Formal Concept Analysis and Hierarchical Classes Analysis

Yaohua Chen, Yiyu Yao

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {chen115y, yyao}@cs.uregina.ca

***Abstract*— Formal concept analysis (FCA) and hierarchical classes analysis (HCA) can be viewed as two approaches of data analysis on data description and summarization, and focus on different characteristics of data in a formal context. In this paper, we propose a framework for classes analysis. The equivalence classes defined on object and attribute universes can be viewed as basic units of information in a formal context. The different data structures in FCA and HCA can be developed based on the equivalence classes. Furthermore, some basic sets of objects and attributes formed by equivalence classes can be used to compose other object and attribute sets, and they demonstrate connections between FCA and HCA. Through our studies, one can obtain an in-depth understanding of data analysis based on FCA and HCA.**

I. Introduction

Formal concept analysis (FCA) and hierarchical classes analysis (HCA) are regarded as two approaches or frameworks to describe, characterize and summarize data for data analysis. Each of them provides a type of visual representation of information and knowledge derived from a simple binary information table, which is called a formal context. They focus on presentation of different characteristics of data by using set-theoretical methods.

FCA provides a formal and graphical way to organize data as formal concepts [12]. In a formal context, a pair of a set of objects and a set of attributes that uniquely associate with each other is called a formal concept. The set of objects is referred to as the extension, and the set of attributes as the intension, of a formal concepts. The family of all formal concepts is a complete lattice. The hierarchy of concepts in a concept lattice can be defined by an order relation on extensions or intensions of concepts. Many researches have been proposed on this area [14], [15], [16].

HCA is another approach on description and summarization of data [2], [3], [4], [6]. In a formal context, equivalence classes, which are viewed as basic information units, can be defined based on an equivalence relation [13], [17]. The graphical hierarchies of the classes are defined and constructed by an order relation on the associated object or attribute sets.

Zhang *et al.* studied a method of attribute reduction in concept lattice. Their studies can be regarded as a related work to data analysis. Some attributes are considered as necessary attributes to form concepts. By removing unnecessary attributes, attribute reduction in concept lattice can be made for discovery of implicit knowledge easier and the representation simpler.

In this paper, we propose a framework for equivalence classes analysis. Some specific sets of objects and attributes can be formed based on equivalence classes. They can be used to compose other object and attribute sets, even the universe of objects and the universe of attributes. Thus, They are considered as basic sets. Some important properties of these basic sets are able to describe the equivalence relation and reflect the order relation among the classes. Furthermore, our studies explore that these basic sets of objects and attributes can be a bridge to connect FCA and HCA. That is, these basic sets not only can reflect the notion of hierarchies of classes, but also can be used to form concepts. In fact, they present some very important characteristics of data in a formal context.

The rest of this paper is organized as following. The next section is an overview of the simple classification in formal contexts. Then, we will discuss the associations between objects and attributes in Section III. In Section IV and V, we review the basic notions of formal concept analysis and hierarchical class analysis. In Section VI, we will show connections between HCA and FCA by considering some basic sets of classes. The conclusion will finally be given in Section VII.

II. Classification based on Formal Contexts

An information table provides a simple and powerful tool for data analysis. All available information and knowledge can be provided and derived by an information table [13]. Generally, an information table is constituted by a finite nonempty set of objects, a finite nonempty set of attributes, and the values of objects on attributes. A binary information table is a sample information table in which each entry contains binary value 0 or 1. Table I is an example of binary information table. Every information table can be represented by a binary information table [1].

Let U be a finite nonempty set representing the universe of objects. Let V be another finite nonempty set representing the universe of attributes. A binary relation R between the two universes U and V is a subset of the Cartesian product $U \times V$. For an object $x \in U$ and an attribute $y \in V$, if $(x, y) \in R$, written as xRy, we say that an object x possesses the attribute y. Therefore, based on the binary relation R, the objects and attributes describe and associate with each other.

TABLE I

A BINARY INFORMATION TABLE TAKEN FROM [4]

	a	b	c	d	e	f	g	h	i	j	k	l
1	1	1	0	0	1	1	0	0	0	0	0	1
2	1	1	0	0	1	1	0	0	0	0	0	1
3	1	1	0	0	1	1	0	0	0	0	0	1
4	0	0	1	1	1	1	0	0	0	1	1	1
5	0	0	1	1	1	1	0	0	0	1	1	1
6	0	0	0	0	0	0	1	1	1	1	1	1
7	0	0	0	0	0	0	1	1	1	1	1	1
8	0	0	1	1	1	1	1	1	1	1	1	1
9	0	0	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1

The triplet $S = (U, V, R)$, where U is the universe of objects, V is the universe of attributes and R is a binary relation between two universes, represents a binary information table and is called a formal context [7], [12].

In a formal context, given an object x, the set of attributes that are possessed by an object x is denoted as xR and defined by:

$$xR = \{y \in V \mid xRy\} \subseteq V.$$

For an attribute y, the set of objects that possess y is denoted as Ry:

$$Ry = \{x \in U \mid xRy\} \subseteq U.$$

By the notation xR, objects may be viewed as equivalent if they have the same set of attributes. Thus, an equivalence relation between objects can be formally defined. Let $E_U \subseteq U \times U$ denote an equivalence relation on U.

Definition 1: Two objects x and x' in the universe U are equivalent if $xR = x'R$, that is,

$$x\ E_U\ x' \Longleftrightarrow xR = x'R.$$

The equivalent objects can be grouped together into a same set, which is called an equivalence class and denoted as $[x]$ [10].

Definition 2: An equivalence class of objects including the object x is defined by:

$$\begin{aligned} [x] &= \{x' \in U \mid x'\ E_U\ x\}, \\ &= \{x' \in U \mid xR = x'R\}. \end{aligned}$$

Another view to look at the equivalence classes is that the equivalence relation E_U partitions the universe of objects U into disjoint subsets, which are the equivalence classes of objects [9], [10], [11].

Within an equivalence class of objects, each element is regarded as indiscernible. Each equivalence class is considered as a whole instead of individuals inside [11], [13]. Thus, an equivalence class may be viewed as a basic unit of information or knowledge.

Furthermore, the partition of the universe U is the family of all equivalence classes of objects, and commonly known as the quotient set U/E_U on U induced by E_U. The universe U is in fact the union of existing equivalence classes of objects.

TABLE II

THE REDUCTION OF TABLE I

	[a]	[c]	[e]	[g]	[j]	[l]
[1]	1	0	1	0	0	1
[4]	0	1	1	0	1	1
[6]	0	0	0	1	1	1
[8]	0	1	1	1	1	1
[10]	1	1	1	1	1	1

Similarly, attributes may also be regarded as equivalent if they are possessed by the same set of objects. The equivalence relation E_V on V can be defined by:

$$y\ E_V\ y' \Longleftrightarrow Ry = Ry'.$$

And an equivalence class of attributes is defined by:

$$\begin{aligned} [y] &= \{y' \in V \mid y'\ E_V\ y\}, \\ &= \{y' \in V \mid Ry = Ry'\}. \end{aligned}$$

The partition of the universe V is the family of all equivalence classes of attributes and the quotient set V/E_V on V induced by E_V. The universe V is the union of existing equivalence classes of attributes.

In this paper, for simplicity, we call the equivalence classes of objects as object classes, and the equivalence classes of attributes as attribute classes.

Since object and attribute classes are viewed as basic units of information, they can be used to reconstruct the formal context instead of individual objects and attributes. Therefore, a reduction of a formal context can be done by using the classes. The reduction of Table I is illustrated in Table II. In a reduced information table, the object and attribute classes associate with each other.

Object and attribute classes can also be defined by a particular type of mappings between U and V. That is, $^b : 2^U \longrightarrow 2^V$ and $^b : 2^V \longrightarrow 2^U$.

Definition 3: For a set of objects $X \subseteq U$ and a set of attributes $Y \subseteq V$, we have,

$$X^b = \{y \in V \mid Ry = X\},$$
$$Y^b = \{x \in U \mid xR = Y\}.$$

For simplicity, the same symbol is used for both mappings. The mapping ^{b} is called basic set assignment [15], which associate an object class with a set of attributes and an attribute class with a set of objects. In fact, the family $\{X^b \neq \emptyset \mid X \subseteq U\}$ is the family of attribute classes, and the family $\{Y^b \neq \emptyset \mid Y \subseteq V\}$ is the family of object classes. Let AT denote the family of all sets of attributes that each associates with an object class, that is, $AT = \{Y \mid Y \subseteq V, Y^b \neq \emptyset\}$. Let OB denote the family of all sets of objects that each associates with an attribute class, that is, $OB = \{X \mid X \subseteq U, X^b \neq \emptyset\}$.

Example 1: In Table II, the object classes and their associated sets of attributes are:

$$\begin{array}{ll} Y_1^b = [1], & Y_1 = \{[a],[e],[l]\}, \\ Y_2^b = [4], & Y_2 = \{[c],[e],[j],[l]\}, \\ Y_3^b = [6], & Y_3 = \{[g],[j],[l]\}, \\ Y_4^b = [8], & Y_4 = \{[c],[e],[g],[j],[l]\}, \\ Y_5^b = [10], & Y_5 = \{[a],[c],[e],[g],[j],[l]\}. \end{array}$$

The attribute classes and their associated sets of objects are:

$$\begin{array}{ll} X_1^b = [a], & X_1 = \{[1],[10]\}, \\ X_2^b = [c], & X_2 = \{[4],[8],[10]\}, \\ X_3^b = [e], & X_3 = \{[1],[4],[8],[10]\}, \\ X_4^b = [g], & X_4 = \{[6],[8],[10]\}, \\ X_5^b = [j], & X_5 = \{[4],[6],[8],[10]\}, \\ X_6^b = [l], & X_6 = \{[1],[4],[6],[8],[10]\}. \end{array}$$

A class may be considered as a basic definable unit in the universe [11], [16]. In other words, under an equivalence relation, a class is the smallest non-empty observable, measurable, or definable subset of 2^U. A union of some classes is called a composed class [10], [11]. By extending the definability of equivalence classes, we assume that a union of some classes is also definable. By adding the empty set $\emptyset$, a new family of all composed classes $\sigma(U/E_U)$, which is a subsystem of 2^U, can be constructed. $\sigma(U/E_U)$ is closed under set complement, intersection, and union. Obviously, a subsystem $\sigma(V/E_V)$ of 2^V can be developed based on the universe V and the equivalence relation E_V. Rough set approximations can be defined based on the subsystem $\sigma(U/E_U)$ [11], [16].

Based on the subsystems on U and V, the set of attributes xR for an object class $[x]$ is the union of some attribute classes, and the set of objects Ry for an attribute class $[y]$ is the union of some object classes.

III. Associations between Objects and Attributes

By extending the notations xR and Ry, we can establish relationships between subsets of objects and subsets of attributes [7], [12]. This leads to two operators, one from 2^U to 2^V and the other from 2^V to 2^U.

Definition 4: Suppose (U, V, R) is a formal context. For a set of objects, we associate it with a set of attributes:

$$\begin{aligned} X^* &= \{y \in V \mid \forall x \in U (x \in X \Longrightarrow xRy)\} \\ &= \{y \in V \mid X \subseteq Ry\} \\ &= \bigcap_{x \in X} xR. \end{aligned} \tag{1}$$

For a set of attributes, we associate it with a set of objects:

$$\begin{aligned} Y^* &= \{x \in U \mid \forall y \in V (y \in Y \Longrightarrow xRy)\} \\ &= \{x \in U \mid Y \subseteq xR\} \\ &= \bigcap_{y \in Y} Ry. \end{aligned} \tag{2}$$

For simplicity, the same symbol is used for both operators. The actual role of the operators can be easily seen from the context.

Düntsch and Gediga referred to * as a modal-style operator, called the sufficiency operator [5], [8]. In fact, it reflects a unique association between an object set and an attribute set. Moreover, X^* for an object set X is the maximal set of attributes possessed by all objects in the set X. That is, the set X^* consists of necessary attributes of an object in X. In other words, an object in X must have attributes in X^*. Similarly, Y^* for an attribute set Y is the maximal set of objects that have all attributes in Y. That is, an attribute in Y must have objects in Y^*, and Y^* consists of necessary objects of an attribute in Y. Furthermore, the sufficiency operator * can also be defined on the quotient sets U/E_U and V/E_V.

Consequently, $[x]^* = xR$ is the set of attributes possessed by elements in the object class $[x]$. $[y]^* = Ry$ is the set of objects having the attributes in the attribute class $[y]$.

The further studies of sufficiency operator and its relationships with other modal-style operators in a formal context are provided by Yao [15].

The sufficiency operator holds the following properties: For $X, X_1, X_2 \subseteq U$ and $Y, Y_1, Y_2 \subseteq V$,

$$\begin{array}{ll} (1) & X_1 \subseteq X_2 \Longrightarrow X_1^* \supseteq X_2^*, \\ & Y_1 \subseteq Y_2 \Longrightarrow Y_1^* \supseteq Y_2^*, \\ (2) & X \subseteq X^{**}, \\ & Y \subseteq Y^{**}, \\ (3) & X^{***} = X^*, \\ & Y^{***} = Y^*, \\ (4) & (X_1 \cup X_2)^* = X_1^* \cap X_2^*, \\ & (Y_1 \cup Y_2)^* = Y_1^* \cap Y_2^*, \end{array}$$

By Property (2), one can know that an object class $[x]$ is included in an object set $[x]^{**}$, and an attribute class $[y]$ is include in an attribute set $[y]^{**}$.

In terms of the basic set assignment, we can re-express operation * as [15]:

$$\begin{aligned} X^* &= \bigcup \{A^b \mid A \subseteq U, X \subseteq A\}, \\ Y^* &= \bigcup \{B^b \mid B \subseteq V, Y \subseteq B\}. \end{aligned}$$

This formulation presents the connection between basic set assignment ^{b} and the modal-style operator *. Moreover, by combing the two operators, one can easily have that $X^{b*} = X$ and $Y^{b*} = Y$, where $X \subseteq U$, $X^b \neq \emptyset$, $Y \subseteq V$ and $Y^b \neq \emptyset$.

IV. Formal Concept Analysis

Formal concept analysis deals with a visual presentation and analysis of data [7], [12]. It follows the traditional notion of concept, namely, a concept is constituted by an extension and an intension. The extension is referred to as instances of a concept. The intension is referred to as properties or features of a concept.

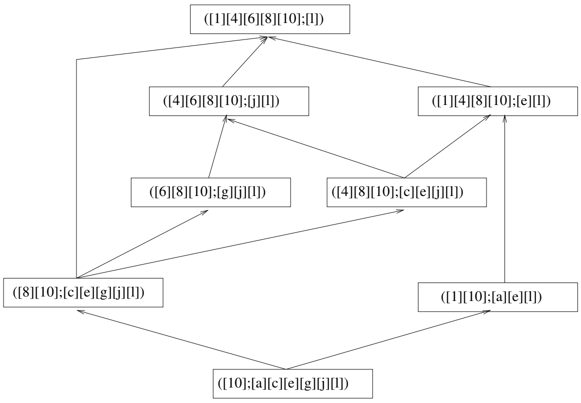

Fig. 1. Formal concept lattice induced from the Table II

A. *Formal Concepts*

FCA focuses on a unique association between the extension represented by a set of objects and the intension represented by a set of attributes. That is, a pair of a set of objects and a set of attributes, which uniquely associate with each other, is a formal concept.

Definition 5: A pair (X, Y), where $X \subseteq U$, $Y \subseteq V$, is called a formal concept in the context (U, V, R) if $X = Y^*$ and $Y = X^*$.

X is referred to as the extension, and Y is referred to as the intension, of the concept. Consequently, a pair (X, X^*), $X \subseteq U$, that satisfies $X = X^{**}$ is a formal concept. Equivalently, a pair (Y^*, Y) satisfying $Y \subseteq V$ and $Y = Y^{**}$ is a formal concept.

The family of all formal concepts induced from a formal context can be constructed as a complete lattice, called concept lattice. The Figure 1 illustrates the concept lattice for the formal context of Table II.

The meet and join of the formal concepts in the lattice are characterized by the following operations [7], [12].

Theorem 1: The formal concept lattice is a complete lattice in which the meet and join are given by:

$$\bigwedge_{t \in T} (X_t, Y_t) = (\bigcap_{t \in T} X_t, (\bigcup_{t \in T} Y_t)^{**}),$$
$$\bigvee_{t \in T} (X_t, Y_t) = ((\bigcup_{t \in T} X_t)^{**}, \bigcap_{t \in T} Y_t),$$

where T is an index set and for every $t \in T$, (X_t, Y_t) is a formal concept.

B. *Hierarchy of Concepts*

The order relation of the concepts can be defined based on the set inclusion relation.

Definition 6: For two formal concepts (X_1, Y_1) and (X_2, Y_2), (X_1, Y_1) is a sub-concept of (X_2, Y_2), written $(X_1, Y_1) \preceq (X_2, Y_2)$, and (X_2, Y_2) is a super-concept of (X_1, Y_1), if and only if $X_1 \subseteq X_2$, or equivalently, if and only if $Y_2 \subseteq Y_1$.

It is easy to know that a more general concept is represented by a larger set of object classes that share a smaller set of attribute classes. Conversely, a more specific concept has a smaller set of object classes and a larger set of attribute classes.

V. Hierarchical Classes Analysis

In a formal context, hierarchical classes analysis is proposed to describe and summarize the data by using the order relation among the equivalence classes [2], [4].

A. *Hierarchies of Classes*

Based on the notation xR, there exists an order relation, denoted as $\sqsubseteq$, between the object classes.

Definition 7: Given two object classes $[x]$ and $[x']$ on U, an order relation $\sqsubseteq$ between them is defined by:

$$[x] \sqsubseteq [x'] \Longleftrightarrow xR \subseteq x'R.$$

$[x]$ is called a sub-class of $[x']$. Conversely, $[x']$ is called a super-class of $[x]$. For simplicity, $[x]$ is $[x']$'s sub-class, and $[x']$ is $[x]$'s super-class. This formulation means that one can order the object classes by using their associated set of attributes. An object class having no sub-class is called object bottom class.

Similarly, we can have the definition of order relation between the attribute classes.

Definition 8: For two attribute classes $[y]$ and $[y']$ on V, the order relation $\sqsubseteq$ between them is defined by:

$$[y] \sqsubseteq [y'] \Longleftrightarrow Ry \subseteq Ry'.$$

$[y]$ is a sub-class of $[y']$ and $[y']$ is a super-class of $[y]$. Or, $[y]$ is $[y']$'s sub-class, and $[y']$ is $[y]$'s super-class. An attribute class having no sub-class is called attribute bottom class.

By analyzing the basic set assignment and the order relation, it is easy to know that if an object class associates with an attribute class, it also associates with all super-classes of the attribute class. Likewise, an attribute class associates with an object class and all its the super-classes [2], [4].

According to the order relation between the classes, a hierarchical structure of the classes for a formal context can be constructed. For example, hierarchies of classes for Table II is illustrated as Figure 2. The hierarchy of attribute classes in the part (b) of Figure 2 is presented upside down in the part (c). The hierarchies of object classes in the part (a) and attribute classes in the part (b) are combined together by linking the bottom classes of two hierarchies with dot lines.

VI. Basic Sets of Classes

The hierarchies of classes may be viewed as a description of data within a formal context. The order relation is a relationship of set inclusion on attribute sets of object classes or object sets of attribute classes.

A set of attributes in AT may be a union of some other sets of attributes in AT. If a set of attributes $Y \in AT$ is

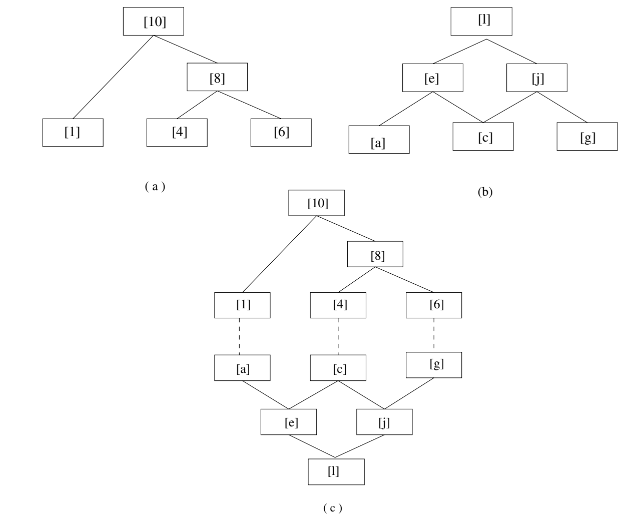

Fig. 2. Hierarchies of classes induced from the Table II

the union of other sets of attributes in AT, then we say that Y is expressed by some other sets of attributes and is an expressible set of attributes, otherwise, it is an inexpressible set of attributes.

Similarly, a set of objects in OB may be a union of some other sets of objects in OB. If a set of objects $X \in OB$ is the union of some other sets of objects in OB, then we say that X is expressed by some other sets of objects and is an expressible set of objects, otherwise, it is an inexpressible set of objects.

According to the definitions of order relation on object and attribute classes, an expressible attribute set $xR \in AT$ must associate with an object super-class $[x]$ because it is the union of the attribute sets $x'R$ of $[x]$'s sub-classes $[x']$. For an object bottom class, its attribute set must be inexpressible because it has no sub-class. In other words, its attribute set is not a union of some other attribute sets in AT. With the same way, an expressible object set $Ry \in OB$ must associate with an attribute super-class $[y]$ because it is the union of the object sets Ry' of the $[y]$'s sub-classes $[y']$. For an attribute bottom class, its object set is inexpressible because it has no sub-class.

An expressible set must be a union of some inexpressible sets. The inexpressible sets can be considered as the basic units for expressing other sets. An inexpressible object set in OB is called an object block. An inexpressible attribute set in AT is called an attribute block. The object and attribute blocks can be formally defined as following.

Definition 9: In a formal context, an object set $X \in OB$ is called an object block if

$$X \neq \bigcup\{A \mid A \in OB, A \neq X, A^b \sqsubseteq X^b\}.$$

The family of all object blocks is denoted by $\mathcal{BO}$.

Definition 10: In a formal context, an attribute set Y in AT is called an attribute block if

$$Y \neq \bigcup\{B \mid B \in AT, B \neq Y, B^b \sqsubseteq Y^b\}.$$

The family of all attribute blocks is denoted as $\mathcal{BA}$.

An object (attribute) expressible set is a union of some object (attribute) blocks. Therefore, one can defined the membership of object (attribute) blocks included in an object (attribute) set.

Definition 11: For an object set $X \in OB$, the set of object blocks included in X is denoted as $M(X)$ and defined by:

$$M(X) = \{A \mid A \in \mathcal{BO}, A \subseteq X\}.$$

Definition 12: For an attribute set $Y \in AT$, the set of attribute blocks included in Y is denoted as $M(X)$ and defined by:

$$M(Y) = \{B \mid B \in \mathcal{BA}, B \subseteq Y\}.$$

By analyzing the object block sets $\mathcal{BO}$, the following properties can be held:

1) $U = \bigcup\{A \mid A \in \mathcal{BO}\}$.
2) For $X \in \mathcal{BO}$, $X^{**} = X$.
3) For $[y]$ and $[y']$, $[y] \sqsubseteq [y'] \Longleftrightarrow M(Ry) \subseteq M(Ry')$.

Property (1) means that the union of all object blocks is the universe U of objects. Since each object set in OB associates with an attribute class. The union of all the attribute classes is the universe V of attributes. Consequently, the union of all the object sets in OB is the universe U of objects. Additionally, the expressible object sets in OB can be composed of object blocks. Therefore, by replacing the expressible object sets with object blocks, the universe of objects U must be the union of all object blocks.

Property (2) means that the pair (X, X^*) is a formal concept. Since the object set $X \in \mathcal{BO}$ associates with an attribute class, that is, $X^b \neq \emptyset$, so we know that $X^{b*} = X$. By considering the Property (3) of the sufficiency operator *, we can have that $X = X^{b*} = X^{b***} = X^{**}$. Thus, based on formal concept analysis, the pair (X, X^*) that satisfies $X = X^{**}$ is a formal concept.

Property (3) means that the order relation between attribute classes can be redefined by using membership of object blocks. An attribute class $[y]$ is a sub-class of an attribute class $[y']$ if the set of object blocks included in Ry is a subset of the set of object blocks included in Ry'. In other words, an attribute super-class must associate with more object blocks than any of its sub-classes does.

Since each block consists of classes, so the elements in a class must be in the same set of blocks. That is, equivalence relation can also be redefined by considering the membership of an object (attribute) in object (attribute) blocks.

Similarly, the attribute block sets $\mathcal{BA}$ have the following properties:

1) $V = \bigcup\{B \mid B \in \mathcal{BA}\}$.
2) For $Y \in \mathcal{BA}$, $Y^{**} = Y$.

3) For $[x]$ and $[x']$, $[x] \sqsubseteq [x'] \Longleftrightarrow M(xR) \subseteq M(x'R)$.

Property (1) indicates that the union of all attribute blocks is the universe of attributes V. Property (2) says that the pair (Y^*, Y) that satisfies $Y^{**} = Y$ is a formal concept. The property (3) shows that the order relation between object classes can be redefined by using membership of attribute blocks. An object super-class must associate with more attribute blocks than any of its sub-classes does.

Example 2: In Table II, the object blocks and attribute blocks are:

$$\begin{array}{ll} X_1 = \{[1], [10]\}, & Y_1 = \{[a], [e], [l]\}, \\ X_2 = \{[4], [8], [10]\}, & Y_2 = \{[c], [e], [j], [l]\}, \\ X_3 = \{[6], [8], [10]\}, & Y_3 = \{[g], [j], [l]\}. \end{array}$$

We know that $U = X_1 \cup X_2 \cup X_3$ and $V = Y_1 \cup Y_2 \cup Y_3$.

By considering the sufficiency operator *, we know that $X_1^* = X_1^{**}$. So, the pair (X_1, X_1^*) is a formal concept.

For the object class $[1]$, it associates with the set of attributes $\{[a], [e], [l]\}$. The membership of attribute blocks for the set of attributes $\{[a], [e], [l]\}$ is:

$$M(\{[a], [e], [l]\}) = \{Y_1\}.$$

For the object class $[10]$, it associates with the whole universe $V = \{[a], [c], [e], [g], [j], [l]\}$. Thus, the membership of attribute blocks for V is:

$$M(V) = \{Y_1, Y_2, Y_3\}.$$

Therefore, the object class $[10]$ is a super-class of the object class $[1]$ because the membership of attribute blocks for $[10]$'s attribute set includes the membership of attribute blocks for $[1]$'s attribute set, that is, $[1] \sqsubseteq [10] \Longleftrightarrow M([a], [e], [l]) \subseteq M(V)$.

The family of object blocks $\mathcal{BO}$ and the family of attribute blocks $\mathcal{BA}$ have similar properties. However, the sizes of them are not necessarily equal. In fact, when every object block associates with an attribute bottom class, and every attribute block associates with an object bottom class, the sizes of $\mathcal{BO}$ and $\mathcal{BA}$ are equal, just like the example demonstrated in Example 2.

VII. Conclusion

FCA and HCA may be considered as two different approaches on visual representation of data. They focus on different data structures and provide two different types of knowledge derived from a formal context based on different notions. We focus on some specific object and attribute sets that are considered as basic sets to compose other sets. We provide a set-theoretical analysis to show some connections between FCA and HCA based on these basic sets. Our studies reveal some basic common characteristics of FCA and HCA, and may be useful for further research on data analysis.

References

[1] Buszkowski, W. Approximation spaces and definability for incomplete information systems, *Rough Sets and Current Trends in Computing, Proceedings of 1st International Conference, RSCTC'98*, 115-122, 1998.

[2] De Boeck, P. and Rosenberg, S. Hierarchical classes: model and data analysis, *Psychometrika*, **53**, 361-381, 1988.

[3] De Boeck, P. and Van Mechelen, I. Traits and taxonomies: a hierarchical classes approach, *European Journal of Personality*, **4**, 147-156, 1990.

[4] De Boeck, P., Rosenberg, S., and Van Mechelen, I. The hierarchical classes approach: a review, in: I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns (Eds.) *Categories and Concepts: Theoretical Views and Inductive Data Analysis*, Academic Press, New York, 265-286, 1993.

[5] Düntsch, I. and Gediga, G. Approximation operators in qualitative data analysis, in: *Theory and Application of Relational Structures as Knowledge Instruments*, de Swart, H., Orlowska, E., Schmidt, G. and Roubens, M. (Eds.), Springer, Heidelberg, 216-233, 2003.

[6] Feger, H. and De Boeck, P. Categories and concepts: introduction to data analysis, in: I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns (Eds.) *Categories and Concepts: Theoretical Views and Inductive Data Analysis*, Academic Press, New York, 203-224, 1993.

[7] Ganter, B. and Wille, R. *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, New York, 1999.

[8] Gediga, G. and Düntsch, I. Modal-style operators in qualitative data analysis, *Proceedings of the 2002 IEEE International Conference on Data Mining*, 155-162, 2002.

[9] Lin, T.Y. Granular computing, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Proceedings of the 9th International Conference*, Lecture Notes in Artificial Intelligence 2639, 16-24, 2003.

[10] Pawlak, Z. Rough sets, *International Journal of Computer and Information Sciences*, **11**, 341-356, 1982.

[11] Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Boston, 1991.

[12] Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts, in: *Ordered sets*, Ivan Rival (Ed.), Reidel, Dordecht-Boston, 445-470, 1982.

[13] Yao, Y.Y. Information granulation and rough set approximation, *International Journal of Intelligent Systems*, **16**, 87-104, 2001.

[14] Yao, Y.Y. Concept Lattices in Rough Set Theory, *Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS2004)*, 796-801, 2004.

[15] Yao, Y.Y. A comparative study of formal concept analysis and rough set theory in data analysis, *Proceedings of Rough Sets and Current Trends in Computing*, 59-68, 2004.

[16] Yao, Y.Y. and Chen, Y.H. Rough set approximations in formal concept analysis, *Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS2004)*, 73-78, 2004.

[17] Zadeh, L.A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **19**, 111-127, 1997.

[18] Zhang, W.X., Wei, L. and Qi, J.J. Attribute reduction in concept lattice based on discernibility matrix, manuscript, 2005.