関係計算を用いたファジィ関係データベースの研究

モハッマド，デニ，アクバル

出版情報：Kyushu University, 2017, 博士（機能数理学），課程博士
バージョン：
権利関係：

A Study on Fuzzy Relational Database Model using Relational Calculus

A Dissertation Submitted to the Graduate School of Mathematics in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Functional Mathematics Kyushu University

By

Mohammad Deni Akbar
Supervisor: Professor Yoshihiro Mizoguchi

九州大学
KYUSHU UNIVERSITY

Graduate School of Mathematics
Kyushu University
2017

Abstract

We introduce a formalization of a fuzzy relational database model using relational calculus on the category of fuzzy relations. The fuzzy relational database is an extension of the relational database using a soft computing technique, fuzzy theory. We also introduce general formulas for the notion of database operations such as 'projection', 'selection', 'injection' and 'natural join' which can be used for both traditional and fuzzy database models. Using our relational formulas of relational calculus, we can denote its properties by simple and correct formulas. Also, we can prove its properties formally using relational calculus. Further, we formalize an equivalence class of the fuzzy relational database. There are many applications of fuzzy relational database such as managing hyperlinks of web pages, customer relationship management, etc. Our motivation is developing formal verification tools for a software system using an equivalence class of fuzzy relational database.

Next, we show the soundness and the completeness of Armstrong's inference rules of the Functional Dependency (Implication or Association rule) with respect to fuzzy relational databases. In 2009, Ishida et al. presented a relational formulation for Armstrong's inference rules and implication in a Dedekind category. They showed a relation-algebraic proof of the soundness and completeness theorem for Armstrong's inference rules in a Dedekind category. We follow their approach especially focusing on fuzzy relations using relational calculus. We introduce a formalization of a fuzzy equivalence relation, a fuzzy implication, and functional dependency. We prove theorems in our formalization of fuzzy concepts using relational calculus. We also show the logical comparison between a fuzzy implication dependency and a functional dependency.

Finally, we demonstrate the truck backer-upper control problem using our formulation of the fuzzy relational database. Every fuzzy state, operation strategies, and solving procedures are described by database tables and formulas of relational calculus. We also show examples of computations of fuzzy relational databases using our Mathematica implementation.

Acknowledgment

First, I would like to express my deepest gratitude to Professor Yoshihiro Mizoguchi, my supervisor as well as the academic advisor, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I am also grateful to Professor Yasuo Kawahara for his valuable suggestions and helpful comments to succeed this research.

I am express my gratitude to Professor Hitoshi Furusawa of Department of Mathematics of Kagoshima University for all of his valuable comments, suggestions, encouragement and his continues support. Since my thesis is the application of his theory in algebraic of fuzzy relations, his comments and corrections are very important in my research.

I am also express my gratitude to Professor Hidefumi Kawasaki of Department of Mathematics of Kyushu University for his kind support and all of comments during my studies in Kyushu University. Also, I would like to thanks for his kind help that give me chance to get experience internship in Toshiba.

My wholeheartedly gratitude goes to my parents and my family, especially to my mother and my wife who was the persons behind all the success in my life, providing me every support, and for all supplications to Allah for my success in PhD.

I would like thanks to all Mizoguchi's lab member, Indonesian community, moslem community in Fukuoka, the officials of various academic and non-academic divisions of Kyushu University for their kind support during my studies. Also, I would like to thanks to Interactive Media Laboratory members in Toshiba company for all the experience, kind help, and discussion during my internship in Toshiba.

Finally, I would like to thank The Ministry of Education, Culture, Sports, Science and Technology (MEXT) Japan, for providing Super Global scholarship to support funding during my studies.

Contents

Chapter 1

Introduction

The relational database model was first introduced by Codd in 1970[Cod70]. One of the advantages of the relational model is soundness for data consistency. A procedure of data processing is sometimes described by dynamic sequences of operations which may have ambiguities in its implementation. Since a procedure in the relational database model is defined by a static formula, we can avoid inconsistencies in their implementations.

Relational databases commonly contain attributes, tuples, database relation, database schema, and database operations. In this paper, we show properties of fuzzy relational databases using relational calculus.

Fuzzy database relation has been introduced by Raju and Majumdar in 1988[RM88], and fuzzy operations model was introduced by Umano and Fukami in 1994[UF94] and Nakata in 2000[Nak00]. Database operator is one of the important aspects in relational database properties. Fuzzy relational database operators('projection', 'join', and 'selection') has been introduced by Umano in 1994[UF94]. In our work, we use relational calculus to compute database operations. We use simpler and clearer formula to compute database operations.

The first concept of fuzzy relation has been invented by Zadeh in 1965[Zad65]. Kawahara, et al. developed an algebraic formalization of fuzzy relations using relational calculus[Kaw88, Kaw95, KM93, MK95a, MK95b]. Okuma, et al. introduced a relational database model using a theory of relational calculus on the Dedekind Category in 2000[OK00]. Since the Dedekind Category includes the category of fuzzy relations, we can formalize a fuzzy relational database model using relational calculus on the category of fuzzy relations.

In Chapter 2, we present basic terminologies and key results related to fuzzy relations and its operations, which will be used throughout this thesis.

In Chapter 3, we introduce a formalization of a fuzzy relational database model using relational calculus on the category of fuzzy relations. We also introduce general formulas for the notion of database operations such as 'projection', 'selection', 'injection' and 'natural join' which can be used for both traditional and fuzzy database models. We prove several elementary properties of database operations using relational calculus. We note that the same formulas of relational calculus can be used for a traditional non-fuzzy relational database model. So we can consider our fuzzy relational database theory using intuition of the traditional database operations. There exist some trials of fuzzy relational database theory, but they are limited in specific topics such as fuzzy graph problem introduced by Kiss in 1991[Kis91] and Mordeson and Nair in 2001[MN01]. The advantage of our framework is showing several relational database operations uniformly using the theory of relational calculus.

In Chapter 4, one of our goals is to formalize an equivalence class of the fuzzy relational database. Using our relational formulas of relational calculus, we can denote its properties by simple and correct formulas. Also, we can prove its properties formally using relational calculus. There are many applications of fuzzy relational database such as managing hyperlinks of web pages, customer relationship management, etc. Our motivation is developing formal verification tools for a software system using an equivalence class of fuzzy relational database. We also extend the formalization database operations in previous chapter such as "projection", "selection", and "natural join" to equivalence class model. We prove several elementary properties of natural join operations using our formalization.

In Chapter 5, we show the proof of the soundness and completeness in Armstrong's inference rules for the fuzzy functional dependency and fuzzy implication (association rule) with respect to fuzzy context table. Okuma and Kawahara introduced the notions of functional dependency and multivalued dependency using Dedekind category[OK00]. In 2009, Ishida et al. presented relational formulations for Armstrong's inference rules in a Schrőder category[IHK08]. Since Schrőder category is special for the boolean case then we considered making using Dedekind category to investigate the soundness and completeness in fuzzy context table. We investigate the approach introduced by Okuma, Kawahara and Ishida especially focusing on fuzzy relations using relational calculus. In 1998, Duntsch and Gediga defined indiscernibility relation. We follow their approach to define the equivalence relation[DG98]. In 1997, Ganter and Wille in their textbook investigated formal concept analysis, they introduced some properties of dependency in the formal context such as implication, functional dependency, etc[GW97]. In this chapter, we try to redefine the implication and functional dependency using equivalence relation in relational calculus. The formalization can be used to analyze equivalence condition between functional dependency and implication. We prove theorems in our formalization of fuzzy concepts using relational calculus. We also show some logical comparison theorems between a fuzzy implication and a functional dependency. Finally, we explain implemented operations in our fuzzy formal concept analysis using Mathematica software. We show some examples in the application of data analysis. Future work, includes constructing a theory of fuzzy relational database theory with computer verified formal proofs using relational calculus.

In Chapter 6, we apply our formulation to an example of the truck backer-upper control problem using fuzzy logic introduced by Freeman in 1994[Fre94]. Every fuzzy states, procedures are described as database tables of the fuzzy relational database theory. Problem-solving procedures are also described by static formulas of the relational calculus on the category of fuzzy relations. Since every property is described statically, the consistency of data can be proved formally. We also implemented operations in the theory of fuzzy relational database using Mathematica Software. Using our Mathematica library, we demonstrate computations in the truck backer-upper control problem.

Chapter 2

Basic Notions of Fuzzy Relational Calculus

There are many applications of fuzzy relations such as fuzzy modeling, fuzzy diagnosis, and fuzzy control. The applications of fuzzy relations can be used in many fields such as psychology, medicine, economics, and sociology. In this chapter, we define and discuss the notions of fuzzy relations. Beginning with a definition of fuzzy relations, we then talk about expressing fuzzy relations in relational calculus. Later we discuss the various properties of fuzzy relations and operations that can be performed with fuzzy relations.

In 1941, relational calculus has been introduced by Tarski[Tar41]. Then, many researchers continue about relational calculus such as Relational set theory by Kawahara[Kaw95], Relational Mathematics by Schmidt[Sch10], etc.

Next, there is much research that extends the relational calculus to the fuzzy relation. Using Dedekind category, they made some formalization of fuzzy relation, such as for cut off relation by Furusawa et al[FKW11], member basis value by Winter et al[WJ16].

In this chapter, we would like to introduce some properties and operations of fuzzy relations using relational calculus. The formalization is very basic and important because we will use the formalization to formalize in the next chapter.

2.1 Fuzzy Relation

In this section, we summarize basic notations for fuzzy relations. We denote the set $\{\mathbf{x} \in \mathbb{R} | 0 \leq \mathbf{x} \leq 1\}$ as [0,1]. The supremum and infimum of a family $\{x_\lambda\}_{\lambda \in \Lambda}$ of elements $x_\lambda \in [0,1]$ is denoted by $\bigvee_{\lambda \in \Lambda} x_\lambda$ and $\bigwedge_{\lambda \in \Lambda} x_\lambda$, respectively. In particular, $x \vee x' = max\{x, x'\}$, $x \wedge x' = min\{x, x'\}$ for $x, x' \in [0,1]$. For two elements $x, x' \in [0,1]$, the relative pseudo-complement $\Rightarrow$ of x relative to

x' defined by $x \Rightarrow x' := [x \leq x'] \vee x'$, where

$$[x \leq x'] := \begin{cases} 1 & \text{if } x \leq x', \\ 0 & \text{otherwise }. \end{cases}$$

Lemma 2.1.1. *Let $x, y, z \in [0, 1]$.*

$$x \leq (y \Rightarrow z) \text{ if and only if } x \wedge y \leq z.$$

Proof.

$$\begin{aligned} (\rightarrow)\ x \leq (y \Rightarrow z) &\leftrightarrow x \leq [y \leq z] \vee z \\ &\leftrightarrow (x \leq [y \leq z]) \vee (x \leq z) \\ &\rightarrow (x = 0 \vee (y \leq z)) \vee (x \leq z) \\ &\rightarrow (x \wedge y \leq z). \\ (\leftarrow)\ x \wedge y \leq z &\leftrightarrow x \leq z \vee y \leq z \\ &\rightarrow x \leq z \vee x \leq [y \leq z] \\ &\rightarrow x \leq [y \leq z] \vee z \\ &\leftrightarrow x \leq (y \Rightarrow z). \end{aligned}$$

□

Definition 2.1.1. *A* **fuzzy relation** *α from a set A to another set B is a fuzzy subset of the Cartesian product $A \times B$. i.e. $\alpha : A \times B \rightarrow [0, 1]$. It is denoted by $\alpha : A \rightharpoondown B$. The* **composition** *$\alpha \cdot \beta : A \rightharpoondown C$ of $\alpha : A \rightharpoondown B$ followed by $\beta : B \rightharpoondown C$ is a fuzzy relation defined by $\alpha \cdot \beta(a, c) := \bigvee_{b \in B} (\alpha(a, b) \wedge \beta(b, c))$. The* **identity** *relation $id_A : A \rightharpoondown A$ is defined by*

$$id_A(a, b) := \begin{cases} 1 & (a = b), \\ 0 & (a \neq b) \ . \end{cases}$$

Definition 2.1.2. *We define a category $FRel$ as follows:*
An object X of $FRel$ is a set. For two objects X and Y, a morphism set $FRel(X, Y)$ is a set of fuzzy relations from X to Y.

It is easy to check that $FRel$ is a category with a composition and an identity of relation.

Definition 2.1.3. *Let X, Y be objects in $FRel$. α, β morphism in $FRel(X, Y)$, $a \in X$, and $b \in Y$. We define fuzzy operations $\#$(invers), $\sqcup$(union), $\sqcap$(intersection), $\sqsubseteq$(subset), $\Rightarrow$(the relative pseudo-complement), and constants $\mathbf{0}_{AB}$(least), ∇_{AB} (greatest) in $FRel$, as follows:*

1. $\alpha^{\#}(b, a) := \alpha(a, b)$
2. $(\alpha \sqcup \beta)(a, b) := \alpha(a, b) \vee \beta(a, b)$,

3. $(\alpha \sqcap \beta)(a, b) := \alpha(a, b) \wedge \beta(a, b)$,

4. $\alpha(a, b) \sqsubseteq \beta(a, b)$ *iff* $\alpha(a, b) \leq \beta(a, b)$,

5. $\mathbf{0}_{AB}(a, b) := 0$, *and*

6. $\nabla_{AB}(a, b) := 1$.

Example 2.1.1. *Consider* $A = \{a, b, c, d\}$, $B = \{2, 3, 4\}$, $C = \{x, y, z\}$. *Let* $\alpha_1 : A \rightharpoonup B$, $\alpha_2 : A \rightharpoonup B$, $\beta : B \rightharpoonup C$ *be fuzzy relations such that we defined by* $\alpha_1(a, 2) = 0.1$, $\alpha_1(a, 4) = 0.4$, $\alpha_1(c, 2) = 0.3$, $\alpha_1(d, 3) = 0.5$, $\alpha_2(a, 2) = 0.3$, $\alpha_2(c, 2) = 0.2$, $\beta(2, y) = 0.5$, $\beta(4, y) = 0.2$, $\beta(3, z) = 0.6$,*and* $\beta(3, x) = 0.4$.

The followings are examples of a union, an intersection and a composition relation:

- *We have* $(\alpha_1 \sqcup \alpha_2)(a, 2) = 0.3$, $(\alpha_1 \sqcup \alpha_2)(a, 4) = 0.4$, $(\alpha_1 \sqcup \alpha_2)(c, 2) = 0.3$ *for* $(\alpha_1 \sqcup \alpha_2) : A \rightharpoonup B$.

- *We have* $(\alpha_1 \sqcap \alpha_2)(a, 2) = 0.1$, $(\alpha_1 \sqcap \alpha_2)(c, 2) = 0.2$ *for* $(\alpha_1 \sqcap \alpha_2) : A \rightharpoonup B$.

- *We have* $(\alpha_1 \cdot \beta)(a, y) = (\alpha_1(a, 2) \wedge \beta(2, y)) \vee (\alpha_1(a, 3) \wedge \beta(3, y)) \vee (\alpha_1(a, 4) \wedge \beta(4, y)) = 0.1 \vee 0 \vee 0.2 = 0.2$ *for* $(\alpha_1 \cdot \beta) : A \rightharpoonup C$.

Definition 2.1.4. *Let* $\alpha : X \rightharpoonup Y$, $\beta : Y \rightharpoonup Z$ *be fuzzy relations for all* $x \in X$, $y \in Y$ *and* $z \in Z$. *The residue composite* $\alpha \rhd \beta : X \rightharpoonup Z$ *is defined by:*

$$(\alpha \rhd \beta)(x, z) = \bigwedge_{y \in Y} (\alpha(x, y) \Rightarrow \beta(y, z))$$

2.2 Category

Definition 2.2.1. *[FK15]* **Dedekind Category** D *is a category satisfying the following axioms D1(Complete Distributive Lattice), D2(Converse), D3(Dedekind Formula), and D4(Residual Composition).*
We denote a morphism $\alpha \in D(X, Y)$ *as* $\alpha : X \rightharpoonup Y$.
D1. *For all pairs of objects X and Y the hom-set* $D(X, Y)$ *consisting of all morphisms of X into Y is a complete distributive lattice. D(X,Y) =* $($*D(X,Y)*, $\sqcap$, $\sqcup$, $\sqsubseteq$, 0_{XY}, $\nabla_{XY})$ *with the least morphism* 0_{XY} *and the greatest morphism* ∇_{XY}.

That is, if $\alpha, \alpha_i \in D(X, Y)$ *for an index set* I, *we have the followings:*

(a) $\sqsubseteq$ *is a partial order on* $D(X, Y)$,

(b) $0_{XY} \sqsubseteq \alpha \sqsubseteq \nabla_{XY}$,

(c) $\sqcup_{i \in I} \alpha_i \sqsubseteq \alpha$ *iff* $\alpha_i \sqsubseteq \alpha$ *for all* $i \in I$,

(d) $\alpha \sqsubseteq \sqcap_{i \in I} \alpha_i$ *iff* $\alpha_i \sqsubseteq \alpha$ *for all* $i \in I$, *and*

(e) $\alpha \sqcap \sqcup_{i\in I}\alpha_i = \sqcup_{i\in I}(\alpha \sqcap \alpha_i)$.

D2. *There is given a converse operation*$^{\#} : D(X,Y) \to D(Y,X)$. *That is, for all morphisms* $\alpha, \alpha' \in D(X,Y), \beta \in D(Y,Z)$, *the following converse laws hold:*

(a) $(\alpha\beta)^{\#} = \beta^{\#}\alpha^{\#}$,

(b) $(\alpha^{\#})^{\#} = \alpha$, and

(c) $\alpha \sqsubseteq \alpha'$, then $\alpha^{\#} \sqsubseteq \alpha'^{\#}$.

D3. *For all morphisms* $\alpha \in D(X,Y), \beta \in D(Y,Z)$ *and* $\gamma \in D(X,Z)$, *the Dedekind formula* $\alpha\beta \sqcap \gamma \sqsubseteq \alpha(\beta \sqcap \alpha^{\#}\gamma)$ *holds.*
D4. *For all morphisms, we denote* $\alpha \in D(X,Y)$ *as* $\alpha : X \rightharpoondown Y$ *and* $\beta \in D(Y,Z)$ *as* $\beta : Y \rightharpoondown Z$, *the residual composite* $\alpha \rhd \beta : X \rightharpoondown Z$ *is a morphism such that* $\gamma \sqsubseteq \alpha \rhd \beta$ *if and only if* $\alpha^{\#}.\gamma \sqsubseteq \beta$ *for all morphisms* $\gamma : X \rightharpoondown Z$.

Example 2.2.1. *We consider a category* Rel_e *whose objects are all nonempty set and in which a hom-set* $Rel_e(X,Y)$ *between objects* X *and* Y *is the set of all (binary) fuzzy relations on* X *if* $X = Y$, *and* $\nabla_{XY} = 0_{XY}$, *otherwise. That is, a hom-set* $Rel_e(X,Y)$ *is singleton set when* X *and* Y *are distinct. Then it is easy to verify that the category* Rel_e *is Dedekind category. The conditions (D1) and (D2) are trivial, then (D3) and (D4) also hold as follows:*

(a). If $X = Y = Z$, then (D3) and (D4) are clear from Proposition 2.2.1.1 and 2.2.1.2.

(b). If $X = Y \neq Z$, then $\beta = 0_{YZ}$, and $\gamma = 0_{XZ}$.
So $\alpha \cdot \beta \sqcap \gamma = \alpha \cdot 0_{YZ} \sqcap 0_{XZ} = 0_{XZ}$ holds. Since $0_{XZ} \sqsubseteq \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$, then we have $\alpha \cdot \beta \sqcap \gamma \sqsubseteq \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$ (D3).
Since $\gamma \sqsubseteq \alpha \rhd \beta \leftrightarrow 0_{XZ} \sqsubseteq \alpha \rhd 0_{YZ}$ and $\alpha^{\#} \cdot \gamma \sqsubseteq \beta \leftrightarrow \alpha^{\#} \cdot 0_{XZ} \sqsubseteq 0_{YZ}$, then we have $\gamma \sqsubseteq \alpha \rhd \beta$ and $\alpha^{\#} \cdot \gamma \sqsubseteq \beta$ for any α, β, γ (D4).

(c). If $X \neq Y$ then $\alpha = 0_{XY}$.
So $\alpha \cdot \beta \sqcap \gamma = 0_{XY} \cdot \beta \sqcap \gamma = 0_{XZ}$ holds. Since $0_{XZ} \sqsubseteq \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$, then we have $\alpha \cdot \beta \sqcap \gamma \sqsubseteq \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$ (D3).
Since $\gamma \sqsubseteq \alpha \rhd \beta \leftrightarrow \gamma \sqsubseteq 0_{XY} \rhd \beta \leftrightarrow \gamma \sqsubseteq \nabla_{XZ}$ and $\alpha^{\#} \cdot \gamma \sqsubseteq \beta \leftrightarrow 0_{YX} \cdot \gamma \sqsubseteq \beta \leftrightarrow 0_{YZ} \sqsubseteq \beta$, then we have $\gamma \sqsubseteq \alpha \rhd \beta$ and $\alpha^{\#} \cdot \gamma \sqsubseteq \beta$ for any α, β, γ (D4).

Proposition 2.2.1. *Let* $\alpha \in FRel(X,Y),\ \beta \in FRel(Y,Z)$ *and* $\gamma \in FRel(X,Z)$ *be fuzzy relations.*

1. $\alpha\beta \sqcap \gamma \sqsubseteq \alpha(\beta \sqcap \alpha^{\#}\gamma)$,
2. $\gamma \sqsubseteq \alpha \rhd \beta$ if and only if $\alpha^{\#} \cdot \gamma \sqsubseteq \beta$, and
3. $(0_{XY} \rhd \beta) = \nabla_{XZ}$.

Proof. 1. Let $x \in X$ and $z \in Z$.

$$\begin{aligned}(\alpha \cdot \beta \sqcap \gamma)(x,z) &= (\alpha \cdot \beta)(x,y) \wedge \gamma(x,z) \\ &= \bigvee_{y \in Y} (\alpha(x,y) \wedge \beta(y,z) \wedge \gamma(x,z)) \\ &= \bigvee_{y \in Y} (\alpha(x,y) \wedge \beta(y,z) \wedge \alpha^{\#}(y,x) \wedge \gamma(x,z)) \\ &\sqsubseteq \bigvee_{y \in Y} (\alpha(x,y) \wedge \beta(y,z)) \wedge \bigvee_{x' \in X} (\alpha^{\#}(y,x') \wedge \gamma(x',z)) \\ &= \bigvee_{y \in Y} (\alpha(x,y) \wedge \beta(y,z)) \wedge (\alpha^{\#} \cdot \gamma)(y,z) \\ &= \bigvee_{y \in Y} (\alpha(x,y) \wedge (\beta \sqcap \alpha^{\#} \cdot \gamma)(y,z)) \\ &= \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)(x,z)\end{aligned}$$

So we have $\alpha \cdot \beta \sqcap \gamma \sqsubseteq \alpha \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$.

2. Let $y \in Y$ and $z \in Z$

$$\begin{aligned}\alpha^{\#} \cdot \gamma(y,z) \sqsubseteq \beta(y,z) &\leftrightarrow \forall_y \forall_z : \alpha^{\#} \cdot \gamma(y,z) \leq \beta(y,z) \\ &\leftrightarrow \forall_y \forall_z : \bigvee_{x \in X} (\alpha^{\#}(y,x) \wedge \gamma(x,z)) \leq \beta(y,z) \\ &\leftrightarrow \forall_y \forall_z \forall_x : \alpha^{\#}(y,x) \wedge \gamma(x,z) \leq \beta(y,z) \\ &\leftrightarrow \forall_y \forall_z \forall_x : \gamma(x,z) \wedge \alpha^{\#}(y,x) \leq \beta(y,z)\end{aligned}$$

By Lemma 2.1.1, we get:

$$\begin{aligned}&\leftrightarrow \forall_y \forall_z \forall_x : \gamma(x,z) \sqsubseteq \alpha^{\#}(y,x) \Rightarrow \beta(y,z) \\ &\leftrightarrow \forall_x \forall_z : \gamma(x,z) \sqsubseteq \bigwedge_{y \in Y} \alpha(x,y) \Rightarrow \beta(y,z) \\ &\leftrightarrow \forall_x \forall_z : \gamma(x,z) \sqsubseteq (\alpha \rhd \beta)(x,z)\end{aligned}$$

So we have $\gamma \sqsubseteq \alpha \rhd \beta \leftrightarrow \gamma \sqsubseteq \alpha \rhd \beta$.

3. Let $x \in X$ and $z \in Z$

$$\begin{aligned}(0_{XY} \rhd \beta)(x,z) &= \bigwedge_{y \in Y} 0_{XY}(x,y) \rhd \beta(y,z) \\ &= \bigwedge_{y \in Y} [0_{XY}(x,y) \leq \beta(y,z)] \vee \beta(y,z) \\ &= \bigwedge_{y \in Y} 1 \vee (\beta(y,z)) = 1 = \nabla_{XZ}(x,z)\end{aligned}$$

So we have $(0_{XY} \rhd \beta) = \nabla_{XZ}$.

□

Proposition 2.2.2. *[FK15]*
Let $\alpha : A \rightharpoondown B$, $\beta : A \rightharpoondown B$ and $\beta_\lambda : A \rightharpoondown B$ $(\lambda \in \Lambda)$ be fuzzy relations. Then we have Basic properties of Dedekind categories:

1. $(\alpha \cdot 0_{YZ} = 0_{XZ})$ *and* $(0_{VX} \cdot \alpha = 0_{VY})$.
2. *If* $(\alpha \sqsubseteq \alpha')$ *and* $(\beta \sqsubseteq \beta')$ *then* $(\alpha \cdot \beta \sqsubseteq \alpha' \cdot \beta')$.
3. *If* $(\alpha \sqsubseteq \alpha')$ *and* $(\beta \sqsubseteq \beta')$ *then* $(\alpha' \rhd \beta \sqsubseteq \alpha \rhd \beta')$.
4. $\alpha \sqcap (\sqcup_k \alpha_k) = \sqcup_k(\alpha \sqcap \alpha_k)$.
5. $\alpha \cdot (\sqcap_j \beta_j) \cdot \gamma \sqsubseteq \sqcap_j \alpha \cdot \beta_j \cdot \gamma$, $\alpha \cdot (\sqcup_j \beta_j) \cdot \gamma = \sqcup_j \alpha \cdot \beta_j \cdot \gamma$.
6. $id_X^{\#} = id_X$, $0_{XY}^{\#} = 0_{YX}$. $\nabla_{XY}^{\#} = \nabla_{YX}$,
7. $(\sqcap_k \alpha_k)^{\#} = \sqcap_k \alpha_k^{\#}$, $(\sqcup_k \alpha_k)^{\#} = \sqcup_k \alpha_k^{\#}$.
8. $(\sqcup_k \alpha_k) \rhd \beta = \sqcap_k(\alpha_k \rhd \beta)$.
9. $\alpha \sqsubseteq \alpha \cdot \alpha^{\#} \cdot \alpha$.
10. $\alpha \cdot \beta \sqcap \gamma \sqsubseteq (\alpha \sqcap \gamma \cdot \beta^{\#}) \cdot (\beta \sqcap \alpha^{\#} \cdot \gamma)$.
11. $[f \cdot (\beta \sqcap \beta') \cdot g^{\#} = (f \cdot \beta \cdot g^{\#}) \sqcap (f \cdot \beta' \cdot g^{\#})]$, *for all* $f : X \to Y$, *and for all* $g : W \to Z$.
12. $[f \cdot (\beta \Rightarrow \beta') \cdot g^{\#} = (f \cdot \beta \cdot g^{\#}) \Rightarrow (f \cdot \beta' \cdot g^{\#})]$, *for all* $f : X \to Y$, *for all* $g : W \to Z$.
13. *If* $f \sqsubseteq g$ *then* $f = g$, *for all* $f, g : X \to Y$.
14. *If* $(u \sqsubseteq id_X)$ *then* $(u^{\#} = u)$,
15. *If* $(u \sqsubseteq id_X)$ *and* $(v \sqsubseteq id_X)$ *then* $(u \cdot v = u \sqcap v)$, *and*
16. *If* $(u \sqsubseteq id_X)$ *and* $(v \sqsubseteq id_X)$ *then* $(u \rhd v) \sqcap id_X = (u \Rightarrow v) \sqcap id_X$.

□

Lemma 2.2.1. *Let $f : A \rightharpoondown B$, $\alpha, \beta : B \rightharpoondown C$, and $g : C \rightharpoondown D$ be fuzzy relations.*

If $ff^{\#} \sqsubseteq id$ and $g^{\#}g \sqsubseteq id$ then we have $f \cdot (\alpha \sqcap \beta) \cdot g = (f \cdot \alpha \cdot g) \sqcap (f \cdot \beta \cdot g)$

□

2.3 Fuzzy Power Set

The **fuzzy power set** $\wp_f(Y)$ of a set X is the set of all fuzzy relations $\pi : I \rightharpoondown Y$, where I denotes a singleton set $\{*\}$. A fuzzy relation π in $\wp_f(Y)$ is called a fuzzy relation into Y . We will identify a point y of Y with a (crisp) fuzzy relation $\hat{y} : I \rightharpoondown Y$ such that $\hat{y}(*, y') = 1$ if $y' = y$ and $\hat{y}(*, y') = 0$ otherwise.

Let $B : I \rightharpoondown Y$ be a fuzzy relation and $y \in Y$. The restriction B_y of B on y is a fuzzy relation into Y such that

$$B_y(*, y') = \begin{cases} B(*, y) & y' = y, \\ 0 & \text{Otherwise} \end{cases}$$

It is trivial that $B = \sqcup_{y \in Y} B_y$.

A fuzzy relation $B : I \rightharpoondown Y$ is **finite** if $B_y = 0_{IY}$ except for a finite number of $y \in Y$, or equivalently if there is a finite subset $J \subseteq Y$ such that $B = \sqcup_{y \in J} B_y$.

Then we have fuzzy power set is equal to a morphism set $FRel(I, Y)$,

$$\wp_f(Y) = FRel(I, Y)$$

Chapter 3

Fuzzy Relational Operators Using Relational Calculus

One of the problem for relational database from Codd[Cod70] is computing the imprecise data. For example, there is a person making survey about 'height' in the class. Ahmad forgot the "exact" value of his height. Since his tall is less than Adam's height and more than Kotaro's height then he just write "tall".

Name	Height
Adam	175
Eva	169
George	161
Kotaro	165
Ahmad	Tall
Dimas	159
Siti	160

Using the relational database, we can not query the table, since the table should be in exact value and also the query should be in "exact" operation.
In fuzzy theory, we can deal with vagueness of data.
For example, we can define fuzziness of data.

For example, we divided become 3 classification "short" from 0 to 155, "medium" from 155 to 165, "tall" from 165 to 180. We also can make function to define the "degree" value of classification.

In this chapter, we introduce the notions of fuzzy relational database and also its operation/query process in relational calculus such as "selection", "projection", "injection", and "natural join".

3.1 Fuzzy Relational Database

In this section, we review a formalization which is introduced by [OK00] and introduce new formalization of database operations.

Definition 3.1.1. *Let A be a finite set of attributes. For any attribute $a \in A$, we define a set domain of attribute a with D_a. We denote the product set $\prod_{a\in X} D_a$ by D_X for finite subset X of A.*

Definition 3.1.2. *A* **database scheme** $R = (A, \{D_a\}_{a \in A})$ *is a pair of an attribute set and a class of attribute domains. Letting* $a \in A$ *and* $Y \sqsubset X \sqsubseteq A$*, we denote a* ***projection function*** *from* D_X *to* D_a *by* $\rho_{Xa} : \prod_{a \in X} D_a \to D_a$ *and a* ***projection function*** *from* D_X *to* D_Y *by* $\rho_{X,Y} = \sqcap_{a \in Y} \rho_{Xa} \cdot \rho^{\#}_{Ya} : D_X \rightharpoondown D_Y$.

Example 3.1.1. *Let* $A = \{a, b, c, d\}$*,* $X = \{a, b, c\}$*, and* $Y = \{b, c\}$ *with domain of attribute* $D_a = \{a_1, a_2\}$*,* $D_b = \{b_1, b_2\}$*,* $D_c = \{c_1\}$*, and* $D_d = \{d_1\}$*. Then:*
$D_X = D_a \times D_b \times D_c = \{(a_1, b_1, c_1), (a_1, b_2, c_1), (a_2, b_1, c_1), (a_2, b_2, c_1)\}$,
$D_Y = D_b \times D_c = \{(b_1, c_1), (b_2, c_1)\}$,
projection function D_X *to* D_a,
$\rho_{Xa} = \{((a_1, b_1, c_1), (a_1)),\ ((a_1, b_2, c_1), (a_1)),\ ((a_2, b_1, c_1), (a_2)),$
$((a_2, b_2, c_1), (a_2))\}$,
projection function D_X *to* D_Y
$\rho_{XY} = \{((a_1, b_1, c_1), (a_1, b_1)), ((a_1, b_2, c_1), (a_1, b_2)),\ ((a_2, b_1, c_19), (a_2, b_1)),$
$((a_2, b_2, c_1), (a_2, b_2))\}$

Proposition 3.1.1. *If* $X \sqsubseteq Y \sqsubseteq Z$*,then:*

$$\rho^{\#}_{Y,X} \cdot \rho_{Y,X} \sqsubseteq id_X,\ id_Y \sqsubseteq \rho_{Y,X} \cdot \rho^{\#}_{Y,X} \text{ and } \rho_{Z,X} = \rho_{Z,Y} \cdot \rho_{Y,X}.$$

Proof. We followed for Okuma and Kawahara[OK00] □

Definition 3.1.3. *A* **database relation** r *is a fuzzy relation* $r : D_A \rightharpoondown D_A$ *which satisfies* $r \sqsubseteq \mathrm{id}_{D_A}$*. We can consider a fuzzy relation* r *as a function* $r : D_A \times D_A \to [0, 1]$ *and for a tuple* $t \in D_A$*, we call* $r(t, t)$ *the fuzzy value of* t*.*

Example 3.1.2. *Let set of attributes* $A = \{X, Y\}$ *with domain of attribute* X*,* $D_X = \{x_1, x_2\}$ *and domain of attribute* Y*,* $D_Y = \{y_1, y_2\}$*. We have database relation* $r \sqsubseteq \mathrm{id}_{D_A}$*,* $r = \{((x_1, y_2), (x_1, y_2), 0.5), ((x_2, y_1), (x_2, y_1), 0.9)\}$*. Then we have 2 tuple* $t_1 = (x_1, y_2)$ *with* $r(t_1, t_1) = 0.5$ *and* $t_2 = (x_2, y_1)$ *with* $r(t_2, t_2) = 0.9$*. Then we can make abbreviation of* r *as showed in Table 3.1.*

r	X	Y	Fuzzy Value
	x_1	y_2	0.5
	x_2	y_1	0.9

Table 3.1: Table of r

Lemma 3.1.1. *Let* $r_1, r_2 : D_A \rightharpoondown D_A$ *be database relation, we have*

$$r_1 = r_1^{\#}, \text{ and } r_1 \cdot r_2 = r_1 \sqcap r_2.$$

Proof. We omitted from Proposition 2.2.2. (15) and (16). □

3.2 Fuzzy Database Operations

3.2.1 Selection

Definition 3.2.1. *Let f and r be database relations f, $r : D_A \rightharpoondown D_A$. The* **selection** $\sigma_f(r)$ *of r with f is the database relation defined by*

$$\sigma_f(r) = r \cdot f : D_A \rightharpoondown D_A.$$

Table 3.2: Table Relation High Experience High Salary

r_H	Name	Job	Experience	Salary	Fuzzy Value
	John	Engineer	8	60000	0.67
	Ashok	Manager	9	70,000	0.80
	Mary	Secretary	8	40,000	0.50
	James	Engineer	12	80,000	1.00
	Robin	Engineer	9	60,000	0.80

Example 3.2.1. *Consider $A = \{Name, Job, Experience, Salary\}$, we abbreviate as $A = \{N, J, E, S\}$, also we have domain of A $D_A = D_N \times D_J \times D_E \times D_S$, and database relation $r_H : D_A \times D_A \to [0, 1]$ (cf. Table 3.2). We select the relation with salary more than or equal to 60000.*
for all $n \in N, j \in J, e \in E$ and $s \in S$

We define $f : D_A \rightharpoondown D_A$, *by* $f((n, j, e, s), (n, j, e, s)) = \begin{cases} 1 & s > 60000, \\ 0 & \textit{otherwise.} \end{cases}$

The **selection** $\sigma_f(r_H) = r \cdot f : D_A \rightharpoondown D_A$ *is shown in Table 3.3. We denote* $\sigma_f(r_H)$ *by* $\sigma_{Salary\ >60000}(r_H)$.

Table 3.3: Fuzzy Selection $\sigma_{Salary>60000}\,(r_H)$

$\sigma_f(r_H)$	Name	Job	Experience	Salary	Fuzzy Value
	Ashok	Manager	9	70000	0.8
	James	Engineer	12	80000	1

Proposition 3.2.1. *Let f, r_1, and r_2 be database relations f, r_1, $r_2 : D_A \rightharpoondown D_A$.*

1. $\sigma_f\,(r_1 \sqcup r_2) = \sigma_f(r_1) \sqcup \sigma_f(r_2)$, *and*
2. $\sigma_f\,(r_1 \sqcap r_2) = \sigma_f(r_1) \sqcap \sigma_f(r_2)$.

Proof. 1. $\sigma_f\,(r_1 \sqcup r_2) = (r_1 \sqcup r_2) \cdot f = (r_1 \cdot f) \sqcup (r_2 \cdot f) = \sigma_f(r_1) \sqcup \sigma_f(r_2)$.

2. Since $f \sqsubseteq \mathrm{id}$, so we get $\sigma_f(r_1 \sqcap r_2) = (r_1 \sqcap r_2) \cdot f = r_1 \sqcap r_2 \sqcap f$ $= (r_1 \cdot f) \sqcap (r_2 \cdot f) = \sigma_f(r_1) \sqcap \sigma_f(r_2)$ by Lemma 3.1.1.

□

Proposition 3.2.2. *Let f, r, and g be database relations $f, g, r : D_A \rightharpoondown D_A$.*

1. $\sigma_{f \sqcup g}(r) = \sigma_f(r) \sqcup \sigma_g(r)$, *and*
2. $\sigma_{f \sqcap g}(r) = \sigma_f(r) \sqcap \sigma_g(r)$.

Proof. 1. $\sigma_{f \sqcup g}(r) = r.(f \sqcup g) = (r.f) \sqcup (r.g) = \sigma_f(r) \sqcup \sigma_g(r)$.

2. Since $r \sqsubseteq \mathrm{id}$, so we get $\sigma_{f \sqcap g}(r) = r \cdot (f \sqcap g) == r \sqcap f \sqcap g = (r \sqcap f) \sqcap (r \sqcap g)$ $= (r \cdot f) \sqcap (r \cdot g) = \sigma_f(r) \sqcap \sigma_g(r)$ by Lemma 3.1.1.

□

3.2.2 Projection

Definition 3.2.2. *Let $X \sqsubseteq A$ and $r : D_A \rightharpoondown D_A$ a database relation. The* **projection** *$\pi_{A,X}(r)$ is the database relation defined by*

$$\pi_{A,X}(r) = \rho^{\#}_{A,X} \cdot r \cdot \rho_{A,X} : D_X \rightharpoondown D_X.$$

Example 3.2.2. *We follow the Example 3.2.1, let $A = \{N, J, E, S\}$, $X = \{J, S\}$, with database relation $r_H : D_A \times D_A \to [0,1]$ be database relation defined by Tabel 3.2. We would like to project the r_H with the set of attributes A to the set of attribute X, then:*

$$\pi_{A,X}(r_H) = \rho^{\#}_{A,X} \cdot r_H \cdot \rho_{A,X}$$

Table 3.4: Fuzzy Projection $\pi_{A,X}(r_H)$

$\pi_{A,Y}(r_H)$	Job	Salary	Fuzzy Value
	Engineer	80000	1
	Manager	70000	0.8
	Secretary	40000	0.5
	Engineer	60000	0.8

Table 3.4 is result of the projection to the set of attribute $X = \{Job, Salary\}$.

Proposition 3.2.3. *Let $X \sqsubseteq A$ and $r_1 : D_A \rightharpoondown D_A$, $r_2 : D_A \rightharpoondown D_A$ be database relations.*

1. $\pi_{A,X}(r_1 \sqcup r_2) = \pi_{A,X}(r_1) \sqcup \pi_{A,X}(r_2)$, *and*
2. $\pi_{A,X}(r_1 \sqcap r_2) \sqsubseteq \pi_{A,X}(r_1) \sqcap \pi_{A,X}(r_2)$.

Proof. 1. $\pi_{A,X}(r_1 \sqcup r_2) = \rho^{\#}_{A,X} \cdot (r_1 \sqcup r_2) \cdot \rho_{A,X} = (\rho^{\#}_{A,X} \cdot r_1 \cdot \rho_{A,X}) \sqcup (\rho^{\#}_{A,X} \cdot r_2 \cdot \rho_{A,X}) = \pi_{A,X}(r_1) \sqcup \pi_{A,X}(r_2)$.

2. By Proposition 2.2.2.5, we have $\pi_{A,X}(r_1 \sqcap r_2) = \rho^{\#}_{A,X} \cdot (r_1.r_2) \cdot \rho_{A,X} \sqsubseteq (\rho^{\#}_{A,X} \cdot r_1 \cdot \rho_{A,X}) \sqcap (\rho^{\#}_{A,X} \cdot r_2 \cdot \rho_{A,X}) = \pi_{A,X}(r_1) \sqcap \pi_{A,X}(r_2)$.

□

Example 3.2.3. *Let $A = \{X, Y\}$ be a set of attributes, with domain of attribute X, $D_X = \{x_1, x_2\}$, and domain of attribute Y, $D_Y = \{y_1, y_2\}$. Consider the database relations $r_1, r_2 : D_A \rightharpoonup D_A$ defined by Table 3.5.*

Table 3.5: Relation r_1 and r_2

r_1

X	Y	Fuzzy Value
x_1	y_1	1
x_2	y_2	1

r_2

X	Y	Fuzzy Value
x_1	y_2	1
x_2	y_1	1

From Table 3.5, then we have, $\pi_{A,Y}(r_1) = \pi_{A,Y}(r_2) = id_Y$ (as we can see in Table 3.6). But, $\pi_Y(r_1 \sqcap r_2) = \oslash \neq id_Y$, so we have $\pi_Y(r_1 \sqcap r_2) \neq \pi_{A,Y}(r_1) \sqcap \pi_{A,Y}(r_2)$.

Table 3.6: Fuzzy Projection r_1 and r_2

$\pi_{A,Y}$

Y	Fuzzy Value
y_1	1
y_2	1

3.2.3 Injection

Definition 3.2.3. *Let $X \sqsubseteq A$ and $r_X : D_X \rightharpoonup D_X$ be a database relation. The* **injection** *$\eta_{X,A}(r_X)$ of r_X to A is the database relation defined by*

$$\eta_{X,A}(r_X) = (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X}) \sqcap \mathrm{id}_A : D_A \rightharpoonup D_A.$$

Example 3.2.4. *We continue from Example 3.2.3, let $U = \{X, Y, Z\}$, domain of attribute Z, $D_Z = \{z_1, z_2\}$. Then we can the result $\eta_{A,U}(r_1)$ and $\eta_{A,U}(r_2)$ shown in Table 3.7*

Table 3.7: Injection Relation r_1 and r_2

$\eta_{A,U}(r_1)$

X	Y	Z	Fuzzy Value
x_1	y_1	z_1	1
x_1	y_1	z_2	1
x_2	y_2	z_1	1
x_2	y_2	z_2	1

$\eta_{A,U}(r_2)$

X	Y	Z	Fuzzy Value
x_1	y_2	z_1	1
x_1	y_2	z_2	1
x_2	y_1	z_1	1
x_2	y_1	z_2	1

Proposition 3.2.4. *Let $X \sqsubseteq A \sqsubseteq B$ and $r_X, r'_X : D_X \rightharpoonup D_X$ be a database relation.*

1. $\eta_{X,A}(r_X \sqcup r'_X) = \eta_{X,A}(r_X) \sqcup \eta_{X,A}(r'_X)$,
2. *If* $r_X \sqsubseteq r'_X$ *then* $\eta_{X,A}(r_X) \sqsubseteq \eta_{X,A}(r'_X)$,
3. $\eta_{X,A}(r_X \cdot r'_X) = \eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X)$,
4. $\eta_{X,A}(r_X \sqcap r'_X) = \eta_{X,A}(r_X) \sqcap \eta_{X,A}(r'_X)$, *and*
5. $\eta_{A,B}(\eta_{X,A}(r_X)) = \eta_{X,B}(r_X)$.

Proof. Suppose that r_X and r'_X be database relation, so $r_X \sqsubseteq id_X$, and $r'_X \sqsubseteq id_X$

1. Using Proposition 2.2.2.5,

$$\begin{aligned}
\eta_{X,A}(r_X \sqcup r'_X) &= \rho_{A,X} \cdot (r_X \sqcup r'_X) \cdot \rho^{\#}_{A,X} \sqcap id_A \\
&= \rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcup \rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X} \sqcap id_A \\
&= \rho_{A,X} \cdot (r_X \sqcup r'_X) \cdot \rho^{\#}_{A,X} \sqcap id_A \\
&= (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcap id_A) \sqcup (\rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X} \sqcap id_A) \\
&= \eta_{X,A}(r_X) \sqcup \eta_{X,A}(r'_X).
\end{aligned}$$

2. $\eta_{X,A}(r_X) = (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcap id_A) \sqsubseteq (\rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X} \sqcap id_A) = \eta_{X,A}(r'_X)$.

3. ($\sqsubseteq$) Since $r_X \cdot r'_X \sqsubseteq r_X$ and $r_X \cdot r'_X \sqsubseteq r'_X$, we have $\eta_{X,A}(r_X \cdot r'_X) \sqsubseteq \eta_{X,A}(r_X)$ and $\eta_{X,A}(r_X \cdot r'_X) \sqsubseteq \eta_{X,A}(r'_X)$. Then we get $\eta_{X,A}(r_X \cdot r'_X) \sqsubseteq \eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X)$

   ($\sqsupseteq$) Since, $\rho^{\#}_{A,X} \cdot \rho_{A,X} \sqsubseteq id_X$, we have,

$$\begin{aligned}
\eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X) &= ((\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X}) \sqcap id_A) \cdot ((\rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X}) \sqcap id_A) \\
&\sqsubseteq \rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X} \\
&\sqsubseteq \rho_{A,X}(r_X \cdot r'_X)\rho^{\#}_{A,X} \\
\eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X) &\sqsubseteq \rho_{A,X}(r_X \cdot r'_X)\rho^{\#}_{A,X}.
\end{aligned}$$

   and also we have,

$$\begin{aligned}
((\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X}) \sqcap id_A) \cdot ((\rho_{A,X} \cdot r'_X \cdot \rho^{\#}_{A,X}) \sqcap id_A) &\sqsubseteq id_A \\
\eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X) &\sqsubseteq id_A.
\end{aligned}$$

   Then we have, $\eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X) \sqsubseteq \rho_{A,X}(r_X \cdot r'_X).\rho^{\#}_{A,X} \sqcap id_A = \eta_{X,A}(r_X \cdot r'_X)$.

4. Using Lemma 3.1.1, $\eta_{X,A}(r_X \sqcap r'_X) = \eta_{X,A}(r_X \cdot r'_X)$. Using (3), $\eta_{X,A}(r_X \cdot r'_X) = \eta_{X,A}(r_X) \cdot \eta_{X,A}(r'_X) = \eta_{X,A}(r_X) \sqcap \eta_{X,A}(r'_X)$

5. ($\sqsubseteq$) We note that $\rho_{B,X} = \rho_{B,A} \cdot \rho_{A,X}$.
We have

$$\begin{aligned}
\eta_{A,B}(\eta_{X,A}(r_X)) &= \rho_{B,A} \cdot (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcap id_A)\rho^{\#}_{B,A} \sqcap id_B \\
&\sqsubseteq (\rho_{B,A} \cdot \rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho^{\#}_{B,A}) \sqcap (\rho_{B,A} \cdot id_A \cdot \rho^{\#}_{B,A}) \sqcap id_B \\
&\sqsubseteq (\rho_{B,A} \cdot \rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho^{\#}_{B,A}) \sqcap id_B \\
&= \rho_{B,X} \cdot r_X \cdot \rho^{\#}_{B,X} \sqcap id_B, \text{ and} \\
\eta_{A,B}(\eta_{X,A}(r_X)) &\sqsubseteq \eta_{X,B}(r_X)
\end{aligned}$$

($\sqsupseteq$) Since $r_X \sqsubseteq id_X$, We have

$$\begin{aligned}
\eta_{X,B}(r_X) &= \rho_{B,X} \cdot r_X \cdot \rho^{\#}_{B,X} \sqcap id_B \\
&= \rho_{B,X} \cdot (r_X \sqcap id_X) \cdot \rho^{\#}_{B,X} \sqcap id_B \\
&\sqsubseteq (\rho_{B,X} \cdot r_X \cdot \rho^{\#}_{B,X}) \sqcap (\rho_{B,X} \cdot id_X \cdot \rho^{\#}_{B,X}) \sqcap id_B \\
&= (\rho_{B,X} \cdot r_X \cdot \rho^{\#}_{B,X}) \sqcap (\rho_{B,X} \cdot \rho^{\#}_{B,X}) \sqcap id_B \\
&= (\rho_{B,A} \cdot \rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho^{\#}_{B,A}) \sqcap (\rho_{B,A} \cdot \rho^{\#}_{B,A}) \sqcap id_B \qquad (*) \\
&\sqsubseteq \rho_{B,A} \cdot (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho^{\#}_{B,A} \sqcap \rho^{\#}_{B,A} \cdot \rho_{B,A} \cdot \rho^{\#}_{B,A}) \sqcap id_B \\
&\sqsubseteq \rho_{B,A} \cdot (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \cdot \rho^{\#}_{B,A} \sqcap id_A \cdot \rho^{\#}_{B,A}) \sqcap id_B \\
&\sqsubseteq \rho_{B,A} \cdot (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcap id_A \cdot \rho^{\#}_{B,A} \cdot \rho_{B,A}) \cdot \rho^{\#}_{B,A} \sqcap id_B \\
&\sqsubseteq \rho_{B,A} \cdot (\rho_{A,X} \cdot r_X \cdot \rho^{\#}_{A,X} \sqcap id_A) \cdot \rho^{\#}_{B,A} \sqcap id_B
\end{aligned}$$

So we have $\eta_{X,B}(r_X) \sqsubseteq \eta_{A,B}(\eta_{X,A}(r_X))$.

(*) We note that $id_B \sqsubseteq \rho_{B,X} \cdot \rho^{\#}_{B,X}$ and $id_B \sqsubseteq \rho_{B,A} \cdot \rho^{\#}_{B,A}$, then we get $\rho_{B,X} \cdot \rho^{\#}_{B,X} \sqcap id_B = \rho_{B,A} \cdot \rho^{\#}_{B,A} \sqcap id_B$.

□

3.2.4 Natural Join

Definition 3.2.4. *Let $X, Y \sqsubseteq A$ and $Z = X \sqcup Y$. We consider database relations $r_X : D_X \rightharpoonup D_X$ and $r_Y : D_Y \rightharpoonup D_Y$. The* **natural join** *$r_X \bowtie r_Y$ of r_X and r_Y is the database relation defined by*

$$r_X \bowtie r_Y = \eta_{X,Z}(r_X) \cdot \eta_{Y,Z}(r_Y) : D_Z \rightharpoonup D_Z.$$

Table 3.8: r_{LIKES}

LIKES

Student	Course	Fuzzy Value
John	DBMS	0.90
Mary	DBMS	0.70
John	AI	0.80
Ashok	AI	0.95

Table 3.9: r_{TEACH}

TEACH

Teacher	Course	Fuzzy Value
Rao	DBMS	0.80
Rao	AI	0.60
Johnson	DBMS	0.60
Johnson	AI	0.90

Table 3.10: Fuzzy Natural Join $r_{LIKES} \bowtie r_{TEACH}$

Student	Teacher	Course	Fuzzy Value
John	Rao	DBMS	0.80
John	Johnson	DBMS	0.60
John	Rao	AI	0.60
John	Johnson	AI	0.80
Marry	Rao	DBMS	0.70
Marry	Johnson	DBMS	0.60
Ashok	Rao	AIS	0.60
Ashok	Johnson	AI	0.90

Example 3.2.5. *Let $LIKES = \{Student, Course\}$ and $TEACH = \{Teacher, Course\}$. Consider $r_{LIKES} : D_{LIKES} \times D_{LIKES} \to [0,1]$ defined by Table 3.8 and $r_{TEACH} : D_{TEACH} \times D_{TEACH} \to [0,1]$ defined by Table 3.9.*

We note that:$LIKES \sqcup TEACH = \{Student, Course, Teacher\}$. The natural join of r_{LIKES} and r_{TEACH} is expressed in Table 3.10

Proposition 3.2.5. *Let $X, Y, Z \sqsubseteq A$ be a attribute set, $r_X, r'_X : D_X \rightharpoonup D_X$, $r_Y, r'_Y : D_Y \rightharpoonup D_Y$, $r_Z : D_Z \rightharpoonup D_Z$ a fuzzy database relation, then:*

1. $(r_X \bowtie r_Y) \bowtie r_Z = \alpha \bowtie (\beta \bowtie \gamma)$.
2. $(\sqcup_{\lambda\in\Lambda}(r_X)_\lambda) \bowtie r_Y = \sqcup_{\lambda\in\Lambda}((r_X)_\lambda \bowtie r_Y)$.
3. $(\sqcap_{\lambda\in\Lambda}(r_X)_\lambda) \bowtie r_Y = \sqcap_{\lambda\in\Lambda}((r_X)_\lambda \bowtie r_Y)$.
4. *If $r_X \sqsubseteq r'_X$ and $r_Y \sqsubseteq r'_Y$ then $r_X \bowtie r_Y \sqsubseteq r'_X \bowtie r'_Y$.*

Proof. 1. We have $(r_X \bowtie r_Y) \bowtie r_Z = (\eta_{X\cup Y, X\cup Y\cup Z}(r_X \bowtie r_Y) \cdot \eta_{Z, X\cup Y\cup Z}(r_Z)$

$$= (\eta_{X\cup Y, X\cup Y\cup Z}((\eta_{X, X\cup Y}(r_X)) \cdot (\eta_{Y, X\cup Y}(r_Y)) \cdot \eta_{Z, X\cup Y\cup Z}(r_Z)$$
$$= ((\eta_{X\cup Y, X\cup Y\cup Z}(\eta_{X, X\cup Y}(r_X))) \cdot (\eta_{X\cup Y, X\cup Y\cup Z}((\eta_{Y, X\cup Y}(r_Y))) \cdot \eta_{Z, X\cup Y\cup Z}(r_Z)$$
$$= (\eta_{X, X\cup Y\cup Z}(r_X) \cdot \eta_{Y, X\cup Y\cup Z}(r_Y)) \cdot \eta_{Z, X\cup Y\cup Z}(r_Z)$$
$$= \eta_{X, X\cup Y\cup Z}(r_X) \cdot (\eta_{Y, X\cup Y\cup Z}(r_Y) \cdot \eta_{Z, X\cup Y\cup Z}(r_Z))$$
$$= \eta_{X, X\cup Y\cup Z}(r_X) \cdot (\eta_{Y\cup Z, X\cup Y\cup Z}(\eta_{Y, Y\cup Z}(r_Y)) \cdot (\eta_{Y\cup Z, X\cup Y\cup Z}(\eta_{Z, Y\cup Z}(r_Z)))$$
$$= \eta_{X, X\cup Y\cup Z}(r_X) \cdot (\eta_{Y\cup Z, X\cup Y\cup Z}(\eta_{Y, Y\cup Z}(r_Y) \cdot \eta_{Z, Y\cup Z}(r_Z))$$
$$= \eta_{X, X\cup Y\cup Z}(r_X) \cdot (\eta_{Y\cup Z, X\cup Y\cup Z}(r_Y \bowtie r_Z))$$
$$= r_X \bowtie (r_Y \bowtie r_Z).$$

Then we get $(r_X \bowtie r_Y) \bowtie r_Z = r_X \bowtie (r_Y \bowtie r_Z)$.

2. Since $\eta_{X,X\cup Y}(\sqcup_{\lambda\in\Lambda}(r_X)_\lambda) = \sqcup_{\lambda\in\Lambda}(\eta_{X,X\cup Y}(r_X)_\lambda)$, we have

$$\begin{aligned}((\sqcup_{\lambda\in\Lambda}(r_X)_\lambda) \bowtie r_Y &= (\sqcup_{\lambda\in\Lambda}\eta_{X,X\cup Y}((r_X)_\lambda)) \cdot \eta_{Y,X\cup Y}(r_Y) \\ &= \sqcup_{\lambda\in\Lambda}(\eta_{X,X\cup Y}((r_X)_\lambda) \cdot \eta_{Y,X\cup Y}(r_Y)) \\ &= \sqcup_{\lambda\in\Lambda}((r_X)_\lambda \bowtie r_Y).\end{aligned}$$

3. Since $\eta_{X,X\cup Y}(\sqcap_{\lambda\in\Lambda}(r_X)_\lambda) = \sqcap_{\lambda\in\Lambda}(\eta_{X,X\cup Y}(r_X)_\lambda)$ we have
$((\sqcap_{\lambda\in\Lambda}(r_X)_\lambda) \bowtie r_Y = (\sqcap_{\lambda\in\Lambda}\eta_{X,X\cup Y}((r_X)_\lambda)) \cdot \eta_{Y,X\cup Y}(r_Y)$
Since $\eta_{X,X\cup Y}((r_X)_\lambda), \eta_{X,X\cup Y}(r_Y) \sqsubseteq \mathrm{id}$, we have

$$\begin{aligned}(\sqcap_{\lambda\in\Lambda}\eta_{X,X\cup Y}((r_X)_\lambda)) \cdot \eta_{Y,X\cup Y}(r_Y) &= \sqcap_{\lambda\in\Lambda}(\eta_{X,X\cup Y}((r_X)_\lambda) \cdot \eta_{Y,X\cup Y}(r_Y)) \\ &= \sqcap_{\lambda\in\Lambda}((r_X)_\lambda \bowtie r_Y).\end{aligned}$$

4. Prop 2.2.2.5 and Prop 3.2.4.2, we have
$r_X \bowtie r_Y = \eta_{X,X\sqcup Y}(r_X) \cdot \eta_{Y,X\sqcup Y}(r_Y) \sqsubseteq \eta_{X,X\sqcup Y}(r'_X) \cdot \eta_{Y,X\sqcup Y}(r'_Y) = r'_X \bowtie r'_Y$.

□

Chapter 4

Equivalence Relational Database and Its Operations

The classical(boolean) relation database has introduced by E.F Codd [Cod70]. According to the classic relation database all the information in it, have to involve precisely defined values (atomic). One of the advantages of the relational model is soundness for data consistency.

In classical database operations, we must find strictly query operation to get certain result. For example, we would like to go to Fukuoka city from Tokyo city. We also have Table 4.1 detail transportation schedule. If we would like to "select" trip that From "Tokyo" to "Fukuoka" with schedule departure at "8 AM" and arrive at "13 AM". In traditional query, we get nothing since there is no option with strict location and time. Using equivalence of fuzzy relational database, we can tolerance "location" and "time". We consider less then 100 Km still in "Tokyo Area" then we have {Shinagawa, Narita, Haneda} still in Tokyo Area. Also, we can tolerance the time such as "morning" from 7 AM until 12 AM and also "noon" from after 12 AM until 2 PM. Using equivalence class, we can find the cooperative option. In this paper, we would like to formalize the operatin then we can get cooperative result.

Transportation	Start Point	Finish Point	Departure	Arrival
Nozomi Shinkansen	Shinagawa	Hakata	08.30	13.30
Peach Airlines	Narita	Fukuoka	10.30	12.00
Skymark Airlines	Haneda	Fukuoka	14.30	15.50
Hikari Shinkansen	Osaka	Hakata	08.00	12.00

Table 4.1: Transportation Schedule

These imprecise information have been focused on Zadeh's fuzzy set theory and fuzzy logic [Zad65]. The fuzzy set theory and fuzzy logic provide mathematical framework to deal with the imprecise information in a fuzzy relational databases. Kawahara, et al. developed an algebraic formalization of fuzzy relations using relational calculus [Kaw88, KF99, MK95b]. Okuma, et al. introduced a relational database model using a theory of relational calculus on the Dedekind Category [OK00]. Since the Dedekind Category includes the category of fuzzy relations, we can formalize a fuzzy relational database model using relational calculus on the category of fuzzy relations. Further, we formalize the notion of database operations using formulas of relational calculus. We note that the same formulas of relational calculus can be used for a traditional non-fuzzy relational database model. So we can consider our fuzzy relational database theory using intuition of the traditional database operations. We show several elementary properties of database operations using relational calculus.

As an extension of the degree of membership concept for sets elements, we have similarity relationship. Here the domain elements are considered as having varying degrees of similarity, replacing the idea of exact equality / inequality. To deal with fuzzy data constraint, Zadeh has introduced the concept of particularization (restriction) of fuzzy relation due to a fuzzy proposition[Zad71]. The particularization of fuzzy relational database due to a set of fuzzy integrity constraints can be computed by combining the fuzzy propositions associated with these integrity using relational calculus.

Using fuzzy similarity relation from Zadeh's model, Shenoi and Melton[SMF90] tried to implement the fuzzy equivalence relation to the relational database. Each definition of notion in their paper are defined formally using relational formulas, and also we show corresponding theorem using relation calculus. In this research, we formalize fuzzy equivalence relation in relational database to make easy for analyzing fuzzy equivalence relational database system. Then, we hope our formalization can help to build prove system by mathematical approach. There are some applications of fuzzy equivalence relational database for clustering method such as: manage hyperlinks of we pages[KMK13], e-commerce for customer relationship management(CRM)[MW07]. Since this paper is mathematical approach to formalize FRDB, the numbering of each equations include in some lemmas, theorems and propositions.

4.1 Similarity Domain and Fuzzy Relation

In 1982, Buckles-Petry introduced representation of similarity for domain of attribute database[BP82]. We would like to formalize the properties of fuzzy similarity threshold and equivalence relation domain using relational calculus.

We note that A **database scheme** $R = (A, \{D_a\}_{a\in A})$ is a pair of an attribute set and a class of attribute domains. Letting $a \in A$ and $Y \sqsubset X \sqsubseteq A$, we denote a **projection function** from D_X to D_a by $\rho_{Xa} : \prod\limits_{a\in X} D_a \to D_a$ and a **projection function** from D_X to D_Y by $\rho_{X,Y} = \sqcap_{a\in Y} \rho_{Xa} \cdot \rho^{\#}_{Ya} : D_X \rightharpoonup D_Y$.

Definition 4.1.1. *Let a be an attribute with domain set D_a, a* ***similarity relation*** *of attribute a is a (homogeneous) fuzzy relation between D_a, $\psi_a : D_a \rightharpoonup$*

D_a

Next, we will define a similarity relation with a threshold value. First, we define threshold

Definition 4.1.2. *Let A be a set of attributes, D_a attribute domain with $a \in A$, and ψ_a a similarity relation of D_a, $\psi_a : D_a \rightharpoondown D_a$. We consider a threshold value, $c \in [0, 1]$. We define a $c-$**threshold relation** of attribute a, $[\psi_a]_c : D_a \rightharpoondown D_a$, by $[\psi]_c(a_1, a_2) := [c \leq \psi_a(a_1, a_2)]$, for all $a_1, a_2 \in a$*

Since, the value of a $c-$threshold relation is 0 or 1, then we can consider a $c-$threshold relation as a boolean binary relation, $[\psi_a]_c : D_a \to D_a$.

Proposition 4.1.1. *Let A be a set of attributes, D_a attribute domain with $a \in A$, ψ_a and ψ'_a similarity relations of D_a, $\psi_a, \psi'_a : D_a \rightharpoondown D_a$, and c a threshold value, then:*

1. $[(\psi_a \sqcap \psi'_a)]_c = [\psi_a]_c \sqcap [\psi'_a]_c$,
2. $[(\psi_a \sqcup \psi'_a)]_c = [\psi_a]_c \sqcup [\psi'_a]_c$, *and*
3. *If* $\psi_a \sqsubseteq \psi'_a$ ***then*** $[\psi_a]_c \sqsubseteq [\psi'_a]_c$.

Proof. Let $x, y \in D_a$,

1. Since we have $(\psi_a \sqcap \psi'_a)(x, y) = (\psi_a(x, y) \wedge \psi'_a(x, y))$,

$$\begin{aligned} [(\psi_a \sqcap \psi'_a)]_c(x, y) &= [c \leq (\psi_a \sqcap \psi'_a)(x, y)] \\ &= [c \leq (\psi_a(x, y) \wedge \psi'_a(x, y))] \\ &= [c \leq \psi_a(x, y)] \wedge [c \leq \psi'_a(x, y)] \\ &= [\psi_a]_c(x, y) \sqcap [\psi'_a]_c(x, y) \end{aligned}$$

2. Since we have $(\psi_a \sqcup \psi'_a)(x, y) = (\psi_a(x, y) \vee \psi'_a(x, y))$, we get

$$\begin{aligned} [(\psi_a \sqcup \psi'_a)]_c(x, y) &= [c \leq (\psi_a \sqcup \psi'_a)(x, y)] \\ &= [c \leq (\psi_a(x, y) \vee \psi'_a(x, y))] \\ &= [c \leq \psi_a(x, y)] \vee [c \leq \psi'_a(x, y)] \\ &= [\psi_a]_c(x, y) \sqcup [\psi'_a]_c(x, y) \end{aligned}$$

3. Since we have $\psi_a(x, y) \sqsubseteq \psi'_a(x, y)$ then $\psi_a(x, y) = \psi_a(x, y) \sqcap \psi'_a(x, y)$ then if we use threshold c we have $[\psi_a]_c(x, y) = [(\psi_a \sqcap \psi'_a)]_c(x, y) = [\psi_a]_c(x, y) \sqcap [\psi'_a]_c(x, y)$. So, we have $[\psi_a]_c(x, y) \sqsubseteq [\psi'_a]_c(x, y)$

□

Definition 4.1.3. *Let A be a set of attributes, D_a attribute domain with $a \in A$. Let θ_a be a (homogeneous) binary relation, $\theta_a : D_a \times D_a \to \{0, 1\}$. θ_a is an **equivalence relation** of attribute a which satisfies following: (i). $id_{D_a} \sqsubseteq \theta_a$, (ii). $\theta_a \cdot \theta_a \sqsubseteq \theta_a$, and (iii). $\theta_a^{\#} = \theta_a$*

Example 4.1.1. *Let a be an attribute, with domain $D_a = \{x_1, x_2, x_3\}$, and $\psi_a : D_a \rightharpoondown D_a$ be similarity relation of D_a as shown in Table 4.2. In Table 4.2 we can see for example $\psi_a(x_1, x_2) = \psi_a(x_2, x_1) = 0.5$, $\psi_a(x_1, x_1) = \psi_a(x_2, x_2) = \psi_a(x_3, x_3) = 1$. We consider threshold value $c = 0.5$, then we can get a table of $[\psi_a]_c$ in Table 4.3.*

	x_1	x_2	x_3
x_1	1	0.5	0
x_2	0.5	1	0
x_3	0	0	1

Table 4.2: $\psi_a : D_a \rightharpoondown D_a$

	x_1	x_2	x_3
x_1	1	1	0
x_2	1	1	0
x_3	0	0	1

Table 4.3: $[\psi_a]_c : D_a \times D_a \to \{0, 1\}$

Since $id_{D_a} \sqsubseteq [\psi_a]_c$, $[\psi_a]_c^{\#} = [\psi_a]_c$, and $[\psi_a]_c \cdot [\psi_a]_c \sqsubseteq [\psi_a]_c$, then $[\psi_a]_c$ is an equivalence relation.

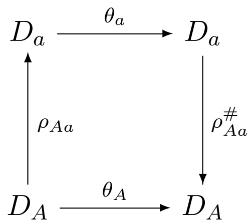
4.2 Equivalence Relational Database

In this section we define a equivalence relational database. We followed the definition of relational database by Shenoi, et al[SMF90] as a set of relation tuples τ. However, unlike an ordinary tuple, each component of a fuzzy tuple is a non-empty set of values drawn from the underlying database domain. Since we use threshold relation to define equivalence relation, then we consider that each element of tuple is a crisp of power set relation.

Definition 4.2.1. *Let D_a be a set of domain attribute with a, θ_a be an equivalence relation of D_a. We define an **equivalence class** $\theta_a(x)$ by $\theta_a(x) = \{y | (x, y) \in \theta_a\}$*

Definition 4.2.2. *Let D_a be a set of domain attribute with a, θ_a be an equivalence relation of D_a. We define the partition $D_a^{\theta_a}$, $D_a^{\theta_a} = \{\theta_a(x) | x \in D_a\}$*

Definition 4.2.3. *Let D_a be a set of domain attribute with a, θ_a be an equivalence relation of D_a. We consider $D_a^{\theta_a}$ be a set of partition of D_a. Let q a total function such that $q_a^{\#} q_a = \mathrm{id}_{D_A^{\theta_a}}$ and $qq^{\#} = \theta_a$, we call q is quotient relation defined by θ_a.*



Proposition 4.2.1. *Let A be set of all attributes, D_a attribute domain with $a \in A$, $D_A = \prod_{a \in A} D_a$. We define $\theta_A = \sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#} : D_A \rightharpoondown D_A$. Then: If $\theta_a : D_a \times D_a \to \{0,1\}$ is an equivalence relation then $\theta_A : D_A \rightharpoondown D_A$ is an equivalence relation.*

Proof. We have to prove 3 conditions:

- $\theta_A^{\#} = \theta_A$.
  Since $\theta_a^{\#} = \theta_a$, then:

$$\begin{aligned}\theta_A^{\#} &= \sqcap_{a \in A}(\rho_A a \theta_a \rho_A a^{\#})^{\#} \\ &= \sqcap_{a \in A} \rho_A a \theta_a^{\#} \rho_A a^{\#} \\ &= \sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#} \\ &= \theta_A\end{aligned}$$

- $\mathrm{id}_A \sqsubseteq \theta_A$.
  Since we have $\mathrm{id}_A \sqsubseteq \rho_{Aa} \cdot \rho_{Aa}^{\#}$ and $\mathrm{id}_a \sqsubseteq \theta_a$. Then:

$$\begin{aligned}\theta_A &= \sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#} \\ &\sqsupseteq \sqcap_{a \in A} \rho_A a \mathrm{id}_a \rho_A a^{\#} \\ &\sqsupseteq \mathrm{id}_A\end{aligned}$$

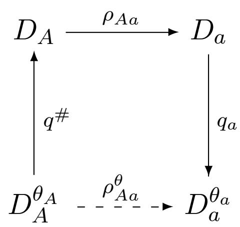
- $\theta_A \cdot \theta_A \sqsubseteq \theta_A$.

$$\begin{aligned}\theta_A \cdot \theta_A &= (\sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#}) \cdot (\sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#}) \\ &\sqsubseteq \sqcap_{a \in A}((\rho_A a \theta_a \rho_A a^{\#}) \cdot (\rho_A a \theta_a \rho_A a^{\#})) \\ &\sqsubseteq \sqcap_{a \in A}((\rho_A a \theta_a \cdot \theta_a \rho_A a^{\#})) \\ &\sqsubseteq \sqcap_{a \in A}((\rho_A a \theta_a \rho_A a^{\#})) \\ &\sqsubseteq \theta_A\end{aligned}$$

Since we have proof 3 conditions, then we can conclude that θ_A is an equivalence relation. □

Then we define θ_A is an equivalence relation of A, θ_A by $\theta_A = \sqcap_{a \in A} \rho_A a \theta_a \rho_A a^{\#} : D_A \times D_A \to \{0,1\}$.

Definition 4.2.4. *Let A be a set of attributes, D_a be a set of domain attribute, and $\theta_a = q_a q_a^{\#}$ be an equivalence relation for all $a \in A$. Then we can get $D_a^{\theta_a}$ be a set of partition of D_a. We can define θ_A be equivalence relation, and also define the partition of a set attributes A by $D_A^{\theta_A} = \prod_{a \in A} D_a^{\theta_a}$. We also define **projection equivalence function** by $\rho_{Aa}^{\theta} : D_A^{\theta_A} \to D_a^{\theta_a}$. Then we define quotient relation of A by $q_A = \sqcap_{a \in A} \rho_{Aa} q_a (\rho_{Aa}^{\theta})^{\#}$ i.e $qq^{\#} = \theta_A$ and $q^{\#} q = \mathrm{id}_{D_A^{\theta_A}}$.*



Proposition 4.2.2. *Let A be set of attributes. If $\theta_A : D_A \times D_A \to \{0,1\}$ is an equivalence relation of A then for all $a \in A$, $\rho^{\#}_{Aa}\theta_A\rho_{Aa}$ is an equivalence relation.*

Proof. We have to prove 3 conditions:

- $(\rho^{\#}_{Aa}\theta_A\rho_{Aa})^{\#} = \rho^{\#}_{Aa}\theta_A\rho_{Aa}$.
  Since $\theta^{\#}_A = \theta_A$, then:

$$\begin{aligned}(\rho^{\#}_{Aa}\theta_A\rho_{Aa})^{\#} &= (\rho^{\#}_{Aa}\theta^{\#}_A\rho_{Aa}) \\ &= (\rho^{\#}_{Aa}\theta_A\rho_{Aa})\end{aligned}$$

- $\mathrm{id}_A \sqsubseteq (\rho^{\#}_{Aa}\theta_A\rho_{Aa})$.
  Since we have $\mathrm{id}_A \sqsubseteq \rho_{Aa} \cdot \rho^{\#}_{Aa}$ and $\mathrm{id}_a \sqsubseteq \theta_a$. Then:

$$\begin{aligned}(\rho^{\#}_{Aa}\theta_A\rho_{Aa}) &\sqsupseteq (\rho^{\#}_{Aa} \cdot \rho_{Aa}) \\ &\sqsupseteq \mathrm{id}_a\end{aligned}$$

- $(\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \cdot (\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \sqsubseteq (\rho^{\#}_{Aa}\theta_A\rho_{Aa})$.

$$\begin{aligned}(\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \cdot (\rho^{\#}_{Aa}\theta_A\rho_{Aa}) &= (\rho^{\#}_{Aa}\theta_A\rho_{Aa}\mathrm{id}_a\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \\ &\sqsubseteq (\rho^{\#}_{Aa}\theta_A\rho_{Aa}\theta_a\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \\ &\sqsubseteq (\rho^{\#}_{Aa}\theta_A\theta_A\theta_A\rho_{Aa}) \\ &\sqsubseteq (\rho^{\#}_{Aa}\theta_A\rho_{Aa}) \\ &\sqsubseteq \theta_A\end{aligned}$$

Since we have proof 3 conditions, then we can conclude that $\rho^{\#}_{Aa}\theta_A\rho_{Aa}$ is equivalence relation. □

Proposition 4.2.3. *Let A be a set of attributes, and θ_A, θ'_A equivalence relations of A. Then:*

$$\theta_A \cdot (\rho_{Aa} \cdot \theta_a \cdot \rho^{\#}_{Aa}) \sqsubseteq \rho_{Aa} \cdot \theta_a \cdot \rho^{\#}_{Aa}$$

Proof. Since $\theta_A = \sqcap_{a\in A}\rho_A a\theta_a\rho_A a^{\#}$ then we have $\theta_A \sqsubseteq \rho_{Aa}\cdot\theta_a\cdot\rho^{\#}_{Aa}$. So we have $\theta_A\cdot(\rho_{Aa}\cdot\theta_a\cdot\rho^{\#}) \sqsubseteq (\rho_{Aa}\cdot\theta_a\cdot\rho^{\#}_{Aa})\cdot(\rho_{Aa}\cdot\theta_a\cdot\rho^{\#}_{A}a)$.
Since $\rho^{\#}_{Aa}\cdot\rho_{Aa} \sqsubseteq \mathrm{id}_a$ and $\theta_a\cdot\theta_a \sqsubseteq \theta_a$ then we get $\theta_A\cdot(\rho_{Aa}\cdot\theta_a\cdot\rho^{\#}) \sqsubseteq \rho_{Aa}\cdot\theta_a\cdot\rho^{\#}_{Aa}$. □

4.2.1 Simple(non redundant) sets

Let $I = \{*\}$ be a singleton set. We identify an element $x \in D_X$ as a function $x : I \to D_X$ with $x(*) = x$. Further, for two functions$x : I \to D_X$ and $y : I \to D_Y$ we define a function $x\top y : I \to D_X \times D_Y$ by $(x\top y)(*) = (x, y)$.

Definition 4.2.5. *Let Q be an object in a Dedekind category $\mathcal{D}$ with unit I. An **I-point** of Q is a **total function** $q : I \rightharpoondown Q$ i.e $q\cdot q^{\#} = \mathrm{id}_I$ and $q^{\#}\cdot q \sqsubseteq \mathrm{id}_Q$.*

Definition 4.2.6. *Let $\theta_a = q_a q_a^{\#}$ be an equivalence relation and $q_a^{\#}q_a = \mathrm{id}_{D_a^{\theta_a}}$ where $q_a : D_a \to D_a^{\theta_a}$. A relation $\tau_a : I \rightharpoondown D_a$ is a θ-*subset *if and only if $\tau_a\cdot q_a$ is an I-point of $D_a^{\theta_a}$. That is, $(\tau_a\cdot q_a)\cdot(\tau_a\cdot q_a)^{\#} = \mathrm{id}_I$ and $(\tau_a\cdot q_a)^{\#}\cdot(\tau_a\cdot q_a) \sqsubseteq \mathrm{id}_{D_a^{\theta_a}}$. The set of all θ_a-subset is denoted by $Rel_{\theta_a}(I, D_a)$.*

Example 4.2.1. *Let consider a set $D_a = \mathbf{Z} = \{0, \pm1, \pm2, \cdots\}$ of integers, a finite set $\mathbf{Z}_5 = \{\dot{0}, \dot{1}, \dot{2}, \dot{3}, \dot{4}\}$ and a function $q_a : D_a \to \mathbf{Z}_5$ defined by $q_a(x) = x \bmod 5$. Then we have $q_a^{\#}q_a = \mathrm{id}_{\mathbf{Z}_5}$ and if we set $\theta_a = q_a q_a^{\#}$. Then we have $\theta_a : D_a \rightharpoondown D_a$ is an equivalence relation, i.e. $\mathrm{id}_{D_a} \sqsubseteq \theta_a$, $\theta_a^{\#} \sqsubseteq \theta_a$ and $\theta_a\theta_a \sqsubseteq \theta_a$. So, we can consider the partition $D_a^{\theta_a} = \mathbf{Z}_5$. Let $\tau_a = \{0, 5, 10\}$ is a θ_a-subset but $\tau'_a = \{0, 1\}$ is not a θ_a-subset. We should write $\tau_a = \{(*, 0), (*, 5), (*, 10)\}$ and $\tau'_a = \{(*, 0), (*, 1)\}$, precisely. We have $\tau_a \in Rel_{\theta_a}(I, D_a)$ and $\tau'_a \notin Rel_{\theta_a}(I, D_a)$.*

Definition 4.2.7. *Let $\theta_a = q_a q_a^{\#}$ be an equivalence relation and $q_a^{\#}q_a = \mathrm{id}_{D_a^{\theta_a}}$ where $q_a : D_a \to D_a^{\theta_a}$. A subset $R \subseteq Rel_{\theta_a}(I, D_a)$ be* simple (non-redundant),

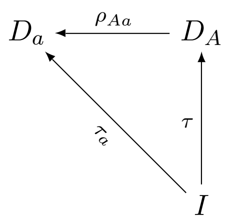
if $\tau_a\cdot q_a = \tau'_a\cdot q_a$ then $\tau_a = \tau'_a$, for all $\tau_a, \tau'_a \in R$.

Example 4.2.2. *Let $\tau_a = \{(*, 0), (*, 5)\}$, $\tau'_a = \{(*, 0)\}$ and $\tau''_a = \{(*, 1)\}$. We note τ_a, τ'_a and τ''_a are θ-subsets. Then $R_0 = \{\tau_a, \tau''_a\}$ is simple but $R_1 = \{\tau_a, \tau'_a, \tau''_a\}$ is not simple.*

Definition 4.2.8. *Let $\theta_a = q_a q_a^{\#}$ be an equivalence relation and $q_a^{\#}q_a = \mathrm{id}_{D_a^{\theta_a}}$ where $q_a : D_a \to D_a^{\theta_a}$ for all $a \in A$. We can define domain of set attributes A by $D_A = \prod_{a\in A} D_a$, and θ_A be an equivalence relation with quotient relatin of set attributes A by q_A, with $qq^{\#} = \theta_A$ and $q^{\#}q = \mathrm{id}_{D_A^{\theta_A}}$. A relation $\tau : I \rightharpoondown D_A$ is a θ-*relation *if and only if $\tau\cdot q$ is an I-point of $D_A^{\theta_A}$. The set of all θ_A-relations is denoted by $Rel_{\theta_A}(I, D_A)$.*

We note that $Rel_{\theta_A}(I, D_A) \neq \prod_{a \in A} Rel_{\theta_a}(I, D_a)$

Let $\rho_{Aa} : D_A \to D_a$. Then we can define a function $\Phi : Rel_{\theta_A}(I, D_A) \to \prod_{a \in A} Rel_{\theta_a}(I, D_a)$ by $\Phi(\tau) = (\tau \cdot \rho_a)_{a \in A}$ for all $a \in A$. And also we can define a function $\Psi : \prod_{a \in A} Rel_{\theta_a}(I, D_a) \to Rel_{\theta_A}(I, D_A)$ by $\Psi(\tau_a)_{a \in A} = \sqcap_{a \in A} \tau_a \rho_{Aa}^{\#}$.



$$Rel_{\theta_A}(I, D_A) \underset{\Psi}{\overset{\Phi}{\rightleftarrows}} \prod_{a \in A} Rel_{\theta_a}(I, D_a)$$

We note $\Phi(\Psi(\tau_a)_{a \in A}) = (\tau_a)_{a \in A}$ for all $(\tau_a)_{a \in A} \in \prod_{a \in A} Rel_{\theta_A}(I, D_A)$. But $\Psi(\Phi(\tau)) = \tau$ does not hold for any $\tau \in Rel_{\theta_A}(I, D_A)$.

Example 4.2.3. *Let* $\tau_1 = \{(*, 0), (*, 2)\}$ *and* $\tau_2 = \{(*, 0), (*, 1)\}$. *Then* $\{\tau_1, \tau_2\} \subset Rel_{\theta_z}(I, \mathbf{Z})$ *and*

$$\Psi(\tau_1, \tau_2) = \{(*, (0, 0)), (*, (0, 1)), (*, (2, 0)), (*, (2, 1))\}.$$

Let $\tau = \{(*, (0, 0)), (*, (2, 1))\}$. *Then* $\tau \in Rel_{\theta_Z}(I, \mathbf{Z} \times \mathbf{Z})$ *and*

$$\Phi(\tau) = (\{(*, 0), (*, 2)\}, \{(*, 0), (*, 1)\}) = (\tau_1, \tau_2).$$

That is $\Psi(\Phi(\tau)) \neq \tau$.

Next, we would like to show equivalence relational database. First, we follow the formalization of relational tuple. Let A be a finite set of attributes with $a \in A$. Thus, a relational tuple is $\tau = (\tau_a, \ldots)$ where $\langle \tau_a : I \rightharpoondown D_a \rangle$ and $\langle \tau_a \neq \emptyset \rangle$. We can consider $\tau : I \rightharpoondown D_A$ where $\langle D_A = \prod_{a \in A} D_a \rangle$ and $\langle \tau = \sqcap_{a \in A} \tau_a \rho_{Aa}^{\#} \rangle$. We note: $\tau_a = \tau \cdot \rho_{Aa}$.

Example 4.2.4. *Let* $A = \{X, Y, Z\}$, $D_X = \{x1, x2\}$, $D_Y = \{y1, y2, y3\}$, *and* $D_Z = \{z1\}$. *Table 4.4 contain only 1 tuple. The tuple means that*

X	Y	Z
$\{x1, x2\}$	$\{y1, y2\}$	$\{z1\}$

Table 4.4: Equivalence Relational Table

$\tau = (\tau_X, \tau_Y, \tau_Z)$ *with* $\tau_X = \{\{*, x1\}, \{*, x2\}\}$, $\tau_Y = \{\{*, y1\}, \{*, y2\}\}$, *and* $\tau_Z = \{\{*, z1\}\}$.
Then we have $\tau = \{\{*, (x1, y1, z1)\}, \{*, (x2, y1, z1)\}, \{*, (x1, y2, z1)\}, \{*, (x2, y2, z1)\}\}$.

Proposition 4.2.4. *Let A be a set of attributes with domain set D_A, and θ_A be an equivalence relation. A relational tuple $\tau : I \rightharpoonup D_A$ is a θ_A-relation if and only if $\tau \neq 0_{ID_A}$ and $\tau^{\#}\tau \sqsubseteq \theta_A$*

Proof. $(\leftarrow)$ Since we have $\tau^{\#}\tau \sqsubseteq qq^{\#}$ and $q^{\#}q = \mathrm{id}_{D_A^{\theta_A}}$, then we can compute

$$(\tau q)^{\#}(\tau q) = q^{\#}\tau^{\#}\tau q \sqsubseteq q^{\#}qq^{\#}q \sqsubseteq \mathrm{id}_{D_A^{\theta_A}}$$

Since $\tau \neq 0_{ID}$ then $\tau \cdot \tau^{\#} = \mathrm{id}_I$ so we have $(\tau q) \cdot (\tau q)^{\#} = \tau qq^{\#}\tau^{\#} \sqsubseteq \tau\tau^{\#} = \mathrm{id}_I$. We have also τq total and $(\tau q)^{\#}(\tau q) \sqsubseteq \mathrm{id}_{D_A^{\theta_A}}$(univalent) then we can conclude that τq is I-point of partition $D_A^{\theta_A}$.

$(\rightarrow)$

- We assume $\tau = 0_{ID}$ then we have $(\tau q) \cdot (\tau q)^{\#} = 0_{ID} \neq \mathrm{id}_I$, so τq is not I-point. Then we use negation proof means that if we assume $\tau q : I - point$ then we get $\tau \neq 0_{ID}$.
- Since we have $id_D \sqsubseteq \theta$ and $qa^{\#} = \theta$, then

$$\begin{aligned} \tau^{\#}\tau &\sqsubseteq qq^{\#}\tau^{\#}\tau qq^{\#} \\ &= q(\tau q)^{\#}(\tau q)q^{\#} \end{aligned}$$

  We consider $\tau q : I - point$ then we have $(\tau q)^{\#}(\tau q) \sqsubseteq \mathrm{id}_{D_A^{\theta_A}}$ and also we have $qq^{\#} = \theta_A$ then we get $\tau^{\#}\tau \sqsubseteq \theta_A$

then we prove that $\tau \neq 0_{ID_A}$ and $\tau^{\#}\tau \sqsubseteq \theta_A$.

□

Definition 4.2.9. *Let A be a set of attributes with $D_A = \prod_{a\in A} D_a$, I a singleton set, and θ_A an equivalence relations of D_A. A* ***fuzzy database relation*** *R associated to θ_A is a subset of $Rel_{\theta_A}(I, D_A)$, i.e $R \subseteq Rel_{\theta_A}(I, D_A)$*

Example 4.2.5. *Let domain of salary $D_S = \{100, 125, 150, 200, 210, 300\}$. We consider partition of salary, $Low = [100, 125]$, $Middle = [126, 150]$, and $High = [151, 300]$, so we consider equivalence relation of S, $\theta_S = \{\{100, 100\}, \{100, 125\}, \{125, 100\}, \{125, 125\}, \{150, 150\}, \{200, 200\}, \{200, 210\}, \{200, 300\}, \{210, 200\}, \{210, 210\}, \{210, 300\}, \{300, 200\}, \{300, 210\}, \{300, 300\}\}$. Then we have partition of domain "Salary", $D_S^{\theta_S} = \{\{100, 125\}, \{150\}, \{200, 210, 300\}\}$. A quotient relation q satisfying with $qq^{\#} = \theta_S$ and $q^{\#}q = \mathrm{id}_{D_S^{\theta_S}}$, then we have $q = \{\{100, \{100, 125\}\}, \{125, \{100, 125\}\}, \{150, \{150\}\}, \{200, \{200, 210, 300\}\}, \{210, \{200, 210, 300\}\}, \{300, \{200, 210, 300\}\}\}$. Let relational tuples $\tau, \tau' : I \rightharpoonup D_S$ defined by $\tau = \{\{*, 200\}, \{*, 300\}\}$, and $\tau' = \{\{*, 100\}, \{*, 150\}\}$. Then:*

- *τ is θ_S-relation*
  *We have $\tau \cdot q = \{\{*, \{200, 210, 300\}\}\}$. Since $\tau \cdot q$ is an **I-point** then τ is θ_S-relation and we get $\tau \in Rel_{\theta_S}(I, D_S)$. Further, We would like to show $\tau^{\#}\tau \sqsubseteq \theta_S$ (Proposition 4.2.4). We have $\tau^{\#}\tau = \{\{200, 200\}, \{200, 300\}, \{300, 200\}, \{300, 300\}\} \sqsubseteq \theta_S$.*

- *τ' is not θ_S-relation*
  *We have $\tau' \cdot q = \{\{*, \{100, 125\}\}, \{*, \{150\}\}\}$. Since $\tau \cdot q$ is a **not I-point** then τ' is a not θ_S-relation and we get $\tau' \notin Rel_{\theta_S}(I, D_S)$. Further, We would like to show $\tau^{\#}\tau \sqsubseteq \theta_S$ since $\tau' \notin Rel_{\theta_S}(I, D_S)$. Then we get $\tau'^{\#}\tau' = \{\{100, 100\}, \{100, 150\}, \{150, 100\}, \{150, 150\}\} \not\sqsubseteq \theta_S$. We can check $\{100, 150\} \notin \theta_S$, and $\{150, 100\} \notin \theta_S$.*

Next we define the notions 'redundant' that has been introduced by Shenoi et al[SMF90]. We use relational formalization to define 'redundant' in fuzzy database relation.

Definition 4.2.10. *Let A be a set of attributes with domain D_A, θ_A be an equivalence relation of D_A and and a quotient relation q that is $qq^{\#} = \theta_A$ and $q^{\#}q = \mathrm{id}_{D_A^{\theta_A}}$. A equivalence relational database $R \subseteq Rel_{\theta_A}(I, D_A)$ be a simple (non-redundant), statisfies with:*

$$\text{if } \tau q = \tau' q \text{ then } \tau = \tau', \text{ for all } \tau, \tau' \in R$$

Proposition 4.2.5. *Let a set $R \subseteq Rel_{\theta_A}(I, D_A)$ be a simple (non-redundant), and q a quotient relation of θ_A, then for all $\tau, \tau' \in R$, $\langle \tau q = \tau' q \rangle$ if and only if $\langle \tau\theta_A = \tau'\theta_A \rangle$*

Proof. $(\rightarrow)$ Assume $\tau q = \tau' q$ then we have $\tau qq^{\#} = \tau' qq^{\#}$ and $qq^{\#} = \theta_A$ then we have $\tau\theta_A = \tau'\theta_A\rangle$

$(\leftarrow)$ Assume $\tau\theta_A = \tau'\theta_A \rightarrow \tau\theta_A q = \tau'\theta_A q$. Since $\theta_A q = qq^{\#}q = q$ then we have $\tau q = \tau\theta_A q = \tau'\theta_A q = \tau' q$.

□

Proposition 4.2.6. *Let a set of attributes $A = \{X, Y\}$, θ_X, θ_Y equivalence relation of X, and Y respectively. We consider a relational tuple $\tau \in Rel_{\theta_A}(I, D_A)$. Then:*

$$\tau \cdot \theta_A = (\tau \cdot \rho_{AX} \cdot \theta_X \cdot \rho_{AX}^{\#}) \sqcap (\tau \cdot \rho_{AX} \cdot \theta_Y \cdot \rho_{AX}^{\#})$$

Proof. Since $\theta_A = (\rho_{AX} \cdot \theta_X \cdot \rho_{AX}^{\#}) \sqcap (\rho_{AX} \cdot \theta_Y \cdot \rho_{AX}^{\#})$

$(\sqsubseteq)$ Using Proposition 2.2.2 then we have $\tau \cdot ((\rho_{AX} \cdot \theta_X \cdot \rho_{AX}^{\#}) \sqcap (\rho_{AX} \cdot \theta_Y \cdot \rho_{AX}^{\#})) \sqsubseteq (\tau \cdot \rho_{AX} \cdot \theta_X \cdot \rho_{AX}^{\#}) \sqcap (\tau \cdot \rho_{AX} \cdot \theta_Y \cdot \rho_{AX}^{\#})$.

($\sqsupseteq$) Since $\tau \in Rel_{\theta_A}(I, D_A)$, using dedekind formula we have:

$$(\tau\rho_{AX}\theta_X\rho^{\#}_{AX}) \sqcap (\tau\rho_{AX}\theta_Y\rho^{\#}_{AX}) \sqsubseteq \tau \cdot ((\rho_{AX}\theta_X\rho^{\#}_{AX}) \sqcap (\tau^{\#}\tau\rho_{AX}\theta_Y\rho^{\#}_{AX}))$$
$$\sqsubseteq \tau \cdot ((\rho_{AX}\theta_X\rho^{\#}_{AX}) \sqcap (\theta_A\rho_{AX}\theta_Y\rho^{\#}_{AX}))$$

Using Propostion 4.2.3 then we have $(\tau\rho_{AX}\theta_X\rho^{\#}_{AX}) \sqcap (\tau\rho_{AX}\theta_Y\rho^{\#}_{AX}) \sqsubseteq \tau \cdot (\rho_{AX}\theta_X\rho^{\#}_{AX}) \sqcap (\rho_{AX}\theta_Y\rho^{\#}_{AX})$

Then we prove $\tau \cdot ((\rho_{AX} \cdot \theta_X \cdot \rho^{\#}_{AX}) \sqcap (\rho_{AX} \cdot \theta_Y \cdot \rho^{\#}_{AX})) = (\tau \cdot \rho_{AX} \cdot \theta_X \cdot \rho^{\#}_{AX}) \sqcap (\tau \cdot \rho_{AX} \cdot \theta_Y \cdot \rho^{\#}_{AX})$ □

Lemma 4.2.1. *Let a set $R \subseteq Rel_{\theta_A}(I, D_A)$ be simple, A a set of attributes with $a \in A$, and θ_A equivalence relation of A. Then: for all $\tau, \tau' \in R$, if $\tau \cdot \rho_{Aa} = \tau' \cdot \rho_{Aa}$ then $\tau = \tau'$.*

Proof. Since $\theta = \sqcap_{a \in A}\rho_a\theta_a\rho^{\#}_a$, using Proposition 4.2.5 and 4.2.6 $\tau \cdot (\sqcap_{a \in A}\rho_a\theta_a\rho^{\#}_a) = \sqcap_{a \in A}\tau\rho_a\theta_a\rho^{\#}_a$. Since $\tau \cdot \rho_{Aa} = \tau' \cdot \rho_{Aa}$ and Proposition 4.2.6 then we got $\tau\theta_A = \tau'\theta_A$, then using Proposition 4.2.5 we have $\tau = \tau'$ □

Let relational tuples $\tau, \tau' \in Rel_{\theta_A}(I, D_A)$. If we consider $\tau_a = \tau \cdot \rho_{Aa}$ and $\tau' = \sqcap_{a \in A}\tau_a\rho_{Aa}$. Since we have $\tau' \cdot \rho_{Aa} = (\sqcap_{a \in A}\tau_a\rho_{Aa}) \cdot \rho_{Aa} = \tau_a = \tau \cdot \rho_{Aa}$ then using Lemma 4.2.1 if $\tau \neq \tau'$ then will be redundant, so we can consider $\tau = \tau' = \sqcap_{a \in A}\tau_a\rho_{Aa}$ to get simple (non-redundant) relation.

We represent database relation as Table. Let A set of attributes, θ_A be an equivalence relation, a database relation $R \subseteq Rel_{\theta_A}(I, D_A)$. We define relational tuple $\tau, \tau' \in R$. We shown a database relation R as Table 4.2.1. For all $a \in A$, and $b \in B$

R	A	B
τ	τ_a	τ_b
τ'	τ'_a	τ'_b

Example 4.2.6. *Let $AS = \{Age, Salary\}$, with:$D_{Age} = \{$19,22,34,38$\}$, and $D_{Salary} = \{$100, 125, 150$\}$. We consider p be Table 4.5, the classical relational database. We define relational tuple of $i - row$ by τ^i, for example 1^{th}-row, $\tau^1 = \{*, (19, 100)\}$, and relational "age" tuple for 1^{th}-row, $\tau^1_{Age} = \{*, 19\}$ and 1-row tuple "salary", $\tau^1_{Salary} = \{*, 100\}$. Then we have $p = \{\{*, (19, 100)\}\}, \{*, (22, 125)\}, \{*, (22, 100)\}, \{*, (34, 150)\}, \{*, (38, 150)\}\}$*

Then we consider θ_{Age} equivalence relation of "age", θ_{Salary} equivalence relation of "salary". Then we have $\theta_A = \sqcap_{a \in A}\rho_{Aa}\theta_a\rho^{\#}_{Aa}$.

We have: $D^{\theta_{age}}_{Age} = \{\{$19, 22$\}, \{$34, 38$\}\}$
$D^{\theta_{salary}}_{Salary} = \{\{$100, 125$\}, \{150\}\}$. We have $q : D_{Age} \times D_{Salary} \rightharpoonup D^{\theta_{Age}}_{Age} \times$

Age	Salary
19	100
22	125
22	100
34	150
38	150

Table 4.5: Relation Age and Salary

Age	Salary
{19,22}	{100,125}
{34,38}	{150}

Table 4.6: Equivalence Relation of Table 4.5

$D^{\theta_{Salary}}_{Salary}$ *quotient relation with* $qq^{\#} = \theta$.
For example:
$\{\{19, 100\}, \{(19, 22), (100, 125)\}\} \in q$, *and also*
$\{\{34, 150\}, \{(34, 38), (150)\}\} \in q$.

Then we get $\tau^1(p), \tau^2(p), \tau^3(p)$ *will redundant in* θ*-relations. Since* $\tau^1(p) \cdot q = \tau^2(p) \cdot q = \tau^3(p) \cdot q = \{\{*, \{\{19, 22\}, \{100, 125\}\}\}$ *will be* I*-point. Also* $\tau^4(p), \tau^5(p)$ *will redundant in* θ*-relations. Since* $\tau^4(p) \cdot q = \tau^5(p) \cdot q = \{\{*, \{\{34, 38\}, \{150\}\}\}$ *will be* I*-point. Then we consider transform from classical* p *to* θ*-relations,* $\hat{p} \subseteq Rel_\theta(I, D)$*. Then we have* $\hat{p} = \{\{\tau^1(p), \tau^2(p), \tau^3(p)\}, \{\tau^4(p), \tau^5(p)\} = \{\{\{*, (19, 100)\}, \{*, (22, 125)\}, \{*, (22, 100)\}\}, \{\{*, (34, 150)\}, \{*, (38, 150)\}\}\}$*. Then* $\tau^1(\hat{p}) = \{\{*, (19, 100)\}, \{*, (22, 125)\}, \{*, (22, 100)\}\}$, $\tau^1_{Age}(\hat{p}) = \{19, 22\}$, $\tau^1_{Salary}(\hat{p}) = \{100, 125\}$*. Also* $\tau^2(\hat{p}) = \{\{*, (34, 150)\}, \{*, (38, 150)\}\}$, $\tau^2_{Age}(\hat{p}) = \{34, 38\}$, $\tau^2_{Salary}(\hat{p}) = \{150\}$*. As we can see* $\hat{p}$ *in Table 4.6.*

After we define 'redundant' model using relational formalization. Next, we use relation formalization corresponding to Theorem 3.2 [SMF90].

Theorem 4.2.1. *Let* $R \subseteq Rel_{\theta_A}(I, D_A)$ *be simple. Then*

$$\text{if } \langle \tau \neq \tau' \rangle \text{ then } \langle \tau \sqcap \tau' = 0_{ID_A} \rangle, \text{ for all } \tau, \tau' \in R$$

Proof. Let assume $\tau, \tau' \in R$ and $\tau \sqcap \tau' \neq 0_{ID_A}$ non empty set(NE_*), then there are exists I-point x of D, such that $x \sqsubseteq \tau \sqcap \tau'$,
Then we have $xq \sqsubseteq (\tau \sqcap \tau')q \sqsubseteq \tau q \sqcap \tau' q$. Since $xq, \tau q, \tau' q : I - point$ then we have $\tau q = \tau' q$ Since R is simple(non-redundant) by Definition 4.2.10, we have $\tau = \tau'$. Then we use contradiction proof means that if we assume $\tau \neq \tau'$ then we have $\tau \sqcap \tau' = 0_{ID_A}$ □

Lemma 4.2.2. *Let $p : I \rightharpoonup D_A$ be a fuzzy relation. If $R \subseteq Rel_{\theta_A}(I, D_A)$ is a simple set such that $\sqcup R = p$, then*

$$\text{for all } \tau \in R. \tau qq^{\#} \sqcap p = \tau.$$

Proof. Since we have $\sqcup R = p$ then we get
$\tau qq^{\#} \sqcap p = \tau qq^{\#} \sqcap (\sqcup_{\tau' \in R} \tau') = \sqcup_{\tau' \in R}(\tau qq^{\#} \sqcap \tau')$
We follows using two cases:

1. Case $\tau' = \tau$:
   Since we have $\tau' = \tau$ and $id_Q \sqsubseteq qq^{\#}$. Then we have $(\tau \sqcap \tau' = \tau \cdot id_Q \sqcap \tau') \sqsubseteq (\tau qq^{\#} \sqcap \tau')$ and also we have $\tau qq^{\#} \sqcap \tau' \sqsubseteq (\tau \sqcap \tau')$. Then we have $\tau qq^{\#} \sqcap \tau' = \tau \sqcap \tau' = \tau$.

2. Case $\tau' \neq \tau$:
   Since R:simple from Definition 4.2.10 and Theorem 4.2.1 then we get $\tau q \sqcap \tau' q = 0_{ID}$. Using Dedekind Formula(DF), we get $\tau qq^{\#} \sqcap \tau \sqsubseteq (\tau q \sqcap \tau' q) q^{\#}$. Since $\tau q \sqcap \tau' q = 0_{ID_A}$ then we get $\tau qq^{\#} \sqcap \tau = 0_{ID_A}$.

From 2 cases, we can make union the results all of cases then we can conclude that $\tau qq^{\#} \sqcap \tau = \tau$. ☐

Notation. Let $x, p : I \rightharpoonup D_A$ be fuzzy relations. We denote an element point x of p as follows

$$\langle x \mathrel{\dot\in} p \rangle \leftrightarrow \langle x \sqsubseteq p \rangle \wedge \langle x : I - point \rangle.$$

Using notation of element point then we can formalize fuzzy database relation.

Corollary 4.2.1. *If a set $R \subseteq Rel_{\theta_A}(I, D_A)$ be a simple, then*

$$R = \{xq^{\#} \sqcap (\sqcup R) | x \dot\in (\sqcup R) q\}$$

Proof. We consider $\tau \in R$ then we have $\tau q \dot\in (\sqcup R) q$. If we assume that $x \dot\in (\sqcup R) q$ and we have $\tau q \dot\in (\sqcup R) q$ then we get $x \cdot q = \tau \cdot q$, then from Lemma 4.2.2 we get

$$xq^{\#} \sqcap (\sqcup R) = \tau qq^{\#} \sqcap (\sqcup R) = \tau.$$

Since we have $\tau \in R$ then we prove that $R = \{xq^{\#} \sqcap (\sqcup R) | x \dot\in (\sqcup R) q\}$ ☐

Theorem 4.2.2. *For each nonzero relation $p : I \rightharpoonup D_A$ there exists a unique simple set $R \subseteq Rel_{\theta_A}(I, D_A)$ such that $\sqcup R = p$.*

Proof. (Existence) Let $p \neq 0_{ID_A}$. Then p is nonempty by (NE_*) and so pq is nonempty, too. Define a set $R \subseteq Rel_{\theta_A}(I, D_A)$ by

$$R = \{xq^{\#} \sqcap p | x \dot\in pq\}.$$

Then relation $xq^{\#} \sqcap p$ is θ-relation, for each I-point $x \dot{in} pq$.
At first we show $(xq^{\#} \sqcap p) q = x$, θ_A-relation:

($\sqsubseteq$) Since we have $x\dot{\in}pq \leftrightarrow x \sqsubseteq pq \wedge x : I - point$ then we get $x = x \sqcap pq$. Then using Dedekind formula we get $x \sqsubseteq (xq^{\#} \sqcap p)q$

($\sqsupseteq$) We have $(xq^{\#} \sqcap p)q \sqsubseteq xq^{\#}q$. Since we have $q^{\#}q = id_{D_A^{\theta_A}}$ then we get $(xq^{\#} \sqcap p)q \sqsubseteq x$

Which proves that $(xq^{\#} \sqcap p)q = x$. If we assume that $(xq^{\#} \sqcap p) = (yq^{\#} \sqcap p)$. We also have $(yq^{\#} \sqcap p)q = y$. Hence $R \subseteq Rel_{\theta_A}(I, D_A)$ and R is simple. Also we have

$$\begin{aligned}\sqcup R &= \sqcup_{x\dot{\in}pq}(xq^{\#} \sqcap p)\\ &= (\sqcup_{x\dot{\in}pq}x)q^{\#} \sqcap p\\ &= pqq^{\#} \sqcap p\end{aligned}$$

Since $id_{D_A} \sqsubseteq qq^{\#}$ then we get $\sqcup R = p$.
(Uniqueness) Assume that $S \subseteq Rel_{\theta_A}(I, D_A)$ is simple and $\sqcup S = p$.

1. Case $S \subseteq R$:
   Let $\sigma \in S$, using Lemma 4.2.2 we have $\sigma = \sigma qq^{\#} \sqcap p$. Since $\sigma q\dot{\in}pq$ we get $\sigma \in R$.

2. Case $R \subseteq S$:
   Let $x \in pq$ and $\sqcup_{\sigma\in S}\sigma$ then we have

   $$x = x \sqcap (\sqcup_{\sigma\in S}\sigma)q = \sqcup_{\sigma\in S}(x \sqcap \sigma q).$$

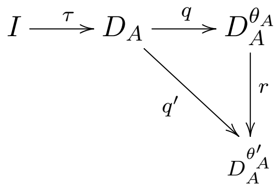
   there are exists σ such that $x \sqcap \sigma q \neq 0_{ID_A^{\theta_A}}$. Since $S \subseteq Rel_{\theta_A}(I, D_A)$, we have σq is an $I - point$ and x is an $I - point$ and we get $x = \sigma q$. Then using Lemma 4.2.2 we get $xq^{\#} \sqcap p = \sigma qq^{\#} \sqcap p = \sigma$. So we get $xq^{\#} \sqcap p \in S$.

Then we have $R = S$ and it is showing uniqueness of R. □

Definition 4.2.11. *Let A set of attributes with domain set D_A, $\theta_A, \theta'_A : D_A \rightharpoondown D_A$ be equivalence relations. θ_A is finer then θ'_A* ***if and only if*** *$\theta_A \sqsubseteq \theta'_A$.*

Let $\theta_A, \theta'_A : D_A \rightharpoondown D_A$ be equivalence relations such that $\theta_A \sqsubseteq \theta'_A$. The quotient map between equivalence relation, we can define as $qq^{\#} = \theta_A$, $q^{\#}q = \mathrm{id}_{D_A^{\theta_A}}$, $q'q'^{\#} = \theta'_A$, $q'^{\#}q' = \mathrm{id}_{D_A^{\theta'_A}}$. Let there are exits relation r between quotient set with $D_A^{\theta_A}$, and quotient with equivalence $D_A^{\theta'_A}$, then we can define:

$$\theta_A \sqsubseteq \theta'_A \rightarrow \exists! r : D_A^{\theta_A} \rightarrow D_A^{\theta'_A}.q' = qr.(r = q^{\#}q')$$



Lemma 4.2.3. *If* $\tau \in Rel_{\theta_A}(I, D_A)$ *and* $\theta_A \sqsubseteq \theta'_A$ ***then*** $\tau \in Rel_{\theta'_A}(I, D_A)$

Proof. We assume τ is θ_A-relation, using Proposition 4.2.4 we have $\tau \neq 0_{ID_A}$ and $\tau^{\#}\tau \sqsubseteq \theta_A$. Since we have $\theta_A \sqsubseteq \theta'_A$ and $\tau^{\#}\tau \sqsubseteq \theta_A$, we have $\tau^{\#}\tau \sqsubseteq \theta'_A$. So, we have $\tau \neq 0_{ID_A}$ and $\tau^{\#}\tau \sqsubseteq \theta'_A$, then using Proposition 4.2.4 we get τ is θ'_A-relation. It means that $\tau \in Rel_{\theta'_A}(I, D_A)$. □

Proposition 4.2.7. *Assume* $R \subseteq Rel_{\theta_A}(I, D_A)$ *and* $R' \subseteq Rel_{\theta_A}(I, D_A)$. *If* $\theta_A \sqsubseteq \theta'_A$ *and* $\sqcup R = \sqcup R'$ *then*

1. $\forall \tau \in R. \exists \tau' \in R'. \tau \in \tau'$.
2. $\forall \tau \in R. \tau' = \sqcup_{\tau \sqsubseteq \tau'} \tau$.

Proof. By Theorem 4.2.2 we may set $R = \{xq^{\#} \sqcap p | x \dot{\in} \tau q\}$ and $R' = \{x'q'^{\#} \sqcap p | x' \dot{\in} \tau' q'\}$ where $p = \sqcup R = \sqcup R'$.

1. Since we have $x \dot{\in} \tau q$ then using Lemma 4.2.2 we get $\tau = xq^{\#} \sqcap p$. Since $id_{D_A^{\theta_A}} \sqsubseteq rr^{\#}$, then $\tau \sqsubseteq xrr^{\#}q^{\#} \sqcap p \sqsubseteq xr(qr)^{\#} \sqcap p$. We consider $\theta \sqsubseteq \theta'$ then we have $x' = xr \dot{\in} \tau' q', qr = q'$, then we got $\tau \sqsubseteq x'q'^{\#} \sqcap p$. Then we can conclude that $\tau \sqsubseteq \tau'$.
2. First, we prove that $x \dot{\in} x'r^{\#} \leftrightarrow xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p$.
   $(\rightarrow)$ $\quad x \dot{\in} x'r^{\#} \rightarrow xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p :$
   Since we have $x \dot{\in} x'r^{\#}$ and $qr = q'$ then we get $xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p$
   $(\leftarrow)$ $\quad xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p \rightarrow x \dot{\in} x'r^{\#}$:
   Since $x \sqsubseteq xrr^{\#}$ and $x \dot{\in} pq$ then we have $x \sqsubseteq (x \sqcap pq)rr^{\#}$. Using Dedekind formula we have $x \sqsubseteq (xq^{\#} \sqcap pq)qrr^{\#}$. We have $xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p$ and $qr = q'$ then we get $x \sqsubseteq (x'q'^{\#} \sqcap pq)q'r^{\#}$. Since $q'^{\#}q' = id_{Q'}$ then we get $x \sqsubseteq x'r^{\#}$. Since x:I-point and $x \sqsubseteq x'r^{\#}$ then we prove that $x \dot{\in} x'r^{\#}$. Since $x \dot{\in} x'r^{\#} \leftrightarrow xq^{\#} \sqcap p \sqsubseteq x'q'^{\#} \sqcap p$ then we have $\sqcup_{\tau \sqsubseteq \tau'} \tau = (\sqcup_{x \dot{\in} x'r^{\#}} x)q^{\#} \sqcap p$. Then $\sqcup_{\tau \sqsubseteq \tau'} \tau = x'r^{\#}q^{\#} \sqcap p$. Since we have $q' = qr$ then we get $\sqcup_{\tau \sqsubseteq \tau'} \tau = x'q'^{\#} \sqcap p$. Since $R' = \{x'q'^{\#} \sqcap p | x' \dot{\in} \tau' q'\}$ then we prove $\sqcup_{\tau \sqsubseteq \tau'} \tau = \tau'$

□

Theorem 4.2.3 (Corresponding to Theorem 3.1 [SMF90]). *Let* $R \subseteq Rel_{\theta_A}(I, D_A)$ *be simple and* $\theta_A \sqsubseteq \theta'_A$. *Then there exists a unique simple set* $S \subseteq Rel_{\theta'_A}(I, D_A)$ *such that* $\sqcup S = \sqcup R$. *(Merge)*

Proof. We consider that $\tau \in R$. Since we have Proposition 4.2.4 then we have $\tau \neq 0_{ID_A}$. We have $\sqcup R : I \rightharpoondown D$ are non-zero relation. Let $\sqcup S = \sqcup R$, then we have $\sqcup S : I \rightharpoondown D_A$ is non-zero relation. Using Theorem 4.2.2 then there exist unique simple set $S \subseteq Rel_{\theta'}(I, D_A)$ with $\theta_A \sqsubseteq \theta'_A$. □

4.3 Equivalence Relational Database Operations

After we define relational model for fuzzy equivalence relations and its characteristic. In this section, we will formalize some operations of equivalence relational database. The fuzzy database operations model was introduced by Umano and Fukami[UF94]. Equivalence relational operators using relational calculus also introduced by Akbar and Mizoguchi[AM16]. In our work, we use relational calculus to compute database operations with equivalence relation with relation scheme $R \subseteq Rel_{\theta}(I, D)$. We use simpler and clearer formula to compute database operations. We also give some example for every operation, we follows approach of Shenoi and Melton[SMF90] that is not counting the fuzzy value to show simple applications.

4.3.1 Union

Proposition 4.3.1. *Let* $R \subseteq Rel_{\theta_A}(I, D_A)$ *and* $R' \subseteq Rel_{\theta'_A}(I, D_A)$ *be simple. Then there exists a unique simple set* $R" \subseteq Rel_{\theta_A \sqcup \theta'_A}(I, D)$ *such that* $\sqcup R" = \sqcup(R \sqcup R')$.

Proof. We consider that $\tau \in R$ and $\tau' \in R'$. Since we have Proposition 4.2.4 then we have $\tau \neq 0_{ID_A}$ and $\tau' \neq 0_{ID_A}$. We have $\sqcup R$, $\sqcup R' : I \rightharpoondown D_A$ are non-zero relation. Let $\sqcup R" = \sqcup(R \sqcup R')$, then we have $\sqcup R" : I \rightharpoondown D$ is non-zero relation. Using Theorem 4.2.2 then there exist unique simple set $R" \subseteq Rel_{\theta \sqcup \theta'}(I, D_A)$. □

Definition 4.3.1. *Let* $R \subseteq Rel_{\theta_A}(I, D_A)$ *and* $R' \subseteq Rel_{\theta'_A}(I, D_A)$ *be simple. There are exists unique simple,* $R" \subseteq Rel_{\theta_A \sqcup \theta'_A}(I, D_A)$ *such that* $\sqcup R" = (\sqcup R) \sqcup (\sqcup R')$. *We define the* $R \underset{\theta_A, \theta'_A}{\sqcup} R'$ *by* $R"$.

Note: $R \sqcup R' \neq R \underset{\theta_A, \theta'_A}{\sqcup} R$ *in general*

Example 4.3.1. *Let* $A = \{Lecturer, Subject, Room\}$, *with:*
$D_{Lecturer} = \{$*Jones,Marry,Codd,David*$\}$,
$D_{Subject} = \{$*Statistics, Algebra, Number Theory, Control System, Database, Cryptography*$\}$, *and* $D_{Room} = \{$*C710, A505, C705, B101*$\}$.
In Table 4.7, we have partition domain with equivalence relation θ *such that:*
$D^{\theta}_{Lecturer} = \{\{$*Jones*$\}, \{$*Marry*$\}, \{$*Codd*$\},, \{$*David*$\}\}$,
$D^{\theta}_{Subject} = \{\{$*Statistics, Algebra, Number Theory, Database, Cryptography*$\}\}$,

$D^{\theta}_{Room} = \{\{$*C710, A505, C705, B101*$\}\}$.
In Table 4.8, we have partition domain with equivalence relation θ' *such that* $D^{\theta}_{Lecturer} = D^{\theta'}_{Lecturer}$, $D^{\theta}_{Room} = D^{\theta'}_{Room}$, *but we have new classification of subjects, we separate the subjects as 'Pure Mathematics Subjects' and 'Application Mathematics Subjects' than we have*
$D^{\theta'}_{Subjects} = \{\{$*Statistics, Algebra, Number Theory*$\}, \{$ *Database, Cryptography*$\}\}$

Lecturer	Subject	Room
{Jones}	{Algebra, Database}	{C710}
{Marry}	{Number Theory, Statistics, Cryptography}	{C705}
{David}	{Algebra, Statistics, Database}	{A505}
{Codd}	{Algebra, Number Theory, Cryptography}	{B101}

Table 4.7: $R \subseteq Rel_{\theta_A}(I, D_A)$

Lecturer	Subject	Room
{Codd}	{Database, Cryptography}	{B101}
{David}	{Statistics, Algebra}	{A505}
{David}	{Cryptography, Control System}	{A505}

Table 4.8: $R' \subseteq Rel_{\theta'}(I, D_A)$

We have $D_{Subject} = \{$*Statistics, Algebra, Number Theory, Control System, Database, Cryptography*$\}$ *and* $D^{\theta'}_{Subjects} = \{\{$*Statistics, Algebra, Number Theory*$\}$, $\{$ *Database, Cryptography*$\}\}$, *then we have* $\theta'_{Subject} \sqsubseteq \theta_{Subject}$ *because not all pairs member of* $\theta_{Subject}$ *are pairs of* $\theta'_{Subject}$ *for example:* $\{Algebra, Statistics\} \in \theta_{Subject}$ *but* $\{Algebra, Statistics\} \notin \theta'_{Subject}$, *etc.*

Since $\theta'_{Subject} \sqsubseteq \theta_{Subject}$, $\theta'_{Lecturer} = \theta_{Lecturer}$, *and* $\theta'_{Room} \sqsubseteq \theta_{Room}$, *then we have* $\theta' \sqsubseteq \theta$, *we get* $\theta' \sqcup \theta = \theta$.

Then we have $R \subseteq Rel_{\theta_A}(I, D_A)$ *be simple (shown Table 4.7). For* $R' \subseteq Rel_{\theta'_A}(I, D_A)$ *be simple (shown Table 4.8), there are exist unique* $S' \subseteq Rel_{\theta_A}(I, D_A)$ *such that* $\sqcup S = \sqcup R$*but* $S' \neq R'$, *we have to transform* R' *to* S'. *Tuple 2nd and tuple 3rd is redundant in relation* $S' \subseteq Rel_{\theta_A}(I, D_A)$, *then we have* $S' = \{\{\{Codd\}, \{Database, Cryptography\}, \{B101\}\},$
$\{\{David\}, \{Statistics, Algebra, Cryptography, ControlSystem\}, \{A505\}\}\}$.
Then we get exist unique $R" \subseteq Rel_{\theta_A}(I, D_A)$ *with* $\sqcup R" = (\sqcup S) \sqcup (\sqcup S')$, *as the result* $R \underset{\theta_A, \theta'_A}{\sqcup} R'$ *(shown in Table 4.9).*

Proposition 4.3.2. *Let* $R \subseteq Rel_{\theta_A}(I, D_A)$, $R' \subseteq Rel_{\theta'_A}(I, D_A)$, *and* $R'' \subseteq Rel_{\theta''_A}(I, D_A)$ *be simple, then:*

1. $R \underset{\theta_A, \theta_A}{\sqcup} R = R$,

2. $R \underset{\theta_A, \theta'_A}{\sqcup} R' = R' \underset{\theta'_A, \theta_A}{\sqcup} R$, *and*

Lecturer	Subject	Room
{Jones}	{Algebra, Database}	{C710}
{Marry}	{Number Theory, Statistics, Cryptography}	{C705}
{David}	{Algebra, Statistics, Database, Cryptography, Control System}	{A505}
{Codd}	{Algebra, Database, Cryptography}	{B101}

Table 4.9: $R \underset{\theta,\theta'}{\sqcup} R'$

3. $(R \underset{\theta_A,\theta'_A}{\sqcup} R') \underset{(\theta_A \sqcup \theta'_A),\theta''_A}{\sqcup} R'' = R \underset{\theta_A,(\theta'_A \sqcup \theta''_A)}{\sqcup} (R' \underset{\theta'_A,\theta''_A}{\sqcup} R'')$.

Proof. We have $\sqcup R, \sqcup R', \sqcup R'' : I \rightharpoondown D_A$ are non zero relation, then:

1. Since we have $\theta_A \sqcup \theta_A = \theta_A$ and $(\sqcup R) \sqcup (\sqcup R) = \sqcup R$, then by Proposition 4.3.1 we get exist unique $R \underset{\theta_A,\theta_A}{\sqcup} R = R$.

2. Since we have $\theta_A \sqcup \theta'_A = \theta'_A \sqcup \theta_A$ and $(\sqcup R) \sqcup (\sqcup R') = (\sqcup R') \sqcup (\sqcup R)$, then by Proposition 4.3.1 we get exist unique $R \underset{\theta_A,\theta'_A}{\sqcup} R' = R' \underset{\theta'_A,\theta_A}{\sqcup} R$.

3. Since we have $(\theta_A \sqcup \theta'_A) \sqcup \theta_A = \theta_A \sqcup (\theta'_A \sqcup \theta_A)$ and $((\sqcup R) \sqcup (\sqcup R')) \sqcup (\sqcup R'') = (\sqcup R) \sqcup ((\sqcup R') \sqcup (\sqcup R''))$, then by Proposition 4.3.1 we get exist unique
$(R \underset{\theta,\theta'}{\sqcup} R') \underset{(\theta_A \sqcup \theta'_A),\theta''_A}{\sqcup} R'' = R \underset{\theta_A,(\theta'_A \sqcup \theta''_A)}{\sqcup} (R' \underset{\theta'_A,\theta''_A}{\sqcup} R'')$.

□

4.3.2 Intersection

Proposition 4.3.3. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$ and $R' \subseteq Rel_{\theta'_A}(I, D_A)$ be simple. Then there exists a unique simple set $R" \subseteq Rel_{\theta_A \sqcup \theta'_A}(I, D_A)$ such that $\sqcup R" = (\sqcup R) \sqcap (\sqcup R')$ and $\sqcup R" \neq 0_{ID_A}$.*

Proof. We consider that $\tau \in R$ and $\tau' \in R'$. Since we have Proposition 4.2.4 then we have $\tau \neq 0_{ID_A}$ and $\tau' \neq 0_{ID_A}$. We have $\sqcup R$, $\sqcup R' : I \rightharpoondown D_A$ are non-zero relation. Let $\sqcup R" = (\sqcup R) \sqcap (\sqcup R')$ and we have $\sqcup R" : I \rightharpoondown D_A$ is non-zero relation. Using Theorem 4.2.2 then there exist unique simple set $R" \subseteq Rel_{\theta \sqcup \theta'}(I, D_A)$.

□

Definition 4.3.2. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$ and $R' \subseteq Rel_{\theta'_A}(I, D_A)$ be simple. There are exists unique simple, $R" \subseteq Rel_{\theta_A \sqcup \theta'_A}(I, D_A)$ such that $\sqcup R" = (\sqcup R) \sqcap (\sqcup R')$ and $\sqcup R" \neq 0_{ID_A}$. We define the $R \underset{\theta_A,\theta'_A}{\sqcap} R'$ by $R"$.*

Note: *$R \sqcap R' \neq R \underset{\theta_A,\theta'_A}{\sqcap} R'$ in general.*

Example 4.3.2. *We continue from Example 4.3.1 with same attribute and domain set A, and equivalence relation θ_A and θ'_A. We would like to compute*

Lecturer	Subject	Room
{David}	{Algebra, Statistics}	{A505}
{Codd}	{Cryptography}	{B101}

Table 4.10: $R \underset{\theta_A,\theta'_A}{\sqcap} R'$

the intersection between Table 4.7 and Table 4.8. Then we get Table 4.10 as the result of $R \underset{\theta_A,\theta'_A}{\sqcap} R'$ with equivalence relation θ' since $\theta_A \sqcup \theta'_A = \theta_A$.

Proposition 4.3.4. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$, $R' \subseteq Rel_{\theta'_A}(I, D_A)$, and $R'' \subseteq Rel_{\theta''_A}(I, D_A)$ be simple, then:*

1. $R \underset{\theta_A,\theta_A}{\sqcap} R = R$,
2. $R \underset{\theta_A,\theta'_A}{\sqcap} R' = R' \underset{\theta_A,\theta'_A}{\sqcap} R$,
3. $(R \underset{\theta_A,\theta'_A}{\sqcap} R') \underset{(\theta_A\sqcup\theta'_A),\theta''_A}{\sqcap} R'' = R \underset{\theta_A,(\theta'_A\sqcup\theta''_A)}{\sqcap} (R' \underset{\theta'_A,\theta''_A}{\sqcap} R'')$,
4. $(R \underset{\theta_A,\theta'_A}{\sqcap} R') \underset{(\theta_A\sqcup\theta'_A),\theta''_A}{\sqcup} R'' = (R \underset{\theta_A,\theta''_A}{\sqcup} R'') \underset{\theta,(\theta'_A\sqcup\theta''_A)}{\sqcap} (R' \underset{\theta'_A,\theta''_A}{\sqcup} R'')$,
5. $(R \underset{\theta_A,\theta'_A}{\sqcup} R') \underset{(\theta_A\sqcup\theta'_A),\theta''_A}{\sqcap} R'' = (R \underset{\theta_A,\theta''_A}{\sqcap} R'') \underset{\theta_A,(\theta'_A\sqcup\theta''_A)}{\sqcup} (R' \underset{\theta'_A,\theta''_A}{\sqcap} R'')$.

Proof. We have $\sqcup R, \sqcup R', \sqcup R'' : I \rightharpoonup D_A$ are non zero relation, then:

1. Since we have $\theta_A \sqcup \theta_A = \theta_A$ and $(\sqcup R) \sqcap (\sqcup R) = \sqcup R$, then by Proposition 4.3.3 we get exist unique $R \underset{\theta_A,\theta_A}{\sqcap} R = R$.

2. Since we have $\theta_A \sqcup \theta'_A = \theta'_A \sqcup \theta_A$ and $(\sqcup R) \sqcap (\sqcup R') = (\sqcup R') \sqcup (\sqcup R)$, then by Proposition 4.3.3 we get exist unique $R \underset{\theta_A,\theta'_A}{\sqcap} R' = R' \underset{\theta_A,\theta'_A}{\sqcap} R$.

3. Since we have $(\theta_A \sqcup \theta'_A) \sqcup \theta_A = \theta_A \sqcup (\theta'_A \sqcup \theta_A)$ and $((\sqcup R) \sqcap (\sqcup R')) \sqcap (\sqcup R'') = (\sqcup R) \sqcap ((\sqcup R') \sqcap (\sqcup R''))$, then by Proposition 4.3.3 we get exist unique
$(R \underset{\theta_A,\theta'_A}{\sqcap} R') \underset{(\theta_A\sqcup\theta'_A),\theta''_A}{\sqcap} R'' = R \underset{\theta_A,(\theta'_A\sqcup\theta''_A)}{\sqcap} (R' \underset{\theta'_A,\theta''_A}{\sqcap} R'')$.

4. Since we have $(\theta_A \sqcup \theta'_A) \sqcup \theta''_A = (\theta_A \sqcup \theta''_A) \sqcup (\theta'_A \sqcup \theta''_A)$ and by Proposition 2.2.2.4 we have $((\sqcup R) \sqcap (\sqcup R')) \sqcup (\sqcup R'') = ((\sqcup R) \sqcup (\sqcup R'')) \sqcap ((\sqcup R') \sqcup (\sqcup R''))$, then by Proposition 4.3.1 and Proposition 4.3.3 we get exist unique $(R \underset{\theta_A,\theta'_A}{\sqcap} R') \underset{(\theta_A\sqcup\theta'_A),\theta''_A}{\sqcup} R'' = (R \underset{\theta_A,\theta''_A}{\sqcup} R'') \underset{\theta_A,(\theta'_A\sqcup\theta''_A)}{\sqcap} (R' \underset{\theta'_A,\theta''_A}{\sqcup} R'')$.

5. Since we have $(\theta_A \sqcup \theta'_A) \sqcup \theta''_A = (\theta_A \sqcup \theta''_A) \sqcup (\theta'_A \sqcup \theta''_A)$ and $((\sqcup R) \sqcup (\sqcup R')) \sqcap (\sqcup R'') = ((\sqcup R) \sqcap (\sqcup R'')) \sqcup ((\sqcup R') \sqcap (\sqcup R''))$, then by Proposition 4.3.1 and Proposition 4.3.3 we get exist unique
$(R \underset{\theta_A,\theta'_A}{\sqcup} R') \underset{(\theta_A \sqcup \theta'_A),\theta''_A}{\sqcap} R'' = (R \underset{\theta_A,\theta''_A}{\sqcap} R'') \underset{\theta_A,(\theta'_A \sqcup \theta''_A)}{\sqcup} (R' \underset{\theta'_A,\theta''_A}{\sqcap} R'')$.

□

4.3.3 Selection

Proposition 4.3.5. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$ be simple. Let f be a function $f : D_A \to \{True, False\}$. There are exists unique simple, $R' \subseteq Rel_{\theta_A}(I, D_A)$ such that $\sqcup R' = \sqcup R \cdot \theta_A \cdot f$*

Proof. We consider that $\tau \in R$. Since we have Proposition 4.2.4 then we have $\tau \neq 0_{ID_A}$. We have $\sqcup R : I \rightharpoondown D_A$ are non-zero relation. Since $f : D_A \to \{True, False\}$, then we consider $\sqcup R' = \sqcup R \cdot \theta_A \cdot f$, so we have $\sqcup R' : I \rightharpoondown D_A$ is non-zero relation. Using Theorem 4.2.2 then there exist unique simple set $R' \subseteq Rel_{\theta_A}(I, D_A)$ □

Definition 4.3.3. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$ be simple. Let f be a function $f : D_A \to \{True, False\}$. There are exists unique simple, $R' \subseteq Rel_{\theta_A}(I, D_A)$ such that $\sqcup R' = \sqcup R \cdot \theta_A \cdot f$. We define the selection $\sigma_f(R \subseteq Rel_{\theta_A}(I, D_A)) = R'$.*

Example 4.3.3. *Let $A = \{Seller, Player, Position\}$, with:*
$D_{Seller} = \{$Inter,Chelsea,Persib$\}$,
$D_{Player} = \{$Essien, Ronaldo, Bale, Robin, Terry, Zulham, Atep, Zaneti$\}$, and
$D_{Position} = \{$CF,LF,RF,CB,LB,RB,CM,AM$\}$.

In Table 4.11, we have partition domain with equivalence relation θ such that:
$D^{\theta Seller}_{Seller} = \{\{$Inter$\}, \{$Madrid$\}, \{$Chelsea$\},, \{$Persib$\}\}$,
$D^{\theta_{Player}}_{Player} = \{\{$Essien, Ronaldo, Bale, Robin, Terry, Zulham, Atep, Zaneti$\}\}$, and
$D^{\theta_{Position}}_{Position} = \{\{CF,LF,RF\}, \{CB,LB,RB\}, \{CM,AM\}\}$.

Seller	Player	Position
{Inter}	{Ronaldo, Bale}	{LF,RF}
{Chelsea}	{Terry, Robbin}	{CB}
{Persib}	{Essien, Atep}	{CM,AM}
{Persib}	{Zulham}	{CF}
{Inter}	{Zaneti}	{RB}

Table 4.11: $R \subseteq Rel_{\theta_A}(I, D_A)$

We would like to select from Table 4.11 only attribute "Position". Then we get Table 4.12 as the result of $\sigma_{Position=LB}(R)$ Since "LB" equivalence with "RB" and "CB" then we got the result Table 4.3.1. In the traditional query, we will not get the result. But using fuzzy equivalence we got cooperative query answering. We got all of list "back" football player.

Seller	Player	Position
{Chelsea}	{Terry, Robbin}	{CB}
{Inter}	{Zaneti}	{RB}

Table 4.12: $\sigma_{Position=LB}(R)$

Proposition 4.3.6. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$ be simple. Let f be a function $f, g : D_A \leftarrow \{True, False\}$..*

1. $\sigma_{f \sqcup g}(R) = \sigma_f(R) \sqcup \sigma_g(R)$, *and*
2. $\sigma_{f \sqcap g}(R) \sqsubseteq \sigma_f(R) \sqcap \sigma_g(R))$.

Proof.
1. Since we have $(\sqcup R) \cdot \theta_A \cdot (f \sqcup g) = (\sqcup R \cdot \theta_A \cdot f) \sqcup (\sqcup R \cdot \theta_A \cdot g)$. Then we get $\sigma_{f \sqcup g}(R) = \sigma_f(R) \sqcup \sigma_g(R)$.
2. Since we have $(\sqcup R) \cdot \theta_A \cdot (f \sqcap g) \sqsubseteq (\sqcup R \cdot \theta_A \cdot f) \sqcap (\sqcup R \cdot \theta_A \cdot g)$. Then we get $\sigma_{f \sqcap g}(R) \sqsubseteq \sigma_f(R) \sqcap \sigma_g(R)$
   □

□

4.3.4 Projection

Proposition 4.3.7. *Let $X \subseteq A$, $R \subseteq Rel_{\theta_A}(I, D_A)$ is simple then there exist unique simple $R' \subseteq Rel_{\theta_X}(I, D_X)$ such that $\theta_X = \rho^{\#}_{A,X}\theta_A\rho_{A,X} : D_X \rightharpoonup D_X$ and $\sqcup R' = \sqcup R \cdot \rho_{A,X}$*

Proof. We consider that $\tau \in R$. Since we have Proposition 4.2.4 then we have $\tau \neq 0_{ID}$. We have $\sqcup R : I \rightharpoonup D$ are non-zero relation. Since $\sqcup R$ is non zero relation, then if we consider $\sqcup R' = \sqcup R \cdot \rho_{A,X}$, so we have $\sqcup R' : I \rightharpoonup D$ is non-zero relation. And also we have $\theta_X = \rho^{\#}_{A,X}\theta_A\rho_{A,X} : D_X \rightharpoonup D_X$, using Theorem 4.2.2 then there exist unique simple set $R' \subseteq Rel_{\theta_X}(I, D_X)$ □

Definition 4.3.4. *Let $X \subseteq A$, $R \subseteq Rel_{\theta_A}(I, D_A)$ is simple then there exist simple $R' \subseteq Rel_{\theta_X}(I, D_X)$ such that $\theta_X = \rho^{\#}_{A,X}\theta_A\rho_{A,X} : D_X \rightharpoonup D_X$ and $\sqcup R' = \sqcup R \cdot \rho_{A,X}$. We define the **Projection** $\pi_X(R)$ by R'.*

Example 4.3.4. *We continue from Example 4.3.3, we have Table 4.11 with collection of attributes set $A = \{Seller, Player, Position\}$, and equivalence relation θ. We consider $B = \{Player, Position\}$, then we get Table 4.13 as the result of $\pi_B(R \subseteq Rel_{\theta_A}(I, D_A))$.*

Proposition 4.3.8. *Let $R \subseteq Rel_{\theta_A}(I, D_A)$, and $R' \subseteq Rel_{\theta'_A}(I, D_A)$ be simple, then:.*

1. $\pi_{A,X}(R) \underset{\theta_A,\theta'_A}{\sqcup} \pi_{A,X}(R') = \pi_{A,X}(R \underset{\theta_A,\theta'_A}{\sqcup} R')$, *and*

Player	Position
{Ronaldo, Bale, Zulham}	{CF,LF,RF}
{Terry, Robbin, Zaneti}	{CB, RB}
{Essien, Atep}	{CM,AM}

Table 4.13: $\pi_B(R \subseteq Rel_{\theta_A}(I, D_A))$.

As we can see in Table 4.13. {Zulham} become one tuple with {Ronaldo, Bale} and also {Zaneti} become one tuple with {Terry, Robbin}. Since {CF} one class with {RF,LF} and {RB} one class with {CB}.

2. $\pi_{A,X}\left(R \underset{\theta_A,\theta'_A}{\sqcap} R'\right) \sqsubseteq \pi_{A,X}(R) \underset{\theta_A,\theta'_A}{\sqcap} \pi_{A,X}(R')$.

Proof. 1. Since we have $\pi_{A,X}(R) \underset{\theta_A,\theta'_A}{\sqcup} \pi_{A,X}(R')$ and using Proposition 2.2.2.5 then we have $\sqcup R'' = (\sqcup R \cdot \rho_{A,X}) \sqcup (\sqcup R' \cdot \rho_{A,X}) = ((\sqcup R) \sqcup (\sqcup R') \cdot \rho_{A,X}$, also using Proposition 2.2.2.5 $\theta_X = (\rho^{\#}_{A,X}\theta_A\rho_{A,X}) \sqcup (\rho^{\#}_{A,X}\theta'_A\rho_{A,X}) = \rho^{\#}_{A,X}(\theta_A \sqcup \theta'_A\rho_{A,X}$. Then we prove $\pi_{A,X}(R) \underset{\theta_A,\theta'_A}{\sqcup} \pi_{A,X}(R') = \pi_{A,X}(R \underset{\theta_A,\theta'}{\sqcup} R')$.

2. Since we have $\pi_{A,X}\left(R \underset{\theta,\theta'}{\sqcap} R'\right)$ and using Proposition 2.2.2.5 then we have $\sqcup R'' = ((\sqcup R) \sqcap (\sqcup R') \cdot \rho_{A,X} \sqsubseteq (\sqcup R \cdot \rho_{A,X}) \sqcap (\sqcup R' \cdot \rho_{A,X})$, also using Proposition 2.2.2.5 $\theta_X = (\rho^{\#}_{A,X}\theta_A\rho_{A,X}) \sqcup (\rho^{\#}_{A,X}\theta'_A\rho_{A,X}) = \rho^{\#}_{A,X}(\theta_A \sqcup \theta'_A\rho_{A,X}$. Then we prove $\pi_{A,X}\left(R \underset{\theta,\theta'}{\sqcap} R'\right) \sqsubseteq \pi_{A,X}(R) \underset{\theta_A,\theta'_A}{\sqcap} \pi_{A,X}(R')$.

□

4.3.5 Injection

Definition 4.3.5. *Let $X \subseteq A$, $R \subseteq Rel_{\theta_X}(I, D_X)$ is simple then there exist simple $R' \subseteq Rel_{\theta_A}(I, D_A)$ where $\sqcup R' = \sqcup R \cdot \rho^{\#}_{A,X}$ and $\theta_A = \rho_{A,X} \cdot \theta_X$.*

Notation: Let R_X and R_Y be relations then $R_X \dot{\sqcap} R_Y = \{\tau_X \sqcap \tau_Y | \tau_X \in R_X, \tau_Y \in R_Y,$ and $\tau_X \sqcap \tau_Y \neq \emptyset\}$.

Proposition 4.3.9. *Let R_X and R_Y be relations then:*

$$(R_X \dot{\sqcap} R_Y) \dot{\sqcap} R_Z = R_X \dot{\sqcap} (R_Y \dot{\sqcap} R_Z)$$

Proof. Let $\tau_X \in R_X, \tau_Y \in R_Y,$ and $\tau_Z \in R_Z$. Since we have $(\tau_X \sqcap \tau_Y) \sqcap \tau_Z = \tau_X \sqcap (\tau_Y \sqcap \tau_Z)$ then we prove $(R_X \dot{\sqcap} R_Y) \dot{\sqcap} R_Z = R_X \dot{\sqcap} (R_Y \dot{\sqcap} R_Z)$ □

4.3.6 Natural Join

Definition 4.3.6. *Let $X, Y \sqsubseteq A$ and $Z = X \sqcup Y$. For database relations $R_X \subseteq Rel_{\theta_X}(I, D_X)$ and $R_Y \subseteq Rel_{\theta_Y}(I, D_Y)$. We define* ***natural join*** *between*

R_X *and* R_Y,

$$R_X \bowtie R_Y = \eta_{ZX}(R_X) \dot{\sqcap} \eta_{ZY}(R_Y) : D_Z \rightharpoondown D_Z$$

Example 4.3.5. *Let* $C = \{$*Buyer, Position*$\}$, $D_{Buyer} = \{$*Madrid, MU, Persegal*$\}$ *with partition* $D^{\theta_{Buyer}}_{Student} = \{\{$*Madrid, MU, Persegal*$\}\}$, $D_{Position} = \{$*CF,LF,RF CB,LB,RB,CM,AM*$\}$ *with partition,* $D^{\theta_{Position}}_{Position} = \{\{$*CF,LF,RF,CB,LB,RB,CM,AM*$\}\}$. *We consider relation* $R_C \subseteq Rel_{\theta'}(I, D_C)$ *be simple and shown in Table 4.14.*

Buyer	Position
{Madrid}	{CF,LF, RB }
{MU}	{RF, AM}
{Persegal}	{CB, CM}

Table 4.14: $R_C \subseteq Rel_{\theta_C}(I, D_C)$

We would like to compute natural join between Table 4.11 and Table 4.14. The we can see Table 4.15 as the result $R \subseteq Rel_\theta(I, D)$ *(Table 4.11)* $\bowtie R_C \subseteq Rel_{\theta_C}(I, D_C)$ *(Table 4.14).*

Proposition 4.3.10. *Let* $X, Y, Z \sqsubseteq A$. *We consider* $R_X \subseteq Rel_{\theta_X}(I, D_X)$, $R_Y \subseteq Rel_{\theta_Y}(I, D_Y)$, $R_Z \subseteq Rel_{\theta_Z}(I, D_Z)$ *be simple, then:*

$$(R_X \bowtie R_Y) \bowtie R_Z = R_X \bowtie (R_Y \bowtie R_Z)$$

Proof.

$$\begin{aligned}(R_X \bowtie R_Y) \bowtie R_Z &= (\eta_{AX}(R_X) \dot{\sqcap} \eta_{AY}(R_Y)) \dot{\sqcap} \eta_{AZ}(R_Z) \\ &= \eta_{AX}(R_X) \dot{\sqcap} (\eta_{AY}(R_Y)) \dot{\sqcap} \eta_{AZ}(R_Z)) \\ &= R_X \bowtie (R_Y \bowtie R_Z)\end{aligned}$$

□

Proposition 4.3.11. *Let* $X, Y \sqsubseteq A$ *with* $Z = X \sqcup Y$, $R_X \subseteq Rel_{\theta_X}(I, D_X)$, $R_Y \subseteq Rel_{\theta_Y}(I, D_Y)$ *be simple, then:*

1. $\pi_{Z,X}(R_X \bowtie R_Y) = R_X$.

Buyer	Seller	Player	Position
{Inter}	{Madrid}	{Ronaldo, Bale}	{LF}
{Inter}	{MU}	{Ronaldo, Bale}	{RF}
{Inter}	{Madrid}	{Zaneti}	{RB}
{Chelsea}	{Persegal}	{Terry, Robin}	{CB}
{Persib}	{Persegal}	{Essien, Atep}	{CM}
{Persib}	{MU}	{Essien, Atep}	{AM}

Table 4.15: Result of Table 4.11 $\bowtie$ Table 4.14

2. $R_Z \sqsubseteq \pi_{Z,X}(R_Z) \bowtie \pi_{Z,Y}(R_Z)$.

Proof. 1. To prove $R_Z \sqsubseteq \pi_{Z,X}(R_Z) \bowtie \pi_{Z,Y}(R_Z)$, we have to prove:

a From definition we have $\theta_Z = (\rho_{Z,X} \cdot \theta_X \cdot \rho^{\#}_{Z,X}) \sqcap (\rho_{Z,Y} \cdot \theta_Y \cdot \rho^{\#}_{Z,Y}) : D_Z \rightharpoondown D_Z$. Then we project to the set of attributes X, then $\theta_X = \rho^{\#}_{Z,X}(\theta_Z)\rho_{Z,X}$. Then we get $\theta_X = \rho^{\#}_{Z,X}(\rho_{Z,X} \cdot \theta_X \cdot \rho^{\#}_{Z,X})\rho_{Z,X}$. Since $\rho^{\#}_{Z,X} \cdot \rho_{Z,X} \sqsubseteq \mathrm{id}_X$ then we prove that $\theta_X = \theta_X$.

b From definition we have $\sqcup R_Z = (\sqcup R_X \cdot \rho^{\#}_{Z,X}) \sqcap (\sqcup R_Y \cdot \rho^{\#}_{Z,Y})$. Then we project to the set of attributes X, then $\sqcup R_X = \sqcup R_Z \cdot \rho_{Z,X}$. Then we get $\sqcup R_X = \sqcup R_Z \cdot \rho_{Z,X} = (\sqcup R_X \cdot \rho^{\#}_{Z,X}) \cdot \rho_{Z,X}$. Since $\rho^{\#}_{Z,X} \cdot \rho_{Z,X} \sqsubseteq \mathrm{id}_X$ then we prove that $\sqcup R_X = \sqcup R_X$.

2. To prove $\pi_{Z,X}(R_X \bowtie R_Y) = R_X$, we have to prove:

a We have $\theta_X = \rho^{\#}_{Z,X}(\theta_Z)\rho_{Z,X}$ and $\theta_Y = \rho^{\#}_{Z,Y}(\theta_Z)\rho_{Z,Y}$. Then we can compute natural join-θ_Z, $\theta_Z = (\rho_{Z,X} \cdot \theta_X \cdot \rho^{\#}_{Z,X}) \sqcap (\rho_{Z,Y} \cdot \theta_Y \cdot \rho^{\#}_{Z,Y}) : D_Z \rightharpoondown D_Z$. Since $\rho^{\#}_{Z,\bullet} \cdot \rho_{Z,\bullet} \sqsubseteq \mathrm{id}_\bullet$ ($\bullet = X$ or Y) then we prove that $\theta_Z = \theta_Z$.

b Since $\sqcup R_X = \sqcup R_Z \cdot \rho_{Z,X}$ and $\sqcup R_Y = \sqcup R_Z \cdot \rho_{Z,Y}$ and $\mathrm{id}_Z \sqsubseteq \rho_{Z,\bullet} \cdot \rho^{\#}_{Z,\bullet}$. Then we get $\sqcup R_Z \sqsubseteq (\sqcup R_X \cdot \rho^{\#}_{Z,X}) \sqcap (\sqcup R_Y \cdot \rho^{\#}_{Z,Y})$.

Then we prove $R_Z \sqsubseteq \pi_{Z,X}(R_Z) \bowtie \pi_{Z,Y}(R_Z)$

□

Proposition 4.3.12. *Let $X, Y \sqsubseteq A$ and $Z = X \sqcup Y$. For database relations $R_1 \subseteq Rel_{\theta_X}(I, D_X)$, $R'_1 \subseteq Rel_{\theta'_X}(I, D_X)$, $R_2 \subseteq Rel_{\theta_Y}(I, D_Y)$, and $R'_2 \subseteq Rel_{\theta'_Y}(I, D_Y)$. We consider $\theta_X \sqsubseteq \theta'_X$, and $\theta_Y \sqsubseteq \theta'_Y$. If $R_1 \sqsubseteq R_2$ and $R' \sqsubseteq R'_2$ then $R_1 \bowtie R' \sqsubseteq R_2 \bowtie R'_2$*

Proof. We have to prove:

a Since $\theta_X \sqsubseteq \theta'_X$ and $\theta_Y \sqsubseteq \theta'_Y$ then we prove $(\rho_{X \sqcup Y,X} \cdot \theta_X \cdot \rho^{\#}_{X \sqcup Y,X}) \sqcap (\rho_{X \sqcup Y,Y} \cdot \theta_Y \cdot \rho^{\#}_{X \sqcup Y,Y}) \sqsubseteq (\rho_{X \sqcup Y,X} \cdot \theta'_X \cdot \rho^{\#}_{X \sqcup Y,X}) \sqcap (\rho_{X \sqcup Y,Y} \cdot \theta'_Y \cdot \rho^{\#}_{X \sqcup Y,Y})$

b Since $R_1 \sqsubseteq R'_1$ then we have $\sqcup R_1 \sqsubseteq \sqcup R'_1$. Also $R_2 \sqsubseteq R'_2$ then we have $\sqcup R_2 \sqsubseteq \sqcup R'_2$. Then we prove $(\sqcup R_1 \cdot \rho^{\#}_{Z,X}) \sqcap (\sqcup R_2 \cdot \rho^{\#}_{Z,Y}) \sqsubseteq (\sqcup R'_1 \cdot \rho^{\#}_{Z,X}) \sqcap (\sqcup R'_2 \cdot \rho^{\#}_{Z,Y})$

□

Chapter 5

Fuzzy Implication and Functional Dependency

Dependency theory is an important theory in data analysis. In 1970, Codd[Cod70] introduced a notion of functional dependency, which is a constraint between two sets of attributes. Ganter and Wille[GW97] defined implication dependency for a formal concept analysis. Formal concept analysis is a method in data mining using lattice theory proposed by Wille[Wil82]. Ishida et al.[IHK08] have been developed a formalization of completeness and soundness for functional dependencies based on Beeri et al[BFH77] using relational calculus. A comparison of formalizations of a functional dependency and an implication dependency was also investigated. We extend its formalization using a fuzzy relational concept. A notion of fuzzy was initially proposed by Zadeh[Zad65]. We define fuzzy equivalence relation using an indiscernibility relation by Düntsch and Günther[DG98] to construct a fuzzy functional dependency. The main purpose of our formalizations is investigating correlations between a functional dependency and an implication dependency. We follow the Ishida's approach to define Armstrong's inference rules, and show the soundness and completeness in our formalization using a fuzzy relation.

5.1 Fuzzy Context

A **fuzzy context** is a fuzzy relation $\alpha : X \rightharpoondown Y$. We identify $x \in X$ with a (crisp) fuzzy relation $\hat{x} : I \rightharpoondown X$ such that $\hat{x}(*, x') = 1$ if $x' = x$ and $\hat{x}(*, x') = 0$ otherwise. Then fuzzy relation $\alpha : X \rightharpoondown Y$, which is equivalent to its intent function $\alpha^{@} : X \to \wp_f(Y)$ defined as $\alpha^{@}(x) = \hat{x}\alpha$ for all $x \in X$. Thus the fuzzy context $\alpha : X \rightharpoondown Y$ is equivalent to an X-indexed set $\mathcal{T} = \{\hat{x}\alpha | x \in X\}$ of fuzzy relations into Y , where $\hat{x}\alpha$ denotes the composite of fuzzy relations $\hat{x} : I \to X$ and $\alpha : X \rightharpoondown Y$.

Table 5.1 is set of tuple, $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$ with $\mathcal{T}_i = \hat{x_i}\alpha$. So, we can write $\mathcal{T} = \{\hat{x_0}\alpha, \hat{x_1}\alpha, \ldots, \hat{x_n}\alpha\} \sqsubseteq \wp_f(Y)$.

	y_1	y_2	y_3	$\cdots$	
x_0	0.1	0.5	0	$\cdots$	$x_0\alpha$
x_1	0.9	1	0.3	$\cdots$	$x_1\alpha$
x_2	0	0.7	0.2	$\cdots$	$x_2\alpha$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	0.4	1	0	$\cdots$	$x_n\alpha$

Table 5.1: Fuzzy Context Table

Then a subset $\mathcal{T}$ of $\wp_f(Y)$ will be called a **fuzzy context on** Y.
We note A is a fuzzy relation into Y is a fuzzy relation $A : I \rightharpoondown Y$.

For a fuzzy context $\mathcal{T}$ on Y we define another fuzzy context $\mathcal{T}^*$ on Y by $\mathcal{T}^* = \{\sqcap\mathcal{C}|\mathcal{C} \subseteq \mathcal{T}\}$. For example, $\{x\alpha|x \in X\}^*$ is the set of all formal concepts for a fuzzy relation $\alpha : X \rightharpoondown Y$. Then we can conclude that $\mathcal{T} \subseteq \mathcal{T}^*$

5.2 Armstrong's Inference Rules

Armstrong's inference rules gives a basic framework of databases to treat the logical structure of dependencies on an attribute set. Let A and B be fuzzy relations into Y . A formal expression $A \rhd B$, namely, an ordered pair of A and B, is called a dependency on the attribute set Y.

$$[\text{A0}]\frac{}{A \rhd A} \qquad [\text{A1}]\frac{A \rhd B}{A \sqcup C \rhd B} \qquad [\text{A2}]\frac{A \rhd B \quad B \sqcup C \rhd D}{A \sqcup C \rhd D}$$

Let $\mathcal{L}$ be a set of dependencies on Y . A derivation (or proof) from $\mathcal{L}$ is a nonempty sequence of dependencies such that $\{A_0 \rhd B_0, A_1 \rhd B_1, \ldots, A_m \rhd B_m\}$, for all $k = 0, 1, \ldots, m$, one of the following holds: **(a)** $A_k = B_k([A0])$ or $A_k \rhd B_k$ is in $\mathcal{L}$, **(b)** $\exists i < k$ such that: $[A1]\ \frac{A_i \rhd B_i}{A_k \rhd B_k}$, **(c)** $\exists i, j < k$ such that: $[A2]\ \frac{A_i \rhd B_i \quad A_j \rhd B_j}{A_k \rhd B_k}$.

A dependency $A \rhd B$ is provable from $\mathcal{L}$, written as $\mathcal{L} \vdash A \rhd B$, if there is a derivation $\{A_0 \rhd B_0, A_1 \rhd B_1, ..., A_m \rhd B_m\}$ from $\mathcal{L}$ such that $A = A_m$ and $B = B_m$.

Let $\mathcal{L}$ be a set of dependencies on Y and A a fuzzy relation into Y . Define a subset $\mathcal{L}_A$ of $\wp_f(Y)$ by $\mathcal{L}_A = \{C : I \rightharpoondown Y|\mathcal{L} \vdash A \rhd C\}$. Since Ishida defined the Armstrong's rules using category theory then we used his lemma for fuzzy relation case.

Lemma 5.2.1 ([IHK08]). *Let B be a finite fuzzy relation into Y . Then $\mathcal{L} \vdash A \rhd B$ if and only if $B \sqsubseteq \sqcup\mathcal{L}_A$.*

Proof. $(\rightarrow)$ Assume $\mathcal{L} \vdash A \rhd B$. Then $B \in \mathcal{L}_A$ by the definition of $\mathcal{L}_A$ and so $B \sqsubseteq \sqcup\mathcal{L}_A$.

($\leftarrow$) As B is finite there is a finite subset $J \subseteq Y$ such that $B = \sqcup_{y \in J} B_y$. Assume $B \sqsubseteq \sqcup \mathcal{L}_A$ and let $y \in J$. Then we have $B_y \sqsubseteq \sqcup \mathcal{L}_A$ and

$$B_y(*, y) = B(*, y) \leq \vee_{C \in \mathcal{L}_A} C(*, y) \doteq \max_{C \in \mathcal{L}_A} C(*, y)$$

($\doteq$ Because the set $[0,1]_n$ of fuzzy values is finite.) Hence $\exists D \in \mathcal{L}_A . B_y(*, y) \leq D(*, y)$, which implies $B_y \sqsubseteq D$. Hence we have $L \vdash A \rhd B_y$, since $\{A \rhd D, D \rhd B_y, A \rhd B_y\}$ is a derivation from $\mathcal{L}$. Therefore $\mathcal{L} \vdash A \rhd B$ holds by the union rule [A3], because $B = \sqcup_{y \in J} B_y$ is a finite union. □

Remark. The above lemma always holds if the attribute set Y is finite.

5.3 Fuzzy Functional Dependency

The formalization of indiscernibility(equivalence) relation using Duntsch approach[DG98] has been introduced by Ishida[IHK08]. We extend its formalization to the fuzzy relation using fuzzy operators to define fuzzy equivalence relation. We also follow some properties to show that our formalization can be used for general relations.

Let $\mathcal{T}$ be a fuzzy context table with set of objects X and set of attributes Y and A be a member fuzzy power set of Y, $A \in \wp_f(Y)$. **Fuzzy indiscernibility relation of** A ($\theta[A]$) defined by:

$$(S,T) \in \theta[A] \text{ if and only if } S \sqcap A = T \sqcap A, \text{ for all } S, T \in \mathcal{T}.$$

Let $S, T, U \in \mathcal{T}$, the fuzzy indiscernibility relation has some characteristics such as reflexive, symmetry, transitive: $(S,S) \in \theta[A]$, $(S,T) \in \theta[A]$ if and only if $(T,S) \in \theta[A]$, if $(S,T) \in \theta[A]$ and $(T,U) \in \theta[A]$ then $(S,U) \in \theta[A]$.

Proof. 1. $(S,S) \in \theta[A] \leftrightarrow S \sqcap A = S \sqcap A$ for all $S \in \tau$ (Trivial),

2. We consider $(S,T) \in \theta[A]$, then we get

$$\begin{aligned} (S,T) \in \theta[A] &\leftrightarrow S \sqcap A = T \sqcap A \\ &\leftrightarrow T \sqcap A = S \sqcap A \\ &\leftrightarrow (T,S) \in \theta[A] \end{aligned}$$

   Then we get $(S,T) \in \theta[A] \leftrightarrow (T,S) \in \theta[A]$

3. Since we have $(S,T) \in \theta[A]$, and $(T,U) \in \theta[A]$, then

$$\begin{aligned} (S,T) \in \theta[A] \wedge (T,U) \in \theta[A] &\leftrightarrow (S \sqcap A = T \sqcap A) \wedge (T \sqcap A = U \sqcap A \\ &\leftrightarrow S \sqcap A = U \sqcap A \\ &\leftrightarrow (S,U) \in \theta[A] \end{aligned}$$

   Then we get $(S,T) \in \theta[A] \wedge (T,U) \in \theta[A] \leftrightarrow (S,U) \in \theta[A]$.

□

Proposition 5.3.1. *Let A and B be fuzzy relation into Y. Then:*
(a) $\theta[\nabla_{IY}] = id_{\wp_f(Y)}$ *and* $\theta[\mathbf{0}_{IY}] = \nabla_{\wp_f(Y)\wp_f(Y)}$, ***(b)*** $\theta[A \sqcup B] = \theta[A] \cap \theta[B]$,
(c) $\theta[A \sqcap B] = \theta[A]\theta[B]$.

Proof. (a) 1. Since $\nabla_{IY} = 1$(maximum) for all relation $I \rightharpoondown Y$, then

$$(S,T) \in \theta[\nabla_{IY}] \leftrightarrow S \sqcap \nabla_{IY} = T \sqcap \nabla_{IY} \leftrightarrow S = T$$

Since $(S,T) \in \theta[\nabla_{IY} \leftrightarrow S = T$ then only $(S,S)\theta[A]$ or $(T,T)\theta[A]$ for all $S,T \in I \rightharpoondown Y$. Since $(S,S) \in id_{\wp_f(Y)}$ for all $S \in \wp_f(Y)$ then we get $\theta[\nabla_{IY}] = id_{\wp_f(Y)}$.
2. Since $\mathbf{0}_{IY} = 0$(minimum), for all relation $I \rightharpoondown Y$ then

$$(S,T) \in \theta[\mathbf{0}_{IY}] \rightarrow S \sqcap \mathbf{0}_{IY} = T \sqcap \mathbf{0}_{IY} \leftrightarrow \mathbf{0}_{IY} = \mathbf{0}_{IY}$$

It means all of pairs of (S,T) will be true for all $S \in \wp_f(Y)$ and $T \in \wp_f(Y)$ then we can get the pairs of $(S,T) \in \nabla_{\wp_f(Y)\wp_f(Y)}$.

(b) Since $S \sqcap [A \sqcup B] = (S \sqcap A) \sqcup (S \sqcap B)$, $T \sqcap [A \sqcup B] = (T \sqcap A) \sqcup (T \sqcap B)$ then we have:
$(S,T) \in \theta[A \sqcup B] \leftrightarrow S \sqcap (A \sqcup B) = T \sqcap (A \sqcup B)$
$\leftrightarrow (S \sqcap A) \sqcup (S \sqcap B) = (T \sqcap A) \sqcup (T \sqcap B)$
$\overset{*}{\leftrightarrow} \langle S \sqcap A = T \sqcap A\rangle \wedge \langle S \sqcap B = T \sqcap B\rangle \leftrightarrow \langle (S,T) \in \theta[A]\rangle \wedge \langle (S,T) \in \theta[B]\rangle$
$\leftrightarrow (S,T) \in \theta[A] \cap \theta[B]$

Then we get $\theta[A \sqcup B] = \theta[A] \cap \theta[B]$

(*)($\leftarrow$): Since $S \sqcap (A \sqcup B) = (S \sqcap A) \sqcup (S \sqcap B)$, then

$$\begin{aligned} &\langle S \sqcap A = T \sqcap A\rangle \wedge \langle S \sqcap B = T \sqcap B\rangle \\ &\rightarrow (S \sqcap A) \sqcup (S \sqcap B) = (T \sqcap A) \sqcup (T \sqcap B) \\ &\rightarrow S \sqcap (A \sqcup B) = T \sqcap (A \sqcup B) \end{aligned}$$

(*)($\rightarrow$): Since $S \sqcap (A \sqcup B) = (S \sqcap A) \sqcup (S \sqcap B)$, then:
$S \sqcap (A \sqcup B) = T \sqcap (A \sqcup B) \rightarrow \exists U \in (S \sqcap (A \sqcup B)) \wedge \exists U \in (T \sqcap (A \sqcup B))$
$\rightarrow \exists U \in (S \sqcap (A \sqcup B)) \sqcap (T \sqcap (A \sqcup B))$
$\rightarrow \exists U \in ((S \sqcap A) \sqcup (S \sqcap B)) \sqcap ((T \sqcap A) \sqcup (T \sqcap B))$
$\rightarrow \exists U \in ((S \sqcap A) \sqcap ((T \sqcap A) \sqcup (T \sqcap B))) \sqcup ((S \sqcap B) \sqcap ((T \sqcap A) \sqcup (T \sqcap B)))$
$\rightarrow \exists U \in (((S \sqcap A) \sqcap (T \sqcap A)) \sqcup ((S \sqcap A) \sqcap (T \sqcap B)) \sqcup$
$(((S \sqcap B) \sqcap (T \sqcap A)) \sqcup ((S \sqcap B) \sqcap T \sqcap B)))$
$\rightarrow \exists U \in (((S \sqcap A) \sqcap (T \sqcap A)) \sqcup ((S \sqcap B) \sqcap (T \sqcap B))$
$\rightarrow \exists U \in (((S \sqcap A) \sqcap (T \sqcap A)) \vee \exists U \in ((S \sqcap B) \sqcap (T \sqcap B))$
$\rightarrow \exists U \in (((S \sqcap A) \sqcap (T \sqcap A)) \wedge \exists U \in ((S \sqcap B) \sqcap (T \sqcap B))$
$\rightarrow S \sqcap A = T \sqcap A \wedge S \sqcap B = T \sqcap B$

(c) From definition, we have $(S,T) \in \theta[A \sqcap B] \leftrightarrow S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B)$

$$
\begin{aligned}
(S,T) \in \theta[A \sqcap B] &\leftrightarrow S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B) \\
&\overset{*}{\leftrightarrow} \exists U \langle S \sqcap A = U \sqcap A \rangle \wedge \langle U \sqcap B = T \sqcap B \rangle
\end{aligned}
$$

$\overset{*}{\leftarrow}$: Since $U \sqcap A = S \sqcap A \rightarrow U \sqcap A \sqcap B = S \sqcap A \sqcap B$ and $U \sqcap B = T \sqcap B \rightarrow T \sqcap A \sqcap B = T \sqcap A \sqcap B$. Then we can get

$$\exists U \langle S \sqcap A = U \sqcap A \rangle \wedge \langle U \sqcap B = T \sqcap B \rangle \rightarrow S \sqcap A \sqcap B = T \sqcap A \sqcap B$$

$\overset{*}{\rightarrow}$: We consider set $U = (S \sqcap A) \sqcup (T \sqcap B)$. Then we have

$$
\begin{aligned}
U \sqcap A &= ((S \sqcap A) \sqcup (T \sqcap B)) \sqcap A \\
&= (S \sqcap A) \sqcup (T \sqcap B \sqcap A) \\
&= (S \sqcap A) \sqcup (S \sqcap B \sqcap A) \quad \{S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B)\} \\
&= (S \sqcap A)
\end{aligned}
$$

$$
\begin{aligned}
U \sqcap B &= ((S \sqcap A) \sqcup (T \sqcap B)) \sqcap B \\
&= (S \sqcap A \sqcap B) \sqcup (T \sqcap B) \\
&= (T \sqcap A \sqcap B) \sqcup (T \sqcap B) \quad \{S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B)\} \\
&= (T \sqcap B)
\end{aligned}
$$

then we can get

$$S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B) \rightarrow \exists U.(S \sqcap A = U \sqcap A) \wedge (U \sqcap B = T \sqcap B)$$

Since $(\rightarrow)$ and $(\leftarrow)$ we have $S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B) \leftrightarrow \exists U.(S \sqcap A = U \sqcap A) \wedge (U \sqcap B = T \sqcap B)$. Then we can get

$$
\begin{aligned}
(S,T) \in \theta[A \sqcap B] &\leftrightarrow S \sqcap (A \sqcap B) = T \sqcap (A \sqcap B) \\
&\overset{*}{\leftrightarrow} \exists U \langle S \sqcap A = U \sqcap A \rangle \wedge \langle U \sqcap B = T \sqcap B \rangle \\
&\leftrightarrow \exists U \langle (S,U) \in \theta[A] \rangle \wedge \langle (U,T) \in \theta[B] \rangle \\
&\leftrightarrow (S,T) \in \theta[A].\theta[B]
\end{aligned}
$$

then we prove $\theta[A \sqcap B] = \theta[A] \cdot \theta[B]$

□

Next, we extend the definition of functional dependency introduced by Codd[Cod70] and Ishida[IHK08] to the fuzzy functional dependency. We also use an fuzzy indiscernibility relation from previous explanation.

A functional dependency $A \rhd B$ is satisfied from $\mathcal{T}$, written as $\mathcal{T} \models_F A \rhd B$, is defined by: $\mathcal{T} \models_F A \rhd B$ if and only if $\langle (S,T) \in \theta[A] \rightarrow (S,T) \in \theta[B] \rangle$, for all S,T $\in \mathcal{T}$.

In the next proposition, we show that the functional dependencies satisfy the Armstrong's inference rules.

Proposition 5.3.2. *Let $\mathcal{T}$ be a fuzzy context on Y . Then*

(A0) $\mathcal{T} \models_F A \rhd A$,

(A1) If $\mathcal{T} \models_F A \rhd B$, *then* $\mathcal{T} \models_F A \sqcup C \rhd B$, *and*

(A2) If $\mathcal{T} \models_F A \rhd B$ *and* $\mathcal{T} \models_F B \sqcup C \rhd D$, *then* $\mathcal{T} \models_F A \sqcup C \rhd D$

Proof. We omit proofs for (A0) and (A1).
(A2) For all $S, T \in \mathcal{T}$, we note $S \sqcap (A \sqcup C) = T \sqcap (A \sqcup C)$ equivalent to $S \sqcap A = T \sqcap A$ and $S \sqcap C = T \sqcap C$. Since we assume $\mathcal{T} \models_F A \rhd B$, we have $S \sqcap (B \sqcup C) = T \sqcap (B \sqcup C)$. Since $\mathcal{T} \models_F B \sqcup C \rhd D$, we have $S \sqcap D = T \sqcap D$ which shows $T \models_F A \sqcup C \rhd D$. □

Proposition 5.3.3. *Let $\mathcal{T}_0 = \{A, \nabla_{IY}\}$ be a particular fuzzy context on Y such that $A \neq \nabla_I Y$. Then:* ***(a)*** *$T_0 \models_F \mathbf{0}_{IY} \rhd C$ if and only if $C \sqsubseteq A$,* ***(b)*** *$T_0 \models_F C \rhd \nabla_{IY}$ if and only if $C \not\sqsubseteq A$.*

Proof. (a) Since $\mathcal{T}_0 = \{A, \nabla_{IY}\}$ then we get: $\mathcal{T} \models_F \mathbf{0}_{IY} \rhd C \leftrightarrow \langle (A, \nabla_{IY}) \in \theta[\mathbf{0}_{IY}] \rangle \to \langle (A, \nabla_{IY}) \in \theta[C] \rangle$. From Proposition 5.8.a we have $\theta[\mathbf{0}_{IY}] = \nabla_{\wp_f(Y)\wp_f(Y)}$ then $\langle (A, \nabla_{IY}) \in \nabla_{\wp_f(Y)\wp_f(Y)} \rangle \to \langle (A, \nabla_{IY}) \in \theta[C] \rangle$. Since $\nabla_{\wp_f(Y)\wp_f(Y)}$ means all pairs(maximum) (A, ∇_{IY}) is equivalence relation then we can just focus $\langle (A, \nabla_{IY}) \in \theta[C] \rangle$ then $\mathcal{T} \models_F \mathbf{0}_{IY} \rhd C \leftrightarrow A \sqcap C = \nabla_{IY} \sqcap C$. Since $C \sqsubseteq \nabla_{IY}$ then =C $A \sqcap C = C$ then we get $C \sqsubseteq A$. We proves $T_0 \models_F \mathbf{0}_{IY} \rhd C \leftrightarrow C \sqsubseteq A$

(b) Since $\mathcal{T}_0 = \{A, \nabla_{IY}\}$ then we get: $\mathcal{T} \models_F C \rhd \nabla_{IY} \leftrightarrow \langle (A, \nabla_{IY}) \in \theta[C] \rangle \to \langle (A, \nabla_{IY}) \in \theta[\nabla_{IY}] \rangle$. From Proposition 5.8.a we have $\theta[\nabla_{IY}] = \mathrm{id}_{\wp_f(Y)}$ then we have $A = \nabla_{IY}$. then $\mathcal{T} \models_F \nabla_{IY} \rhd C \leftrightarrow \langle (A, \nabla_{IY}) \in \theta[C] \rangle \to \langle A = \nabla_{IY} \rangle$. Since we assume $A \neq \nabla_{IY}$ then we use rule transposition, we get

$$\langle (A, \nabla_{IY}) \in \theta[C] \rangle \to \langle A = \nabla_{IY} \rangle \leftrightarrow \langle A \neq \nabla_{IY} \rangle \to ((A, \nabla_{IY}) \notin \theta[C]$$
$$\leftrightarrow ((A, \nabla_{IY}) \notin \theta[C] \leftrightarrow A \sqcap C \neq \nabla_{IY} \sqcap C$$

Since $C \sqsubseteq \nabla_{IY}$ then $A \sqcap C \neq \nabla_{IY} \sqcap C \leftrightarrow A \sqcap C \neq C \leftrightarrow C \not\sqsubseteq A$. which proves $T_0 \models_F C \rhd \nabla_{IY} \leftrightarrow C \not\sqsubseteq A$.

□

5.4 Fuzzy Implication

Ganther and Wille[GW97] initially proposed implication on the lattice context. Ishida et al[IHK08] have been formalized of fuzzy implication using relational calculus. We extend its formalization to the fuzzy relation using fuzzy operators to define fuzzy implication. We also follow some properties to show that our formalization can be used for general case. **A fuzzy implication** $A \rhd B$ is satisfied from $\mathcal{T}$, written as $\mathcal{T} \models_G A \triangleright B$, is defined by: $\mathcal{T} \models_G A \triangleright B$ if and only if $\langle A \sqsubseteq T \to B \sqsubseteq T \rangle$ for all $T \in \mathcal{T}$. .

Proposition 5.4.1. *$\mathcal{T} \models_G A \triangleright B$ if and only if $\mathcal{T}^* \models_G A \triangleright B$.*

Proof. ($\leftarrow$) It is trivial from $\mathcal{T} \subseteq \mathcal{T}^*$.
($\to$) Let $U = \sqcap \mathcal{C} \in \mathcal{T}^*$ for $\mathcal{C} \subseteq \mathcal{T}$ then we have $A \sqsubseteq U$ equivalent to $A \sqsubseteq C$ ($\forall C \in \mathcal{C}$). Since $\mathcal{T} \models_G B \triangleright C$ ($\forall C \in \mathcal{C}$) and $B \sqsubseteq U$. So we have $\mathcal{T}^* \models_G A \triangleright B$. □

In the next proposition, we show that the fuzzy implication satisfy the Armstrong's inference rules.

Proposition 5.4.2. *Let $\mathcal{T}$ be a fuzzy context on Y . Then*

(A0) $\mathcal{T} \models_G A \triangleright A$

(A1) If $\mathcal{T} \models_G A \triangleright B$ then $\mathcal{T} \models_G A \sqcup C \triangleright B$,

(A2) If $\mathcal{T} \models_G A \triangleright B$ and $\mathcal{T} \models_G B \sqcup C \triangleright D$ then $\mathcal{T} \models_G A \sqcup C \triangleright D$.

Proof. (A0) Trivial from definition.
(A1) We assume $\mathcal{T} \models_G A \triangleright B$ and we have $A \sqcup C \sqsubseteq T \to A \sqsubseteq T$ Then we get $A \sqcup C \sqsubseteq T \to B \sqsubseteq T$ Hence $\mathcal{T} \models_G A \sqcup C \triangleright B$.
(A2) Let $T \in \mathcal{T}$. We note $A \sqcup C \sqsubseteq T$ is equivalent to $A \sqsubseteq T$ and $C \sqsubseteq T$. Since $\mathcal{T} \models_G A \triangleright B$, we get $B \sqcup C \sqsubseteq T$. Since $\mathcal{T} \models_G B \sqcup C \triangleright D$ then we get $D \sqsubseteq T$. Hence $\mathcal{T} \models_G A \sqcup C \triangleright D$. □

Proposition 5.4.3. *Let $\mathcal{T}_0 = \{A, \nabla_{IY}\}$ be a particular fuzzy context on Y such that $A \neq \nabla_I Y$. Then:* ***(a)*** *$\mathcal{T}_0 \models_G \mathbf{0}_{IY} \triangleright C$ if and only if $C \sqsubseteq A$,* ***(b)*** *$\mathcal{T}_0 \models_G C \triangleright \nabla_{IY}$ if and only if $C \not\sqsubseteq A$.*

Proof.

(a) Since $\mathcal{T}_0 = \{A\}$, from definition we have $\mathcal{T}_0 \models_G \mathbf{0}_{IY} \triangleright C \leftrightarrow (\mathbf{0}_{IY} \sqsubseteq A \to C \sqsubseteq A)$. Since $\mathbf{0}_{IY} \sqsubseteq A$ will be true in every set of A, then we get $\mathcal{T}_0 \models_G \mathbf{0}_{IY} \triangleright C \leftrightarrow C \sqsubseteq A$.

(b) Since $\mathcal{T}_0 = \{A\}$, from definition $\mathcal{T}_0 \models_G C \triangleright \nabla_{IY} \leftrightarrow (C \sqsubseteq A \to \nabla_{IY} \sqsubseteq A)$. Since we have $A \sqsubseteq \nabla_{IY}$ then we have $(C \sqsubseteq A \to \nabla_{IY} = A)$. Since we assume $A \neq \nabla_{IY}$ then using rule transposition we get $\nabla_{IY} \neq A \to C \not\sqsubseteq A$.. Then we proves that $\mathcal{T}_0 \models_G C \triangleright \nabla_{IY} \leftrightarrow C \not\sqsubseteq A$.

□

5.5 Soundness and Completeness

Armstrong[Arm74] and Beery[BFH77] was proposed the inference rules of formalization should be sound and complete. Next, we follow the proof soundness and completeness by Ishida's approach extending to fuzzy relations.

Theorem 5.5.1. *Let $\mathcal{L}$ a set of dependencies and $A \rhd B$ a dependency on a finite set Y . Then the following equivalence holds: $\mathcal{L} \vdash A \rhd B$ if and only if $\forall \mathcal{T} \subseteq \wp_f(Y).(\mathcal{T} \models_\bullet \mathcal{L} \to \mathcal{T} \models_\bullet A \rhd B)$, where $\bullet = F$ or G.*

Proof. ($\to$) Assume $\mathcal{L} \vdash A \rhd B$ and $\mathcal{T} \models_\bullet \mathcal{L}$. Then $\mathcal{T} \models_\bullet A \rhd B$ holds, because of the basic facts (A0), (A1) and (A2) in Proposition 5.3.2 and 5.4.2.
($\leftarrow$) Assume $\forall \mathcal{T}.(\mathcal{T} \models_\bullet \mathcal{L} \to \mathcal{T} \models_\bullet A \rhd B)$.
(I) In the case of $\sqcup \mathcal{L}_A = \nabla_{IY}$.
Since $B \sqsubseteq \nabla_{IY}$, we get $B \sqsubseteq \sqcup \mathcal{L}_A$. From Lemma 5.2.1, we get $\mathcal{L} \vdash A \rhd B$
(II) In the case of $\sqcup \mathcal{L}_A \neq \nabla_{IY}$.
We choose a fuzzy context $\mathcal{T}_0$ which satisfying conditions Proposition 5.3.3(a), 5.3.3(b), 5.4.3(a) and 5.4.3(b). Then we will see $\mathcal{T}_0 \models_\bullet \mathcal{L}$, that is, $\mathcal{T}_0 \models_\bullet C \rhd D$ for all $C \rhd D \in \mathcal{L}$.

(II-i) In the case of $C \sqsubseteq \sqcup \mathcal{L}_A$.
Since $\mathcal{L} \vdash A \rhd C$ and $C \rhd D \in \mathcal{L}$, we get $D \sqsubseteq \sqcup \mathcal{L}_A$. Next, by Proposition 5.3.3(a) or 5.4.3(a), we get $\mathcal{T}_0 \models_\bullet \mathbf{0}_{IY} \rhd D$. Since $\mathcal{T}_0 \models_\bullet C \rhd \mathbf{0}_{IY}$, we get $\mathcal{T}_0 \models_\bullet C \rhd D$. Hence if $C \sqsubseteq \sqcup \mathcal{L}_A$ then $\mathcal{T}_0 \models_\bullet C \rhd D$

(II-ii) In the case of $C \not\sqsubseteq \sqcup \mathcal{L}_A$.
Since we have conditions in Proposition 5.3.3(b) and 5.4.3(b), we have $\mathcal{T}_0 \models_\bullet \nabla_{IY} \rhd D$ then we proves if $C \not\sqsubseteq \sqcup \mathcal{L}_A$ then $\mathcal{T}_0 \models_\bullet C \rhd D$

Since $\mathcal{L} \vdash A \rhd A$, we get $A \sqsubseteq \sqcup \mathcal{L}_A$ by Lemma 5.2.1. Next, we use Proposition 5.3.3(a), 5.4.3(a) and $\mathcal{T}_0 \models_\bullet A \rhd B$, we get $\mathcal{T}_0 \models_\bullet \mathbf{0}_{IY} \rhd B$. By again we have $B \sqsubseteq \sqcup \mathcal{L}_A$ by Proposition 5.3.3(a) or 5.4.3(a). This complete the proof of $\mathcal{L} \vdash A \rhd B$. □

5.6 Comparison

In this section, we show the difference between an implication and a functional dependency, by using examples.

Example 5.6.1. *Let $X = \{x1, x2, x3, x4, x5\}$ be a domain of objects, $Y = \{y1, y2, y3, y4\}$ be a domain of attributes. Consider $\alpha : X \rightharpoondown Y$ be a fuzzy relation defined by Table 5.2. We assume $A = \{(y1, 0.9)\},\ B = \{(y3, 0.5)\},\ C = \{(y2, 0.9)\},\ D = \{(y4, 0.4)\}$ be fuzzy relations into Y (i.e. $A, B, C, D \in \wp_f(Y)$) From example we got:*

1. *$\mathcal{T} \models_F A \rhd B$ and $\mathcal{T} \not\models_G A \rhd B$*
2. *$\mathcal{T} \models_G C \rhd D$ and $\mathcal{T} \not\models_F C \rhd D$*

Then we can conclude that:$\mathcal{T} \models_F A \rhd B$ is not equivalent with $\mathcal{T} \models_I A \rhd B$

	y1	y2	y3	y4
x1	0.5	0.5	0.4	0.4
x2	0.6	0.6	1	1
x3	0.1	0.7	1	1
x4	0.9	0.9	0.4	0.4
x5	0.7	0.7	0.2	0.2

Table 5.2: Fuzzy Relation $\alpha : X \rightharpoondown Y$

Next, we would like to observe the equivalent condition of fuzzy functional and implication dependency. We review and extend the equivalent condition in the case of boolean[IHK08] to the general or fuzzy relation case. Let $\mathcal{T}$ be a fuzzy context on Y. Define another fuzzy context $\overline{\mathcal{T}}$ on Y by $\overline{\mathcal{T}} = \{(S \Rightarrow T) \sqcap (T \Rightarrow S) | S, T \in \mathcal{T}\}$.

Theorem 5.6.1. *$\mathcal{T} \models_F A \rhd B$ if and only if $\overline{\mathcal{T}} \models_G A \rhd B$.*

Proof. First we note that $S \sqcap A = T \sqcap A$ is equivalent with $(S \sqcap A \sqsubseteq T)$ and $(T \sqcap A \sqsubseteq S)$. By Lemma 2.1.1, we get $S \sqcap A = T \sqcap A$ is equivalent to $A \sqsubseteq (S \Rightarrow T) \sqcap (T \Rightarrow S)$.
For all $V \in \overline{\mathcal{T}}$ there exists a pair of fuzzy relations $S, T \in \mathcal{T}$ such that $V = (S \Rightarrow T) \sqcap (T \Rightarrow S)$. So we have $A \sqsubseteq V$ is equivalent with $S \sqcap A = T \sqcap A$.
($\rightarrow$) Since $\mathcal{T} \models_F A \rhd B$, if $S \sqcap A = T \sqcap A$ then $S \sqcap B = T \sqcap B$. Then if $A \sqsubseteq V$ then $B \sqsubseteq V$ which means $\overline{\mathcal{T}} \models_G A \rhd B$.
($\leftarrow$) Since $\overline{\mathcal{T}} \models_G A \rhd B$, we get if $A \sqsubseteq V$ then $B \sqsubseteq V$. Then if $S \sqcap A = T \sqcap A$ then $S \sqcap B = T \sqcap B$ which means $\mathcal{T} \models_F A \rhd B$. □

Theorem 5.6.2. *(a) If $\nabla_{IY} \in \mathcal{T}$, then $\mathcal{T} \models_F A \rhd B$ implies $\mathcal{T} \models_G A \rhd B$.*

(b) If $\mathcal{T} \sqsubseteq \overline{\mathcal{T}} \sqsubseteq \mathcal{T}^*$, then $\mathcal{T} \models_G A \rhd B$ if and only if $\mathcal{T} \models_F A \rhd B$.

Proof. (a) Assume $\nabla_{IY} \in \mathcal{T}$, then $\mathcal{T} \sqsubseteq \overline{\mathcal{T}}$, because $(S \Rightarrow \nabla_{IY}) \sqcap (\nabla_{IY} \Rightarrow S) = S$ for $S \in \mathcal{T}$. Since $\mathcal{T} \models_F A \rhd B$, we have $\overline{\mathcal{T}} \models_G A \rhd B$ by Theorem 5.6.1.

(b) ($\leftarrow$) Since $\mathcal{T} \sqsubseteq \overline{\mathcal{T}}$ then we proves if $\overline{\mathcal{T}} \models_G A \rhd B$ then $\mathcal{T} \models_G A \rhd B$.
($\rightarrow$) From Proposition 5.4.1 we have $\mathcal{T}^* \models_G A \rhd B$. Since $\overline{\mathcal{T}} \sqsubseteq \mathcal{T}^*$ we have $\overline{\mathcal{T}} \models_G A \rhd B$. So we proved that $\overline{\mathcal{T}} \models_G A \rhd B$. By Proposition 5.6.1 we know that $\overline{\mathcal{T}} \models_G A \rhd B$ is equivalent to $\mathcal{T} \models_F A \rhd B$. Then we proved $\mathcal{T} \models_G A \rhd B$ if and only if $\mathcal{T} \models_F A \rhd B$.

□

Chapter 6

Application: Fuzzy Process on Truck Backer-Upper

We apply our formulation to an example of the truck backer-upper problem using fuzzy logic introduced by Freeman[Fre94]. Every fuzzy states, procedures are describe as database tables of the fuzzy relational database theory. Problem solving procedures are also described by static formulas of the relational calculus on the category of fuzzy relations. Since every properties are described statically, the consistency of data can be proved formally. We also implemented operations in the theory of fuzzy relational database using Mathematica. Using our Mathematica library, we show a demonstration of the truck backer-upper problem.

In this chapter, we try to implement fuzzy database table and database operations. We show the advantage of fuzzy database table to solve control problem. In our framework, we show fuzzy membership functions using fuzzy database table and define formalization of the defuzzification using fuzzy database operation. The difficulty of our works is how to define continue function of membership value. The interesting of our implementation is our formalizations of fuzzy database table and operations useful to implement in control problem and show the moving clearly.

One of simple control problem is a truck backer-upper. We will work on here is a simple version taken from the work of [Kos92]. The object of the control system is to back up the truck so that it arrives perpendicular to the target position (x_T, y_T). We consider the target position is (60,100). The point (x, y) is the center of the rear of the truck, ϕ is the angle of the truck axis to the horizontal, and δ is the steering angle measured from the truck axis. The controller takes as input of the position of the truck, specified by the pair (x, ϕ), and outputs the steering angle δ.

[Fre94] introduced several fuzzy membership functions for positions, truck angle, and steer angles. In our framework work, we define those membership using our

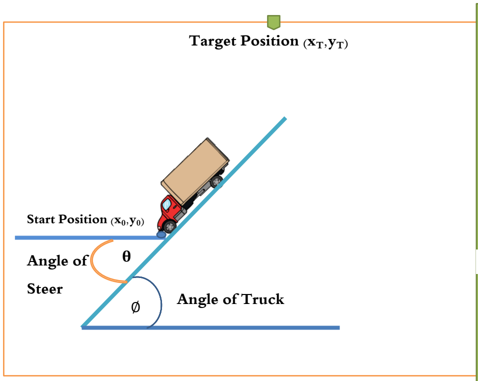


Figure 6.1: Model of Truck

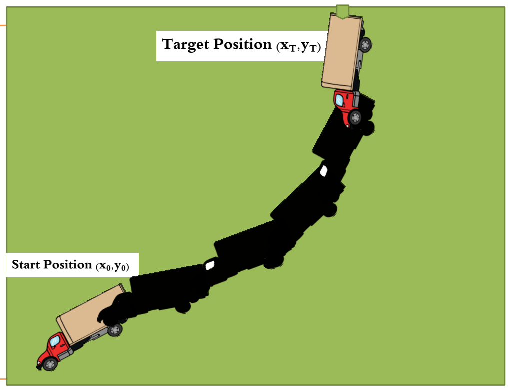


Figure 6.2: Moving Problem

fuzzy relational database model. So, we have to design fuzzy relational database for parking problem of truck backer-upper from every start position (x, y, ϕ) to reach target position(x=60, y=100, $\phi = 90$). Using our formalization we try to show automatic moving truck as we show in Figure 6.2.

6.1 Formulation Using Our Fuzzy Relational Model

In fuzzy truck backer-upper problem, there are input and output elements. The input element are X-axis truck position('XPosition'), and angle of truck('APosition'). Based on information('XPosition' and 'APosition') of truck and target position, we can decide fuzzy rules to compute angle of steer so the truck will move to get target position. For example, if XPosition=50(in the near "Center" from target position), and APosition=100(near "Vertical") so the angle of steer should be small negative or near 0^o. So, we have to define fuzzy relational database table for a classification of XPosition, APosition, SPosition and fuzzy rules.

Let $PT = \{XPosition, PItem\}$. 'PItem' is classification of range X-axis position, we consider D_{PItem} = {Left('LE'), Center('CE'), Right('RI')}, $D_{XPosition} = [0, 100]$, and $r_{PT} : D_{PT} \rightharpoondown D_{PT}$ be a database relation defined by Table 6.1.

Let $AT = \{APosition, AItem\}$. 'AItem' is classification of range angle of truck, we consider:

D_{AItem} ={Right Below('RB'), Right Upper('RU'), Vertical('VE'), Left Upper('LU'), Left Below('LB')}

$D_{APosition}$ =$[-180, 180]$, and $r_{AT} : D_{AT} \rightharpoondown D_{AT}$ is defined by Table 6.2.

r_{AT} is a database relation with attribute {APosition, AItem}.

Table 6.1: Table of classification X Position (r_{PT})

r_{PT}	XPosition	PItem	Fuzzy Value
	$0 \le x \le 30$	LE	1
	$30 < x \le 60$	LE	-1/30 x + 2
	$40 \le x \le 60$	CE	1/20 x - 2
	$60 < x \le 80$	CE	-1/20 x+4
	$60 \le x \le 80$	RI	1/20 x-3
	$80 < x \le 100$	RI	1

Table 6.2: Table of Classification Angle(ϕ) Position(r_{AT})

r_{AT}	APosition	AItem	Fuzzy Value
	$-100 \le a \le -45$	RB	1/55 a + 100/55
	$-45 < a \le 10$	RB	-1/55 a + 10/55
	$-10 \le a \le 35$	RU	1/45 a - 10/45
	$35 < a \le 90$	RU	-1/55 a+90/55
	$60 \le a \le 90$	VE	1/30 a-2
	$90 < a \le 120$	VE	/30 a+4
	$100 \le a \le 155$	LU	1/55 a-100/55
	$155 < a \le 190$	LU	-1/55 a+190/55
	$170 \le a \le 225$	LB	1/55 a-170/55
	$225 < a \le 280$	LB	-1/55 a+280/55

Let $ST = \{SPosition, SItem\}$. 'SItem' is classification of range angle of steer, we consider D_{SItem} = {Negative('NE'), Zero('ZE'), Positive('PO')}, $D_{SPosition} = [-30, 30]$, and $r_{ST} : D_{ST} \rightharpoondown D_{ST}$ be a database relation defined by Table 6.3.

Table 6.3: Table of Classification angle Steer(r_{ST})

r_{ST}	SPosition	SItem	Fuzzy Value
	$-30 < s \le -15$	NE	1/15 s + 2
	$-15 < s \le 0$	NE	-1/15 s
	$-5 < s \le 0$	ZE	1/5 s + 1
	$0 < s \le 5$	ZE	-1/5 s + 1
	$0 < s \le 15$	PO	1/5 s
	$15 < s \le 30$	PO	1

Example 6.1.1. *In Table 6.1, the first row* $(0 \le x \le 30,\ LE,\ 1)$ *means* $r_{PT}(((x, LE), (x, LE))) = 1$ *for* $0 \le x \le 30$.

And the second row $(30 < x \le 60, LE, -1/30x+2)$ *means* $r_{PT}(((x, LE), (x, LE))) = -x/30 + 2$ *for* $30 < x \le 60$.

6.2 Fuzzy Rules

A fuzzy rules table is a relation between some attributes input table(PItem and AItem) with attribute output table(classification of steer angle 'SItem'). Using a fuzzy rules table, we also can design a defuzzification to compute angle of steer.

Let $FR = \{AItem, PItem, SItem\}$. Consider $r_{FR} : D_{FR} \rightharpoondown D_{FR}$ be a database relation of fuzzy rules defined by Table 6.4.

Table 6.4: Table of Fuzzy Rule (r_{FR})

r_{FR} AItem	PItem	SItem	Fuzzy Value
RB	LE	PO	1
RB	CE	PO	1
RB	RI	PO	1
RU	LE	NE	1
RU	CE	PO	1
RU	RI	PO	1
VE	LE	NE	1
VE	CE	ZE	1
VE	RI	PO	1
LU	LE	NE	1
LU	CE	NE	1
LU	RI	PO	1
LB	LE	NE	1
LB	CE	NE	1
LB	RI	NE	1

6.3 Fuzzy Algorithm

Different from the Freeman method, we use database operations (Selection, Projection, and Natural Join) to construct a fuzzy system.

Let $A = PT \sqcup AT \sqcup ST$, for $a \in A$ and $t_0 \in D_A$, we define a database relation $Eq(t_0) : D_a \rightharpoondown D_a$ by

$$Eq(t_0)(t,t) = \begin{cases} 0 & (t \neq t_0), \\ 1 & (t = t_0). \end{cases}$$

Let $A = PT \sqcup AT \sqcup FR$, $x \in D_{XPosition}$ and $\phi \in D_{APosition}$. A database relation $f : D_X \rightharpoondown D_X$ for the selection σ_f is defined by $f = \eta_{XPosition,X}(Eq(x)) \cdot \eta_{APosition,X}(Eq(\phi))$. The **Output** operations r_{out} can be defined by the equation using database operations as follows:

$$r_{out}(x, \phi) = \pi_{A,SItem}(\sigma_f(r_{PT} \bowtie r_{AT} \bowtie r_{FR})).$$

Example 6.3.1. *Let the start position of truck at* $(x, y) = (20, 0)$ *and the start angle of truck* $\phi = -90$*, The selection* $\sigma_{XPosition=20,\ APosition=-90}$ *of natural join* $(r_{PT} \bowtie r_{AT} \bowtie r_{FR})$ *is showed in Table 6.5. The output* $r_{Out}(20, -90)$ *is showed in Table 6.6.*

Table 6.5: $\sigma_{XPosition=20,\ APosition=-90}(r_{PT} \bowtie r_{AT} \bowtie r_{FR})$

XPosition	PITem	APosition	AItem	SItem	Fuzzy Value
20	LE	-90	RB	PO	2/11

Table 6.6: $r_{Out}(20, -90)$

SItem	Fuzzy Value
PO	2/11

Then, we got the result of $r_{out}(20, -90)(PO, PO) = 2/11$*. It means the steer position will be in 'Positive' with the fuzzy value* $2/11$*.*

The **defuzzification** process is a process to compute a real value of steer angle using a classification in the output $r_{out}(x, \phi)$. In this case, we use *Mamdani-Centroid* for the defuzzification process.

Let $D_a \sqsubseteq R$, where R is a set of all real numbers. For a database relation $f : D_a \rightharpoondown D_a$, we define the fuzzy sum $S(f)$ and the centroid $E(f)$ as follows:

$$\begin{aligned} S_a(f) &= \sum\{f(t,t) \in [0,1] \mid t \in D_a\}, \\ E_a(f) &= \sum\{t \cdot f(t,t) \in [0,1] \mid t \in D_a\}/S(f). \end{aligned}$$

A database relation r_c is defined by

$$r_c = \pi_{ST,SPosition}(r_{ST} \bowtie r_{out}(x, \phi)) : D_{ST} \rightharpoondown D_{ST}.$$

We get the **defuzzification** δ_c **of** r_c by $\delta_c = E_a(r_c)$.

SimulateTruck (x, y, ϕ);
Input : Start position of truck (x, y, ϕ)
Output : Set of update position (x_t, y_t, ϕ_t)
Variable: r_c, δ_c
while *not TargetPosition*(x_t, y_t) **do**
$r_{out}(x_t, \phi_t) = \pi_{X,SItem}(\sigma_{X\ Position=x \wedge APosition=\phi_t}(r_{PT} \bowtie r_{AT} \bowtie r_{FR})).$;
$r_c = \pi_{ST,SPosition}(r_{ST} \bowtie r_{out}(x, \phi_t)).$;
$\delta_c = E_{SPosition}(r_c)$;
$x_{t+1} = x_t + 5cos(\delta_c)$,
$y_{t+1} = y_t + 5sin(\delta_c)$,
$\phi_{t+1} = \phi_t + \delta_c.$;
$t = t + 1$;
end
return collection set $\{(x_0, y_0, \phi_0), (x_1, y_1, \phi_1), \ldots, (x_n, y_n, \phi_n)\}$;

Figure 6.3: Algorithm Moving of Truck Backer-Upper

6.4 Result

In this section, we would like to explain more detail the example of fuzzy truck backer-upper. Using algorithm in Figure 6.3, we follow the algorithm and fuzzy database tables to make the movement of truck backer-upper automatically.

The followings are equations of moving truck[Fre94]:

$$\begin{aligned} x_{t+1} &= x_t + 5cos(\delta_c), \\ y_{t+1} &= y_t + 5sin(\delta_c), \text{ and} \\ \phi_{t+1} &= \phi_t + \delta_c. \end{aligned}$$

Controlling the angle of the steer δ_c, the coordinate (x, y) and angle of truck ϕ will be updated, the truck moves to the target position $(60, 100)$.

Example 6.4.1. *We consider the start position* $(x, y) = (20, 0)$ *with the angle of truck* ϕ*=-90. We got the output* $r_{Out}(20, -90) = \{(PO, PO), (PO, PO), 2/11\}$ *as shown in Example 6.3.1. Next, we continue to get a real value of the steer angle using the defuzzification process. In the defuzzification process, there are some steps to compute an angle of steer* δ_c*, we use Mathematica 10.0.1 to compute the defuzzification process and the algorithm is shown in Appendix. In the Appendix, we use function '*`defuzzy`*' to compute the angle of steer* δ_c*. In the Appendix, the result of* `defuzzy[ST, rST, rOut[20,-90]]` *is 15.8016, it means angle of steer* δ_c *is 15.8016 when the position* $x = 20$ *and* $\phi = -90$*(*$r_{Out}(20, -90)$*). Then, the position of the truck move from* $(x_0, y_0, \phi_0) = (20, 0, -90)$ *to the new position* $(x_1, y_1, \phi_1) = (20 + 5.cos\ 15.8016,\ 0 + 5.sin\ 15.8016,\ -90 + 15.8016)$*. The new position causes updating the new*

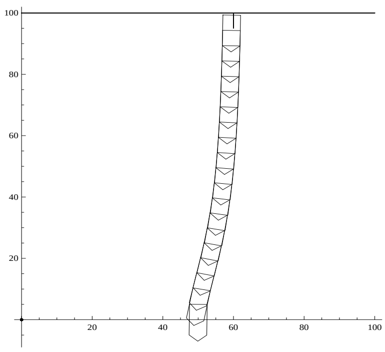


Figure 6.4: Moving result with start pos.
$(x_0 = 50, y_0 = 0, \phi_0 = 100)$

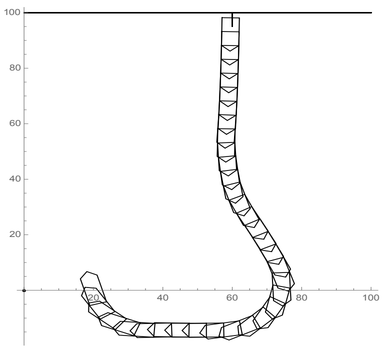


Figure 6.5: Moving result with start pos.
$(x_0 = 20, y_0 = 0, \phi_0 = -90)$

output $r_{out}(x_1, \phi_1)$ *then we get the new* δ_c*. This algorithm process of the truck backer-upper will stop if the truck reach the target position* $(x_t, y_t, \phi_t) = (60, 100, 90)$.

The result of the truck with the start position $(x, y) = (20, 0)$ *and the angle of truck* ϕ*=-90 showed in the Figure 6.5. We showed the result of the truck with the start position* $(x, y) = (50, 0)$ *and the angle of truck* ϕ*=100 in Figure 6.4. The results showed an application of our algorithm using fuzzy relational database table to a fuzzy control problem.*

Chapter 7

Conclusion and Future Work

Fuzzy relational database is a combination of relational database and fuzzy theory. We have introduced the notions of fuzzy database relational model using relational calculus. Since we used Dedekind category, we can use our notions in general case of database relational model. We also continued to introduce the notions of database operations("selection", "projection", "injection", and "natural join"). We showed some proposition of database operations and we can verify it using relational calculus. It shows that relational calculus can be used to analyze by fuzzy relational database model using symbolic computation.

After we define the query language for the fuzzy relational database with "single" value for every tuple attribute. In Chapter 4, we extended the model of fuzzy equivalence database. We introduced the notions for some properties of fuzzy equivalence relational model. We have defined simple(non-redundant) table using equivalence class to transform from classical database to fuzzy equivalence relation database model. We also extended the database operation for fuzzy equivalence relational database model. We showed that fuzzy equivalence relational database can be used for simplifying fuzzy relational database model. This method also can be used for clustering dataset. Further, we can use our notions to define dependency for fuzzy equivalence relational database model.

In chapter 5, we used different case from the previous chapter. We tried to analyze fuzzy formal context table. We defined fuzzy indiscernibility relation to construct fuzzy functional dependency. We also showed equivalent condition between fuzzy functional dependency and fuzzy implication in the formal context table. We showed also some example to explain the equivalent condition.

In Chapter 6, we have introduced a new method using the relational database to solve a simple control problem. In Freeman's method, He introduced several fuzzy membership functions for positions, truck angles, and steer angles. They also defined fuzzy rule tables and introduced several computations for each operation. In our framework, each position, truck angles, steer angles and a fuzzy rule table are both represented as tables of the fuzzy relational database. Further, we do not use problem specific computations. The decision procedure is defined using relational database operations, such as projection, selection, and natural join. Using relational calculus, we can extend notions of the traditional relational database to fuzzy relational database. Relational calculus is used to make simple institutional notions into the fuzzy database tables. We implemented our notions in relational calculus in Mathematica Software. We showed the movement of the truck backer-upper. It means that our formalizations in relational calculus can be applied.

Bibliography

[AM16] Mohammad Deni Akbar and Yoshihiro Mizoguchi. A formalization of a fuzzy relational database model using relational calculus. *Bulletin of Informatics and Cybernetics*, 48(6):83–104, 2016.

[Arm74] William Ward Armstrong. Dependency structures of data base relationships. In *IFIP Congress*, pages 580–583, 1974.

[BFH77] Catriel Beeri, Ronald Fagin, and John H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *Proceedings of the 1977 ACM SIGMOD International Conference on Management of Data*, SIGMOD '77, pages 47–61, New York, NY, USA, 1977. ACM.

[BP82] Billy P. Buckles and Frederick E. Petry. A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7(3):213 – 226, 1982.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, jun 1970.

[DG98] Ivo Duntsch and Gunther Gediga. Uncertainty measures of rough set prediction. *Artificial Intelligence*, 106(1):109 – 137, 1998.

[FK15] Hitoshi Furusawa and Yasuo Kawahara. Point axioms and related conditions in dedekind categories. *Journal of Logical and Algebraic Methods in Programming*, 84(3):359 – 376, 2015. 13th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS 2012).

[FKW11] Hitoshi Furusawa, Yasuo Kawahara, and Michael Winter. Dedekind categories with cutoff operators. *Fuzzy Sets and Systems*, 173(1):1 – 24, 2011.

[Fre94] James A. Freeman. Fuzzy system for control applications : The truck backer upper. *The Mathematica Journal*, 4:64 – 69, 1994.

[GW97] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.

[IHK08] Toshikazu Ishida, Kazumasa Honda, and Yasuo Kawahara. Implication and functional dependency in intensional contexts. *Bulletin of informatics and cybernetics*, 40:101–111, dec 2008.

[Kaw88] Yasuo Kawahara. Applications of relational calculus to computer mathematics. *Bulletin of Informatics and Cybernetics*, 23(1-2):67–78, 1988.

[Kaw95] Yasuo Kawahara. Relational set theory. In *Category Theory and Computer Science, 6th International Conference, CTCS '95, Cambridge, UK, August 7-11, 1995, Proceedings*, pages 44–58, 1995.

[KF99] Yasuo Kawahara and Hitoshi Furusawa. An algebraic formalization fuzzy relations. *Fuzzy Sets and Systems*, 101:125–135, 1999.

[Kis91] A. Kiss. An application of fuzzy graphs in database theory. *Pure Mathematics and Applications*, 1(3-4):337–342, 1991.

[KM93] Yasuo Kawahara and Yoshihiro Mizoguchi. Relational structures and their partial morphisms in view of single pushout rewriting. In *Dagstuhl Seminar on Graph Transformations in Computer Science*, 1993.

[KMK13] Dimitris K. Kardaras, Xenia J. Mamakou, and Bill Karakostas. *Fuzzy Equivalence Relation Based Clustering and Its Use to Restructuring Websites' Hyperlinks and Web Pages*, pages 52–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Kos92] Bart Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Inc., 1992.

[MK95a] Yoshihiro Mizoguchi and Yasuo Kawahara. Relational graph rewritings. *Theoritical Computer Science*, 141:311–328, 1995.

[MK95b] Masao Mori and Yasuo Kawahara. Fuzzy graph rewritings. *RIMS KOKYUROKU*, 918:65–71, 1995.

[MN01] J. N. Mordeson and P. S. Nair. *Fuzzy Mathematics*. Studies in Fuzziness and Soft Computing. Physica-Verlag, 2001.

[MW07] Andreas Meier and Nicolas Werro. A fuzzy classification model for online customers. *Informatica (Slovenia)*, 31(2):175–182, 2007.

[Nak00] Michinori Nakata. *Formulation of Division Operators in Fuzzy Relational Databases*, pages 144–156. Physica-Verlag HD, Heidelberg, 2000.

[OK00] H. Okuma and Y. Kawahara. Relational aspects of relational database dependencies. *Bulletin of Informatics and Cybernetics*, 16:91–104, 2000.

[RM88] K. V. S. V. N. Raju and Arun K. Majumdar. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Syst.*, 13(2):129–166, June 1988.

[Sch10] Gunther Schmidt. *Relational Mathematics*. Cambridge University Press, New York, NY, USA, 2010.

[SMF90] S. Shenoi, A. Melton, and L. T. Fan. An equivalence classes model of fuzzy relational databases. *Fuzzy Sets Syst.*, 38(2):153–170, November 1990.

[Tar41] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, September 1941.

[UF94] Motohide Umano and Satoru Fukami. Fuzzy relational algebra for possibility-distribution-fuzzy-relational model of fuzzy data. *J. Intell. Inf. Syst.*, 3(1):7–27, February 1994.

[Wil82] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470. Reidel, 1982.

[WJ16] Michael Winter and Ethan C. Jackson. Categories of relations for variable-basis fuzziness. *Fuzzy Sets Syst.*, 298(C):222–237, September 2016.

[Zad65] Lofti A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zad71] Lotfi. A. Zadeh. Similarity relations and fuzzy orderings. *Inf. Sci.*, 3(2):177–200, April 1971.

List of Figures

List of Tables