}(Bind(f_2))) = \sigma_{\rho s}(Bind(f_1))\theta_{\rho j}\sigma_{\rho sj} f_2.$$

- If $\forall a \in f_1, b \in f_2, \rho_s\ (a \vee b) = \rho_s\ (b)$, then we have:

$$\sigma_{\rho s}(Bind(f_1)\theta_{\rho j}(Bind(f_2))) = f_1\theta_{\rho j}\sigma_{\rho s}(Bind(f_2)).$$

### B. OPTIMIZATION

The proposed propositions enable the fuzzy spatiotemporal XML queries with reasonable sequence of operations. Generally, the strategies are moving the selections and the projections to the leaves of the input query tree as many as possible by using propositions, and integrate the concatenations of selections and projections into single selections, or single projections, or projections after selections. Moving the selections can use Proposition 1, Proposition 3, Proposition 5, Proposition 6, and Proposition 8; moving the projections can use Proposition 2, Proposition 5, and Proposition 7; integrate the concatenations of selections and projections can use Proposition 3, Proposition 4, and Proposition 5. The optimization strategies of fuzzy spatiotemporal data query are given in the following.

**Algorithm 1** Optimization Strategies

**Input**: Initial query tree of Query $Q$
**Output**: Query tree of Query $Q$ after optimization

```
Let Leaf(Q) = move Q to the bottom of the query tree
for i = 1 to l //l is the maximum number of optimization loops
Leaf(Q) ← σ (Q)
   //moving the selections to the leaves of the input query tree according to Proposition 1, Proposition 3, Proposition 5, Proposition 6, and Proposition 8
Leaf(Q) ← π (Q)
   //moving the projections to the leaves of the input query tree Proposition 2, Proposition 5, and Proposition 7.
σ (Q) ← σ(σ(Q))
   // integrate the concatenations of selections according to Proposition 3.
π (Q) ← π (π (Q))
   // integrate the concatenations of projections according to
   Proposition 4.
σ(π (Q)) ← π (σ(Q))
   //integrate the concatenations of selections and
   projections according to Proposition 3, Proposition 4.
end for
```

A proposition may be used in different strategies such as Proposition 5 because an input query may be enabling with reasonable sequence of operations by using selections before projections or projections before selection according to the practical applications.

## VI. TRANSFORMATION OF XQUERY EXPRESSIONS

XQuery, which is the query language for XML data, is very rich in the variety of provided query expressions. In order to build complex queries, XQuery provides a query structure named FLWOR expressions, which corresponds to initials of keywords identifying the clauses of this kind of expressions "for", "let", "where", "order by", and "return". The basic structure of many queries is the FLWOR expression.

In this section, we show how XQuery expressions can be transformed into algebraic expressions. We first show how each clause of a FLWOR expression is transformed, and then give examples to investigate how to express the fuzzy spatiotemporal XML queries. The syntax of FLWOR expression is as follows:

*for clause | let clause*
[*where <where_condition>* ]
[*order by <order_expression>*]
*return return_expression*

### A. THE "FOR" CLAUSE

A "for" clause may contain a single or multiple variables, associated with an expression whose value is the binding sequence for the variable. A "for" clause is transformed into an algebraic expression which returns a different tree for each possible binding.

*Definition 19:* Suppose there is a "for" clause with a single variable binding: *for $i in doc*("F") $\lambda_1[\gamma_1]\ldots\lambda_n[\gamma_n]$, where $F$ is the input fuzzy spatiotemporal XML data tree, $\lambda_i(1 \le i \le n)$ is a path expression and $\gamma_i(1 \le i \le n)$ is a condition. Such a clause can be transformed into the following algebraic expression:

$$\sigma_{[\gamma i]}(\pi_{\lambda i}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''}))))).$$

*Definition 20:* Suppose there is a "for" clause with multiple variable bindings: *for $i, j in doc*("$F$") $\lambda_1[\gamma_1]\ldots\lambda_n[\gamma_n]$, where $F$ is the input fuzzy spatiotemporal XML data tree, $\lambda_i$, $\lambda_j$ $(1 \le i \le n, 1 \le j \le n)$ are path expressions and $\gamma_i$, $\gamma_j$ $(1 \le i \le n, 1 \le j \le n)$ are conditions. Such a clause can be transformed into the following algebraic expression:

$$\sigma_{[\gamma_i]}(\pi_{\lambda i}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''})))))$$
$$\infty\sigma_{[\gamma j]}(\pi_{\lambda j}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''}))))).$$

### B. THE "LET" CLAUSE

A "let" clause may also contain a single or multiple variables. Unlike a "for" clause, a "let" clause binds each variable to the entire result of its associated expression without iteration. A "let" clause is transformed into an algebraic expression which returns a single tree.

*Definition 21:* Suppose there is a "let" clause with a single variable binding: *let*$i*:* = *doc*("F") $\lambda_1[\gamma_1]\ldots\lambda_n[\gamma_n]$, where $F$ is the input fuzzy spatiotemporal XML data tree, $\lambda_i$ $(1 \le i \le n)$ is a path expression and $\gamma_i(1 \le i \le n)$ is a condition. Such a clause can be transformed into the following algebraic expression:

$$TreeF_{\text{root}}(\sigma_{[\gamma i]}(\pi_{\lambda i}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''})))))).$$

*Definition 22:* Suppose there is a "let" clause with multiple variable bindings: *let*$i*:* = *doc*("F") $\lambda_1[\gamma_1]\ldots\lambda_n[\gamma_n]$, *let*$j := *doc*("$F$")$\lambda_1[\gamma_1]\ldots\lambda_m[\gamma_m]$, where $F$ is the input fuzzy spatiotemporal XML data tree, $\lambda_i, \lambda_j$ $(1 \le i \le n, 1 \le j \le m)$ are a path expressions and $\gamma_i, \gamma_j(1 \le i \le n, 1 \le j \le m)$ are conditions. Such a clause can be transformed into the following algebraic expression:

$$TreeF_{\text{root}}(\sigma_{[\gamma i]}(\pi_{\lambda i}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''}))))))\infty$$
$$TreeF_{\text{root}}(\sigma_{[\gamma j]}(\pi_{\lambda j}\ldots(\sigma_{[\gamma 1]}(\pi_{\lambda 1}(Bind(\text{``}F\text{''})))))).$$

### C. THE "WHERE" CLAUSE

An optional "where" clause contain one or more variables as well, which serves as a filter for the tuples of variable bindings.

*Definition 23:* Suppose there is a "where" clause with a single variable binding: *where $i*$\lambda[\gamma]$, where $F_a$ is the algebraic expression representing the input fuzzy spatiotemporal XML data trees, $\lambda$ is a path expression, $\lambda_i$ is a path expression that locates the nodes bound to the variable $i, and $\gamma$ is a condition. Such a clause can be transformed into the following algebraic expression:

$$\sigma_{\lambda i\lambda[\gamma]}(F_a).$$

*Definition 24:* Suppose there is a "where" clause with multiple variable bindings: *where* $ $i\lambda_i[\gamma_i]$ *and* $ $j\,\lambda_j[\gamma_j]$, where $F_a$ is the algebraic expression representing the input fuzzy spatiotemporal XML data trees, $\lambda_i$ *and* $\lambda_j$ are a path expressions, $\lambda_i'$ and $\lambda_j'$ are a path expression that locates the nodes bound to the variable $i and $j, and $\gamma$ is a condition. Such a clause can be transformed into the following algebraic expression:

$$\sigma_{\lambda i'\lambda i[\gamma i]}(F_a)\theta\sigma_{\lambda j'\lambda j[\gamma j]}(F_a).$$

### D. THE "ORDER BY" CLAUSE

An "order by" clause sorts the result of FLWOR expressions. The transformation of an "order by" clause can be done using the previously defined ordering operator. The transformation of this clause is quite straightforward.

### E. THE "RETURN" CLAUSE

A typical "return" clause specifies the result of the FLWOR expression, which can be transformed using the tree construction operator. Sometimes a FLWOR expression can be nested inside a "return" clause.

*Definition 25:* Suppose there is a "return" clause, $F$ is the input fuzzy spatiotemporal XML data tree, $E$ is an expression that can represent the internal "for", "let", "where", and "order by" clauses.

- If there is no nested clause in the "return" clause, then we have:

$$TreeF_{\text{root}}(*)(\Sigma\lambda_i(\pi_\lambda(Bind(\text{``}F\text{''})))).$$

- If there are nested clauses in the "return" clause, then we have:

$$Tree(E(result)\theta F_{\text{root}}(*)(\Sigma\lambda_i(\pi_\lambda(Bind(\text{``}F\text{''}))))).$$

### F. EXAMPLE

In this subsection, we will give a specific description of the fuzzy spatiotemporal XML query processing according to the definitions of fuzzy spatiotemporal XML algebraic

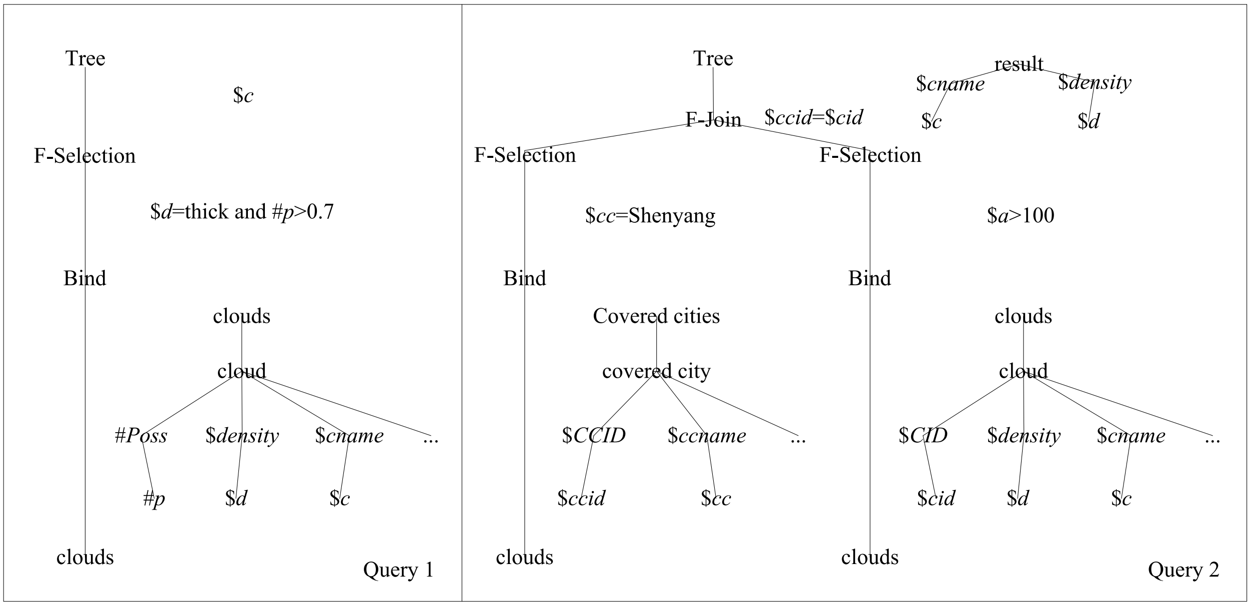



**FIGURE 5.** Fuzzy spatiotemporal algebraic operation.

operations presented before. We use the spatiotemporal XML document in our previous work [21] to investigate how to express the fuzzy spatiotemporal XML query. For convenience, we only use the nature language to express the fuzzy XML queries. In fuzzy spatiotemporal XML data trees, there may be the same name of element and attribute. In order to specify the element and the attribute, we use "$" to express XML element, and "#" to represent XML attribute.

*Query 1:* Return the name of a cloud, whose density is "thick" and the "Poss" more than 0.7.

According to Definition 12, 17, 18, we have

$$Treecname(/cloud/[\$d = \text{``}thick\text{''}](/cloud/[\#p > 0.7](/cloud(Bind(clouds))))).$$

*Query 2:* Return the name and density of clouds, whose area is larger than 100 (km$^2$) in Shenyang. At the same time, the clouds belong to Wiz Khalifa.

According to Definition 12, 13, 14, 17, 18, we have

Tree answer ((/cloud/$c), (/density/$d)) (
$\sigma$/[cloud/#CID= covered city/#CCID] (
$\sigma$/ cloud/[$cc= Shenyang] (
$\pi$/ cloud (Bind (clouds))) $\infty\sigma$/area/[$a>100] (
$\pi$/ covered cities (Bind (cities)))).

Fig.5 shows a simplified, yet substantially expressive, fragment of description of the fuzzy spatiotemporal algebraic operations. Our algebra is well defined and powerful enough to capture the semantics of XQuery.

## VII. EXPERIMENTS

### A. EXPERIMENTAL SETTING

We implemented all the evaluations in C++ on Microsoft Visual Studio 10.0, and performed on a Windows 7 system with 3.2 GHz i7 processor with 4 GB RAM.

**TABLE 1.** Queries used for two data sets.

| ID | Query types | Queries |
|---|---|---|
| Q1 | Query with general node labels only | …//title |
| Q2 | Query with temporal node labels | …//ATTR/ name [#$t_s$] $\vert$ [#$t_e$] |
| Q3 | Query with spatial node labels | …//position/$x \vert y$ |
| Q4 | Query with spatiotemporal node labels | …//position[#$t_s$] $\vert$ [#$t_e$] |
| Q5 | Query with fuzzy general node labels | …//title/[#p>0.7] |
| Q6 | Query with fuzzy temporal node labels | …//ATTR/ name [#$t_s$] $\vert$ [#$t_e$] / [#p>0.7] |
| Q7 | Query with fuzzy spatial node labels | …//position/$x \vert y$ / [#p>0.7] |
| Q8 | Query with fuzzy spatiotemporal node labels | …//position[#$t_s$] $\vert$ [#$t_e$] / [#p>0.7] |

We used two data sets for experimental evaluations. The first data set is the generated FXMark data set [27], stemming from the well-known XMark data set [28]. The second data set is a synthetic data set (abbr. Random), which contains fuzzy spatiotemporal node labels.

For each data set, we select representative queries for evaluations. Table 1 shows the queries used in the experiments.

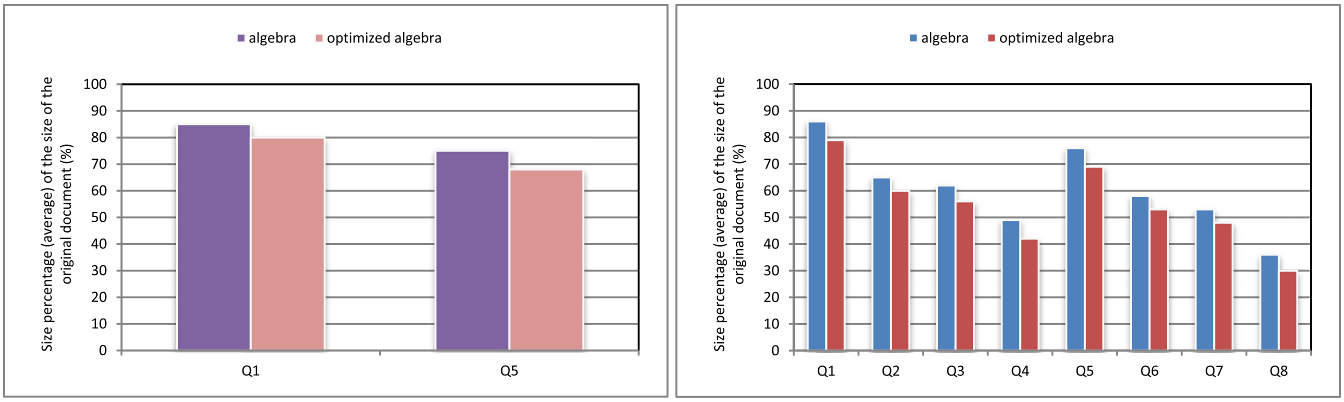
### B. EXPERIMENTAL RESULTS

In the first group of experiments, we evaluate the correctness. The evaluations are composed of Q1 to Q8 shown in Table 1, with their expected results according to the XQuery semantics. Table 2 shows the experimental results of query algebra and optimized query algebra over the two data sets. Because there are no spatiotemporal node labels in FMark, evaluations are only performed on Q1 and Q5 over FMark. It can be observed from Table 2 that the precision of each query over the two data sets are all 100%. An immediate observation from Table 2 is that our algebra does preserve the semantics of the queries.

In the second group of experiments, we evaluate the effectiveness. The experiments are conducted to evaluate the relative reduction of document size in file and the relative reduction of document size in memory.

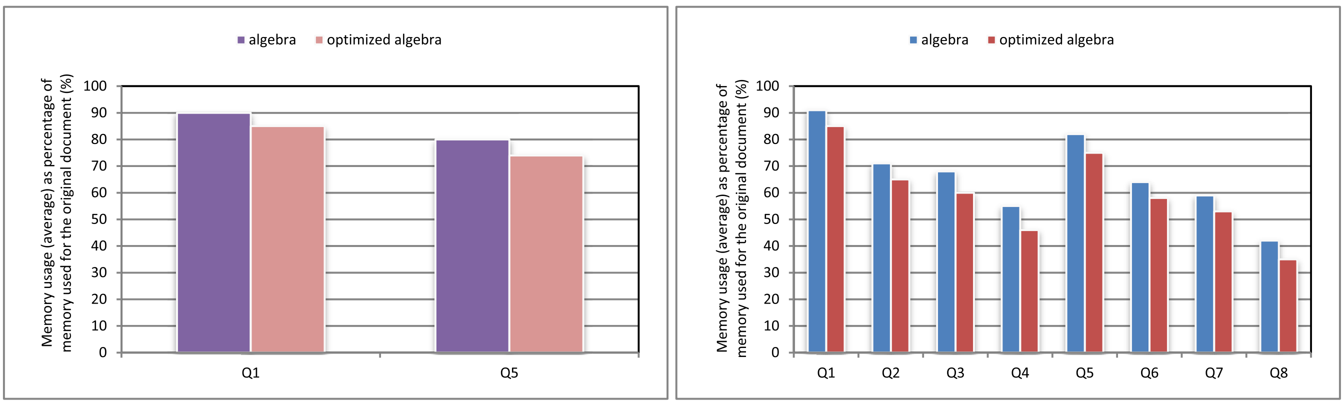
**TABLE 2. Precision of queries.**

| | | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|---|---|---|---|---|---|---|---|---|---|
| algebra | FMark | 100% | - | - | - | 100% | - | - | - |
| | Random | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| optimized algebra | FMark | 100% | - | - | - | 100% | - | - | - |
| | Random | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |




(a) (b)

**FIGURE 6. Size percentage (average) of the size of the original document. (a) FMark. (b) Random.**




(a) (b)

**FIGURE 7. Memory usage (average) as percentage of memory used for the original document. (a) FMark. (b) Random.**

Fig. 6 shows the average size percentage of the size of the original document. We report results over both FMark and Random data sets. An obvious observation from Fig. 6(a) and Fig. 6 (b) is that the algebra and the optimized algebra can reduce the document size in file over both FMark and Random. What's more, the optimized algebra reduces more document size than algebra. Specifically, in Fig. 6(b), it can be observed that Q8 reduces the most document size in file and Q1 reduces the least document size in file. This can be explained by the fact that the more fuzzy spatiotemporal node labels are (temporal, spatial, or fuzzy node labels) the more document size reduce. For this reason, Q8 reduces the most document size in file because it contains fuzzy, spatial, and temporal node labels. Q4 reduces the second most document size in file because it contains spatiotemporal (spatiotemporal, spatial or temporal) node labels. Q2 and Q3 reduce almost the same document size in file because Q2 contains temporal node labels and Q3 contains spatial node labels. Actually, Q3 reduces slightly more document size in file than Q2 for the reason that there are more cases in spatial node labels than that in temporal node labels. The same circumstances are Q6 and Q7. Furthermore, Q6 and Q7 reduce slightly more document size in file than Q2 and Q3 owing to the additional fuzzy node labels. Q5 reduces more document size in file than Q1 because of the additional fuzzy node labels. Circumstances are analogous for Q1 and Q5 in Fig. 6(a).

Fig. 7 shows the average memory usage as percentage of memory used for the original document. It can be observed that the memory usage over FMark and Random is consistent

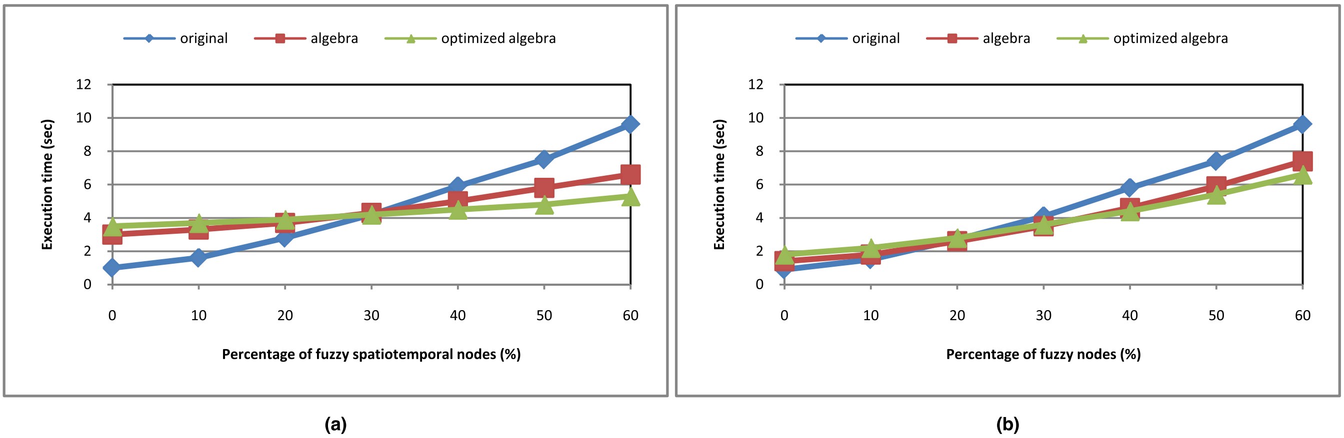



**FIGURE 8.** **Execution time (average) with different percentage of fuzzy (spatiotemporal) nodes. (a) Fmark. (b) Random.**

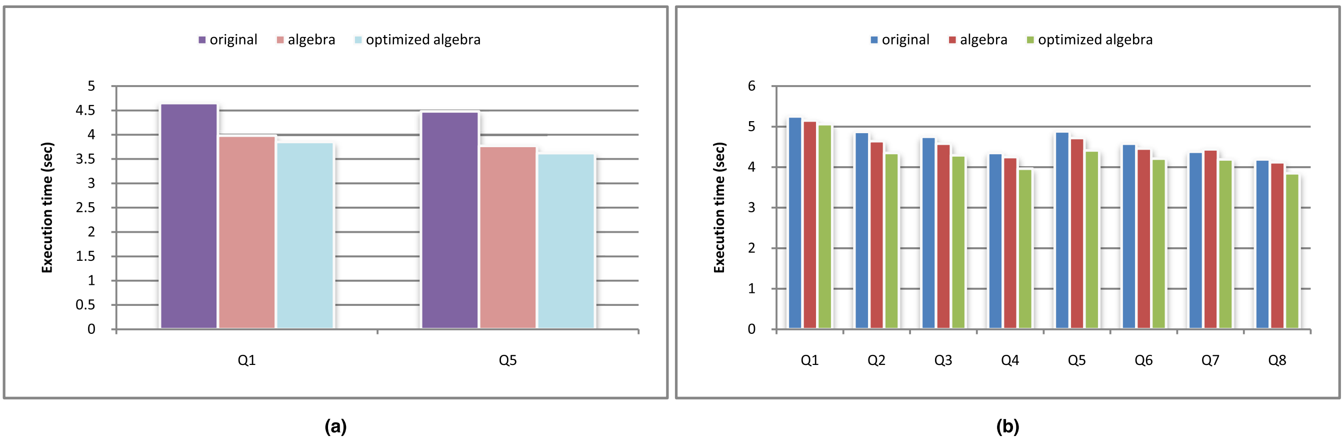



**FIGURE 9.** **Execution time (average) of each query. (a) FMark. (b) Random.**

with the size percentage of the size of the original document. Algebra and optimized algebra tend to use slightly more memory than their size due to their parsing and loading in fuzzy spatiotemporal XML documents.

In the last group of experiments, we evaluate the processing time. Fig. 8 shows execution time with different percentage of fuzzy nodes over FMark and execution time with different percentage of fuzzy spatiotemporal nodes over Random, shown in Fig. 8(a) and Fig. 8(b). It can be observed that the execution time of the original query is less than the execution time using algebra and optimized algebra when the percentage of fuzzy (spatiotemporal) nodes is small; the execution time of the original query is more than the execution time using algebra and optimized algebra when the percentage of fuzzy (spatiotemporal) nodes becomes large. Furthermore, the execution time using algebra is less than the execution time using optimized algebra when the percentage of fuzzy (spatiotemporal) nodes is small; the execution time using algebra is more than the execution time using optimized algebra when the percentage of fuzzy (spatiotemporal) nodes becomes large. This might look surprising, but can be explained by the parsing and loading time in fuzzy (spatiotemporal) XML documents. On the other hand, the execution time over Random outperforms that over FMark. This can be explained by the fact that there is a great deal of fuzzy spatiotemporal nodes in Random. What's more, the execution time over FMark and Random is consistent with the size percentage of the size of the original document shown in Fig. 6 and the memory usage as percentage of memory used for the original document shown in Fig. 7.

It is noted that the execution time are the average execution time of Q1 to Q8. Actually, the execution time of Q1 to Q8 has different performances. In order to find their differences, we perform evaluations of their execution time over FMark shown in Fig. 9(a) and over Random shown in Fig. 9(b). We can observe that the execution time over FMark and Random is consistent with the size percentage of the size of the original document shown in Fig. 6 and the memory usage as percentage of memory used for the original document shown in Fig. 7. Algebra and optimized algebra cost less execution time, while optimized algebra consumes less execution time

**TABLE 3.** Comparing corresponding approaches in terms of supported elements.

| | Spatial elements | Temporal elements | Fuzziness | algebra | | |
|---|---|---|---|---|---|---|
| | | | | Algebraic operators | Algebraic equivalences | Transformation of XQuery expressions |
| [3] | Y | Y | Y | N | N | N |
| [16] | N | N | N | Y | N | N |
| [24] | N | N | N | Y | Y | N |
| [20] | Y | Y | N | N | N | N |
| [10] | Y | Y | N | N | N | N |
| [17] | N | N | N | Y | N | Y |
| [8] | N | N | Y | Y | Y | Y |
| [25] | N | N | N | Y | N | N |
| Our work | Y | Y | Y | Y | Y | Y |

than algebra. In other words, using the optimized algebra can speed up query processing.

As a conclusion, performances can be improved using the proposed algebra, specially the optimized algebra. The more fuzzy spatiotemporal nodes are, the higher performances improve.

## VIII. COMPARATIVE STUDY

In this section, we compare corresponding approaches with our work in terms of supported elements. By comparing with respect to other relational algebra and XML algebra, advantages of our work can be observed. Comparing details between corresponding approaches and our work in terms of supported elements are shown in Table 3.

Comparing corresponding approaches with our work in terms of their supported elements we conclude that the approaches [3], [10], [20] have adequate support for measures in spatial and temporal elements. However, their capabilities to support all measures in algebra are limited. In addition, the approach [3] support fuzziness, while the approaches [10], [20] do not. The approaches [16], [17], [24], [25] do not support measures in spatial elements, temporal elements, and fuzziness. However, they have capabilities to support algebra. For example, the approaches [16], [25] have adequate support for algebraic operators, but their efforts are not extended to algebraic equivalences and transformation of XQuery expressions. The approach [24] has adequate support for algebraic operators and algebraic equivalences, but their efforts are not extended to transformation of XQuery expressions. The approach [17] has adequate support for algebraic operators and transformation of XQuery expressions, but their efforts are not extended to algebraic equivalences. Finally, the approach [8] has adequate support for all measures in algebra and fuzziness, yet their capabilities to support spatial and temporal elements are limited.

Accordingly, to the best of our knowledge, there are no researches concentrating on fuzzy spatiotemporal algebraic operations in XML. Our paper aims at filling this gap.

By comparisons, our work has adequate support for all measures in not only fuzzy spatiotemporal elements but also algebraic elements. Compared with their work, our model and algebra are well defined and powerful enough to support tree patterns, order-sensitive fuzzy spatiotemporal XML queries.

## IX. CONCLUSION

We have proposed a fuzzy spatiotemporal XML data model and presented an algebra for manipulating fuzzy spatiotemporal XML data. In spite of the complex structure of the trees involved, the algebra has a couple of operators using the same basic structure for their parameters. In addition, the algebra is able to handle existence of order within a tree. Furthermore, we also introduce several formal characterizations of fuzzy spatiotemporal algebraic equivalences, and study how XQuery expressions can be transformed into algebraic expressions.

Algebraic operations on fuzzy spatiotemporal data in spatiotemporal applications provide basic foundations for spatiotemporal XML data management. To query the eligible information, we need to know the appropriate input and output formats. Our primary purpose in defining it is to use it as the basis for query evaluation and optimization, which are also our future work.

## REFERENCES

[1] S. Benferhat, J. Ben-Naim, O. Papini, and E. Würbel, "An answer set programming encoding of prioritized removed sets revision: Application to GIS," *Appl. Intell.*, vol. 32, no. 1, pp. 60–87, 2010.

[2] R. Mehrotra and A. Sharma, "Evaluating spatio-temporal representations in daily rainfall sequences from three stochastic multi-site weather generation approaches," *Adv. Water Resour.*, vol. 32, no. 6, pp. 948–962, Jun. 2009.

[3] H. Adeli and X. Jiang, "Neuro-fuzzy logic model for freeway work zone capacity estimation," *J. Transp. Eng.*, vol. 129, no. 5, pp. 484–493, Sep. 2003.

[4] L. Bai, L. Yan, and Z. M. Ma, "Querying fuzzy spatiotemporal data using XQuery," *Integr. Comput.-Aided Eng.*, vol. 21, no. 2, pp. 147–162, Apr. 2014.

[5] M. Koyuncu and A. Yazici, "IFOOD: An intelligent fuzzy object-oriented database architecture," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5, pp. 1137–1154, Sep. 2003.

[6] A. Sözer, A. Yazıcı, H. Oğuztüzün, and O. Taş, "Modeling and querying fuzzy spatiotemporal databases," *Inf. Sci.*, vol. 178, no. 19, pp. 3665–3682, 2008.
[7] P. Senellart and S. Abiteboul, "On the complexity of managing probabilistic XML data," in *Proc. 26th ACM SIGACT-SIGMOD-SIGART Symp. Princ. Database Syst.*, Beijing, China, 2007, pp. 283–292.
[8] Z. M. Ma, J. Liu, and L. Yan, "Fuzzy data modeling and algebraic operations in XML," *Int. J. Int. Syst.*, vol. 25, no. 9, pp. 925–947, 2010.
[9] F. Rizzolo and A. A. Vaisman, "Temporal XML: Modeling, indexing, and query processing," *VLDB J.*, vol. 17, no. 5, pp. 1179–1212, Aug. 2008.
[10] B. Huang, S. Yi, and W. T. Chan, "Spatio-temporal information integration in XML," *Future Gener. Comput. Syst.*, vol. 20, no. 7, pp. 1157–1170, Oct. 2004.
[11] Z. M. Ma and L. Yan, "Fuzzy XML data modeling with the UML and relational data models," *Data Knowl. Eng.*, vol. 63, no. 3, pp. 972–996, 2007.
[12] C. Kahraman, "Fuzzy theory and technology with applications," *J. Intell. Fuzzy Syst.*, vol. 19, nos. 4–5, pp. 231–233, Dec. 2008.
[13] A. Baykasoğlu and T. Göçken, "A review and classification of fuzzy mathematical programs," *J. Intell. Fuzzy Syst.*, vol. 19, no. 3, pp. 205–229, Aug. 2008.
[14] H. Prade and C. Testemale, "Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries," *Inf. Sci.*, vol. 34, no. 2, pp. 115–143, Nov. 1984.
[15] C. Beeri and Y. Tzaban, "SAL: An algebra for semistructured data and XML," in *Proc. Int. Workshop Web Databases*, Philadelphia, PA, USA, 1999, pp. 37–42.
[16] G. Buratti and D. Montesi, "A data model and an algebra for querying XML documents," in *Proc. 17th Int. Conf. Database Expert Syst. Appl.*, Krakow, Poland, Sep. 2006, pp. 482–486.
[17] H. V. Jagadish, L. V. S. Lakshmanan, D. Srivastava, and K. Thompson, "TAX: A tree algebra for XML," in *Proc. 8th Int. Workshop Database Program. Lang.*, Frascati, Italy, 2001, pp. 149–164.
[18] M. Magnani and D. Montesi, "XML and relational data: Towards a common model and algebra," in *Proc. 9th Int. Database Eng. Appl. Symp.*, Montreal, QC, Canada, Jul. 2005, pp. 96–101.
[19] X. Liu and Y. Wan, "Storing spatio-temporal data in XML native database," in *Proc. 2nd Int. Workshop Database Technol. Appl.*, Wuhan, China, Nov. 2010, pp. 1–4.
[20] M. Franceschet, A. Montanari, and D. Gubiani, "Modeling and validating spatio-temporal conceptual schemas in XML schema," in *Proc. 18th Int. Conf. Database Expert Syst. Appl.*, Regensburg, Germany, Sep. 2007, pp. 25–29.
[21] L. Bai, L. Yan, and Z. M. Ma, "Fuzzy spatiotemporal data modeling and operations in XML," *Appl. Artif. Intell.*, vol. 29, no. 3, pp. 259–282, 2015.
[22] S. Paparizos, Y. Wu, L. V. S. Lakshmanan, and H. V. Jagadish, "Tree logical classes for efficient evaluation of XQuery," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Providence, RI, USA, 2004, pp. 71–82.
[23] D. Che and R. M. Sojitrawala, "DUMAX: A dual mode algebra for XML queries," in *Proc. 2nd Int. Conf. Scalable Inf. Syst.*, Suzhou, China, 2007, pp. 1–4.
[24] M. Fernandez, J. Simeon, and P. Wadler, "An algebra for XML query," in *Proc. 20th Int. Conf. Found. Softw. Technol. Theor. Comput. Sci.*, New Delhi, India, 2000, pp. 11–45.
[25] L. Wang, S. Wang, B. Murphy, and E. A. Rundersteiner, "Order-sensitive XML query processing over relational sources: An algebraic approach," in *Proc. 9th Int. Database Eng. Appl. Symp.*, Montreal, QC, Canada, Jul. 2005, pp. 175–184.
[26] E. Hung, L. Getoor, and V. S. Subrahmanian, "PXML: A probabilistic semistructured data model and algebra," in *Proc. 19th Int. Conf. Data Eng.*, Bangalore, India, Mar. 2003, pp. 467–478.
[27] J. Liu, Z. M. Ma, and R. Ma, "Efficient processing of twig query with compound predicates in fuzzy XML," *Fuzzy Sets Syst.*, vol. 229, pp. 33–53, Oct. 2013.
[28] *XMARK the XML-Benchmark Project*. Accessed: 2018. [Online]. Available: http://monetdb.cwi.nl/xml/index.html
[29] L. Bai, L. Yan, and Z. M. Ma, "Determining topological relationship of fuzzy spatiotemporal data integrated with XML twig pattern," *Appl. Intell.*, vol. 39, no. 1, pp. 75–100, 2013.

**LUYI BAI** received the Ph.D. degree from Northeastern University, China. He is currently an Associate Professor with Northeastern University at Qinhuangdao, China. He has published papers in several journals such as *Integrated Computer-Aided Engineering*, *Fuzzy Sets and Systems*, *Applied Intelligence*, and *Applied Artificial Intelligence*. His current research interests include uncertain databases, fuzzy spatiotemporal XML data management, and knowledge graph. He is a member of CCF.

**LIN ZHU** was with Northeastern University (Qinhuangdao), China. Her research interest includes spatiotemporal databases and social networks.