Chapter 22
Granular Concept Mapping and Applications

Sumalee Sonamthiang, Kanlaya Naruedomkul, and Nick Cercone

Abstract. This chapter presents a granular concept hierarchy (GCH) construction and mapping of the hierarchy for granular knowledge. A GCH is comprised of multilevel granular concepts with their hierarchy relations. A rough set based approach is proposed to induce the approximation of a domain concept hierarchy of an information system. A sequence of attribute subsets is selected to partition a granularity, hierarchically. In each level of granulation, reducts and core are applied to retain the specific concepts of a granule whereas common attributes are applied to exclude the common knowledge and generate a more general concept. A granule description language and granule measurements are proposed to enable mapping for an appropriate granular concept that represents sufficient knowledge so solve problem at hand. Applications of GCH are demonstrated through learning of higher order decision rules.

Keywords: Information granules, granular knowledge, granular concept hierarchy, granular knowledge mapping, granule description language, higher-order rules, multilevel partitioning, attribute selection

22.1 Introduction

An *information system* in a rough set paradigm [10], [11] is a basic knowledge representation method in an attribute-value system. An information system is represented in a table in which a row keeps an object and each column keeps the

Sumalee Sonamthiang
Institute for Innovative Learning, Mahidol University, Nakhon Pathom, Thailand 73170
e-mail: tew197@gmail.com

Kanlaya Naruedomkul
Mathematics Department, Faculty of Science, Mahidol University, Bangkok, Thailand 10400
e-mail: scknr@mahidol.ac.th

Nick Cercone
Department of Computer Science and Engineering, Faculty of Science and Engineering,
York University, Toronto, Ontario, Canada M3J 1P3
e-mail: ncercone@yorku.ca

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 585–603.
springerlink.com

value of the corresponding attribute. The tabular representation simplifies recording the objects into an information system, especially in real-time transactions by capturing a transaction separately and using a single global representation for every record in every situation. However, an occurrence of a transaction may be related to other transactions in the problem space. Representation in this fashion is seen as a flat and unconnected structure that hides the meaningful relations in the data.

An information analysis based on an attribute-value system considers the values of the attribute subsets to extract relationships within data. The relationships can be classified into two types: internal and external. An internal relationship is the relation between attributes' values within a single object, whereas an external relationship provides connections between many objects. Classical rule discovering methods extract internal relationships from a decision table; however, the obtained rules represent fragmented knowledge and hide the meaningful relationships among objects of a universe. An example of an internal relation rule is expressed by: $IF < x,a >= v_{a1} THEN < x,d >= v_{d_1}$. The rules obtained from internal relationships can be superfluous. Postprocessing is necessary to reduce rules' conflicts, shorten the rule premise, shrink the size of the rule set, or group together similar rules. The rules obtained from internal relations represent fragmented knowledge and remain embodied in hidden meaningful relationships among objects in a universe. Postprocessing to improve quality of the rules has been studied, *for example*, evaluation of association rules' importance [9] and mining higher-order decision rules [26].

Unlike internal relationships, an external relationship among objects does not only provide knowledge of higher-order rules but also for concept approximation.

Table 22.1. Some examples of animal data set

label	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	...	class
aardvark	1	0	0	1	0	0	1	1	...	1
antelope	1	0	0	1	0	0	0	1	...	1
bass	0	0	1	0	0	1	1	1	...	4
bear	1	0	0	1	0	0	1	1	...	1
boar	1	0	0	1	0	0	1	1	...	1
buffalo	1	0	0	1	0	0	0	1	...	1
calf	1	0	0	1	0	0	0	1	...	1
carp	0	0	1	0	0	1	0	1	...	4
catfish	0	0	1	0	0	1	1	1	...	4
cavy	1	0	0	1	0	0	0	1	...	1
cheetah	1	0	0	1	0	0	1	1	...	1
chicken	0	1	1	0	1	0	0	0	...	2
clam	0	0	1	0	0	0	1	0	...	7
crab	0	0	1	0	0	1	1	0	...	7
crayfish	0	0	1	0	0	1	1	0	...	7
crow	0	1	1	0	1	0	1	0	...	2

For example, given animal data in the information system in Table 22.1, a human categorizes and conceptualizes a concept differently.

One may give a name of concept represented by class 1 as mammal class without concerning surrounding data. But for a machine, it can learn that only the attribute value *milk=1* is sufficient to determine the class 1 precisely (based on given data sets). In this example, however, some attributes' values such as *egg=0*, *hair=1*, *toothed*=1 are correlated with (but not necessary dependent) *milk=1*. These features can be seen as dominate attribute subset in which together the attribute subset has more gravity to draw animal class abstraction. Therefore, a machine can form a concept by using the most dominant attribute subset on the decision class to mimic human granular conceptualization. On the other hand, if one is asked to differentiate animal in class 1, one needs to granulate knowledge relative to more of the detailed features such as size, domestic/wild, and legs based on given data.

Various attribute subsets can be considered to obtain external relations from different dimensions; thus, groups of related objects can be discovered. The knowledge obtained from this type of relationship is represented as clusters attached with each clusters' description[4],[5], [20]. Moreover, the relationships among the objects can be local or global; specifically, relations can be extracted in many levels of granularity. We hypothesize that discovering external relationships between objects in a universe can be used to approximate the connections of objects and form multilevel granular concepts.

Rough set theory (RST) [10], [11] provides a formal framework that focuses on both internal and external relations. For extensions on rough sets please refer to [12], [13], [14]. In rough sets, the indiscernibility relation expresses the external relations between objects and the relation can be used to form a granular concepts. Rough sets also influence Granular Computing (GrC), an emerging paradigm for computing of concept approximation [1], [15]. *A granular concept represents sufficient information to solve a problem at hand. How coarse or how specific should a granular concept be to convey such sufficient information?*

In this study, a granular concept hierarchy (GCH) and granular concept mapping are presented. GCH is a multilevel of granularity of a domain knowledge in hierarchical structure. This structure provides rich information for a problem solver and mapping mechanism to search for an appropriate level of granularity. GCH comprises of a root node, a set of nonroot nodes, a non empty set of leaves, and the hierarchy relations. A node in a tree can be seen as a granule in which instances in the node hold similar properties to a certain degree, and they are part of their parent. Thus, a parent holds the common properties of its children, and the siblings have a certain degree of similarity to each other by the common properties.

We present two algorithms to construct a GCH. The first algorithm is to recursively partition an information system into a GCH. The second algorithm computes the selection of a sequence of attribute subsets which is necessary to partition a granularity hierarchically. *Common attributes* (defined as the subset of attributes that forms indiscernibility relations among the objects of a granule) and the attributes' values are united to form the granular concept's description. At each level of granulation, reducts and core are applied to retain the specific concepts of a granule, whereas common attributes are applied to exclude the common knowledge and generate a more general concept. We also present a granule description language that provides semantic encoding as well as an interpretation of which semantics a granule concept conveys. The semantics are encoded by rough set approximation. Degree of coarseness/specificity of a granular concept, then, can be interpreted for a target concept.

The chapter is outlined as follows: In the next section, related works in hierarchical information granulation are explored. A formal definition of a GCH and an example are given in Section 22.3. Section 22.4 details two algorithms to construct a GCH hierarchically. Section 22.5 reports our evaluation and results of higher-order rules learning from a GCH of an artificial Zoo database. Finally, Section 22.6 presents conclusion and discussion of possible extensions.

22.2 Related Study

In this section, previous studies on multilevel granular concept approximation are reviewed. There are various approaches to approximate uncertain concepts from uncertain data. Four main approaches are focused which are rough sets, fuzzy sets, near sets, and shadowed sets.

The fuzzy sets and shadowed sets provide contributions to GrC [1] for information processing by using continuous membership grades induction [2], [16],[17],[27]. Hierarchical fuzzy sets and shadowed sets can be identified by further refinement of the sets. Multilevel fuzzy sets can be approximated based on previous layer of fuzzy sets in order to obtain multilevel granular concepts. Therefore, defuzzification is needed to map for granular fuzzy concept indexing. It is preferable if the approximated concept mechanism provide descriptive knowledge and knowledge evaluation for hierarchical granular mapping.

RST was proposed by Zdzisław Pawlak (1926-2006) in 1982. The theory is to model indiscernible (similar) objects and forms a basic granule of knowledge about a domain, based on given observations (see, *e.g.*,[18],[21]). However, the observations can be imperfect: inconsistent, insufficient and uncertain. These characteristics of observations, consequently, cause basic granules being rough which are defined as rough sets. Defining rough sets does not require priori probabilistic information about data. Moreover, the rough sets permits induction of rules about uncertainty [6], namely the certain classes as certain rules, and the uncertain classes as possible rules.

In [19], James F. Peters proposed a special theory of near sets. Near sets are disjoint sets that resemble each other to a certain degree. Resemblance of near sets can be obtained using a probe function such as closeness (qualitatively near) between objects as well as other probe functions that return values of object features such as color, shape, texture, and duration. The closeness is determined by the objects' features. Near sets is an extension of rough sets with nearness function for granular concept approximation. A granular knowledge approximation based on near sets is to identify family of an instance x to an instance y by *link relation*. Therefore, a granule X and a granule Y are near sets to each other if and only if mapping the link relation of x with y is *sufficiently* large. The author also proposed a framework to enable searching for relevant nearness relation relative to the problem being solved through a distance measurement. Based on the rough sets, we proposed an idea of domination attribute subset partitioning to granulate an information granule into lower level granules. Like the near set approach, using domination attributes has advantages in linking a family of an object together. We also apply reduct and core attributes to retain specific information until the lowest level of granulation.

Hoa and Son [7] introduced a complex concept approximation approach based on a layered learning method together with RST. The authors used taxonomy as the domain knowledge and attribute values in the data_set to guide composing attributes into intermediate concepts until the target concept is obtained. The target concepts are the concepts in the decision attribute. However, the domain taxonomies are usually unavailable to guide the layer learning and need to be discovered before applying this approach.

A study of granularity-based formal concepts is presented in [21], [25], for example. In [25], the authors defined a formal concept by a pair consisting of its intension and extension $(\phi, m(\phi))$, where ϕ is a logical rule of a subset of attributes with the attributes' values and $m(\phi)$ is a granule obtained by partitioning the universe of objects using the attribute subset ϕ. Moreover, Yao [24] presented an approach to hierarchical granulation based on rough sets called stratified rough approximation. The stratified rough set approximation is a simple multi-level granulation based on nesting of one-level granulation (*e.g.*, granulation by the equivalence relation). Yao [24] presented three methods for multi-layered granulation which are as follows:

- nested rough set approximations induced by a nested sequence of equivalence relations,
- stratified rough set approximations induced by hierarchies, and
- stratified rough set approximations induced by neighborhood systems.

In the nested granulation approach, the granulation starts with indiscernibility relations on a set of objects represented by attribute-value vectors. Then the subsequent indiscernibility relations are defined by successively removing attributes from the set of remaining attributes. Sequencing of attribute subsets for partitioning is determined by dependencies between condition attributes. The sequence of attribute subsets for partitioning affects a granules' extension and the hierarchy structure. By this approach, the obtained hierarchy structures are predefined by the attributes' dependencies. Thus, the approach is unconcerned about the objects similarities which are very important in the sense of clusters. Moreover, there are some information systems that have no attribute dependency. In the stratified rough set approximations induced by hierarchies, levels of hierarchies provide the sequence of granulation. As mentioned earlier, the hierarchy of a domain may be unavailable. In [23], the authors also described the use of neighborhood systems to induce hierarchial partitioning. The neighborhood system $NS(x)$ is a nested family of subsets of the universe, with each neighborhood representing a specific level of similarity to x.

However, an information system can contain a huge number of attributes. The issue of attribute subset selection for measuring the closeness, similarity or proximity in an information system is not studied.

Yao [24] recommended a motivating idea for our approach, that is, stratified approximation can be used to search for an appropriate level of accuracy for an application. Therefore, a map of granular concepts which provides rich information about domain structure is developed.

22.3 Granular Concept Hierarchy

In this section, we shall formally define elements and the hierarchy structure of our granular concept mapping approach. Definitions of a GCH with its syntax and semantics are given and detailed in the next subsection. Then, the granular knowledge evaluation is also presented through the semantics of the target concept approximation using a rough set-based approach.

22.3.1 Formal Definitions of a Granular Concept Hierarchy

A GCH is a hierarchical granular knowledge organization that provides multilevel granular knowledge units, evaluation of knowledge, and knowledge mapping mechanism.

A hierarchy of granular concept mapping is formally defined as a quadruple

$$GCH = < G, R, T, \alpha >, \tag{22.1}$$

where G is a non-empty set of *nodes*, and the nodes themselves are non-empty set. R denotes a relationship between two nodes. T denotes the target concept of a node, and α denotes the accuracy approximation of the target concept T .

R is a binary relation of *parent-child* and *child-parent* relation on g. If $< g, g' > \in R$ then g is the parent of g' and g' is a child of g. There is a designated element r of G called *root*. r holds the universe of elements such that $\neg r = \emptyset$. A branch $BR = g_0, g_1, g_2, ..., g_n$ is the maximal sequence of element of G such that $g_0 = r$, and for every $i \geq 0, < g_i, g_{i+1} > \in R$. Nodes g which $R(g) = \emptyset$ are called *leaves*. The level of g, denoted by $\|g\|$, is defined by n if and only if there is a branch $BR = g_0, g_1, g_2, ..., g_n$, where $g = g_n$. Obviously, $\|r\| = 0$.

T is the target concept of granule which is defined by a set of decision attribute values.

α is knowledge evaluation of a granule g. The knowledge evaluation in our approach is defined by accuracy of rough approximation:

$$\alpha(g) = \frac{|LOWER(g)|}{|UPPER(g)|}. \tag{22.2}$$

$LOWER(g)$ is lower approximation and $|UPPER(g)|$ is upper approximation of a granule g induced by a subset of attribute. $|X|$ denotes the cardinality of a set X. The approximation accuracy is in the range of $0 \leq \alpha(X) \leq 1$, and $\alpha(\emptyset) = 1$.

A GCH comprises of nodes in which the coarsest concept is represented at the root level, whereas the most specific concept is represented at the leaf levels. We articulate a concept by using the idea of the most dominant attribute subset: the more dominant degree attribute subset, the more gravity to draw the objects into concepts by that subset. Once a concept is granulated by the most dominant attributes subset, we obtain the more specific concepts which are drawn by *common attribute subset*. The common attribute subset forms the indiscernibility relations among the concept's extension. This structure allows mapping of appropriate granular knowledge in order to solve a problem at hand. The essences of GCH knowledge organization are as follows.

- In order to map to an appropriate granular knowledge, the problem solver must identify satisfaction criterions. One of satisfaction criterion is that the granular knowledge is evaluated by *sufficient knowledge* for solving a particular problem. If the problem is to find decision rules to predict unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them have smaller boundary regions. If the problem is to predict missing values of condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. One may define a satisfaction criterion by setting precision tolerance of applying the granular knowledge. This criterion permits reducing cost of computation where precisiation is expensive or unavailable.
- Because GCH provides multilevel of granular knowledge ranging from the coarsest level at the root and the most specific level at the leaves, GCH structure provides system of granular knowledge mapping through a tree traversal. Searching for a granular concept in a GCH can be achieved through several techniques such as the depth first search and breadth first search.
- Core attributes are essential to form the more specific concepts since they contains specific characteristics of an object. In GCH construction, core attributes are preserved to retain such specific concepts until the latest granulation.

We shall define the syntax and semantics of GCH and present algorithms to construct a GCH as follows.

22.3.2 Syntax and Semantics of a Granular Concept

This section explains what knowledge is represented in the granular concepts and how to interpret and evaluate knowledge in a granular concept. The section is started by definitions of basic notions, followed by syntax and semantics of a granular concept.

Definition 22.1. *Let g be a node in a map of granular concepts M and g is a decision table, $g \subseteq D$. A common attribute of g is the attribute that forms the indiscernibility relation on $g \times g$. The set of common attributes is denoted by $CA, CA \subseteq A$.*

Definition 22.2. *The set of target concepts of g, denoted by $\tau(g)$, is defined by the set of decision values in the decision attribute of $x \in g$.*

$$\tau(g) = \bigcup v_d | < x, d >= v_d, \forall x \in g. \tag{22.3}$$

Definition 22.3. *The most dominant target concept, $\hat{\tau}$, is defined by the decision value of the largest decision class in g.*

Definition 22.4. *A granular concept description phrase of g, denoted by $\pi(g)$, comprises of atomic predicates. A predicate is defined by a pair of a common attribute's name and a value of the attribute. Each predicate is conjuncted by the $\wedge$ operator to form a phrase.*

$$\pi(g) = ca_0(V_{ca_0}) \wedge ca_1(V_{ca_1}) \wedge ca_2(V_{ca_2}) \wedge ... \wedge ca_n(V_{ca_n}), \tag{22.4}$$

where $ca_i \in CA$ and $|CA| = n, 1 \leq i \leq n$.

Definition 22.5. *A granular concept description language of g is denoted by $\lambda(g)$. The language $\lambda(g)$ is generated by traversing M from g_0 to g. The phrases of the traversed granules are $\wedge$ conjuncted successively to form $\lambda(g)$.*

$$\lambda(g) = \pi(g_0) \wedge \pi(g_1) \wedge ... \wedge \pi(g). \tag{22.5}$$

Definition 22.6. *Syntax of a granular concept g is denoted by a pair:*

$$\psi =< \phi(g), \lambda(g) > \text{ if and only if } x \models \lambda(g), \forall x \in g. \tag{22.6}$$

$\phi(g) = \{x | x \in g\}$ is called concept's extensions and every member of $\phi(g)$ is understood by $\lambda(g)$. Note that $\lambda(g)$ is the granular concept's intension.

Definition 22.7. *Semantics of a granular concept is the accuracy of rough approximation of the granule toward the most dominant target concepts and the concept's intension. The semantics of g is denoted by $\xi(g)$*

$$\xi(g) = \frac{|LOWER(g)|}{|UPPER(g)|}, \tag{22.7}$$

where

$$LOWER(X) = \bigcup [g]_B | x \in g, [g]_B \subseteq g,$$

$$UPPER(X) = \bigcup [g]_B | x \in g, [g]_B \cap g \neq \emptyset,$$

$$[g]_B = \bigcup \{[< a, v >] | a \in B, B = CA \cup \{d\}\}, f(x, a) = \hat{\tau}.$$

A granular concept is indexed by its intension. The granular concept conveys a semantic of being a target concept ($\hat{\tau}$.). To interpret the concept's semantic, one can measure the rough approximation accuracy toward the target concept based on the granular concept intension. Example 1 provides an illustrative explanation.

Example 1. The decision table of Flu diagnosis in Table 22.2 contains four condition attributes of symptoms {*Temperature, Headache, Nausea, Cough*} and one decision attribute {*Flu*}.

Table 22.2. Flu diagnosis

Cases	Temperature	Headache	Nausea	Cough	Flu
1	high	yes	no	yes	yes
2	very high	yes	yes	no	yes
3	high	no	no	no	no
4	high	yes	yes	yes	yes
5	normal	yes	no	no	no
6	normal	no	yes	yes	no

There are six cases of patient. If the first partitioning is $\{Headache\}^*$, two granular concepts of g_1 and g_2 are obtained as shown in Table 22.3. If the equivalence relation is used to discern patients, there is one *common attribute* $CA = \{Headache\}$ for both g_1 and g_2. The description language is $\lambda(g_1) = Headache(yes)$ and $\lambda(g_2) = Headache(no)$. The target concept of g_1 is *having Flu*, and the semantics conveyed by g_1 is *the patients who have headache also get flu*, with the accuracy of approximation is 3/4. For g_2, the target concept is *having no Flu*. The semantics of *having no Flu* is 2/2 of the patients who have no headache.

Table 22.3. Granulated concepts of Flu diagnosis

$g_1 : \lambda(g_1) = Headache(yes)$					
Cases	Temperature	Headache	Nausea	Cough	Flu
1	high	**yes**	no	yes	yes
2	very high	**yes**	yes	no	yes
4	high	**yes**	yes	yes	yes
5	normal	**yes**	no	no	no
$g_2 : \lambda(g_2) = Headache(no)$					
Cases	Temperature	Headache	Nausea	Cough	Flu
3	high	**no**	no	no	no
6	normal	**no**	yes	yes	no

By the definitions, granular concepts can be approximated and interpreted to obtain their semantic. The next section gives details of GCH construction. A recursive partitioning algorithm is proposed as well as an attribute subset selection algorithm to partition the granularity hierarchically.

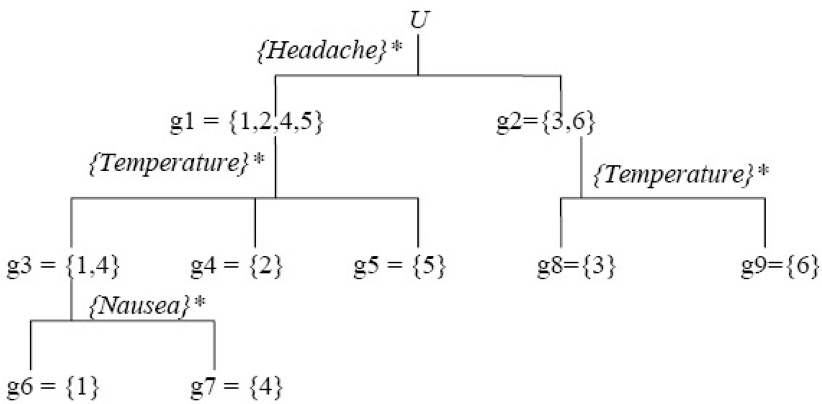

Fig. 22.1. A granular concept hierarchy for the Flu case base

22.4 Granular Concept Hierarchy Construction

The GCH construction is a recursive granulation in top-down manner. Specifically, the recursive construction is given in Section 22.4.1, where the symbols are the ones defined in Section 22.3. We also present an algorithm for attribute subset selection based on the most dominant degree of the attribute subset as illustrated in Section 22.4.2. In Section 22.4.3, the mapping for an appropriate granularity on the GCH is detailed.

22.4.1 An Algorithm for Recursive Granulations

Given a data set as an information system format, all observations start in one cluster, and granulations are performed recursively as one moves down the hierarchy. We design a recursive hierarchy construction as given in Algorithm 22.1. The input to this algorithm is an information system and the output is a GCH. The process begins with finding common attribute subset. Then, a temporary decision (*TempD*) table is derived from the current decision table by removing the common attributes. The *TempD* is not necessary if there is no common attribute. The attribute sequencing is accomplished through local attributes subset selection in the recursive partitioning. We select the most dominant attribute subset based on the attributes' values available in the decision table. We determine the domination using Algorithm 22.2.

Algorithm 22.1. Granular Concept Hierarchy Construction

Input : a decision table, $D = < g, (A \cup \{d\}), (V_a)_{a \subset A}, V_d, f >$.
Output: a granular concept map, $M = < G, R >$.
$g \longleftarrow D$.
$g_0 \longleftarrow g$. // the root of the hierarchy
$TempD \longleftarrow D$.
$B \longleftarrow \emptyset$. // attribute subset for partitioning
$CA \longleftarrow \emptyset$. // the set of common attributes
$granulatedStatus(g) = false$. //mark the granulated concepts
Function GCHconstruct(g)
begin
  if *(g is discernible)* **then**
    1. Find common attribute subset CA of g;
    2. Generate a granule description phrase //See Definition 22.4.
    3. **if** *($CA \neq \emptyset$)* **then**
      $A \longleftarrow A - AC$
      $TempD \longleftarrow < g, (A \cup \{d\}), (V_a)_{a \subset A}, V_d, f >$
    4. $B \longleftarrow MostDAselect(TempD)$.
    //Select the most dominant attribute subset B, see Algorithm 22.2.
    5. Partition the *TempD* by B, $\{B\}* = g_1, g_2, ..., g_n$.
    6. Generate relations of $< g, g_1 >, < g, g_2 >, ..., < g, g_n > \in R$.
    for *all* $g_i, < g, g_i > \in R$ **do**
      7. Find $\hat{\tau}$, the most dominant target concept of g_i.
      // see Definition 22.3
      8. Compute semantics of g_i.
      // see Definition 22.7
      9. $granulatedStatus(g_i) \longleftarrow false$.
    10. $granulatedStatus(g) \longleftarrow true$; //mark g as granulated.
    11. $g \longleftarrow g_1$.
    12. $TempD = < g, (A \cup \{d\}), (V_a)_{a \subset A}, V_d, f >$
  else
    Make a leaf granule.
end
for *all* g_i, $(R(g_i) = R(g))$ *and* $(granulatedStatus(g_i) = false)$ **do**
  GCHconstruct(g_i).

The selected attributes subset is then used to partition *TempD* and assign relationships between the obtained granules (children) and the original granule (parent). If a granule cannot be partitioned by the indiscernibility relation, a leaf node is generated.

22.4.2 An Algorithm for Level-Wise Attribute Selection

In this section, we present an algorithm for attribute subset selection which the selected attribute subset is used in partitioning a granule at each level by

Algorithm 22.2. Most Dominant Attribute Subset Selection

Input : a decision table (*TempD*), CORE, parameter N and ε.
Output: The most dominant attribute subset B // this attribute subset will be used to partition the *TempD* in Algorithm 22.1.

$MostDA \longleftarrow \emptyset$.
$TopDA \longleftarrow \emptyset$.
$B \longleftarrow \emptyset$.
for *each* $a \in A$ **do**
  for *each* $v_a \in V_a$ **do**
    1. $[x]_{a \cup d} \longleftarrow \bigcup\{[a,v] | f(x,a) = v_a, f(x,d) = v_d\}$
  2. $domDegree(a_i) \longleftarrow \arg\max(|[x]_{a \cup d}|)$
$MostDA \longleftarrow \arg\max(domDegree(a_i))$.
$TopDA \longleftarrow TopN \arg\max(domDegree(a_i))$
$B \longleftarrow MostDA$
for *each TopB, TopB* $\subseteq$ *TopDA and* $|TopB| > 1$ **do**
  $[x]_{TopB} \longleftarrow \bigcup\{[a,v] | f(x,a) = v_a, a \in TopB\}$
  if $\arg\max(|[x]_{topB}|) > \varepsilon$ **then**
    $B \longleftarrow topB$
if $B - CORE \neq \emptyset$ **then**
  $B \longleftarrow B - CORE$
Return B

Algorithm 22.1. Algorithm 22.2 is designed to compute the most dominant attribute subset selection.

The rough set exploration system (RSES version 2.2) [3] is used to calculate reducts of the universe. Then *CORE* can be derived from intersection of all reducts. Given a decision table (temporary), we find the N most dominant attributes toward the decision class. *CORE* is used to preserve the specific feature(s) of instances in the granule by retaining *CORE* until the latest granulations. N can be tuned up to the number of condition attributes to compose a concept. In other words, our algorithm allows a flexible number of attributes in a subset for partitioning. We use co-occurrence counting of attributes' values and decision classes to determine the domination degree. Once the most N dominant attributes are obtained, we determine the co-occurrences within the N attributes to find if any combination of them can be used to approximate a concept by threshold ε. A count of co-occurrence among condition attributes' values implies the degree of which these attribute values can be used to compose a common concept. We tune the ε by the number of instances in working granule. The subset of attributes with the greatest domination degree, and the greatest domination degree is greater than the threshold, is selected to partition the current granule. If no domination degree of the N combination attributes meets the threshold ε, the single most dominant attribute is selected.

Example 2 illustrates the recursive construction of a GCH using Algorithm 22.1 and Algorithm 22.2.

Example 2. A GCH construction for the Flu diagnosis decision table (Table 22.1) is described step by step. The granulation starts by partitioning the universe (Table 22.1). In this example, the equivalence relation is used. The size of attribute subset to partition is one ($N = 1$) since the number of condition attributes is relatively small. The objects in the universe are discernible by the equivalence relation. Thus, we find reducts for this table which are, $\{$*Temperature*, *Headache*, *Nausea*$\}$, $\{$*Temperature*, *Nausea*, *Cough*$\}$, and $\{$*Headache*, *Nausea*, *Cough*$\}$, and core is $\{$*Nausea*$\}$. There is no common attribute value in this granule. We select the first attribute subset by determining the degree of attribute dominations. *Headache* has the highest domination degree ($domDegree = 3$) compared with the rest of the condition attributes ($domDegree = 2$). Thus, the first attribute subset to partition is $\{Headache\}$ and $g_1 = \{1,2,4,5\}$ and $g_2 = \{3,6\}$ are obtained. Then we continue granulate g_1 selecting the most dominant attributes for g_1. *Temperature*,*Nausea* and *Cough* attributes have the same degree of domination. *Nausea* is the core; thus, it is retained at this granulation. We can select *Temperature* or *Cough* to partition g_1. If we apply *Temperature*, we obtain granule $g_3 = \{1,4\}$, $g_4 = \{5\}$, $g_5 = \{2\}$ which are children of g_1. The granule g_4 and g_5 are indiscernible so they are leaf granule. We then granulate g_3 by finding common attribute subset which is $\{Cough\}$. The *Cough* attribute can be now removed. The remaining attribute $\{Nausea\}$ is then used to partition g_3 to obtain $g_6 = \{2\}, g_7 = \{3\}$. Since all siblings are now leaf nodes we can return to the higher levels. We continue granulate g_2. Note that the temporary table can be generated as the common attribute $\{Headache\}$ is removed. Like partitioning g_1, *Nausea* is retained. If we partition g_2 by *Temperature*, the indiscernible granule $g_8 = \{3\}$ and $g_9 = \{6\}$ are obtained. Fig. 22.1 shows the GCH for the Flu diagnosis domain.

22.4.3 Mapping for Appropriate Granularity in a GCH

Our approach of GCH does not only provide a multilevel of granular concept representation of variables, but also enables searching and evaluating techniques for a granular variable. In order to solve a problem, an application can perform a search in the GCH for an *appropriate* level of granularity. As a result, the *appropriate* level of granularity is evaluated by sufficient knowledge for solving a particular problem. If the problem is to obtain decision rules to diagnose unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them have smaller boundary regions. If the problem is to predict missing values of condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. Searching for a granular concept in a GCH can be achieved through several techniques such as the depth first search and breadth first search.

Table 22.4. Higher-order rules for the Zoo database

Rules	Accuracy	Coverage
Rule 1: IF animals have the same values in condition attribute $\{C,E,J,K\}$ THEN they are in the same class of $\{R\}$	0.956	0.979
Rule 2: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{B,D,F,H,I,M,O,P\}$ THEN they are in the same class of $\{R\}$	0.972	0.209
Rule 3: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{D,L,M,P\}$ and animals have the same values in condition attributes $\{B,F,G,I,N,O\}$ THEN they are in the same class of $\{R\}$	1.0	0.266
Rule 4: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{D,P\}$ and animals have the same values in condition attributes $\{G,I,M,Q\}$ THEN they are in the same class of $\{R\}$	0.961	0.293
Rule 5: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{F\}$ and animals have the same values in condition attributes $\{B,D,M,O\}$ THEN they are in the same class of $\{R\}$	0.988	0.562
Rule 6: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{B,D,I,L,M,N,O\}$ THEN they are in the same class of $\{R\}$	0.998	0.534

G1: {squirrel, fruitbat, vampire, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, dolphin, porpoise, wallaby, sealion, platypus, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl, calf, goat, pony, reindeer, pussycat}
    G11: {fruitbat, vampire}
    G12: {platypus}
    G13: {squirrel, hare, vole, mole, opossum, cavy, hamster seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer elephant ,giraffe, oryx , boar ,cheetah ,leopard ,lion ,lynx ,mongoose, polecat, puma ,raccoon, wolf, mink, girl, dolphin, porpoise, calf, goat, pony, reindeer}
        G131: {dolphin, porpoise}
        G132: {squirrel, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl calf, goat, pony, reindeer}
            G1321: {squirrel, gorilla, wallaby, girl, hare, vole, cavy, hamster, antelope, buffalo, deer, elephant, giraffe, oryx, calf, goat, pony, reindeer, mole, opossum, aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose polecat, puma, raccoon, wolf, pussycat}
                G13211: {hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx, mole, opossum aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
                    G132111: {mole, opossum, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
                    G132112: {aardvark, bear}
                    G132113:{hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx}
                G13212: {squirrel, gorilla, wallaby}
                G13213: {girl}
                G13214: {cavy, hamster, calf, goat, pony, reindeer, pussycat}
                    G1321411: {cavy}
                    G1321412: {hamster calf goat pony reindeer}
                    G1321413: {pussycat}
            G1322: {seal, sealion}
            G1323: {mink}
G2: {chicken, dove, duck, lark, parakeet, pheasant, sparrow, wren, kiwi, crow, gull, hawk, skimmer, skua, ostrich, flamingo, swan, penguin, rhea, vulture}
    G21: {lark, pheasant, sparrow, wren, duck, kiwi, crow, hawk, gull, skimmer, skua}
        G211: {kiwi}
        G212: {lark, pheasant, sparrow, wren, duck, crow, hawk, gull, skimmer, skua}
            G2121: {lark, pheasant, sparrow, wren, crow, hawk}
                G21211: {lark, pheasant, sparrow, wren}
                G21212: {crow, hawk}
            G2122: {duck, gull, skimmer, skua}
                G21221: {duck}
                G21222: {gull, skimmer, skua}
    G22: {chicken, dove, parakeet}
    G23: {ostrich, flamingo, swan, rhea, vulture, penguin}
        G231: {ostrich, flamingo, rhea, vulture}
            G2311: {ostrich, rhea}
                G23111: {ostrich}
                G23112: {rhea}
            G2312: {flamingo, vulture}
                G23121: {flamingo}
                G23122: {vulture}
        G232: {swan penguin}
            G2321: {swan}
            G2322: {penguin}
G3: {pitviper, seasnake, slowworm, tortoise, tuatara, carp, haddock, seahorse, sole, bass, catfish, chub, dogfish, herring, pike, piranha, stingray, tuna, frog, frog, newt, toad}
    G31: {bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna}
        G311: {carp, haddock, seahorse, sole}
        G312: {bass, catfish, chub, herring, piranha, dogfish, pike, tuna}
            G3121: {bass, catfish, chub, herring, piranha}
            G3122: {dogfish, pike, tuna}
        G313: {stingray}
    G32: {pitviper, slowworm, tortoise, tuatara}
    G33: {seasnake}
    G34: {frog, frog, newt, toad}
        G341: {frog, newt}
            G3411: {frog}
            G3412: {newt}
        G342: {frog}
        G343: {toad}
G4: {flea, termite, gnat, honeybee, housefly, ladybird, moth, wasp, clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish}
    G41: {clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish, flea, termite}
        G411: {seawasp, starfish, flea, termite, slug, worm, crab, crayfish, lobster, clam}
            G4111: {flea, termite, slug, worm}
                G41111: {flea, termite}
                G41112: {slug worm}
            G4112: {seawasp}
            G4113: {starfish, crab, crayfish, lobster, clam}
                G41131: {clam}
                G41132: {starfish, crab, crayfish, lobster}
                    G411321: {crab}
                    G411322: {starfish}
                    G411321: {crayfish, lobster}
        G412: {scorpion}
        G413: {octopus}
    G42: {gnat, ladybird}
        G421: {gnat}
        G422: {ladybirdl}
    G43: {housefly, moth, wasp}
        G431: {housefly, moth}
        G432: {wasp}
    G44: {honeybee}

Fig. 22.2. A granular concept hierarchy for the Zoo data set

22.5 Evaluation

We evaluate the usefulness of a GCH through higher-order decision rules learning. The definition of higher-order rules are introduced by Yao [26]. A higher order rule expresses connections of different objects based on their attribute values. An example of a higher-order rule is ”*if object x is related to object y with respect to an attribute set a, then x is related to y to another attribute set b.*” Yao recommends a mining of higher-order rules from a transformed decision table, where an entity is a pair of objects from the original table. However, transforming the n objects table generates $\frac{n!}{((n-2)!*2!)}$ pairs of objects. We present an alternative approach to extract higher-order decision rules from a GCH where no transformation process is required.

The data set used in the experiment is the Zoo database from the UCI machine learning repository. This database contains 101 objects, 17 condition attributes and one decision attribute. The condition attributes include 16 boolean-valued attributes and a numerical attribute. The decision attribute contains 7 classes of animal type. There is no missing value in this datas et. We construct a GCH for the Zoo data set as shown in Fig 22.2.

There can be several groups of animals that hold the same attributes' values in a subset of condition attributes. For example, there are 6 groups of animals clustered by attribute set $\{C,E,J,K\}$ which are Feathers, Milk, Backbone, Breathes respectively. These attributes draw a concept of *mammal* when $C=0, E=1, J=1$, and $K=1$. The concept of *bird* is drawn when $C=1, E=0, J=1$, and $K=1$, the concept of *amphibia* is formed by $C=0, E=0, J=1$, and $K=1$. The *arthropod* (bug) concept is formed by $C=0, E=0, J=0$, and $K=1$. The concept of *fish* is formed by $C=0, E=0, J=1$, and $K=0$. The concept of being *crustacean* is formed when $C=0, E=0, J=0$, and $K=0$. Note that, these groups will be granulated until all the member of the group are indiscernible. The concept descriptions of the animal groups are used to generate the higher-order rules. Once the hierarchy is obtained, a depth first tree search is performed to find the maximum level of accuracy of each branch. The higher-order rules are obtained from conjunctive connection of granular concepts' intensions along the visited branches. The extracted higher-order decision rule set for the Zoo data base is given in the first column in Table 22.3. Number of conjunction shows the level of hierarchy, starting from level 0 at the root. The higher-order rules are applied to the total of 5,050 pairs of animals, and there are 1,177 pairs of animals that belong to the same class. We measure the rules' accuracy and coverage which were used by [22] as follows:

$$accuracy(premise \Rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(premise)|}, \quad (22.8)$$

$$coverage(premise \Rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(conclusion)|}, \quad (22.9)$$

where $\phi(g)$ is the granule's extension, and $|x|$ denotes the cardinality of the set x. The results of the rules' accuracy and coverage are shown in the second and the third column of Table 22.4, respectively.

We shall discuss the interestingness of the higher-order decision rules as follows. The higher-order rule is the type of knowledge in more abstract level. This knowledge should be firstly applied to solve a problem. Naturally, given two animals, one can differentiate them by the concepts, not by the detailed of each attribute value if not necessary. The higher-order rules provide the concepts upon the domain which the rules can be applied for only some groups. The rules obtained from our approach have much higher accuracy degree than the coverage degree. This is because of the tree traversal searches for the maximum accurate level of each branch, where their children do not have smaller boundary regions than the parents. Once the target granules are found, the granules' language can be used to express the connections between objects in the same granule directly. The connections are multi-dimension which reflect the relationships between attributes in the attribute subset (*e.g.*, dependencies) and also the relationships between the condition attribute subset and the decision attribute. On the other hand, if one prefers the rule with higher coverage degree, the bread first search for the coarser granules can be achieved.

22.6 Conclusion

An approach to automatically construct a GCH from a decision table is presented. A GCH represents knowledge in different level of specificness/coarseness. A granular concept is formally defined for its syntax, semantic, and interpretation. With this rich information, an application can map from a granular concept that conveys sufficient information to solve a problem. The usefulness of the GCH is shown from the ability to extract higher-order rules from the GCH structure without postprocessing required. Extensions of this work include granular concept mapping based on a conceptual network of a domain for real world applications such as the educational and instructional area.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing. An introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centered information processing. IEEE Transactions on Fuzzy Systems 16(2), 320–330 (2008)
3. Bazan, J.G., Szczuka, M.S.: The Rough Set Exploration System. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)

4. Biswas, G., Weinberg, J.B., Fisher, D.H.: ITERATE: a conceptual clustering algorithm for data mining. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 28(2), 219–230 (1998)
5. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
6. Grzymała-Busse, J.W., Siddhaye, S.: Rough set approaches to rule induction from incomplete data. In: Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2004, Perugia, Italy, July 4-9, vol. 2, pp. 923–930 (2004)
7. Hoa, N.S., Son, N.H.: Rough set approach to approximation of concepts from taxonomy. In: Proc. of Knowledge Discovery and Ontologies Workshop, KDO 2004 at ECML/PKDD 2004, pp. 13–24 (2004)
8. Hu, X., Cercone, N.: Learning in relational databases: A rough set approach. J. Computational Intelligence 2, 323–337 (1995)
9. Li, J., Cercone, N.J.: A Rough Set Based Model to Rank the Importance of Association Rules. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005, Part II. LNCS (LNAI), vol. 3642, pp. 109–118. Springer, Heidelberg (2005)
10. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
11. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
12. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
13. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Sciences 177(1), 28–40 (2007)
14. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177(1), 41–73 (2007)
15. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley (2008)
16. Pedrycz, W., Vukovich, G.: Granular computing with shadowed sets. International Journal of Intelligent Systems 17, 173–197 (2002)
17. Pedrycz, W.: Shadowed sets: Representing and processing fuzzy sets. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 28(1), 103–109 (1998)
18. Peters, J.: Approximation and perception in ethology-based reinforcement learning. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Granular Computing, pp. 671–688. John Wiley & Sons, Ltd., Chichester (2008)
19. Peters, J.: Near Sets. special Theory about Nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
20. Priss, U.: Formal Concept Analysis in Information Science. In: Cronin, B. (ed.) Annual Review of Information Science and Technology, vol. 40, pp. 521–543. Information Today, Inc. on behalf of ASIS&T, Medford (2006)
21. Skowron, A., Peters, J.: Rough-granular computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Granular Computing, pp. 225–328. John Wiley & Sons, Ltd., Chichester (2008)
22. Tsumoto, S., Tanaka, H.: Discovery of Approximate Medical Knowledge Based on Rough Set Model. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 468–476. Springer, Heidelberg (1998)
23. Yao, Y.Y.: Granular computing using neighborhood systems. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) Advances in Soft Computing: Engineering Design and Manufacturing, The 3rd On-line World Conference on Soft Computing, WSC3, June 21-30, pp. 539–553. Springer, London (1999)

24. Yao, Y.Y.: Information Granulation and Rough Set Approximation. International Journal of Information Systems 16, 87–104 (2001)
25. Yao, Y. Y., Yao, J.T.: Granular Computing as a Basis for Consistent Classification Problems. In: Proceedings of PAKDD 2002 Workshop on Toward the Foundation of Data Mining, pp. 101–106 (2002)
26. Yao, Y.Y.: Mining high order decision rules. In: Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.) Rough Set Theory and Granular Computing, pp. 125–135. Springer, Berlin (2003)
27. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)