A Logic Language of Granular Computing

Yiyu Yao and Bing Zhou
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {yyao, zhou200b}@cs.uregina.ca

Abstract

Granular computing concerns human thinking and problem solving, as well as their implications to the design of knowledge intensive systems. It simplifies complex real world problems by considering different levels of granularity. Basic granules represent the elementary units of human knowledge. A granular structure reflects structural connections of different pieces of knowledge. The main objective of this paper is to show how to construct granular structures in a logic setting. This is done by interpreting notions of granular computing by using a logic language $\mathcal{L}$. Two types of granular structures are constructed based on the basic granules, called $\cap$-closure granular structure and $\cup$-closure granular structure. An information table is used to give a concrete interpretation of the language $\mathcal{L}$ for constructing granular structures.

1. Introduction

Cognitive science [15] and cognitive informatics [18, 19] study the human intelligence and its computational process. As an effective way of thinking, we typically focus on a particular level of abstraction and ignore irrelevant details. This not only enables us to identify differences between objects in the real world, but also helps us to view different objects as being the same, if low-level detail is ignored. Granular computing [1, 5, 10, 21, 23, 26, 28, 30, 32, 33] can be seen as a formal way of modeling this human thinking process. It is a practical way to perceive complex real world problems based on simple computational models. One can view granular computing from three different angles: a way of structured thinking, a method of structured problem solving, and a paradigm of structured information processing [28]. They focus on three different perspectives. Structured thinking focuses on modeling human perception of the reality. Structured problem solving focuses on methods and strategies for finding solutions. Structured information processing focuses on the implementation of computer based systems. These three perspectives together form the granular computing triangle [30], and the center of it is granular structures.

A granular structure provides structured description of a system or an application under consideration. In a specific application, individuals sharing the same properties can be put into the same granule. Basic granules represent the basic knowledge of human intelligence. They are the main focuses or basic observations of a problem in the real world. Once the basic granules are properly identified, one can investigate the relations between them and define the related computational operations based on them. In this way, structural knowledge is formed to help us to see the relationships between different parts of the problem, and hence understand the real world problem more clearly.

During the past decade, many researchers focused on the semantic and algorithmic aspects of granularity in modeling and reasoning, such as the process of information granulation and computing with granules [4, 8, 11, 13]. In 1985, Hobbs [3] briefly presented a logic based study of granularity. In this paper, we introduce a logic language of granular computing and show the construction of granular structures for some concrete granular computing models.

Rough set analysis and formal concept analysis are two concrete models of granular computing for information representation and data analysis [6, 7, 9, 20, 27]. Rough set analysis studies relationships between objects and their attribute values in an information table. Formal concept analysis studies the relationships of objects and attributes in a formal context. In spite of their differences, they share the same basic notions, such as concept definability, and the process of constructing definable sets of objects and attributes. By extracting the high-level similarities from the two methods, we propose a logic language $\mathcal{L}$ as a more general logic approach to granular computing.

The language $\mathcal{L}$ is an extension of the decision logic language used by Pawlak in rough set theory [7]. Instead of expressing the atomic formulas by a particular concrete type of conditions, we treat them as primitive notions that can be interpreted differently. This flexibility enables us to describe granules in different applications. The language is interpreted in the Tarski's style through the notion of a model and satisfiability [2, 9, 22, 24, 25]. The model is a nonempty domain consisting of a set of individuals. The basic granules of the model are represented by atomic formulas. An individual satisfies a formula if the individual has the properties as specified by the formula. The introduction of the language $\mathcal{L}$ brings new insights into the notion of definability in rough set analysis and formal concept analysis. That is, a granule is definable if there is a formula in the language $\mathcal{L}$ that defines it, and undefinable otherwise. The knowledge representation and data analysis are no longer restricted to an information table or a formal context. One can directly work on relations between objects and their properties by addressing them with basic granules.

By using the language $\mathcal{L}$, we can formally define granules in the model M. We can construct two types of granular structures. One is constructed by a top-down process by dividing larger granules into smaller and lower level granules, we call it an $\cap$-closure granular structure. The other is constructed by a bottom-up process by forming larger and higher level granules with smaller granules, we call it an $\cup$-closure granular structure. As an illustration, we interpret the language $\mathcal{L}$ in an information table and analyze two types of granular structures.

2. The Logic Language $\mathcal{L}$

In this section, we introduce a logic language $\mathcal{L}$ of granular computing by adopting and modifying the decision logic language used in rough set theory [7, 24].

2.1. Formulation

The language $\mathcal{L}$ is constructed from a finite set of atomic formulas, denoted by $\mathcal{A} = \{p, q, ...\}$. Each atomic formula may be interpreted as representing one piece of basic knowledge. We assume that they are the elementary units that one uses to represent and understand a real world problem. The physical meaning of atomic formulas becomes clearer in a particular application. In general, an atomic formula corresponds to one particular property of an individual under discussion. The construction of atomic formulas is an essential step of knowledge representation. The set of atomic formulas provides a basis on which more complex knowledge can be represented. Compound formulas can be built recursively from atomic formulas by using logic connectives. If ϕ and ψ are formulas, then so are $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.

The semantics of the language $\mathcal{L}$ is defined in the Tarski's style by using the notions of a model and satisfiability. The model is a nonempty domain consisting of a set of individuals, denoted by $M = \{x, y, ...\}$. The meaning of formulas can be given recursively. For an atomic formula p, we assume that an individual $x \in M$ either satisfies p or does not satisfy p, but not both. For an individual $x \in M$, if it satisfies an atomic formula p, we write $x \models p$, otherwise, we write $x \not\models p$. The satisfiability of an atomic formula by individuals of M is viewed as the knowledge describable by the language $\mathcal{L}$. An individual satisfies a formula if the individual has the properties as specified by the formula. Let ϕ and ψ be two formulas, the satisfiability of compound formulas is defined as follows:

$$
\begin{array}{lll}
(1). & x \models \neg\phi & \text{iff} \quad x \not\models \phi, \\
(2). & x \models \phi \wedge \psi & \text{iff} \quad x \models \phi \text{ and } x \models \psi, \\
(3). & x \models \phi \vee \psi & \text{iff} \quad x \models \phi \text{ or } x \models \psi, \\
(4). & x \models \phi \rightarrow \psi & \text{iff} \quad x \models \neg\phi \vee \psi, \\
(5). & x \models \phi \leftrightarrow \psi & \text{iff} \quad x \models \phi \rightarrow \psi \text{ and} \\
 & & \qquad x \models \psi \rightarrow \phi.
\end{array}
$$

To emphases the roles played by the set of atomic formulas $\mathcal{A}$ and the set of individuals M, we also rewrite the language $\mathcal{L}$ as $\mathcal{L}(\mathcal{A}, M)$.

A fundamental difference between the language $\mathcal{L}$ and other decision logic languages is the treatment of the set of atomic formulas. In early works, atomic formulas are defined based on other notions. For example, atomic formulas can be defined in an information table based on the values of attributes [7, 24]. In this paper, we treat atomic formulas as a primitive notion. Many concrete examples of this language can be obtained by specific definitions of atomic formulas. The construction of the set of atomic formulas and the model M depends on a particular application. For modeling different problems, we may choose different sets of atomic formulas and models. The language $\mathcal{L}$ therefore is flexible and enables us to describe a variety of problems.

With the notion of satisfiability, one can introduce a set-theoretic interpretation of formulas of the language $\mathcal{L}$. If ϕ is a formula, the meaning of ϕ in the model M is the set of individuals defined by:

$$m(\phi) = \{x \in M \mid x \models \phi\}. \tag{1}$$

That is, $m(\phi)$ is the set of individuals satisfying the formula ϕ. This establishes a correspondence between logic connectives and set-theoretic operators. The following properties hold [7]:

$$
\begin{aligned}
&(\text{C1}). \quad m(\neg\phi) = -m(\phi), \\
&(\text{C2}). \quad m(\phi \wedge \psi) = m(\phi) \cap m(\psi), \\
&(\text{C3}). \quad m(\phi \vee \psi) = m(\phi) \cup m(\psi), \\
&(\text{C4}). \quad m(\phi \rightarrow \psi) = -m(\phi) \cup m(\psi), \\
&(\text{C5}). \quad m(\phi \leftrightarrow \psi) = (m(\phi) \cap m(\psi)) \cup \\
&\qquad\qquad\qquad (-m(\phi) \cap -m(\psi)),
\end{aligned}
$$

where $-m(\phi) = M - m(\phi)$ denotes the set complement of $m(\phi)$.

In the study of concepts [14, 16, 17], many interpretations have been proposed and examined. The classical view regards a concept as a unit of thoughts consisting of two parts, namely, the intension and extension of the concept [2, 17, 20]. By using the language $\mathcal{L}$, we obtain a simple and precise representation of a concept in terms of its intension and extension. That is, a concept is defined by a pair $(\phi, m(\phi))$. The formula ϕ is the description of properties shared by individuals in $m(\phi)$, and $m(\phi)$ is the set of individuals satisfying ϕ. A concept is thus described jointly by its intension and extension. This formulation enables us to study concepts in a logic setting in terms of intensions and in a set-theoretic setting in terms of extensions.

2.2. Two Sub-languages

The language $\mathcal{L}$ provides a formal method for describing and interpreting rules in data mining and machine learning [31]. In many situations, one is only interested in certain types of rules. For example, rules contain only the logical connective $\wedge$. This requires us to consider the restriction of the language $\mathcal{L}$ to certain logical connectives. In this paper, we consider two sub-languages of $\mathcal{L}$. One uses only the conjunctive connective $\wedge$, written as $\mathcal{L}_\wedge(\mathcal{A}, M, \wedge)$.The other uses only the disjunctive connective $\vee$, written as $\mathcal{L}_\vee(\mathcal{A}, M, \vee)$.

3. Granular Structures

In the language $\mathcal{L}$, each formula is associated with a subset of M. This subset may be viewed as a granule in M. Therefore, we can study granular structures of M through the language $\mathcal{L}$. A granular structure at least contains three basic components [26, 28]:

- internal structure of a granule;
- collective structure of a family of granules;
- hierarchical structure of a web of granules.

The language $\mathcal{L}$ enables us to study such structures in logic terms.

3.1. Internal Structure

Granules are the building blocks to form a granular structure. The internal structure of a granule represents the characterization of the granule. Analyzing the internal structure of a granule helps us to understand why individuals are draw together.

The indiscernibility relation is a fundamental notion in rough set analysis [6, 7, 9]. By using the logic language $\mathcal{L}$, we can formally define an indiscernibility relation on the model M. For a subset $\mathcal{A}_0 \subseteq \mathcal{A}$, two individuals $x, y \in M$ are indistinguishable if no formula in $\mathcal{A}_0$ can distinguish them. Let us define a mapping from M to $\mathcal{A}$ as follows:

$$m'(x) = \{p \in \mathcal{A} \mid x \models p\}.$$

That is, $m'(x)$ is the set of atomic formulas satisfied by x. For a subset $\mathcal{A}_0 \subseteq \mathcal{A}$, the indiscernibility relation can be defined by:

$$x \sim_{\mathcal{A}_0} y \text{ iff } m'(x) \cap \mathcal{A}_0 \ = \ m'(y) \cap \mathcal{A}_0$$

The language $\mathcal{L}$ can be used to reason about intensions. Each formula represents an intension of a concept. For two formulas ϕ and ψ, we say that ϕ is more specific than ψ, and ψ is more general than ϕ, if and only if $\models \phi \rightarrow \psi$, namely, ψ logically follows from ϕ. In this case, the meaning sets of ϕ and ψ have the relation $m(\phi) \subseteq m(\psi)$. Moreover, $m(\phi)$ is called a sub-granule of $m(\psi)$ and $m(\psi)$ a super-granule of $m(\phi)$.

In the language $\mathcal{L}$, logic formulas are used to characterize definability. The definability of a set of individuals or a granule can be defined formally. We say a subset or a granule $X \subseteq M$ is definable in the model M if and only if there exits a formula ϕ in the language $\mathcal{L}$ such that,

$$X = m(\phi).$$

Otherwise, it is undefinable. Accordingly, the family of all definable sets or granules is defined as:

$$Def(\mathcal{L}(\mathcal{A}, M)) = \{m(\phi) \mid \phi \in \mathcal{L}(\mathcal{A}, M)\}.$$

In this way, the language $\mathcal{L}$ only enables us to define certain subsets of M. For an arbitrary subset of M, we may not be able to construct a formula for it. In other words, depending on the set of atomic formulas, the language $\mathcal{L}$ can only describe a restricted family of subsets of M.

3.2. Collective Structure

Granules in the same level are formed with respect to a certain level of abstraction and collectively show a certain structure. The collective structures are related to granules in other levels.

We can classify granules by the number of atomic formulas in their intensions. In the sub-language $\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)$, a granule involving k atomic formulas is called an k-conjunction. An k-conjunction granule is more general than its ($k+1$)-conjunctions, and more specific than its ($k-1$)-conjunctions. In the sub-language $\mathcal{L}_{\vee}(\mathcal{A}, M, \vee)$, a granule involving k atomic formulas is called an k-disjunction. In this case, an k-disjunction granule is more general than its ($k-1$)-disjunctions, and more specific than its ($k+1$)-disjunctions.

3.3. Hierarchical Structure

Granules can be ordered based on their generalities or sizes. For example, in set-theoretic setting, the size of a granule can be defined by its cardinality. One can define operations on granules so that smaller granules can form larger granules, and larger granules can be decomposed into smaller granules. Elementary granules are the most basic granules which are defined by atomic formulas in the language $\mathcal{L}(\mathcal{A}, M)$. Based on them, smaller granules are defined by formulas in the sub-language $\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)$ where atomic formulas are connected by the conjunctive connective $\wedge$, and larger granules are defined by formulas in the sub-language $\mathcal{L}_{\vee}(\mathcal{A}, M, \vee)$ where atomic formulas are connected by the disjunctive connective $\vee$. In this case, a granule is said to be definable if it is the intersection or union of some elementary granules [29]. Connections between granules can be represented as binary relations. For example, the relation could be an order relation [26] interpreted as "more general than" or "more specific than".

Granules in different levels are linked by the order relations and operations on granules. A higher level contains granules that are ordered before granules in a lower level, and granules in a lower level are ordered after granules in a higher level. Granules in a higher level can be decomposed into many smaller granules with more detail in a lower level, and conversely granules in a lower level can form more abstract larger granules in a higher level. The connections of different levels form a multi-level hierarchical structure.

3.4. Two Types of Granular Structures

The three components as a whole is referred to as a granular structure. Based on different relations between granules, there are two ways to construct a granular structure. One is constructed by a top-down process, we call it an $\cap$-closure granular structure. The other is constructed by a bottom-up process, we call it an $\cup$-closure granular structure.

Let $GS_{\cap}(\mathcal{L})$ denote the $\cap$-closure granular structure of the model M. We can formally define it by the sub-language $\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)$, written as:

$$GS_{\cap}(\mathcal{L}_{\wedge}) = (Def(\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)), \cap),$$

where $Def(\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge))$ is the family of granules defined by the sub-language $\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)$.

The process of constructing an $\cap$-closure granular structure is a top-down process, which involves dividing a larger granule into smaller and lower level granules. Each granule is labeled by the formulas of the sub-language $\mathcal{L}_{\wedge}(\mathcal{A}, M, \wedge)$. At the top level, the most general granule is labeled by the formula `T`, which is satisfied by every individual. The next level are the elementary granules labeled by atomic formulas. The intersections of two elementary granules produce the next level of granules labeled by the conjunction of the two atomic formulas, and so on. Finally, at the bottom level, we close the structure by a most specific granule formed by the intersection of all elementary granules.

Let $GS_{\cup}(\mathcal{L})$ denote the $\cup$-closure granular structure of the model M. We can formally define it by the sub-language $\mathcal{L}_{\vee}(\mathcal{A}, M, \vee)$, written as:

$$GS_{\cup}(\mathcal{L}_{\vee}) = (Def(\mathcal{L}_{\vee}(\mathcal{A}, M, \vee)), \cup),$$

where $Def(\mathcal{L}_{\vee}(\mathcal{A}, M, \vee))$ is the family of granules defined by the sub-language $\mathcal{L}_{\vee}(\mathcal{A}, M, \vee)$.

The process of constructing an $\cup$-closure granular structure is a bottom-up process, which involves the process of forming a larger and higher level granule with smaller and lower level granules. At the bottom level, a most specific granule is labeled by the formula `T'`, which is not satisfied by any individual. The upper level are the elementary granules labeled by atomic formulas. The unions of two elementary granules produce the upper level of granules labeled by the disjunction of the two atomic formulas, and so on. Finally, at the top level, we close the structure by a most general granule formed by the union of all elementary granules.

4. Interpreting the Logic Language $\mathcal{L}$ in an Information Table

In this section, we use an information table as a concrete granular computing model to show the usefulness of the language $\mathcal{L}$. We analyze granular structures in rough set theory [7, 9].

4.1. Information Tables

An information table provides a convenient way to describe a finite set of objects by a finite set of attributes [7]. Formally, an information table can be expressed as:

$$S = (U, At, \{V_a \,|\, a \in At\}, \{\{R_a\} \,|\, a \in At\}, \{I_a \,|\, a \in At\}),$$

where

U is a finite nonempty set of objects,
At is a finite nonempty set of attributes,
V_a is a nonempty set of values for $a \in At$,
$\{R_a\}$ is a family of binary relations on V_a,
$I_a : U \to V_a$ is an information function.

Each information function I_a maps an object in U to a value of V_a for an attribute $a \in At$.

The above definition of an information table considers more knowledge and information about relationships between values of attributes. Each relation R_a can represent similarity, dissimilarity, or ordering of values in V_a [2]. The equality relation $=$ is only a special case of R_a. The standard rough set theory uses the trivial equality relation on attribute values [7].

Pawlak and Skowron [9] consider a more generalized notion of an information table. For each attribute $a \in At$, a relational structure $\Re_a$ over V_a is introduced. A language can be defined based on the relational structures. A binary relation is a special case of relational structures.

4.2. Constructing Granular Structures in an Information Table

In an information table, if we are interested in the relationships of objects of the universe U in terms of their attribute values, we can construct the language $\mathcal{L}$ by using U as the model M. That is, individuals of M are objects in the universe U. The set of atomic formulas are constructed as follows. With respect to an attribute $a \in At$ and an attribute value $v \in V_a$, an atomic formula of the language $\mathcal{L}$ is denoted by (a, R_a, v). An object $x \in U$ satisfies an atomic formula (a, R_a, v) if the value of x on attribute a is R_a-related to the value v, that is $I_a(x)\ R_a\ v$, we write:

Object	Height	Hair	Eyes	Class
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

Table 1. An Information Table

$$x \models (a, R_a, v) \text{ iff } \quad I_a(x)\ R_a\ v.$$

We denote the language as $\mathcal{L}(\{(a, R_a, v)\}, U)$. The granule corresponding to the atomic formula (a, R_a, v), namely, its meaning set, is defined as:

$$m(a, R_a, v) = \{x \in U \mid I_a(x) R_a v\}.$$

Granules corresponding to the compound formulas are defined by Equation (1).

The language for standard rough set theory [6, 7, 9] is given by $\mathcal{L}(\{(a, =, v)\}, U)$ with atomic formulas in the form of $(a, =, v)$. An object $x \in U$ satisfies an atomic formula $(a, =, v)$ if the value of x on attribute a is v, that is, $I_a(x) \ = \ v$. We write:

$$x \models (a, =, v) \text{ iff } \quad I_a(x) \ = \ v.$$

The granule,

$$m(a, =, v) \ = \ \{x \in U \mid I_a(x) = v\}.$$

corresponds to the atomic formula $(a, =, v)$.

We can simply construct the $\cap$-closure and $\cup$-closure granular structures in an information table by using the equality relation. Formally, we can rewrite the $\cap$-closure granular structure as:

$$GS_{\cap}(\mathcal{L}) = (Def(\mathcal{L}_{\wedge}((a, =, v), U, \wedge)), \cap).$$

Similarly, we can rewrite the $\cup$-closure granular structure as:

$$GS_{\cup}(\mathcal{L}) = (Def(\mathcal{L}_{\vee}((a, =, v), U, \vee)), \cup).$$

Example 1 Table 1 is an information table taken from [12]. Each object is described by four attributes. The column labeled by "Class" denotes an expert's classification of the objects. The possible values for three

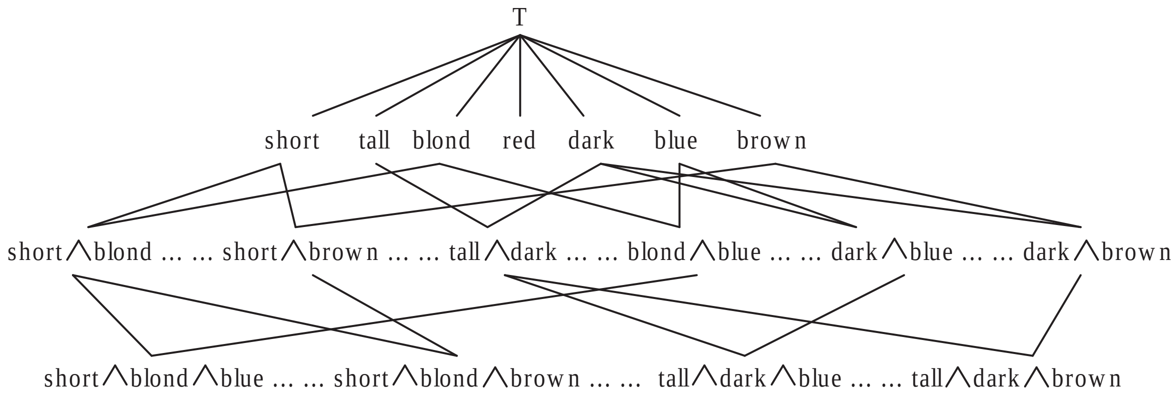

Figure 1. An example of ∩-closure granular structure

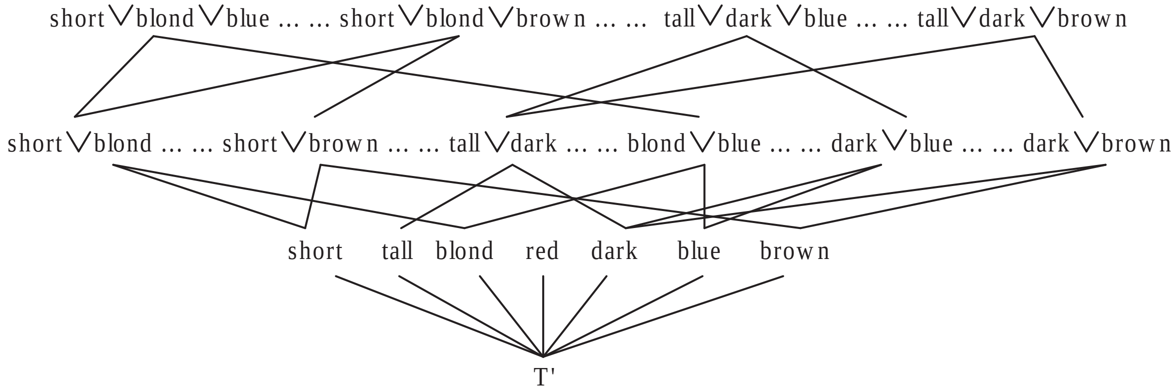

Figure 2. An example of ∪-closure granular structure

attributes {Height, Hair, Eyes} are:

$$\begin{aligned} V_{\text{Height}} &= \{\text{short}, \text{tall}\}, \\ V_{\text{Hair}} &= \{\text{blond}, \text{dark}, \text{red}\}, \\ V_{\text{Eyes}} &= \{\text{blue}, \text{brown}\}. \end{aligned}$$

If the attribute Height is chosen, we can partition the universe into the following equivalence classes or elementary granules:

$$\{o_1, o_2, o_8\},\ \{o_3, o_4, o_5, o_6, o_7\},$$

corresponding to atomic formulas:

$$\begin{aligned} &(\text{Height}, =, \text{short}), \\ &(\text{Height}, =, \text{tall}), \end{aligned}$$

respectively. Similarly, the use of attribute Hair produces the following equivalence classes or elementary granules:

$$\{o_1, o_2, o_6, o_8\},\ \{o_3\},\ \{o_4, o_5, o_7\},$$

corresponding to atomic formulas:

$$\begin{aligned} &(\text{Hair}, =, \text{blond}), \\ &(\text{Hair}, =, \text{red}), \\ &(\text{Hair}, =, \text{dark}), \end{aligned}$$

respectively. For the attribute Eyes, we have:

$$\{o_1, o_3, o_4, o_5, o_6\},\ \{o_2, o_7, o_8\},$$

corresponding to atomic formulas:

$$\begin{aligned} &(\text{Eyes}, =, \text{blue}), \\ &(\text{Eyes}, =, \text{brown}), \end{aligned}$$

respectively.

Smaller granules are set-theoretic intersections of elementary granules. For example, sets

$$\begin{aligned} &\{o_1, o_2, o_8\} \cap \{o_1, o_2, o_6, o_8\} = \{o_1, o_2, o_8\}, \\ &\{o_3, o_4, o_5, o_6, o_7\} \cap \{o_4, o_5, o_7\} \cap \{o_2, o_7, o_8\} \\ &\qquad = \{o_7\} \end{aligned}$$

are smaller granules with the corresponding compound formulas $(\text{Height}, =, \text{short}) \wedge (\text{Hair}, =, \text{blond})$, and $(\text{Height}, =, \text{tall}) \wedge (\text{Hair}, =, \text{dark}) \wedge (\text{Eyes}, =, \text{brown})$, respectively.

Figure 1 draws part of the $\cap$-closure granular structure for Table 1. In the figure, we assume that an attribute appears at most once in each formula of $GS_{\cap}(\mathcal{L})$. An atomic formula is simply represented by the attribute value. For example, the atomic formula $(\text{Height}, =, \text{short})$ is simply written as short.

Example 2 In Table 1, larger granules are set-theoretic unions of elementary granules. For example, sets

$$\{o_1, o_2, o_8\} \cup \{o_4, o_5, o_7\} = \{o_1, o_2, o_4, o_5, o_7, o_8\},$$
$$\{o_3, o_4, o_5, o_6, o_7\} \cup \{o_3\} \cup \{o_2, o_7, o_8\} = \{o_2, o_3, o_4, o_5, o_6, o_7, o_8\},$$

are larger granules for the corresponding compound formulas $(\text{Height}, =, \text{short}) \vee (\text{Hair}, =, \text{dark})$, and $(\text{Height}, =, \text{tall}) \vee (\text{Hair}, =, \text{red}) \vee (\text{Eyes}, =, \text{brown})$, respectively.

Figure 2 draws part of the $\cup$-closure granular structure for Table 1.

5. Conclusion

Granular computing is a new area of research. Its main purpose is to model, state, and solve real world problems at multiple levels of granularity. A fundamental notion of granular computing is granular structures described in terms of granules, levels, and hierarchical structures. In order to formally define granular structures, we introduce a logic language $\mathcal{L}$. Formulas of the language is recursively constructed from a set of atomic formulas, representing the basic or elementary observations. The meaning of formulas is defined in Tarski's style by using a model M. It is assumed that an individual in M either satisfies a formula or does not satisfy a formula.

The logic language provides a method to define a granule as a pair $(\phi, m(\phi))$ consisting of a formula of the language $\mathcal{L}$ and a subset of M. Furthermore, the notions of indiscernibility and definability can be defined. Two sub-languages of $\mathcal{L}$ are introduced, each of them uses only one logic connective. They lead to the introduction of two types of granular structures.

The results from this study show that it is useful to study formal and concrete models of granular computing. In addition, it may be necessary to further study logic approaches to granular computing.

References

[1] Bargiela, A., Pedrycz W. Granular computing: an introduction, Kluwer Academic Publishers, Boston, 2002.

[2] Demri, S., Orlowska, E. Logical analysis of indiscernibility, in: *Incomplete Information: Rough Set Analysis*, Orl owska, E. (Ed.), Physica Verlag, Heidelberg, 347-380, 1997.

[3] Hobbs, Jerry R. Granularity, *Proceedings, Ninth International Joint Conference on Artificial Intelligence*, 432-435, 1985.

[4] Klir, G.J. Basic issues of computing with granular probabilities, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, 101-105, 1998.

[5] Lin, T.Y., Yao, Y.Y. and Zadeh, L.A. (Eds.) Data mining, rough sets and granular computing, Physica-Verlag, Heidelberg, 2002.

[6] Nguyen, H. S., Skowron, A., Stepaniuk, J. Granular computing: a rough set approach, *Computational Intelligence*, **17**, 514-544, 2001.

[7] Pawlak, Z. *Rough Sets - Theoretical Aspects of Reasoning About Data*, Kluwer Publishers, Boston, Dordrecht, 1991.

[8] Pawlak, Z. Granularity of knowledge, indiscernibility and rough sets, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, 106-110, 1998.

[9] Pawlak, Z., Skowron, A. Rough sets: some extensions, *Information Science,* **177**, 28-40, 2007.

[10] Pedrycz, W. (Ed.) Granular computing: an emerging paradigm, Physica-Verlag, Heidelberg, 2001.

[11] Polkowski, L. and Skowron, A. Towards adaptive calculus of granules, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, 111-116, 1998.

[12] Quinlan, J. R. Learning efficient classification procedures and their application to chess end-games, in: *Machine Learning: An Artificial Intelligence Approach*, Michalski, J.S. et al.(Eds.), Morgan Kaufmann, **1**, 463-482, 1983.

[13] Skowron, A. and Stepaniuk, J. Information granules and approximation spaces, manuscript, 1998.

[14] Smith, E.E. Concepts and induction, in Posner, M.I. (Ed.), *Foundations of Cognitive Science*, The MIT Press, Cambridge, Massachusetts, 501-526, 1989.

[15] Simon, H. A., Kaplan, C. A. Foundations of cognitive science, in *Foundations of cognitive science*, M. I. Posner's (Ed.), Cambridge, MA: MIT Press, 1-47, 1989.

[16] Sowa, J.F. *Conceptual Structures, Information Processing in Mind and Machine*, Addison-Wesley, Reading, Massachusetts, 1984.

[17] Van, Mechelen, I., Hampton, J., Michalski, R.S. and Theuns, P. (Eds.), *Categories and Concepts, Theoretical Views and Inductive Data Analysis*, Academic Press, New York, 1993.

[18] Wang, Y. Cognitive informatics: a new transdisciplinary research field, *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy*, **4**, 115-127, 2003.

[19] Wang, Y. On cognitive informatics, *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy*, **4**, 151-167, 2003.

[20] Wille, R. Concept lattices and conceptual knowledge systems, *Computers Mathematics with Applications*, **23**, 493-515, 1992.

[21] Yao, J.T., Yao, Y.Y. Induction of classification rules by granular computing, *Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing*, LNAI **2475**, 331-338, 2002.

[22] Yao, Y.Y. Modeling data mining with granular computing, *Proceedings of the 25th Annual International Computer Software and Applications Conference*, 638-643, 2001.

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

[24] Yao, Y.Y., Liau, C.-J., A generalized decision logic language for granular computing, *FUZZ-IEEE'02 in The 2002 IEEE World Congress on Computational Intelligence*, 1092-1097, 2002.

[25] Yao, Y.Y. A step towards the foundations of data mining, in: *Data Mining and Knowledge Discovery: Theory, Tools, and Technology V*, Dasarathy, B.V. (Ed.), The International Society for Optical Engineering, 254-263, 2003.

[26] Yao, Y.Y. Granular computing, *Proceedings of The 4th Chinese National Conference on Rough Sets and Soft Computing*, **31**, 1-5, 2004.

[27] Yao, Y.Y. A comparative study of formal concept analysis and rough set theory in data analysis, *International Conference on Rough Sets and Current Trends in Computing (RSCTC'2004)*, 59-68, 2004.

[28] Yao, Y.Y. Three perspectives of granular computing, *The Proceedings, International Forum on Theory of GrC from Rough Set Perspective, Journal of Nanchang Institute of Technology*, **25**, 16-21, 2006.

[29] Yao, Y.Y. A note on definability and approximations, *Transactions on Rough Sets VII*, 274-282, 2007.

[30] Yao, Y.Y. The art of granular computing, *Proceeding of the International Conference on Rough Sets and Emerging Intelligent Systems Paradigms*, *LNAI*, **4585**, 101-112, 2007.

[31] Yao, Y.Y. Zhou, B. Chen, Y.H. Interpreting low and high order rules: a granular computing approach, *LNAI*, **4585**, 371-380, 2007.

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

[33] Zadeh, L.A. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, *Soft Computing*, **2**, 23-25, 1998.