

A Graph Based Approach To Identify Objects Using Identifying Attribute

Anita Kushwaha*, R. S. Pandey

Birla Institute of Technology, MESRA, Ranchi, B-7 Industrial Area, PO TSL, Naini, Allahabad-211010

*Corresponding author, e-mail: a.kushwaha@bitmesra.ac.in

Abstract

We have proposed a method to identify objects in database schema using association degree with other objects. We have also used identifying attribute of associations in graph to specify a unique path to resolve ambiguity of Fuzzy Object Functional Dependencies. Recently Fuzzy Concepts were used in Object Oriented Data Models. The Object Identifier allows distinguishing between similar objects. Functional Dependencies play a dominant role to uniquely identify objects. Moreover object identification has now become a modeling concept rather than database concept so starting a search for objects with a set of values is possible. We have also investigated the presence of identifying attributes in fuzzy object schema and its implications.

Keywords: association degree, object database schema, fuzzy object functional dependencies, cycle-free graph, possibility distribution

Copyright © 2017 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Whenever there is presence of fuzzy associations between different object instances of object type system, it becomes necessary to adopt a different method to distinguish between objects. We examine an Object identification method based on fuzzy relationships with identifying attributes using graph. Object Identification has to be treated as a modeling concept not a database concept. Distinguishing of objects by simply distinguishing between database instances and values in the models do no work as it allows for fuzzy associations between differing instances of objects-also containing fuzzy members of an object system. Other problems that usually occur in identification of objects are recursive and deeply nested structures and presence of identifying attributes i.e. descriptive attribute.

We examine some of these problems that occur in Object identification when there is also presence of identifying attribute [1, 2], [12] using a graph based approach. It is possible to address every object of a type in each database state by specifying some of its properties at the level of values and relationships according to the semantic units. Identifying values gives direct access to the object or they define object as the starting point for the retrieval of the object by navigating along relationship among objects in a given database state. Many differing instances of Objects participate in a fuzzy relationship with some known association degrees. Object identifiers are created by object oriented systems and are hidden from the user i.e. a data query language is not allowed access directly to those identifiers. Though Object identifiers allows a distinguishing of objects even if they coincide in their values or their values are equivalent to each other with a threshold α ($\alpha \in [0, 1]$) they are not sufficient. We present a new approach to object identification based on identifying attribute. It is based on the assumption that every n-ary relationship can be easily broken down into binary relationship. Application domains suggest value based identification criteria (VBIC) for objects for modeling of the objects. In Recent past, several authors emphasized the value based identification of objects [3-7], [10]. An interesting problem is how the structuring of data in a Fuzzy object-oriented schema i.e.-Inheritance influences value based identification. Things become complicated when there is presence of identifying attributes in relationships. Added complexity is there due to objects containing fuzzy members-fuzzy attributes or fuzzy function members. We investigate this problem and cover the concept with some innovative examples and modified new definitions and a short algorithm in the light of Fuzzy associations. We find that some varied research effort are there in multi-

database system [11] in which object identification is done through new language constructs in which administrator describe solution to the identity problem. Some research efforts also [21] have been taken to develop a method to find appropriate knowledge using data-mining technique as given in the works of Rahman, Desa, Wibowo and Haris. Also we find that Mari'f and Mulyanto [22] have developed an affinity network to identify correct system. We find that in [19] language with value invention & expressive power specific to object oriented databases is discussed. Some research efforts are there in F-logic [20] which is a Database logic which accounts for clean declarative fashion for most object oriented features as object identity, complex objects, inheritance and methods.

The organization of this paper is as follows: Section 2 presents basic knowledge with respect to Fuzzy Object Functional Dependencies. Section 3 describes Fuzzy Object Functional Dependencies. Section 4 describes an algorithm to build Fuzzy relation. Section 5 describes the actual Algorithm to find the cycle-free rank based path which holds Fuzzy Object Functional dependencies in object schema graph. Finally, the Section 6 presents result and discussion and section 7 concludes the paper.

2. Research Method

2.1. Semantic Measure of Fuzzy Data

According to Ma Zongmin as in [1-3], [10], [13] a fuzzy value of attribute X of the class is represented by possibility distribution as follows:

$\pi_X = \{\pi_X(u_1)/u_1, \pi_X(u_2)/u_2 \dots \pi_X(u_n)/u_n\}$. Here $U = \{u_1, u_2, \dots, u_n\}$ is universe of discourse. $\pi_X(u_i)/u_i$, $u_i \in U$, denotes the possibility that x takes value u_i . Let π_A and π_B be two fuzzy data on U based on possibility distribution. The degree that π_A semantically includes π_B denoted by $SID(\pi_A, \pi_B)$ is defined as:

$$SID(\prod A, \prod B) = \sum_{i=1}^n \downarrow (\pi_A(u_i) - \pi_B(u_i)) \uparrow \pi_B(u_i)$$

Let $SE(\pi_A, \pi_B)$ be the degree that π_A and π_B are equivalent to each other. Then $SE(\pi_A, \pi_B) = \min(SID(\pi_A, \pi_B), SID(\pi_B, \pi_A))$. For a given threshold $(0 < \beta \leq 1)$, if $SE(\pi_A, \pi_B) \geq \beta$ it is said that π_A and π_B are equivalent to each other with degree β .

2.2. Fuzzy Object-Oriented Database Schema

An object schema [1], [15] consists of binary object types and relationships between these types including inheritance. Sometimes the associations itself can be a Fuzzy association. Also it can contain an identifying attribute which again may be crisp or a fuzzy member of the fuzzy or crisp association. An Object O has a finite set $attr(O)$ of attributes with a domain assigned to each attribute. Relationships may have cardinalities as additional constraints. It can be any one of the type i.e 1:1, 1:M, M:1, M:N. We assume that the relationships to be uniquely named. For our purpose it is sufficient to consider an inheritance hierarchy as a set of binary relationships with additional cardinalities for mandatory and optional hierarchies. Let I be a countable infinite set of object identifiers. An object o of type O is a pair $o = (i, v)$ with $i \in I$ and v is a tuple with attribute set $attr(O)$, called object value. Sometimes presence of Identifying attribute (descriptive attribute) further complicates the issue. Many times we find that the identifying attribute cannot be reasonably associated with any of the objects participating in the relationships or associations. We have to consider it as the attribute of the relationship or the association itself. An example of identifying attribute for relationship is-Let there be objects *Student* and *course* which participates in the relationship set *registered_for*. We may wish to store descriptive attribute *for_credit* with the relationship *registered_for* to record whether the student has taken the course for credit or is auditing the course. We can neither associate the *for_credit* attribute with *Student* object or with the *course* object. It seems it has to be the attribute of relationship *registered_for*. It can be shown by a line connecting the two objects. (See *University database schema* of Figure 1. here *course* is the identifying attribute of *student* and *registration* objects shown by a line connecting the two objects.) The identifying attribute can also be a fuzzy attribute. For e.g. *qualifying_height* attribute between *Candidate* object and

Training object for the relationship *appeared_for*. The qualifying height can take values from the domain {small, medium, high}. Next we give some new definitions with respect to fuzzy associations between different instances of differing objects. **Definition 1:** In presence of Fuzzy associations existing between different instances of differing objects, we model a fuzzy object by a quintuple (id, k, v, μ, μ_r) . Let I be countable infinite set of object identifiers. A fuzzy object o of fuzzy object type is a 5-tuple (id, k, v, μ, μ_r) with $id \in I$, k is a tuple of key attribute which can be crisp or fuzzy attributes, v is tuple (a_2, a_3, \dots, a_n) with $a_i \in \text{dom}(A_i)$, $A_i \in \text{attr}(O)$ -called Object value and $v = v-k$, μ is a membership degree of the object o belonging to object type O and μ_r is a product of $r \times \mu$, which is a association degree of an object instance with other object instance. The extension $\text{ext}(O)$ of fuzzy object type O is a finite set of fuzzy objects. For example for a fuzzy object oriented schema of university given in Figure 1. Here course is the identifying attribute. $\text{ext}(\text{Student}) = \{(1, [\text{CS101}, 'aaa', \{0.9/\text{low}, 0.5/\text{avg}\}], \mu_1, \mu_r), (2, [\text{CS102}, 'bbb', \{0.7/\text{good}, 0.5/\text{excellent}\}], \mu_2, \mu_r)\}$. Here value of attribute type is a fuzzy value on domain {low, poor, good, excellent} based on possibility distribution.

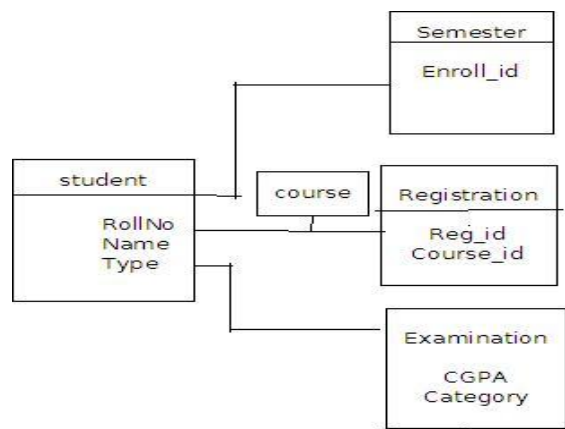


Figure 1. University database

There's a disjunctive XOR between Part-time and Full-time courses in the fuzzy relationships Registered ($\chi=0.8$) and enrolled ($\chi=0.9$) as shown in Figure 2. Prerequisite (not shown in the Figure 2) is a fuzzy attribute with domain {low, avg, medium, high} of registration object.

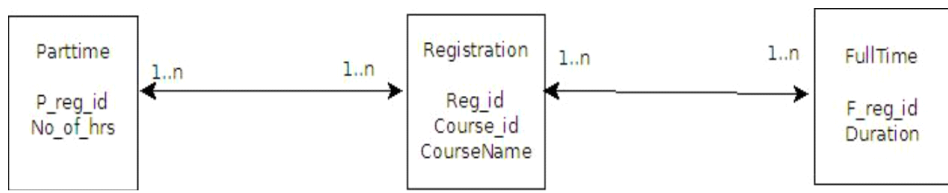


Figure 2. Part-time and Full-time Students Registered Relationship

Specialization or Generalization depicts Inheritance from super-class to subclass. For example a course may be running at different branches of University. Therefore (course, branch) together determines Professor. If (course_id, branch_Addr) are keys of course and branch, a VBIC as in [3] for Professor is given. This provides identification criteria besides identification of Professor by attribute Emp_no which may be useful in some application. Neither course_id nor Branch_Addr are sufficient to identify a Professor.

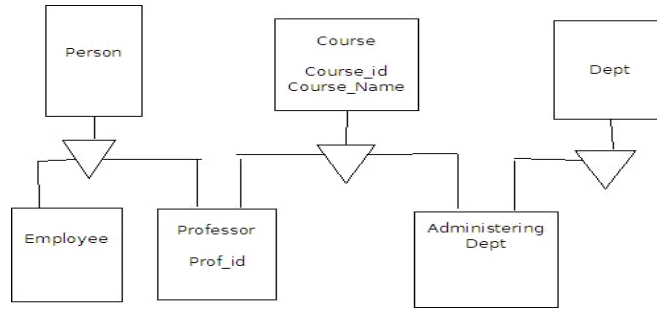


Figure 3. VBIC for Professor

We also find that there can be a possibility of presence of cycles in the representation of the object types. For example: A Professor type can be a person type and predecessor of professor can again be a person type. This results in a cyclic representation of Professor and it's predecessor. This accounts for the added complexity to the object schema. **Definition 2:**The extension $ext(r)$ of a fuzzy relationship r between two fuzzy object types O_1, O_2 with extensions $ext(O_1)$ and $ext(O_2)$ is a finite set of fuzzy links $(id_1, id_2, \mu_r) \in I \times I \times \chi$ with degree of association $\chi \in [0, 1]$, Here $id_1 \in I$ ($ext(O_1)$) and $id_2 \in I$ ($ext(O_2)$). Let $t = ext(r)$. If a cardinality constraint is specified for r then each extension of r has to comply with the constraint. Since every n -ary relationship can be broken down into binary relationship even in presence of identifying attribute, this notion suffices. The state (S) of a fuzzy object-oriented database schema S consists of all extensions $ext(O)$ and $ext(r)$ such that $I((ext(O_1) \cap I (ext(O_2))) = \emptyset, \forall O_1, O_2 \in S$. In order to present a fuzzy object-oriented database schema we use a fuzzy schema graph similar to [5].

2.3. A Fuzzy Schema Graph

A fuzzy schema graph [9], [14], [16] can be defined with respect to fuzzy associations as $G_s = (V, E, I, D)$ of fuzzy object oriented database schema S which is a χ -and edge-labeled graph, where set of nodes V corresponds to fuzzy object types of S and set of edges E represents the fuzzy relationships of S , I is a χ -and edge-labeling function, $I(e) = (r_n, \chi)$ here r_n is a name of the fuzzy relationship r and χ ($\chi \in [0, 1]$) is degree of association between two fuzzy object types belong to the fuzzy relationship r and $d(e) = D$. Let state for S be given, a fuzzy link chain (with respect to a path $\Gamma = O_1 e_1 O_2 e_2 O_3 \dots e_{n-1} O_n$ with $O_i \in V, e_j \in E, e_j$ represents a fuzzy relationship between two fuzzy object types O_j and O_{j+1} with degree of association $\chi, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n-1\}$ between a fuzzy object $o_1 \in ext(O_1)$ and a fuzzy object $o_n \in ext(O_n)$ is sequence $\pi = o_1 I_1 o_2 I_2 \dots I_{n-1} o_n$ such that the following holds: $o_j \in ext(O_i), I_j \in ext(r_j)$ where I_j is a pair of identifiers of o_j and o_{j+1} and r_j being the label of edge e_j . For each fuzzy object type O , let $Set_s(O) = 2^{attr(O)} \cup \{\{O\}\}$, O written as a shorthand notation for the singleton $\{O\}$. For a fuzzy object-oriented database and $D_s = \bigcup set_s(O)$. We shall assume a fuzzy object-oriented database schema to be non-empty i.e it contains at least a fuzzy object type.

2.4. Fuzzy Relation

An $ext(O)$ can be represented by a fuzzy relation $R_{ext(O)}$ with set of attributes $\Omega_R = attr(O) \cup \{id_o\} \cup \mu_o$ here id_o is called the identifier attribute μ_o is a membership attribute and $R_{ext(O)} = \{t \mid t \text{ tuple over } \Omega_R \wedge (\exists (i, v, \mu_{01}) \in ext(O) \exists (i, w, \mu_{02}) \in ext(O_2) (t[id_o] = id_1 \wedge t[id_{o2}] = id_2) \wedge t[\mu_o] = \min(\mu_{01} \times \chi, \mu_{02} \times \chi)) \wedge ((id_1, v, \mu_{01}) \text{ and } (id_2, w, \mu_{02}) \text{ relate to each other with degree } \chi)$. Similarly an extension $ext(r)$ of a fuzzy relationship r between two fuzzy objects O_1 and O_2 can be represented by a fuzzy object O_1 and O_2 can be represented by a fuzzy relation $R_{ext(r)}$ with $\Omega_R = \{id_{o1}, id_{o2}, \mu_{01}\}$ and $R_{ext(r)} = \{t(id_1, id_2, \mu_o) \mid t \text{ tuple over } \Omega_R \wedge (\exists (i, v, \mu_o) \in ext(O_1) (\exists (id_2, w, \mu_{02}) \in ext(O_2)) (t[id_o] = id_1 \wedge t[id_{o2}] = id_2 \wedge t[\mu_o] = \min(\mu_{01} \times \chi, \mu_{02} \times \chi)) \wedge ((id_1, v, \mu_{01}) \text{ and } (id_2, w, \mu_{02}) \text{ relate to each other with a degree } \chi))\}$. Let $t = ext(r)$. The domains of the fuzzy relation attributes are given by the domains of the corresponding attributes of the fuzzy object type. We have to define $ext(t)$. Objects which can be represented by fuzzy relation $ext(t)$ with $\Omega_R = \{id_1, id_2, \dots, id_n, \mu_{01}, \mu_{02}, \dots, \mu_{0n}\}$ and $ext(t) = \{t(id_1, id_2, \dots, id_n, \mu_{01}, \mu_{02}, \dots, \mu_{0n}) \mid t \text{ tuples over } \Omega_R \wedge \exists (id_1, v, \mu_{01}) \in ext(O_1) \exists (id_2, v, \mu_{02}) \in ext(O_2) \dots \exists (id_n, v, \mu_{0n}) \in ext(O_n) (t[id_{o1}] = id_1 \wedge t[id_{o2}] = id_2 \wedge \dots \wedge t[id_{on}] = id_n \wedge t[\mu_{01}] = \min(\mu_{01} \times \chi_1, \mu_{02} \times \chi_1) \wedge (t[\mu_{02}] = \min(\mu_{01} \times \chi_2, \mu_{02} \times \chi_2) \dots (t[\mu_{0n}] = \min(\mu_{01} \times \chi_n, \mu_{02} \times \chi_n))$. Let R_{t_o} be a fuzzy

relation with set of attributes Ω_R , $X \subseteq \Omega_R$. A tuple $t \in R_{fo}$ is total on X if $t[C] \neq \perp$ for each $C \in X$. $NF(R_{fo}, X) = \{t \in R_{fo} \mid t[C] \neq \perp \forall C \in X\}$ called NULL filter relational. $WNF(R_{fo}, X) = \{t \in R_{fo} \mid (\forall t)(\exists C \in X)(t[C] = \perp)\}$. A tuple $t \in R_{fo}$ is undefined on X if $t[C] = \perp$ for each $C \in X$. Here, we use symbol \perp to represent missing links of a fuzzy object or a null value in the sense of value does not exist. $R_{fo}[X]$ denotes the usual projection of R_{fo} on X . Let t_1, t_2 be two tuples over Ω_R . For a given equivalence threshold α , t_1 subsumes t_2 on X if $(\forall C \in X) (SE(t_1[C], t_2[C]) \geq \alpha \vee t_2[C] = \perp)$. To give an example, Let $t_1 = \{(0.9/1, 1.0/2, 0.9/3), 2, 3\}$, $t_2 = \{(0.8/1, 1.0/2, 0.8/3), 2, \perp\}$, $t_3 = \{(0.8/1, 0.8/2, 1.0/3), \perp, 3\}$, $t_4 = (\perp, 2, \perp)$ are tuples over the attributes $X = \{A, B, C$ (value domain of A is fuzzy)}. For $\alpha = 0.8$ we have $SE(t_1[A], t_2[A]) = 0.928 > \alpha$, $SE(t_1[B], t_2[B]) = 1 > \alpha$ and $t_2[C] = \perp$. Therefore t_1 subsumes t_2 on set of attributes $\{A, B, C\}$. Similarly t_1 subsumes t_3, t_4 on set of attributes $\{A, B, C\}$ but t_1 neither does subsumes t_3 nor does t_3 subsumes t_2 on set of attributes $\{A, B, C\}$.

3. Fuzzy Object Dependencies based on Equivalence Class Partitioning

Let us take two objects type student and registration in university database. An example relation for it is given in Table 1:

Example 1: We find attribute RollNo has value 'be/CS/101' only on tuples t_1 and t_2 so they form an equivalence class $[t_1]_{[RollNo]} = [t_2]_{[RollNo]} = \{1, 2\}$. The whole partition with respect to RollNo is $\pi\{RollNo\} = \{\{1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}\}$. Name is $\pi\{Name\} = \{\{1\}, \{2, 3, 4\}, \{5, 6\}, \{7, 8\}\}$ Reg_id is $\pi\{Reg_id\} = \{\{1, 3, 4, 6\}, \{2, 5, 7\}, \{8\}\}$. CourseName is $\pi\{CourseName\} = \{\{1, 4, 7\}, \{2\}, \{3\}, \{5\}, \{6\}, \{8\}\}$. RollNoName is $\pi\{RollNoName\} = \{\{1\}, \{2\}, \{3, 4\}, \{5\}, \{6\}, \{7, 8\}\}$. The partition with respect to $\pi\{NameReg_id\} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$. The functional dependency $S \rightarrow A$ holds if and only if π_S refines π_A . The dependency $\{Name Reg_id\} \rightarrow \{RollNo\}$ holds.

Table 1. Students and Their Registration

Object id	Rollno	Reg_id	Name	CourseName
1	be/CS/101	CS101	'aaa'	DBMS
2	be/CS/101	CS102	'bbb'	C
3	be/CS/102	CS101	'bbb'	Soft Engg.
4	be/CS/102	CS101	'bbb'	DBMS
5	be/CS/102	CS102	'aaa'	C++
6	be/CS/103	CS101	'aaa'	OS
7	be/CS/103	CS102	'ddd'	DBMS
8	be/CS/103	CS104	'ddd'	Compiler dgn

We find attribute RollNo has value 'be/CS/101' only on tuples t_1 and t_2 so they form an equivalence class $[t_1]_{[RollNo]} = [t_2]_{[RollNo]} = \{1, 2\}$. The whole partition with respect to RollNo is $\pi\{RollNo\} = \{\{1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}\}$. Name is $\pi\{Name\} = \{\{1\}, \{2, 3, 4\}, \{5, 6\}, \{7, 8\}\}$ Reg_id is $\pi\{Reg_id\} = \{\{1, 3, 4, 6\}, \{2, 5, 7\}, \{8\}\}$. CourseName is $\pi\{CourseName\} = \{\{1, 4, 7\}, \{2\}, \{3\}, \{5\}, \{6\}, \{8\}\}$. RollNoName is $\pi\{RollNoName\} = \{\{1\}, \{2\}, \{3, 4\}, \{5\}, \{6\}, \{7, 8\}\}$. The partition with respect to $\pi\{NameReg_id\} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$. The functional dependency $S \rightarrow A$ holds if and only if π_S refines π_A . The dependency $\{Name Reg_id\} \rightarrow \{RollNo\}$ holds. We can compare $\pi\{Name Reg_id\}$ and $\pi\{RollNo\}$ and check whether $\pi\{Name Reg_id\}$ refines $\pi\{RollNo\}$. In this example the dependency holds since each equivalence class is totally contained by some equivalence class in $\pi\{RollNo\}$. The dependency $\{RollNo\} \rightarrow \{Name\}$ **does not** hold. The equivalence class $[t_3]_{[RollNo]} = \{3, 4, 5\}$ is not contained in any equivalence class in $\pi\{Name\} = \{\{1\}, \{2, 3, 4\}, \{5, 6\}, \{7, 8\}\}$. The important point is that the dependency $\{RollNo Reg_id\} \rightarrow \{Reg_id\}$ holds which helps in Object identification even if key attribute is missing.

3.1. Fuzzy Object-Functional Dependencies

Fuzzy Functional Dependencies (FOFD) allows us to standardize fuzzy object class [2]. It also restricts fuzzy data redundancies and prevents information loss. Besides it allows expressing constraints stating that objects of the fuzzy object type are identifiable by their value or part of their value in the same way as tuples in a relation can be distinguished by looking at their values in key attributes. However in the case if the attribute values of a fuzzy object are

insufficient to identify it in the ext (O) in state s(S), fuzzy relationships to other objects and values of these objects in s(S) can be taken into account. Generalizing this approach results in a fuzzy Object Functional dependency (FOFD) $f: \Delta \rightarrow \Gamma$ with $\Delta, \Gamma \subseteq D_s$, Where any fuzzy object type of S may contribute to Δ and Γ . Fuzzy Object types between which such a kind of dependency exists do not have to be directly connected in the fuzzy schema graph G_s , a path between them is sufficient. The ambiguity of FOFD can occur if between any two fuzzy objects types appearing Δ or between any fuzzy object type of Δ and a fuzzy object type appearing in Γ more than one path may exist in G_s . A different path corresponds to different semantics [1]. Obviously fuzzy functional dependency [1] can be satisfied with respect to one path and not with respect to another one. Thus paths have to be specified together with fuzzy object functional dependency as given in Figure 4. A graph based approach seems to be appropriate to deal with the mentioned problems. The extension of fuzzy object oriented model as F-model [9] in which fuzzy objects and fuzzy associations are represented. A few definitions with appropriate modifications with respect to fuzzy functional dependencies are as follows: **Definition 3:** Let S be a fuzzy object-oriented database schema with fuzzy schema graph $G_s = (V_{df}, E_{df}, \eta_{df}, d_{df})$ $f = (G_{df}, V_{df})$ is a fuzzy object functional dependency of S with the following properties: The FOFD graph $G_{df} = (V_{df}, E_{df}, \eta_{df}, d_{df})$ is a χ - and edge labeled tree of node set V_{df} ($V_{df} \neq \emptyset, V_{df} \subseteq V$). η_{df} is the restriction of l to E_{df} and $d_{df}(e) = D.V_{df}: V_{df} \rightarrow D_s \times D_s$ is a node labeling function such that for each $O \in V_{df}$ with V_{df} defined and $V_{df}(O) = (\delta, \gamma)$. $\delta, \gamma \in \text{Sets}(O)$. v_{df} has to be defined for at least every leaf node of G_{df} .

For a fuzzy object type O, $v_{df}(O) = (\delta, \gamma)$ and $\delta \neq \emptyset, (\gamma \neq \emptyset)$ is called a source (sink) fuzzy object type of f. (δ part of fuzzy object are used to determine other types; γ -parts of fuzzy object type are determined by other types). The specification of FOFDs by trees of nodes of the fuzzy schema graph [17, 18] guarantees that there are no ambiguities with respect to connections between sink object type and source object types as in [3]. Since a fuzzy object can be a source object type or a sink object type of FOFD, the node labels are chosen as pairs. Each component of node label consists either of an attribute set or of the fuzzy object type itself. Let $\Delta \rightarrow \Gamma$ be a set notation of V_{df} , where Δ is set of node labels δ ($\delta \neq \emptyset$) and Γ is set of all node labels γ ($\gamma \neq \emptyset$). Δ is called the left side and Γ the right side of the FOFD f. A fuzzy object type O is referred to by FOFD f (or involved in the FOFD f) if O itself or any subset of its attribute set appears in the left side or right side of f. f is called canonical FOFD if only one fuzzy object is involved in Γ . f is called local (global) if just one (more than one) fuzzy object type is involved in f. A FOFD $f = (G_{df}, v_{df})$ can be represented by $f: \Delta \rightarrow \Gamma$ with FOFD graph G_{df} and set notation $\Delta \rightarrow \Gamma$. Let $\Delta \rightarrow \Gamma$ be a set notation of V_{df} , where Δ is set of node labels δ ($\delta \neq \emptyset$) and Γ is set of all node labels γ ($\gamma \neq \emptyset$). Δ is called the left side and Γ the right side of the FOFD f. A fuzzy object type O is referred to by FOFD f (or involved in the FOFD f) if O itself or any subset of its attribute set appears in the left side or right side of f. f is called canonical FOFD if only one fuzzy object is involved in Γ . f is called local (global) if just one (or more than one) fuzzy object type is involved in f. A FOFD $f = (G_{df}, v_{df})$ can be represented by $f: \Delta \rightarrow \Gamma$ with FOFD graph G_{df} and set notation $\Delta \rightarrow \Gamma$.

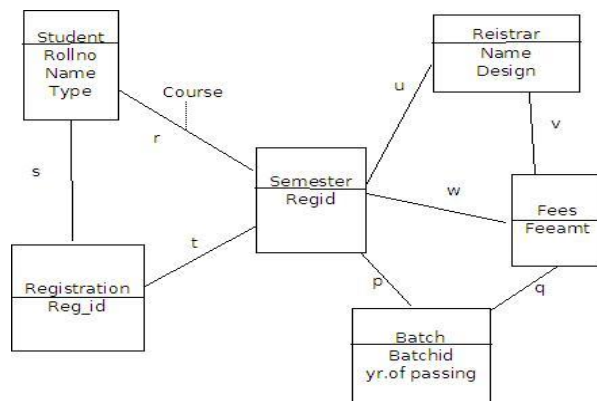


Figure 4. Object schema Graph showing six objects some of which may be fuzzy and their association's p, q, r, s, t, u, v, w and identifying attributes course

Ambiguity of FOFD can be occurred if between any fuzzy object types appearing or between any fuzzy type and a fuzzy object type appearing in more than one path may exist in G_s as given in [1].

4. Algorithm to Build Fuzzy Relation

Input $f: \{\delta_1, \delta_2 \dots \delta_n\} \rightarrow \{\gamma_1, \gamma_2 \dots \gamma_k\}$ with FOFD graph
 $G_f = (V_{df}, E_{df}, \eta_{df}, d_{df})$ and node-labeling function V_{df} ; a equivalence threshold α .
Output: Fuzzy relation R_{fo}
Method:

Let $\delta_i' = \delta_i$ if δ_i is an attribute set $\delta_i' = \{id_o\}$ if $\delta_i = O, O \in OT_s, \Delta' := \cup_{j=1}^k \delta_i'$. Let $\gamma_j' = \gamma_j$ is an attribute set $\gamma_j' = \{id_o\}$ if $\gamma_j = O, O \in OT_s, \tau' := \cup_{j=1}^n \gamma_j'$. If F is a local FOFD referring to a fuzzy Object type then $R_{fo} = R_{ext(o)}$. If F is a global FOFD then R_{fo} is defined as follows: Let $\{o_1 \dots o_m\} \subseteq V_{df}$ be the set of all sink fuzzy Object types and $\Gamma = \{id_{o1}, id_{o2}, \dots, id_{om}\}$ the set of their identifier attributes. For a node $O \in V_f$, let $o = \delta \cup \gamma \cup \{id_o\}$ if $v_{df}(O) = (\delta, \gamma)$ and $o = \{id_o\}$ if $v_{df}(O)$ undefined.

- Select a start node $O \in V_{df}, R_{fo} = R_{ext(o)}[\phi_o]$ For $\forall e \in E_{df}$ do If $e = (O, O_i)$ and $\eta_{df}(e) = r$ then begin
 $R_{fo} = R_{fo} \triangleright \triangleleft \alpha. R_{ext(r)}$; $E_{df} = E_{df} - \{e\}$ and $d_{df}(e) = D$;
 End;
 $V_{df} = V_{df} - \{O\}$;
 While $(V_{df} \neq \emptyset)$ do begin
 Select a node $O_j \in V_{df}$ with $id_{o_j} \in \Omega_{R_{fo}}$. $R_{fo} = R_{fo} \triangleright \triangleleft \alpha. R_{ext(o_j)}[\phi_o]$
 For $e' \in E_{df}$ do
 If $e' = (O_j, O_k)$ and $\eta_{df}(e') = r'$ then Begin $R_{fo} = R_{fo} \triangleright \triangleleft \alpha. R_{ext(r)}$;
 $E_{df} = E_{df} - \{e'\}$, $d_{df}(e) = D$;
 End;
 $V_{df} = V_{df} - \{O_j\}$;

Example 2: Consider the fuzzy object-oriented database schema given in Figure 5, a FOFD f of this schema as follows:

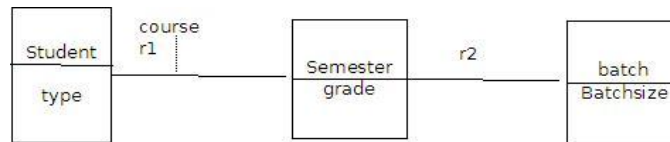


Figure 5. Student, Semester and Batch Fuzzy objects with association r1 and r2 holding between them and identifying attribute course in association r1

Consider the schema in figure 5 consisting of fuzzy object types student, semester, batch with fuzzy attributes type, grade and batch size respectively and two fuzzy relationships r1, r2. Let extensions be given as follows: $Ext(Student) = \{(1, \{1.0/a, 0.7/b, 0.6/c\}), (2, \{1.0/a, 0.7/b, 0.7/c\}), (3, \{0.6/a, 1.0/b, 0.7/c\})\}$. $Ext(Semester) = \{(4, \{0.6/a, 1.0/c\}), (5, \{0.6/a, 0.7/b, 0.1/c\}), (8, \{0.7/a, 1.0/b, 0.6/c\})\}$. $Ext(Batch) = \{(9, \{0.6/a, 0.6/b, 1.0/e\}), (10, \{0.7/a, 0.7/c, 1.0/f\}), (11, \{0.6/a, 0.7/b, 1.0/g\})\}$. $Ext(r1) = \{(1,5), (1,6), (2,7), (3,8)\}$. $Ext(r2) = \{(6,9), (7,10)\}$. Base on different concepts about linkage as in [1], we have fuzzy objects $(1, \{1.0/a, 0.7/b, 0.6/c\}), (2, \{1.0/a, 0.7/b, 0.7/c\}) \in Student$ has a partial linkage with respect to Batch fuzzy objects. Fuzzy objects $(3, \{0.6/a, 1.0/b, 0.7/c\}) \in student$ has only an insufficient link chain with respect to Batch objects; Fuzzy object $(4, \{0.6/a, 1.0/c\}) \in semester$ object has no link chain at all. For example the application of the fuzzy full outer join to two fuzzy relation $R_{ext(batch)}$ and $R_{ext(r2)}$ yields the following fuzzy relation $R, R = R_{ext(batch)} \triangleright \triangleleft_{0.9} R_{ext(r)}$.

$Id_{student}$	Id_{batch}	C	D
6	9	{0.6/a, 0.6/b, 1.0/e}	{0.5/a, 0.6/b, 0.4/c}
7	10	{0.7/a, 0.7/c, 1.0/f}	{0.5/a, 0.6/b, 0.4/c}
	11	{0.6/a, 0.7/b, 1.0/g}	{0.5/a, 0.6/b, 0.4/c}

5. Algorithm to Find Cycle Free Rank based Path which Holds FOFD in Object Schema Graph

$G = (V, E, I, D)$ is weighted directed graph

$W(u, v)$ is the cost of directed edge from node u to v computed as follows

$W = \beta + \text{unique}(1|0)$ where β is the weight assigned to degree of association between relationships and $\text{unique}(1|0)$ is the weight assigned to the identifying (descriptive) attribute if any present on the path from node u to node v . The value of descriptive attribute is 1, if it is prime attribute and has only unique values and 0 otherwise. If the descriptive attribute is present we assign equal weights to β as well as D as they may contribute equally in identifying an object uniquely in a particular path associated with a functional dependency.

```

begin
Initialize
    S: source node, T: sink node
/*Source and sink nodes are determined based on particular functional dependency
satisfying the particular path in the graph G.*/
/*K : number of shortest paths to find, *Pu a path from S to u, B: heap data structure
containing paths,
/*P: set of shortest paths from S to t
    Countu=number of shortest paths found to node u,*P=empty
* Countu = 0 for all u in V
    Insert path Ps={S} into B with cost 0
    While B is not empty && count t < k
        Let Pu be the shortest cost path in B with cost C
        B= B- {Pu}; countu = countu +1
    If u=t then p=p U Pu
    If countu< k then
For each vertex adjacent to u
Let Pv be the new path with cost C + W (u,v) formed by concatenating edge (u,v) to the
path Pu
Insert Pv into B
End

```

6. Result and Discussion

Vu Duc Quang, Doan Van Ban, Ho Cam Ha in [2] have also proposed object-identification model for functional dependencies but they have not incorporated the role of relationships between the objects to remove the functional dependencies. The relationships and descriptive attribute can be used to find the functional dependencies in case of dependencies cycles. In this paper we have added relationship and descriptive attribute in modeling of fuzzy object-oriented database schema and proposed a method for finding the best path to remove cyclic functional dependencies.

7. Conclusion and Future Work

In this paper we use identifying attribute in object Graph for selection of better path in functional dependencies. We have also included relationship with their degree of association between objects in our identification model. We focused on FOFD with cycle free paths in order to keep thing simple. However there are meaningful examples of constraints suggesting a cyclic representation also appearing in FOFD. Different approaches to the different definition of satisfiability and identification of objects in cyclic representation can be taken into account for the extension.

References

- [1] Gloria Bordogna, Dario Lucarella, Gabriella Pasi. *A Fuzzy Object Oriented Data Model*. In Proc. Of Third IEEE Int. Conf on Fuzzy Systems. 1994; 3 13-3: 18,

- [2] Vu Duc Quang, Doan Van Ban, Ho Cam Ha, *A method of Object Identification based on Fuzzy Object functional dependencies in Fuzzy Object Oriented databases*. International Conference on Knowledge and Systems Engineering. 2012; 00: 46-53.
- [3] Serge Atiboul, Paris C Kanellakis. Object Identity as a Query Language primitive. *Journal of the ACM (JACM)*. 1998; 45(5): 798-842.
- [4] Hans-Joachim Klein, Jochen Rasch. *Value Based Identification and Functional Dependencies for Object Databases*. BIWIT'97 Proceedings of the 3rd Basque International Workshop on Information Technology (BIWIT'97). 1997: 22.
- [5] Doan Van Ban, Ho Cam Ha, Vu Duc Quang. *Querying Fuzzy Object Oriented Data based on Fuzzy Association Algebra*. Third International Conference on Knowledge and Systems Engineering (KSE). 2011. doi:10.1109/KSE.2011.15.
- [6] M Gogolla. *A Declarative Approach to Object Identification*, Proceeding 14* Intl.Conf. On Object-oriented and Entity-Relationship Modeling, *Gold Coast*, 1995, *Lecture Notes in Computer Science* 1021, pp 65-76, Springer-Verlag.
- [7] S Abiteboul and J Van den Bussche. *Deep Equality Revisited*, Proc. 4th Intl.Conf on Deductive and Object-oriented Databases. *Singapore*, *Lecture Notes in computer Science* 1013. 1995: 213-228, Springer-Verlag.
- [8] M Lacroix, A pirotte, Generalized Joins. *ACM SIGMOID Record (FDT)*. 1976; (3): 15-16.
- [9] Selee Na, Seog Park. A Fuzzy Association Algebra Based on Fuzzy Object oriented Data Model. *IEEE*. 1996. 0730-3157/96.
- [10] Edward Mezyk, Olgierd Unold. Mining fuzzy rules using an Artificial Immune system with Fuzzy partition Learning. *Applied Soft Computing*. 1975; 11(2011): 1965–1974.
- [11] William Kent, Rafi Ahmed, Joseph Albert, Mohammad Ketabchi, Ming-Chien Shan. *Object identification in multidatabase systems*. DOI: 10.1016/B978-0-444-89879-1.50023-3.
- [12] Francois Bancijhon. *Object oriented database systems*. Proceedings of the seventh ACM SIGACT-SIGART symposium on Principles of database systems, Austin, Texas, United States. 1988: 152-162.
- [13] B Bouchon-Meunier, Ho Thuan, Dang Thanh Ha. *Fuzzy Logic and Applications*. Vietnam National University. 2007
- [14] P Boase, O Pivert. SQLf: a relational database language for fuzzy query. *IEEE Transaction on Fuzzy System model*. 1995; 3(1): 1-19.
- [15] Christophe Lecluse, Philippe Richard, Fernando Velez. *O2-an object oriented Data Model*. Proceedings of the ACM SIGMOID. 1988; 17(3): 424-433.
- [16] G Shaw, S Zdonic. A object algebra for Object-oriented databases. *Proc.6th Int.Conf.on Data Engineering*. 1990: 154-162.
- [17] Doan Van Ban, Ho Cam Ha, Duc Quang, Normalizing object classes in Fuzzy object-oriented database schema, vjs.ac.vn/index.php/jcc/article/view/497
- [18] Mike P. Papazoglou, Stefano Spaccapietra, Zahir Tari. *Advances in Object oriented Modeling*, *Article*. 2000.
- [19] Victor Vianu. Rule based languages. *Annals of Mathematics and Artificial Intelligence*. 1997; 19(1): 215-259.
- [20] Michael Kifer, Georg Lausen. F-Logic: a higher order language for reasoning about object, inheritance and scheme. *SIGMOID'89 Proceedings of the 1989 ACM SIGMOID international conference on Management of data*. 1989: 134-146.
- [21] Fauziah Abdul Rahman, Mohammad Ishak Desa, Antoni Wibowo, Norhaidah Abu Haris. *Knowledge Discovery Database (KDD)-Data Mining Application in Transportation*. Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014). Indonesia. 2014: 20-21.
- [22] Muhammad Rifqi Ma'arif, Agus Mulyanto. Improving Recommender System Based on Item's Structural Information in Affinity Network. *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014)*, Yogyakarta, Indonesia. 2014: 20-21.