What is Granular Computing?
Ancient Practices, Partial Theories and Future Directions

Tsau Young (T. Y.) Lin
Department of Computer Science
San Jose, California 95192, USA
tylin@cs.sjsu.edu

Article Outline

Glossary
I. Definition
II. Introduction
III. Classical Examples
IV. Formal Granular Models
V. Two Dual Operators
VI. Three Semantic Views
VII. Quotient Structures
VIII. Future Directions
Glossary

Granulation: Granulation is an operation or a process of forming granules, with a granule being a collection of objects (points) that are drawn together by some constraints, such as indistinguishability, similarity or functionality.

Granular Structure: Granular Structure is the collection of granules, in which the internal structure of each granule is visible. Informally speaking, granular structure is a collection of white box granules.

Quotient Structure: A quotient structure is the mathematical structure of the collection of granules, in which each granule is regarded as an element (point) of a set, but the interactions among granules are preserved. Informally speaking, quotient structure is a collection of black box granules.

The collection of \{... − 2, 0, 2,...\} and \{... − 3, 1, 3,...\} is a granular structure. Let E be the first subset (even integers) and O be second subset (the odd integers.) Then the collection of E and O is the quotient structure.

Neighborhood System (Local Granular Model): A domain of interests (a classical set) \( U \) is called the universe. To each point \( p \) in the universe, a family of subsets is assigned. Such a family (could be empty) and each such a subset is called a neighborhood system \( NS(p) \) at \( p \) and a neighborhood at \( p \) respectively. The collection \( \beta \) of such a family at every point of the universe is called a neighborhood system \( NS(U) \) of the universe. Neighborhood and neighborhood system are pre-GrC language; in granular computing, they are called granule and the granular structure respectively. The pair \( (U, \beta) \) is called a local granular model, since each granule is associated with some points.

Topological Neighborhood System: A neighborhood system is called a topological neighborhood system, if it satisfies the axioms of topology.

Binary Neighborhood System (Binary Granular Model): A binary neighborhood system is a neighborhood system defined by a binary relation \( R \). A (right) neighborhood is defined as follows: \( B(p) = \{x \mid (p, x) \in R\} \). The collection \( B(p) \) at each \( p \) is the (right) binary neighborhood system. Similarly, we can define left version: A left neighborhood system \( C \) is defined by the \( C(p) = \{x \mid (x, p) \in R\} \) at every point \( p \). Note that the right and left neighborhood systems determine each other. The pair \( (U, \beta) \) is called a binary granular model, where \( \beta \) is B, C or R.

Bag: A bag is similar to a set, but allows an element to appear more than once. For example \( \{1, 2, 1, 2, 1, \} \) is a bag, but not a set. If a bag contains \( n \) elements, we may say it is a \( n \)-bag. For example, previous bag is 5-bag.

Relational Structure (Relational Granular Model): A set of classical sets is called a universe and denoted by \( U \). A Cartesian product of a \( n \)-bag of \( U \) is called a \( n \)-product
set. A n-ary relation is a subset of a n-product set. A collection $\beta$ of n-ary relations ($n$ could very) is called a relational structure. The pair $(U, \beta)$ is called a Relational Granular Model. Note that this relational structure is the relational structure (without functions) in the First Order Logic.

**Partial Covering (Global Granular Model)**: Let $U$ be a classical set, called the universe. Let $\beta = \{F^1, F^2, \ldots\}$ be a family of subsets. Then the pair $(U, \beta)$ is called Global Granular Model.

**Equivalence Relation**: A binary relation $R$ is called an equivalence relation, if it has the following properties: Let $u, v$ and $w$ be elements of $U$.

- reflexive: $u R u$
- symmetric: $u R v$ implies $v R u$
- transitive: $u R v$ and $u R w$ implies $u R w$

**Partition**: A partition $P$ of a classical set $U$ is a collection of subsets that are mutually disjoint and their union is $U$. Each subset is called an equivalence class. This name is derived from the fact that partition is equivalent to the following equivalence relation: We define $u R v$, if and only if $u$ and $v$ belongs to the same equivalence class. Such $R$ is the equivalence relation corresponding to the partition $P$. Note that a partition is a special type of granular structure, so an equivalence class is a special granule.

1 **Definition of the Subject and Its Importance**

Granular Computing (GrC) is still in its inception stage, we use motivation as its statements of importance.

*How Important is Granular Computing?* Granulation seems to be a natural methodology deeply rooted in human thinking. Many daily "things" are routinely granulated into sub"things;" human body has been granulated into head, neck, and so forth. The notion is intrinsically fuzzy, vague and imprecise. To formalize it is difficult, mathematicians idealized/simplified it into the notion of partitions (=equivalence relations), and have developed it into a fundamental part of mathematics, e.g., congruence in Euclidean geometry, quotient groups, quotient rings, and etc in algebra. Nevertheless, the notion of partitions (see glossary), which absolutely does not permit any overlapping among its granules, seems to be too restrictive for real world problems. Even in natural science, classification does permit small degree of overlapping; there are beings that are both appropriate subjects of zoology and botany. So a more general theory, namely, Granular Computing (GrC) is needed.

*What is Granular Computing (GrC)?* There is no formal definition yet; we will not offer a hasty one. We believe in incremental developments. Its development, though not as glorious, may be similar to that of classical geometry. In the history of geometry, various geometries, e.g., Euclidean, hyperbolic, elliptic geometries, appeared first by its own right. Then these
theories converge to the unified formal theory in Kleins Erlangen program. So we present here some piecemeal definitions, theories and directions of GrC from various angles: We will describe this subject by explaining: What is the intended concept, What has been formalized, What are the key constituents, What are the appropriate interpretations/semantic views, Associated Concepts, and Applications.

1) The intended concept is defined by an Inductive "Definition:" Granular Computing is a computing theory or mathematics on the concept of granulation, which, at this point, can only be "defined" implicitly by a set of classical examples; see Section 3.

   In essence, GrC is a label to denote a set of common, even ancient, concepts and practices.

2) A system of Formal Models that realizes nearly all examples, is presented; see Section 4. It is a system of ”convenient models” in the sense that they can be derived from a more general model, but for convenient, they are modeled independently on its own rights. Unfortunately, the uncovered example represents a vast class of common practices.

2) Two Dual Operators: In granulating a problem, a dual action, namely, Integrating sub-solutions is invoked. They are dual to each other. The integration, which may be related to the Information Integration in database, has been neglected; see Section 5. Here, we note some unusual observations.

   1. Recursive granulation, without due course, may take Np hard time (or never) to reach the bottom.

   2. Integrating a set of given sub-solutions may lead to more than one unified solutions. This is an unexplored problem, we use the EXTENSION Functor in Homological Algebra to illustrate the nature of the problem.

3) Three Semantics Views: Granules may be interpreted from

   1. Uncertainty Theory: A granule is a unit of lacking precise knowledge. Some lessons may be learned from the uncertainty in quantum mechanics.

   2. Knowledge Engineering: A granule is a unit of Basic Knowledge (Information).

   3. Principle of Computing (How-to-Solve/Compute-it): A granule is a sub-problem or software unit. It is a special type of basic knowledge.

   Each view may have its own GrC theory; For example, Concept Approximations are useful in Second View, while Information Hiding is in Third View; see Section 6.

4) The classical examples represent some application. New Applications in Computer Security, Web Technology and others have been reported.

Working Formal Models are Illustrated: The most general model is expressed in category theory. However, for easy-to-understand purpose, we explain the Second GrC model first.
1) **Second GrC Model** Let $U$ be a classical set, called the universe. Let $\beta = \{F^1, F^2, \ldots \}$ be a family of subsets. Then the pair $(U, \beta)$, called Global Granular Model, is the formal definition of Second GrC Model.

To understand the next model, Table 1, that explains the generalization process, is helpful.

2) **Fifth GrC Model**: Suppose we have

1. $U = \{U^h_j, h, j, = 1, 2, \ldots \}$ is a family of classical sets, called the universe.
2. $U^j_1 \times U^j_2 \times \ldots$ is a Cartesian product of sets, $j = 1, 2, \ldots$.
3. An $n$-ary relation is a subset $R^j \subseteq U^j_1 \times U^j_2 \times \ldots U^j_n$.
4. $\beta = \{R^1, R^2, \ldots \}$ is a family of $n$-ary relations ($n$ could vary).

Then the pair $(U, \beta)$, called Relational Granular Model, is the formal definition of Fifth GrC Model.

In Second GrC model, each subset $F^j$ is what Zadeh defined as a *granule* that is a clump of objects which are drawn together by indistinguishability, similarity or functionality [20], [6]. Here, we have implicitly assumed that such drawing forces are symmetric (the order of the elements are irrelevant). In Fifth GrC Model, we have removed the symmetric assumption, so a granule is a tuple. Those tuples in the same $n$-ary relation are those granules that are accidentally drawn together by the same force. So each relation models/generalizes an ”indistinguishability, similarity or functionality” constraint.

Next, we can generalize the category of sets to a general Category; we skip the detail as it is straightforward.

## 2 Introduction

*What is Granular Computing (GrC)?* The best approach is to trace how the intuitions have been evolved. Or this purpose, let us recall the event. For convenience, we will speak as if we are the third person. In the academic year 1996-97, when Lin had his sabbatical leave at Berkeley, Zadeh suggested granular mathematics to be his research area. To limit the scope, Lin proposed the term granular computing [21]. So, at the beginning, roughly it is the computable
part of granular mathematics.

What is granular mathematics (GrM)? Zadeh in his 1979 paper [19] had implicitly explained his view. Here, we take a simple approach: It is a new mathematics, in which some "points" are replaced by "granules." We call this process, namely, selecting the granules, the granulation. The collection of these granules is called the granular structure

What is a granule or a granulation? There is an obvious candidate, partition of a set $U$, which is defined mathematically as a collection of subsets that are mutually disjoint and their union is the total set $U$. This geometric concept of a partition is equivalence to an algebraic notion equivalence relation. Namely, two elements are said to be equivalent if and only if they are in the same subset of a partition. So these subsets have been called equivalence classes in mathematics. For general granulation, there is no formalized definition yet, we will give some examples.

3 Classical Examples

- Ancient Examples
  1. Granulation of human body: head, neck, body, hand and etc
  2. Granulation of Space and Times: Granules of (intuitive) Infinitesimals that led to the invention of Calculus

No theory for the first example; the obvious fuzzy models do not work well. There are two established theories for the second example, namely, non-standard analysis [16] and topology [12].

- Modern Examples
  3. Local Granules of Uncertainty - from Quantum Mechanics
  4. Local Granules of Basic Knowledge - the access list for each file in the discretionary access control models in Computer Security
  5. Global Granules of Knowledge - Simplicial complexes in combinatorial topology. Simplexes are the given basic knowledge on the collection of vertices.

These are "serious" examples: The first example is from natural science. The second example is a common data structure in many computer systems. Both versions are in local form, namely, the granules (including the degenerated cases) are associated to each point. The third one is a global version. It is a common mathematical structure in combinatorial topology which is finding its way to web technology [4].

- Further Examples
Fuzzy sets, as generalizations of classical sets, naturally can be viewed as granules, and so we have generalized further

6 A granule can be a membership function, a function or even a generalized function (such as Dirac delta function).

Traditionally, "How to solve it" [15] has not been any part of formal mathematics, however, "how to compute" is an integral part of computing. So it should be included into GrC.

7 A granule can be a sub-Turing machine.

4 Formal Granular Models

Based on these examples, eight models are discussed. They are basically "convenient models" in the sense that they can be derived from a more general model, but for convenient, they are modeled independently.

4.1 Crisp/Fuzzy Set Based Models

4.1.1 Models for the Ancient Examples

Probably the most lively example is the early notion of infinitesimal granules, which led to the invention of Calculus by Newton and Leibniz. Actually the idea was much more ancient; it was in the mind of Archimedes, Zeno, and etc. Yet the solutions were relatively modern. Two formalizations had emerged. One was the concept of limit (19th century) that led to the notion of topology. The other one was the non-standard analysis, which formally realized the original intuition (20th century). Intuitively,

- The modern notion of the infinitesimal granules in the non standard world is equivalent to the that of Topological Neighborhood System (TNS) in standard world.

The notion of topology can be defined in two ways: (1) A topology $\tau$ is a family of subsets, called open sets, that satisfies the (global version) axioms of topology. (2) A topology, called topological neighborhood system (TNS), is an assignment that associate each point $p$ a family of subsets, TNS($p$), that satisfies the (local version) axioms of topology. These two definitions led to two formal GrC models: In 1988-89, from different context, namely, approximate retrieval and computer security, Lin generalized TNS to the Neighborhood Systems(NS) by simply dropping the (local) axioms of topology [1], [3], [2]. Each neighborhood was treated as a unit of uncertainty. By mapping NS onto Zadehs intuitive statements, Lin used NS as his first mathematical GrC model. The model $(V, U, B, C)$ is called Granular Structure, where $B$ is a NS and $C$ is the family of meaningful names of those neighborhoods [6], [7], [8]. The term pre-topology is also used to address these computer/mathematical systems, because NS and etc have the same framework as topology, except the topological axioms.
**Definition 1** First GrC Model: The Pair \((U, \beta)\), where \(\beta\) is a NS, is called Local GrC Model.

This is a special case of a granular structure, where \(U = V\) and \(C = \emptyset\).

A fuzzy set can be regarded as a Local Granular Model by suitably considering sub-collections of \(\alpha\)-cuts [18] as neighborhood system. For example, points outside of the support have empty set as its NS, while points in core (membership value=1) have whole \(\alpha\)-cuts as its NS.

Again by dropping the global axioms of topology, we have Second GrC model. Note that Second GrC model is a special case of First GrC model: Each \(p\) may regard the collection of members in \(\beta\), that contains \(p\), as its neighborhood system.

**Definition 2** Second GrC Model: The Pair \((U, \beta)\), where \(\beta\) is a family of subsets of \(U\), is called Global GrC Model. This family a partial covering.

### 4.1.2 Models for the Modern Examples

Let us start with the third example, simplicial complex [17]. It consists of a set of vertices and a family of subsets, called simplexes, that satisfies the closed condition. So simplicial complex is an example of Second GrC Model. Perhaps, it is helpful to note that

- the closed condition of simplicial complex is the apriori principle in association (rules) mining.

This observation seems to imply that simplicial complex (hence GrC) is a natural notion in data mining. Next, let us turn to the first example. In quantum mechanics, there is the well known principle of uncertainty. Two observables \(A\) and \(B\) cannot be simultaneously be observed with accuracy. For position and momentum observables, the uncertainty is \((\delta A)(\delta B) \geq 1/2\). This inequality defines a binary relation (BR) on \(\delta A\) and \(\delta B\). In other words, the two quantities of uncertainty are related by a binary relation. In computer security, the Discretionary Access Control Model (DAC) assigns to each user \(X\) a family of users, \(Y_i, i = 1, \ldots\), who can access \(X\)’s data. In other words, each \(X\) is assigned a granule of friends. In abstraction, each \(X\) is assigned a subset, \(B(X) = \{Y_i, i = 1, \ldots\}\), of “Basic knowledge” (a set of friends). Such a set \(B(X)\) is called a (right) binary neighborhood and the collection \(\{B(X) | \forall X \in U\}\) the binary neighborhood system (BNS). So DAC has a BNS structure, which is equivalent to a binary relation(BR):

\[ BR = \{(X,Y) \mid Y \in B(X) \text{ and } X \in V\}. \]

Conversely, a binary relation defines a BNS as follows:

\[ X \rightarrow B(X) = \{Y \mid (X,Y) \in B\} \]

So both modern examples give rise to the BNS, which was called a binary granular structure in [6]. We would like to note that based on this (right) BNS, the (left) BNS can also be defined:
\[ C(X) = \{ Y \mid X \in B(Y) \} \text{ for all } X \in V. \]

Note that BNS is a special case of NS, namely, it is the case when the collection \( \text{NS}(p) \) is a singleton that is, \( B(p) \) alone. So the Third GrC Model defined below is a special case of First GrC Model. The algebraic notion, binary relations, in computer science, is often represented geometrically as graphs, networks, forest and etc. So the following model has captured most of the mathematical structure in computer science.

**Definition 3** Third GrC Model: The Pair \((U, \beta)\), where \( \beta \) is a BNS, is called a Binary Granular Model.

Next, instead of a single binary relation, we consider the case \( \beta \) is a set of binary relations. It was called a [binary] knowledge base [6]. Such a collection naturally defines a NS. So a Fourth Model induces or maps, say by \( g \), to a First Model. On the other hand, a First Model may also induce or map, say by \( f \), to a Fourth Model. So First and fourth models are equivalent, but \( g \) and \( f \) are not the converses to each other.

**Definition 4** Fourth GrC Model: the Pair \((U, \beta)\), where \( \beta \) is a set of binary relations, is called Multi-Binary Granular Model.

Next, we note a very interesting case. A subset can be viewed as a very special type of equivalence relation, namely, its partition is the subset and its compliment. So Second GrC Model, as is about a set of very special type of equivalence relations, is a special form of Fourth GrC Model.

### 4.2 Logic/Relation Based Models

In [20] Lotfi Zadeh assert: . . . a granule being a clump of objects (points) which are drawn together by indistinguishability, similarity or functionality.” The phrase ”drawn together” (n objects) is a tuple in an n-ary relation. Those tuples in the same relation are drawn by the same force (constraint). Note that this is closely related to logic. In fact Fifth Models is the relational structure (without functions) in First Order Logic. Each granule (tuple)may represent a ”committee” (un-homogeneous roles) in social network.

**Definition 5** Fifth GrC Model:

1. \( U = \{ U^h_j, h, j, = 1, 2, \ldots \} \) is a family of classical sets, called the family of the universes.
2. \( U^j_1 \times U^j_2 \times \ldots \) is a Cartesian product of sets, \( j = 1, 2, \ldots \).
3. An n-ary relation is a subset \( R^j \subseteq U^j_1 \times U^j_2 \times \ldots U^j_n \).
4. \( \beta = \{ R^1, R^2, \ldots \} \) be a family of n-ary relations (n could vary).

The pair \((U, \beta)\), called Relational Granular Model, is a formal definition of Fifth GrC Model.
4.3 Function/Algorithm Based Models

Now, we consider the case granules are functions, we normally require the collection of granules(functions) to have the universal approximation property, namely, any function in the universe can be approximated by the functions in the collections. The membership functions selected in fuzzy controls do have such properties. In neural networks, the functions generated by the activation functions also have such property [13]

Definition 6 Sixth GrC Model: Granules can be membership functions, functions and even generalized functions.

Definition 7 Seventh GrC Model: Granules are Turing machines

4.4 Category Theory Based Models

Now we generalize the category of sets to general Category that includes functional and algorithmic GrC.

Let us set up some language of Category Theory. A category consist of

1. A class of objects, and

2. A set Mor(X, Y) of morphisms for every ordered pair of objects X and Y, which satisfies certain properties. For this paper, the formal details are not important; we only need the language loosely.

Here are some examples.

1. The Category of Sets: The objects are classical sets. The morphisms are the maps.

2. The Category of Power Sets: The object \( U_X \) is the power set \( P(X) \) of a classical set \( X \). Let \( U_Y \) be another object, where \( Y \) is another classical set. The morphisms are the maps, \( P(f) : U_X \rightarrow U_Y \) that are induced by maps \( f : X \rightarrow Y \).

Let \( \text{CAT} \) be a given category.

Definition 8 Final GrC Model:

1. \( \mathcal{C} = \{ C^h_j, h, j = 1, 2, \ldots \} \) is a family of objects in the Category \( \text{CAT} \).

2. \( C^1_j \times C^2_j \times \ldots \) is a product of objects, \( j = 1, 2, \ldots \).

3. An n-ary relation object \( R^j \) is a sub-object of \( C^1_j \times C^2_j \times \ldots C^n_j \).

4. \( \beta = \{ R^1, R^2, \ldots \} \) be a family of n-ary relations (n could vary).

The pair \( (\mathcal{C}, \beta) \), called Categorical Granular Model, is a formal definition of Final GrC Model
4.5 Overview of Early GrC Models

Schematically we summarize the relationships based on Granular Structure as follows:
“⇒⇐” is a two way generalization but they are not inverse to each other.
“⇒, ⇑, and ⇓” are one way generalizations.
“GM” means Granular Models.

\[
\begin{array}{c}
\text{Global GM} 
\Rightarrow 
\text{Local GM} \\
\downarrow \\
\text{Multi Binary GM} 
\Rightarrow 
\Leftarrow 
\text{Local GM} 
\downarrow \\
\uparrow \\
\text{Binary GM} 
\Rightarrow 
\text{Relation GM} \\
\end{array}
\]

5 The Dual Operator: Integration

Granulation has been addressed in previous two sections. In each granulation of a problem, a dual action, namely, Integration of sub-solutions is invoked. They are dual to each other. The integration, which may be related to the Information Integration in database, has not been explored. We will use the extension function to illustrate the nature of the problems.

1) Integration without additional information structure.

Let the universe \( U \) be the set \( Z = \{\ldots, 1, 0, 1, \ldots\} \) of integers. The \( U \) has been decomposed into two sub-problems, namely, \( \{\ldots -2,0,2,\ldots\} \) and \( \{\ldots -3,1,3,\ldots\} \). Let \( E \) and \( O \) denote the first and the second subsets. Then,

1. The following collection

\[ \{ \ldots -2,0,2,\ldots, \ldots -3,1,3,\ldots \} \]

is called granular structure. Informally, it is a collection of white boxes, where the internal of the subsets are visible. As the two subsets are mutually disjoint, so the collection is a classical set. Note that if the subsets were non-disjoint, then the mathematical structure of collection may not be a classical set.

2. The collection \( Q = \{E, O\} \) is called quotient structure. Informally it is a set of black boxes, where the internal of the subsets are invisible (information hiding).

From the point of view of problem solving,

- the quotient structure represents the recipe (higher level instructions) of how to put the solutions of sub-problems into the total solution. The quotient set is the "Main Program" that consists of a set of sub-program calls. Intuitively \( Q \) represent high level knowledge
Let \( \text{Int}(E) \) denote the internal structure of \( E \), that is, \( \text{Int}(E) \) is viewed as a set by itself, forgetting it as a subset of \( U \). In this case, \( \text{Int}(E) = \text{Int}(O) = \mathbb{Z} \). Let us summarize what we are given

1. \( \text{Int}(E) = \text{Int}(O) = \mathbb{Z} \)
2. \( \text{Int}(E) \) and \( \text{Int}(O) \) are mapped to a partition \( E, O \) of an unknown \( U \).
3. This partition of unknown \( U \) has a \textit{known} quotient structure \( Q = \{ E, O \} \).

Schematically we are given the following situation:

\[
\begin{array}{c}
Z = \begin{cases}
\text{Int}(E) & \longrightarrow & E \\
\text{Int}(O) & \longrightarrow & O
\end{cases}
\end{array}
\quad \subseteq U \longrightarrow Q
\]

Can \( U \) be constructed back?

The answer is yes. It is the Cartesian product \( Q \times Z \). From set theoretical point of view, the constructed \( Q \times Z \) and the originally given \( U \) are the same set, that is, they are equivalent to each other in the setting of classical set theory.

2) Integration with additional information structure.

Let us consider a second view on the same universe (the set of integers.) But this time, the universe carries additional information, namely, the additive structure of integers, namely, the additive group \((\mathbb{Z}, +)\). This universe is denoted by \((U, +)\). Then

1. \( \text{Int}(E) \) is the additive group \((\mathbb{Z}, +)\) of integers, and \( \text{Int}(O) \) is a set \( \mathbb{Z} \) of integers.
2. The quotient structure \((Q, +) = \{ E, O; + \} \) is an additive group: \( E+E=E, E+O=O+E=O, O+O=E \). This \((Q, +)\) is often called the integer mod 2, and denoted by \((\mathbb{Z}_2, +)\).

Again, we are given a similar situation

\[
\begin{array}{c}
\{ (\mathbb{Z}, +) = \text{Int}(E) & \longrightarrow \text{homomorphism} & E \\
\mathbb{Z} = \text{Int}(O) & \longrightarrow \text{map} & O
\end{array}
\} \quad \subseteq (U, +) \longrightarrow (Q, +)
\]

Can \((U, +)\) be reconstructed back?

The answer is more than YES; there are \textit{two} solutions!! The two solutions are: \((\mathbb{Z}_2 \times \mathbb{Z}, +)\) and \((\mathbb{Z}, +)\). They are not the same as additive groups. This fact can be expressed by the extension functor, namely, \( \text{EXT}(\mathbb{Z}, \mathbb{Z}_2) \neq 0 \).

The important question is

Could such a functor be extended to formal granular models?
The answer is yes. There are positive and negative observations on binary granular models; for example, the completeness of a knowledge representation [10] implies that the integration is unique; the details will be in future report.

Finally, we would give few words of caution, the ”granulate and conquer” is quite different from ”divide and conquer.” For example,

- Granulate from the top could take Np-hard time or never to reach the bottom.

This issue is normally addressed in the topic ”dynamic programming” in standard data structure course. We have used ”topological divide” to solve the binary granulation [9].

6 Semantic Views

Granules may be interpreted from three views.

1. Uncertainty Theory: A granule is a unit of lacking precise knowledge. Some lessons may be learned from the uncertainty in quantum mechanics. Both L.A. Zadeh and T. Y. Lin, who coined the label, started from the uncertainty theory. Zadeh has a grand project [21].

2. Knowledge Engineering: A granule is a unit of Basic Knowledge (Information).

3. Principle of Computing (How-to-Solve/Compute-it): A granule is a sub-problem or software unit. It is a special type of basic knowledge.

Each view may have its own GrC theory: Some fundamental operators.

1) Information Hiding: It is a transformation of granular structures into quotient structures (see glossary).

A quotient structure is the mathematical structure of the collection of granules, in which each granule is regarded as an element(point), but the interactions among granules are kept and transformed into the interactions among elements. For example, in group theory a quotient group is a collection of cosets, in which each coset is regarded as an element and the multiplication of cosets is abstracted into the multiplication of elements in the quotient group. Using software engineering language, a granule in quotient structure is a black box, while in granular structure, it is a while box. These are easy case, for granulations that have non-empty overlappings, the quotient structure is not easy to determine; see Section ??.

2) Information Integration: This has been addressed in Section 5. Here we will be more interested in the integraton with additional information structures.

3 Concept Approximations- We view it as a way to approximate an unknown concept from the given Basic Knowledges. So in GrC both operations ”and (\(\cap\)) and ”or (\(\cup\)) are used. In RST, it uses ”or (\(\cup\)) only.
Let \( X \) be a variable that varies through all possible finite intersections of all the members in the granular structure.

1. Upper approximation: \( C[X] = \overline{\beta}[X] = \{ p : \forall p \in X \& \forall \beta(p) \cap X \neq \emptyset \} \).

2. Lower approximation: \( I[X] = \underline{\beta}[X] = \{ p : \forall p \in X \& \exists a \beta(p) \subseteq X \} \).

3. Closed set based notion: \([？] \) used closed closure operator. It applies closure operator repeatedly (transfinite many steps) until the resultants stop growing. The space is called Frechet(V)-space or (V)-space.

\[
Cl[X] = X \cup C[X] \cup C[C[X]] \cup C[C[C[X]]] \ldots \text{(transfinite)}.
\]

For such a closure, it is a closed set

This definition works for all models such as First, Second, Third, Fourth. In these models, we regard arbitrary subsets as new concepts, and the granules as basic knowledge. In Fifth, we consider any subset in a product space as a new concept, and the relations (subsets of product space) as basic knowledge. For functions, we regard an arbitrary function (think of it as a fuzzy set) as a new concept, and granular structures as basic knowledge (it is a Schauder base in functions space). It is unknown as how to define the concept approximation in general category and Turing machines(algorithmic).

4 Knowledge Representations: This amount to give each point in quotient structure a meaningful name, called it naming map. More precisely, Knowledge representation is a composition of Information hiding (a map from granular structure to quotient structure) and the naming map.

Knowledge representation is another way to discover new knowledge by organizing the basic knowledge in appropriate fashion, such as, information table (this is the same as relation instance or relational table in database theory). In GrC, the table has additional algebraic and topological structures. There are new concept that RST does not have.

7 Quotient Structure

An important structure associate to a Granular Model is the quotient structure. In granular model, a granule is a white box in the sense that the content of every granule is visible in the model. One can also view granules as points, where their contents are invisible in the model. For example, a partition is a collection of equivalence classes, where each equivalence class is a subset of the universe. On the other hand, one can regard the equivalence class as a point or an element of a set. Then the collection of such points is called quotient set. In general, if a granule in a granular model is abstracted to a point, and further the interactions among granules are abstracted to the interactions of points, then the collection of such points is called the Quotient Structure of the model.

Here is a theorem that illustrate the phenomena

**Theorem 1** The Quotient Structure of
1. a Second GrC model is a simplicial complex.

2. A RST is a classical set, called quotient set.

3. a Third GrC model is a BNS.

4. a First GrC model is a NS.

The second item of Theorem 1 is a classical result. Our expositions will focus on Item 3 and 4; they are various forms of (pre)topology. First, we observe that

Proposition The collection of complete inverse image of a given mapping \( f : U \rightarrow V \) defines a partition. Namely, the collection \( \{ f^{-1}(f(p)) \mid p \in U \} \) forms a partition.

So a pre-topology induce a partition on \( U \). This partition will be called the derived partition or induced partition; for the case of BR see [6]. Let \( B : U \rightarrow P(U) \) and \( N : U \rightarrow P(P(U)) \) be a BNS and NS respectively. Then the complete inverse images of B and N are partitions (equivalence relation) and are denoted by \( E_B \) and \( E_N \) respectively. So for BNS and NS, the derived partition gives rise a quotient set \( U/E_B, U/E_N \). The (pre-)topology BNS and NS of \( U \) naturally induce a (pre-)topology on the quotient set \( U/E_B \) and \( U/E_N \), [3], [5]. This explains the Item 3 and 4. The first item involves combinatorial topology [12]. The members of the partial covering \( \beta \) are regarded as abstract vertices. All subfamilies of partial covering, whose intersection is non-empty, are regarded as abstract simplexes. These collection of simplexes (vertices are simplexes of dimension 0) naturally form a simplicial complex.

Example 1 Let \((U, \beta)\) be a Granular Structure of a partial covering, where

1. the universe is: \( U = \{(1,0,0),(0,1,0),(0,0,1)\} \), where three points Euclidean space is denoted by \( x, y, z \) in this order.

2. Granular Structure is \( \beta = \{\{x\},\{y\},\{z\},\{x,y\},\{x,z\},\{y,z\}\} \), where the granules are denoted by \( a, b, \text{ and } c \) in the given order.

The Quotient Structure is a special granular model, called simplicial complex, that consists of

1. the set of vertices: \( V = \{a, b, c\} \) is the COV

2. the set of simplexes: \( X_1 = \{\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\}\} \) that satisfies the closed condition.

Note that these pairs, \( \{a,b\}, \{a,c\}, \{b,c\} \) exist because the intersections \( a \cap b, b \cap c, a \cap c \) are non-empty. By regarding \( \{a,b,c\} \) as abstract vertices, and the three pairs as the 1-simplexes (open segments.), the pair \( (V,X_1) \) forms a simplicial complex. It is a hollow triangle that consist of two vertices and three sides(without the interior of the triangle) This Quotient Structure implies that any two granules do have non-trivial interactions, but no three granules interact together.

Since granules in a partition have no intersections, so the Item 2 of the theorem do not have higher dimensional simplexes.
8 Future Directions

Granular Computing is a label of ancient practices. So we define it by a set of classical examples. Therefore the formalization of these examples are the most urgent task. We can nearly formalize all the examples, except

1) the very first example, which actually represents a vast category of ancient practices, namely, granulating daily "things" to sub"things." It seems an appropriate concept of qualitative fuzzy set is right direction.

1) the very first example, which actually represents a vast category of ancient practices, namely, granulating daily "things" to sub"things." It seems an appropriate concept of "qualitative fuzzy set, Which is one form of type II fuzzy set, is right direction.

2) Each granulation, in fact, invoke a dual action integration that integrates the sub-solutions into the total solution. It is unclear if many technique and concepts of Information Integration, which is currently a very active area in Database, may be useful in Integration. In this article, we use homological algebra to illustrate one of its important issues.

3) Note that the experiences of modeling the second example, granulating the space and time into infinitesimal granules, are precious. Two methodologies were developed, topology and non-standard analysis.

4) The third example is from natural science, it should be weigh heavily for possible abstraction. The binary granular model only captures the surface. A deeper model is desirable. Professor Chang, a physicist, has given us an excellent exposition on this topics in this collection.

5) Semantically speaking, granulations can be viewed from three directions: uncertainty theory, knowledge engineering and principles of computing. A granule is a unit of lacking knowledge, a piece of basic knowledge (information) or a software unit. Each interpretation may have its own theory.

6) Quotient structure provides a formal view of Information hiding. It plays a major role in Integration. Professor Zhang-Zhang have an article in this collection.

In this article, we have examined seven classical examples of granulations. They include the following heavy examples: the ancient granulation of human body (e.g., head, neck . . .), infinitesimal granulation of space or time (e.g., a circle is a polygon of infinitesimal sides), uncertainty principle in quantum mechanics, and etc. All examples, but the very first one, are supported by seven mathematical models.

Structure-wise speaking, as we have observed, these examples are "convenient models;" there are only five fundamental models, the first (topology, continuous mathematics), fifth (relational/logic), sixth (functional), seventh(Turing machine) and a categorical model that unified all models.

At the end, we would conclude this introduction with a vision of Zadeh taken from [22].
8.1 Zadeh’s Vision

Basically, Zadeh views Granular Computing as a mode of computing in which the objects of computation are the granular variables. Let $X$ be a variable which takes values in the universe of discourse, $U$. Informally, a granule is a clump of elements of $U$ which are drawn together by indistinguishability, similarity or proximity. For example, an interval is a granule; so is a fuzzy interval; so is a Gaussian distribution; and so is a cluster of elements of $U$. A granular variable is a variable that takes granules as values. If $G$ is a value of $X$, then $G$ is referred to as a granular value of $X$. If $G$ is a singleton, then $G$ is a singular value of $X$. A linguistic variable is a granular variable whose values are labeled with words drawn from a natural language. For example, if $X$ is the temperature, then 101.3 is a singular value of the temperature, while "high" is a granular (linguistic) value of temperature.

A granular value of $X$ may be interpreted as a representation of one state of imprecise knowledge about the true value of $X$. In this sense, Granular Computing may be viewed as a system of concepts and techniques for computing with variables whose values are not known precisely.

A concept that helps to precisiate the concept of granule is that of generalized constraint. The concept of a generalized constraint is the centerpiece of Granular Computing.

A generalized constraint is an expression of the form $X$ isr $R$, where $X$ is the constrained variable, $R$ is the constraining relation, and $r$ is an indexical variable that identifies the modality of the constraint. The principal modalities are: possibilistic ($r = blank$); veristic ($r = v$); probabilistic ($r = p$); usuality ($r = u$); random set ($r = rs$); fuzzy graph ($r = fg$); bimodal ($r = bm$); and group ($r = g$). The primary constraints are possibilistic, veristic and probabilistic. The standard constraints are bivalent possibilistic, bivalent veristic and probabilistic. Standard constraints have a position of centrality in existing scientific theories.

A generalized constraint, GC($X$), is said to be open if $X$ is a free variable, and is closed (grounded) if $X$ is instantiated. A proposition is a closed generalized constraint. For example, "Lily is young," is a closed possibilistic constraint in which $X$ = Age(Lily); $r = blank$; and $R$ = young is a fuzzy set. Unless indicated otherwise, a generalized constraint is assumed to be closed.

A generalized constraint may be generated by combining, projecting, qualifying, propagating and counterpropagating other generalized constraints. The set of all generalized constraints together with the rules governing combination, projection, qualification, propagation and counterpropagation constitute the Generalized Constraint Language (GCL). In Granular Computing, computation or equivalently deduction, is viewed as a sequence of operations involving combination, projection, qualification, propagation and counterpropagation of generalized constraints. An instance of projection is a deduction of GC($X$) from GC($X$, $Y$); an instance of propagation is a deduction of GC(f($X$)) from GC($X$), where f is a function or a functional; an instance of counterpropagation is deduction of GC($X$) from GC(f($X$)); an instance of combination is deduction of GC(f($X$, $Y$)) from GC($X$) and GC($Y$); and an instance of qualification is computation of $X$ isr $R$ when $X$ is a generalized constraint. An example of probability qualification ($X$ is small) is likely. An example of veristic (truth) qualification ($X$ is small) is not very true.
The principal deduction rule in Granular Computing is the possibilistic extension principle:

\[ f(X) \text{ is } A \longrightarrow g(X) \text{ is } B, \]

where \( A \) and \( B \) are fuzzy sets, and \( B \) is given by \( \mu_B(v) = \sup_{u \in U} (\mu_A(f(u))) \), subject to \( \nu = g(u) \). \( \mu_A \) and \( \mu_B \) are the membership functions of \( A \) and \( B \), respectively.

A key idea in Granular Computing may be expressed as the fundamental thesis: Information is expressible as a generalized constraint. The traditional view that information is statistical in nature may be viewed as a special, albeit important, case of the fundamental thesis.

A proposition is a carrier of information. As a consequence of the fundamental thesis, the meaning of a proposition is expressible as a generalized constraint. This meaning postulate serves as a bridge between Granular Computing and NL-Computation, that is, computation with information described in a natural language.

The point of departure in NL-Computation is (a) an input dataset which consists of a collection of propositions described in a natural language; and (b) a query, \( q \), described in a natural language. To compute an answer to the query, the given propositions are precisiated through translation into the Generalized Constraint Language (GCL). The translates, which express the meanings of given propositions are generalized constraints. Once the input dataset is expressed as a system of generalized constraints, Granular Computing is employed to compute the answer to the query.

As a simple illustration, assume that the input dataset consists of the proposition "Most Swedes are tall," and the query is "What is the average height of Swedes?" Let \( h \) be the height density function, meaning that \( h(u)du \) is the fraction of Swedes whose height lies in the interval \([u, u+du] \). The given proposition "Most Swedes are tall," translates into a generalized constraint on \( h \), and so does the translate of the query "What is the average height of Swedes?" Employing the extension principle, the generalized constraint on \( h \) propagates to a generalized constraint on the answer to \( q \). Computation of the answer to \( q \) reduces to solution of a variational problem. A concomitant of the close relationship between Granular Computing and NL-Computation is a close relationship between Granular Computing and the computational theory of perceptions. More specifically, a natural language may be viewed as a system for describing perceptions. This observation suggests a way of computing with perceptions by reducing the problem of computation with perceptions to that of computation with their natural language’s descriptions, that is, to NL-Computation. In turn, NL-Computation is reduced to Granular Computing through translation/precisiation into the Generalized Constraint Language (GCL).

An interesting application of the relationship between Granular Computing and the computational theory of perceptions involves what may be called perception-based arithmetic. In this arithmetic, the objects of arithmetic operations are perceptions of numbers rather than numbers themselves. More specifically, a perception of a number, \( a \),

is expressed as usually \((\star a)\), where \( \star a \) denotes "approximately \( a \)." For concreteness, \( \star a \) is defined as a fuzzy interval centering on \( a \), and usually is defined as a fuzzy probability. In this setting, a basic question is: What is the sum of usually \((\star a)\) and usually \((\star b)\)? Granular Computing and, more particularly, granular arithmetic, provide a machinery for dealing with questions of this type.
Imprecision, uncertainty and partiality of truth are pervasive characteristics of the real world. As we move further into the age of machine intelligence and automated reasoning, the need for an enhancement of our ability to deal with imprecision, uncertainty and partiality of truth is certain to grow in visibility and importance. It is this need that motivated the genesis of Granular Computing and is driving its progress. In the coming years, Granular Computing and NL-Computation are likely to become a part of the mainstream computation and machine intelligence.

References


