Granular Computing and Rough Set Theory*

Lotfi A. Zadeh

Department of EECS, University of California
Berkeley, CA 94720-1776
Tel.: 510-642-4959; Fax: 510-642-1712
zadeh@eecs.berkeley.edu

To the memory of Professor Zdzisław Pawlak

Extended Abstract

Granulation plays an essential role in human cognition and has a position of centrality in both granular computing and rough set theory. Informally, granulation involves partitioning of an object into granules, with a granule being a clump of elements drawn together by indistinguishability, equivalence, similarity, proximity or functionality. For example, an interval is a granule; so is a fuzzy interval; so is a gaussian distribution; so is a cluster of points; and so is an equivalence class in rough set theory. A granular variable is a variable which takes granules as values. If G is 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 temperature, then 101.3 is a singular value of temperature, while "high" is a granular (linguistic) value of temperature.

Basically, granular computing is a mode of computation in which the objects of computation are granular variables. A granular value, X, may be interpreted as a representation of the 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 either not known precisely or need not be known precisely.

A concept which serves to precisiate the concept of a granule is that of a generalized constraint. The concept of a generalized constraint is the centerpiece of granular computing.

A generalized constraint is an expression of the form X is$r$ R, where X is the constrained variable, R is the constraining relation, and r is an indexical variable which serves to identify 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

* Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant and the BISC Program of UC Berkeley.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 1–4, 2007.

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 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 $X = Age(\text{Lily})$; $r = blank$; and $R =$ young is a fuzzy set. Unless indicated to the contrary, 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 deduction of $GC(X)$ from $GC(X,Y)$; an instance of propagation is 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 is$r$ R when X is a generalized constraint. An example of probability qualification is (X is *small*) is *likely*. An example of veristic (truth) qualification is (X is *small*) is *not very true*.

The principal deduction rule in granular computing is the possibilistic extension principle: $f(X)$ is $A \longrightarrow g(X)$ is B, where A and B are fuzzy sets, and B is given by $\mu_B(v) = sup_u(\mu_A(f(u)))$, subject to $v = g(u)$. μ_A and μ_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 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* $({}^*a)$, where ${}^*a$ denotes "*approximately* a." For concreteness, ${}^*a$ 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* $({}^*a)$ and *usually* $({}^*b)$? Granular computing and, more particularly, granular arithmetic, provide a machinery for dealing with questions of this type.

Granular computing is based on fuzzy logic. Fuzzy logic has endured many years of skepticism and derision largely because fuzziness is a word with pejorative connotations. Today, fuzzy logic is used in a wide variety of products and systems ranging from digital cameras, home appliances and medical instrumentation to automobiles, elevators, subway trains, paper making machinery and traffic control systems. By this measure, fuzzy logic has achieved success.

There are two basic rationales which underlie the success of fuzzy logic. Indirectly, the same rationales apply to granular computing and rough set theory. The second rationale is referred to as "The fuzzy logic gambit." To understand the rationales it is necessary to differentiate between two meanings of precision: precision in value, v-precision; and precision in meaning, m-precision. For example, if X is a real-valued variable, then the proposition X is in the interval $[a, b]$, where a and b are precisely defined numbers, is v-imprecise and m-precise. Additionally, we have to differentiate between mh-precisiation, that is, human-oriented m-precisiation, and mm-precisiation, that is, machine-oriented m-precisiation. For example, a dictionary definition of stability may be viewed as an instance of mh-precisiation, while Lyapunov's definition of stability is an instance of mm-precisiation of stability. Furthermore, v-imprecisiation may be imperative (forced) or intentional (deliberate). For example, if I do not know Lily's age and describe her as young, then v-imprecisiation is imperative (forced). If I know her birthday but choose to describe her age as young, then v-imprecisiation is intentional (deliberate).

Let X be a variable taking values in U. U may be a space of numbers, functions, relations, distributions, etc. Consider two cases.

Case 1: Values of X are not known precisely, i.e., X is v-imprecise, denoted as $^*X$.

Case 2: Values of X are known precisely, i.e., X is v-precise.

In *Case* l, I have some information, $Inf(^*X)$, about values of $^*X$. I mm-precisiate $Inf(^*X)$ by using an information description language, IDL. IDL may be the language of bivalent logic and probability theory, BL + PT; or the language of fuzzy logic, FL; or a natural language, NL. NL may be mm-precisiated through translation into FL. FL is a superlanguage of (BL + PT) in the sense that it has a much higher expressive power than (BL + PT).

In *Case* 1, the use of FL as the information description language serves to enhance the accuracy of description of values of *X, especially when $^*X$ takes values in the space of functions, relations or distributions. This is Rationale 1 for the use of fuzzy logic as an information description language when the values of $^*X$ are not known precisely.

Turning to *Case* 2, we observe that, in general, precision carries a cost. If there is a tolerance for imprecision, we can exploit it by sacrificing precision through v-imprecisiation of X. This is what we do when we perform data compression, summarization and other information-reduction operations. More generally, we v-imprecisiate X to $^*X$ to reduce cost. By so doing, we reduce *Case* 2 to *Case* 1. Then we mm-precisiate $^*X$ through the use of NL as an information description language. This is the essence of Rationale 2 for the use of fuzzy logic when the values of a variable are known precisely. In this context, the fuzzy logic gambit may be stated as:

If there is a tolerance for imprecision, exploit it through v-imprecisiation followed by mm-precisiation.

The fuzzy logic gambit is Rationale 2 for the use of fuzzy logic when the values of a variable are known precisely.

It is of historical interest to note that my 1965 paper "Fuzzy sets" was motivated by Rationale l. My 1973 paper, "Outline of a new approach to the analysis of complex systems and decision processes," was motivated by Rationale 2. Today, most applications of fuzzy logic employ the concepts of a linguistic variable and fuzzy if-then rule sets – concepts which were introduced in the 1973 paper.

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 rough set theory, and is driving their progress. In coming years, granular computing, rough set theory and NL-Computation are likely to become a part of the mainstream of computation and machine intelligence.