Formalization of Generalized Constraint Language: A Crucial Prelude to Computing With Words

Abstract: The Generalized Constraint Language (GCL) , introduced by Zadeh, serves as a basis for computing with words (CW). It provides an agenda to express the imprecise and fuzzy information embedded in natural language and allows reasoning with perceptions. Despite its fundamental role, the definition of GCL has remained informal since its introduction by Zadeh (Zadeh 2004, 2005), and to our knowledge, no attempt has been made by CW community to formulate a rigorous theoretical framework for GCL . Such formalization is necessary for further theoretical and practical advancement of CW for two important reasons: first, it provides the underlying infrastructure for the development of new inference rules based on sound theories. Second, it determines the scope of the language as well as its set of well-formed formula and hence facilitates the translation of natural language expressions into GCL. This paper is an attempt to step in this direction by providing formal recursive syntax together with a compositional semantics for GCL. Furthermore, the soundness of Zadeh's inference rules is proved in the proposed formal language.

Keywords: computing with words, formal language, generalized constraint language, test score semantics, fuzzy logic

Abbreviations: Computing with Words: CW; Generalized Constraint: GC; Generalized Constraint Language: GCL

1. Introduction

After Zadeh's seminal paper 1 which introduced the machinery of computing with words, there have been many researchers that tried to develop the CW framework in theoretical or practical directions. The articles published reflect very different points of view and research directions regarding CW. A recent discussion on what CW means to the different pioneer researchers in the area has been published in 2. Generally, the current literature of CW can be classified as follows:

- *The articles that focus on practical aspects of CW*: These include the works by Kacprzyk and Zadrozny 3 which studied the application of CW in fuzzy database querying and linguistic data summarization, or the study of CW in decision making which was promoted by Herrera and his collaborators 6 .
- *The articles that deal with the linguistic aspects of CW*: These articles 7 studied the relation of CW to cognitive sciences, computational linguistics, and ontology.
- *The articles that focus on developing reasoning methods for CW:* these articles have tried to implement an inference engine for CW for systematic application of deduction rules to a linguistic knowledge base 11
- *The articles that capture underlying uncertainty of the words:* These articles 14 mostly used interval type II fuzzy sets to model the uncertainty embedded in the membership function of words.
- *The articles that propose alternative approaches to computing with words*. The output of the inference engine in CW is in terms of a fuzzy subset over the universe of discourse of the output variable. For some applications, where the output is in natural language, the fuzzy subset needs to be retranslated back to words and approximated within the domain of granule values of the original variable 18. The main challenge faced here is the loss of information caused by this approximation. There have been many articles in the literature that focused on minimizing such loss (19). But there have also been a few articles that have focused on developing alternative representation and/or reasoning approaches for CW to eliminate such loss 25 .
- *The articles that develop a formal model of computation for CW:* These articles 30 interpreted the word "computing" in CW as a formal computational model rather than a "reasoning" framework. They extended the classical methods of computation, such as: finite state and push-down automata, Turing machines, and formal grammars, to accept "words" as fuzzy subsets over the input alphabet hence implementing the computation with words.

Despite the intensive research in the area, the ultimate goal of building a CW computational engine has thus far proven to be elusive. It is the view of the present authors that one important reason for this difficulty is that CW lacks a rigorous theoretical foundation, and that without such a foundation, further advancements in the field will necessarily be weak and lack cohesion. This paper presents a roadmap to achieve such a foundation for CW. The first step in this direction is to develop a formal language for generalized constraints-- that is, to establish a formal syntax that determines a set of well-formed formula, together with a compositional semantics, which will determine the meaning of a complex sentence from its constituents.

The advantage of such formalization is twofold: First, it facilitates automated or semi-automated translation of natural language expressions into GCL. Without a firm definition of the latter it is not possible to assess the representational capability of GCL --- that is, to determine what can or cannot be represented in GCL and whether a translated proposition is equivalent in meaning to the original one. It must be emphasized, however, that an inclusive translation from natural language to GCL is not feasible. The laws of the syntax and semantics of natural language are much more complicated to be grasped by GCL. Natural language statements are context-dependent and contain non-truth functional connectives (Francis 2006) which cannot be transformed into GCL. As a result, the proper goal in this area is to translate a controlled form of natural language 33 with a restricted grammar and semantics. Second, formalization of GCL leads to the development of new sound deduction rules. In 34 Zadeh stresses that the current set of deduction rules are not complete and lists instances of the rules that need to be added to the system. Once a formal compositional semantics is defined for GCL such extensions are possible. One example is given in the penultimate section of the present paper.

The rest of the article is organized as follows. The next section provides some introductory remarks on GCL, Zadeh's test score semantics and the structure of graded values. The following section presents our formal GCL language. In the fourth section, we show that Zadeh's inference rules are sound in the proposed language, and point out the potential that a formal GCL creates for further development of sound inference rules.

2. Preliminaries

Our point of departure for defining a formal GCL is to take Zadeh's informal description of the generalized constraints and the test score semantics and try to formalize them. Before proceeding, we also need to describe the structure of the truth values that we will use. This section reviews GCL and test score semantics and describes the structures of the truth values used in many-valued logics.

2.1 Generalized Constrained Language

Computing with Words (CW) introduces an emerging paradigm shift in information representation and processing and aims at utilizing the tolerance of uncertainty in words to facilitate complex reasoning and human-like decisions [35] . The roots of CW were formed in 1965, when Zadeh proposed the notion of fuzzy sets to model the meaning of inexact words in natural language. The idea then was further extended throughout the years by development in various fuzzy domains such as: fuzzy logic, linguistic variables, fuzzy relations, fuzzy arithmetic, fuzzy quantifiers, and fuzzy probability.

The phrase "computing with words" was first coined by Zadeh [1]:
"A computational system in which the objects of computations are words and propositions drawn from natural language. It is inspired by the human remarkable capability to perform a wide variety of physical and mental tasks without any measurements and any computations."
The main feature distinguishing CW from other logical systems such as propositional or predicate logic is its ability to model and perform computation on the imprecise words used in natural language. Instead of representing the proposition: "gas is expensive" as "expensive(gas)" in predicate logic and giving it a truth value of either 0 or 1, CW models the meaning of expensive and represents it as a fuzzy set over the universe of discourse of the values associated with the price of gas. Furthermore, the fuzzy-based modeling of CW allows expressing a rich set of quantifiers such as "most", "few", "many", "several", and so forth, while predicate logic fails to do so, hence it makes advancement towards reasoning with natural language expressions.

The core of CW is to represent the meaning of a proposition in the form of a generalized constraint. The idea is that a majority of the propositions and phrases used in natural language can be viewed as imposing a constraint on the values of some linguistic variables such as: time, price, taste, age, relation, size, and appearance. For example the sentence: "most Indian foods are spicy" constrains the two variables: (1)

the taste of Indian food, and (2) the portion of the Indian foods that are spicy. In general a GC is in form of:

X isr R

Where *X* is a linguistic (or constrained) variable whose values are constrained by the linguistic (granule) value *R.* A linguistic variable can take various forms: it can be a relation (such as: (X,Y)), a crisp function of another variable (such as: f(Y)), or it can be another GC.

The small *r* shows the semantic modality of the constraint, that is: how X is related to R. various modalities are characterized by Zadeh, among them are:

- possibility (*r* : *blank)*: where *R* denotes the possibility distribution of *X* (Dubois and Prade 1988), *e.g., "X is large."*.
- verity (*r*: *"v")*: where R denotes the truth distribution of *X* (Yager 2000)*, e.g., "X is large isv very true"*.
- identity (r: "="): *X* and *R* are identical variables.
- fuzzy graph (r: "isfg"): *R* is a fuzzy estimation of a function. This modality corresponds to a collection of fuzzy if then rules that share the same variables in their premises and consequences.
- probability (r: "p"), Where *R* is the fuzzy probability distribution of *X* (Bugajski 1996)*, e.g., "(x is large) isp likely"*.

Table 1 shows some examples of natural language propositions and their meaning representation in GC. Notice that a proposition in natural language may constrain one or more variables.

Table 1. Examples of Natural Language Propositions and their representations in GCs

Natural Language Proposition	**Generalized Constraints**
The price of oil has inverse relation with the oil production.	$\underbrace{\text{Relation(price(oil), production(oil))}}_{X}$ is $\underbrace{\text{inverse}}_{R}$
Most Indian foods are spicy	$\underbrace{\text{count}_x\text{ (}\underbrace{\text{taste (x)}}_{X_1}\text{ is }\underbrace{\text{spicy}}_{R_1}\text{ \| }\underbrace{\text{x}}_{X_2}\text{ is }\underbrace{\text{indian-Food}}_{R_2}\text{)}}_{X_3}$ is $\underbrace{\text{most}}_{R_3}$
It is Likely to rain today	$\underbrace{\text{(}\underbrace{\text{weather(today)}}_{X1}\text{ is }\underbrace{\text{rainy}}_{R1}\text{)}}_{X2}$ is**p** $\underbrace{\text{likely}}_{R2}$

A collection of GCs together with a set of logical connectives (such as: and, or, implication, and negation) and a set of inference rules form the generalized constraint language (GCL). The inference rules regulates the propagation of GCs. Table 2 lists instances of inference rules introduced for GCL. As shown in this table, each rule has a syntactic part and a semantic part. The syntactic part shows the general abstract form (also called the protoform) of the GCs of the premises and the conclusion of the rule, while the semantic part is a semantic condition which makes the rule valid. The inference rules of CW are adopted and formalized from various fuzzy domains such as: fuzzy probability, fuzzy logic, fuzzy relation, fuzzy quantifiers, and fuzzy arithmetic.

Table 2. Instances of CW inference Rules. u and v are the universes of discourse of x and y.

Inference rule	**Syntactic part**	**Semantic part**
Conjunction Rule	$\dfrac{\begin{array}{c}x \text{ is } A \\ y \text{ is } B\end{array}}{(x, y) \text{ is } C}$	$\mu_C(u, v) = \mu_A(u) * \mu_B(v)$
Projection Rule	$\dfrac{(x, y) \text{ is } A}{x \text{ is}}$	$\mu_B(u) = \sup_v(\mu_A(u, v))$
Extension Principle	$\dfrac{X \text{ is } A}{f(X) \text{ is } B}$	$\mu_B(z) = \sup_U(\mu_A(u))$ $\text{subject to}: z = f(u)$
Compositional Rule of Inference	$\dfrac{\begin{array}{c}X \textit{ is } A \\ (Y, X) \textit{ is } B\end{array}}{Y \textit{ is } C}$	$\mu_C(v) = \sup_u(\mu_A(u_i) * \mu_B(v, u))$
Fuzzy graph Interpolation	$\dfrac{\begin{array}{l}\sum_i \textit{if } X \textit{ is } A_i \textit{ then } Y \textit{ is } B_i \\ X \textit{ is } A\end{array}}{Y \textit{ is } B}$	$\mu_B(v) = \sup_i(m_i * B_i)$ $m_i = \sup_u(\mu_A(u) * \mu_{A_i}(u)),$ $i = 1...n$
Fuzzy Syllogism	$\dfrac{\begin{array}{l}Q_1 A\text{'s are } B\text{'s} \\ Q_2 (A \& B)\text{'s are } C\text{'s}\end{array}}{Q_3 A\text{'s are } (B \& C)\text{'s}}$	$Q_3(z) = \sup_u(Q_1(w_1) * Q_2(w_2))$ $\text{subject to}: z = w_1 \times w_2$ w_1,w_2, and z are the universes of discourse of Q_1, Q_2,and Q_3 , respectively

2.2 Meaning Representation via Test Score Semantics

Zadeh defined test score semantics 36 as an informal semantics for representing the meaning of natural language statements. Like other meaning representation systems, test score semantics relates a linguistic entity to its denotation in some assigned universe of discourse. This approach assumes that each predicate induces a constraint on a group of object variables in the universe of discourse and is denoted by (in general) a fuzzy relation. A collection of fuzzy (or crisp) relations models the state of the world and constitutes what is referred to as an *explanatory database*. The meaning of a proposition or a predicate is then determined by the degree to which it is compatible with the explanatory database. In other words, the test score of a proposition p with respect to an explanatory database , ED, is a point in the interval [0,1], and indicates the degree to which p is *compatible* with ED. Representing the meaning of a proposition, p, in the test score semantics includes: (1) identifying the object variables whose values are constrained by p, (2) identifying the constraints imposed by p on the constraint variables, (3) finding the degree of compatibility (or test score) of p with respect to its denotation in the explanatory database, and (4) aggregating the partial test scores obtained for each constraint. As an illustration, suppose we would like to assess the test score of the simple proposition: "Ellie and Mary are friends." This proposition induces the constraint *"friend"* on the relation of two object variables denoting two individuals "Ellie" and "Mary". If the explanatory database in this case contains a fuzzy relation "Friends" as follows, then test score of the proposition "Ellie and Mary are friends" with respect to this database is 0.9.

Table 3. Example of fuzzy relation denoting the predicate "Friend" in explanatory database

Friend	**Individual 1**	**Individual 2**	**Degree of Membership**
	Alice	John	.4
	Mona	Ellie	.7
	Jessica	Sarah	.2
	Ellie	Mary	.9

In some cases, it is possible that more than one relation must be tested to determine the test score of a proposition. Consider the proposition "gas is expensive," with an explanatory database containing two relations: (1) the crisp relation "Price" with the attributes: "item" and "unit price", which shows the unit price of a number of items in the universe of discourse, and (2) the fuzzy relation "Expensive" with the attributes "price" and "membership degree" which shows the degree of costliness for each value of the price. In order to find the test score of the proposition "gas is expensive" with respect to this explanatory database, one needs to first find the value of the price of gas from the “Price” relation and then find the corresponding membership degree from the fuzzy relation “Expensive”.

When a proposition induces two or more constraints then the combined test score is computed according to the following rules (where ts_1 and ts_2 are the test scores induced by two different constraints C_1 and C_2 in proposition *p*):

- Modification rules, including: (1) negation: the test score of “not C_1” is $1-ts_1$, (2) concentration: the test score of “ very C_1” is $ts_1^{\ 2}$, and (3) diffusion: the test score of “more or less C_1” is $ts_1^{\ 0.5}$.
- Compositional rules, including: (1) conjunction: the test score of “C_1 and C_2” is: $\min(ts_1, ts_2)$ (2) disjunction: the test score of “C_1 or C_2” is: $\max(ts_1, ts_2)$, (3) implication: the test score of “if C_1 then C_2” is: (1-ts_1+ts_2).

2.3 Algebraic structure of truth Values

To define a cohesive formal semantics for a logical language we first fix an algebraic structure of truth values. In classical logic a proposition is unambiguously true or false. Hence the set of truth values is simply the set $\{0,1\}$ and the algebra of the truth values is a Boolean algebra 37. In contrast, in fuzzy logic, a proposition can be only partially true, and the set of truth values form a partially ordered set. The most general accepted structure of truth values in fuzzy logic which fits the intuitive notion of implication and conjunction is that of *residuated lattice* 38. A residuated lattice is an algebra: $\mathbf{L} = (L, \vee, \wedge *, \rightarrow, 0, 1)$, such that:

- $(L, \vee, \wedge, 0, 1)$ is a lattice with 0 and 1 being the least and the greatest elements respectively.
- $(L, *, 1)$ is a commutative monoid. That is, * is a commutative, associative, and monotone in both arguments, and it has the identity $a * 1 = a$ for each $a \in L$.
- The pair * and $\rightarrow$ satisfy the adjunction property, that is: $a * b \leq c \quad iff \quad a \leq b \rightarrow c$. For $a, b, c \in L$.

The operation * is known as *t-norm* and $\rightarrow$ is called its residuation. Table 4 shows some further properties proved for L (Novák et al. 1999) .

Table 4. Properties of t-norm and its residuation, $a, b, c \in L$

(1) $a * b \le a$	(2) $a * b \le a \wedge b$
(3) $b \le a \to b$	(4) $a * (a \to b) \le b$
(5) $b \le a \to (a * b)$	(6) $a * (a \to 0) = 0$
(7) if $a \le b$ then $a \to c \le b \to c$ and $c \to a \le c \to b$	(8) $a \le b$ *iff* $a \to b = 1$
(9) $(a \vee b) * c = (a * c) \vee (b * c)$	

Each residuation defines its corresponding negation operation as follows:

$\neg a = a \to 0 \qquad , a \in L$

The dual operation of t-norm is called t-conorm and is denoted by $\oplus$. Given a t-norm a complementary t-conorm is defiend as:
$(a \oplus b) = 1 - ((1 - a) * (1 - b))$

A residuated lattice with a continuous t-norm is called a *BL-Algebra* if it satisfies the following properties (Hájek 1998):

- $a \wedge b = a * a \to b$
- $a \vee b = (a \to b) \to b \wedge (b \to a) \to a$
- $a \to b \vee b \to a = 1$

Some important examples of BL-algebras are the Gödel, Łukasiewicz, and product algebras. Table 5 displays the t-norms defined for these algebra along with their corresponding residuation, negation, and t-conorm operations.

Table 5. Three important BL-Algebras

Algebra	**T-Norm**	**T-Conorm**	**Residuation**	**Negation**
Łukasiewicz	$a *_L b = 0 \vee (a + b - 1)$	$a \oplus_L b = 1 \wedge (a + b)$	$a \to_L b = 1 \wedge (1 - a + b)$	$\neg_L a = 1 - a$
Gödel	$a *_G b = a \wedge b$	$a *_G b = a \vee b$	$a \to_G b = \begin{cases} 1 & if\ a \le b \\ 0 & a > b \end{cases}$	$\neg_G a = \begin{cases} 1 & if\ a = 0 \\ 0 & a > 0 \end{cases}$
Product	$a *_P b = a \times b$	$a *_P b = a + b - ab$	$a \to_P b = \begin{cases} 1 & if\ a \le b \\ \frac{b}{a} & a > b \end{cases}$	$\neg_P a = \begin{cases} 1 & if\ a = 0 \\ 0 & a > 0 \end{cases}$

3. Formalization of GCL

This section presents our formal syntax and semantics for GCL. We have used Zadeh's original symbols and notations, with a slight modification, to define terms, atomic formula, coordinators, and formation rules. The proposed language is used in the next section to formulate and extend Zadeh's inference rules for CW.

3.1 GCL Syntax

The formalization of syntax includes identifying a set of valid *well-formed formula*. The well-formed formulas are defined by induction, using a set of symbols and a set of formation rules which describes how the symbols may be combined to form a valid formula.

3.1.1 Symbols

- A non-empty finite set of sorts, denoted by $S = \{s_1, s_2, .., s_n\}$. The sorts determine different types of variables. While we could use a single sort and define many sorts by means of crisp unary predicates, it is conventional to use many-sorted structures, to facilitate type-checking. Each non-empty sequence of sorts: $(s_1, s_2, ..., s_n)$ is called a *rank* 39.
- For each sort s_i, a countable set of variables, denoted by the lowercase letters at the end of the alphabet: $x, y, z, ...$ The subscripts are used to distinguish the variables: $x_0, x_1, x_2 ...$ .Variables are interpreted as individual objects in the universe of discourse of each sort, for example: "Mary" and "John" of sort PEOPLE, "New York" and "Chicago" of sort CITIES, 35.5 of sort REAL-NUMBERS, and so forth.
- A countable number of crisp function symbols with arity greater than 0, denoted by lowercase letters: f, g, h,... . Each function of arity n has a rank: $(s_0, s_1, .., s_n)$, which is the sequence of sorts of variables in its domain, and it has a sort: s_{n+1}, which is the sort of its range. Function symbols of valence 0 are constant symbols, and are often denoted by lowercase letters at the beginning of the alphabet a, b, c. Each constant symbol has a sort.
- A countable set of possibilistic granule values, each having a rank $(s_0, s_1, .., s_n)$. The granule values are denoted by the uppercase letters at the beginning of the alphabet: $\mathrm{A, B, C...}$. The granule value may be interpreted as a fuzzy value (such as: "tall" , "slim", "tasty", "about 6"…) or a crisp value such as: ("6", (3,6), etc.).

- A countable set of veristic granule values, such as: “true”, “certainly true”, “positive”, and so forth.
- A countable set of quantifiers denoted by $Q_0, Q_1, Q_2 \cdots$ (such as: “many”, “most”, “almost all”, “all”, “some”,…). Two classes of quantifiers are distinguished: 1) absolute, which are expressed with a number, e.g., “approximately 10”, “much more than three”, “almost 5” etc, and 2) relative quantifiers which are expressed with proportion: “most”, “all”, “many”, “some”, “few”, “several”, “numerous”, etc.
- The possibility constraint symbol, denoted by “is”.
- The equality constraint symbol denoted by “=”.
- The verity constraint symbol, denoted by “isv”.
- The fuzzy graph constraint: “isfg”.
- The parentheses, brackets, and other punctuation symbols.
- Connectives and operations: “and” for conjunction, “or” for disjunction, “not” for negation, and “If …Then” for implication.
- Fuzzy modifiers, denoted by $m_0, m_1, m_2 \cdots$, such as: “very”, “more or less”, “much”, “somewhat” “to some extend”, and so forth.

3.1.2 Formation Rules

The formation rules define terms and valid formula of GCL. These rules can be mapped in to a context free grammar.

Terms:

Terms are defined inductively as follows (note that each term is associated with a sort):

- If x is a variable, then x is a term with sort s_i.
- If $t_1, t_2, ...t_n$ are terms and f is a crisp function with rank ($s_1, .., s_n$) and sort s_f, then $f(t_1, t_2, ...t_n)$ is a term with sort s_f.

Formulas:

- If $t_1, t_2, ...t_n$ are terms and A is a possibilistic granule value of the same rank, then “$(t_1, t_2, ...t_n)$ is A” is a formula.
- If t_1 and t_2 are terms then “$t_1 = t_2$” is a formula, where t_1 and t_2 are of the same rank.

- If $(t_1, t_2, \ldots t_n)$ are terms and A is a possibilistic granule value of the same rank, then "t is mA" is a formula, where *m* is a fuzzy modifier.
- If t_1 and t_2 are terms then "(t_1, t_2) *isfg* $\{(A_1, B_1), (A_2, B_2), \ldots, (A_n, B_n)\}$" is a formula. Where each A_i and B_i are possibilistic granule values with the same rank as t_1 and t_2, respectively.
- If Q is an absolute or relative quantifier and φ is a formula then "$Q_{x:s}\ \varphi$" is a formula. Where x is a variable in φ, and *s* is its sort.
- If Q is a <u>relative</u> quantifier and φ_1, φ_2 are formula then "$Q_{x:s}\ (\varphi_1, \varphi_2)$" is a formula and formulates a binary quantifier.
- If φ is a formula then "*not* φ" is a formula.
- If φ_1 and φ_2 are formulas then "φ_1 and φ_2" is a formula.
- If φ_1 and φ_2 are formulas then "φ_1 *or* φ_2" is a formula.
- If φ_1 and φ_2 are formulas then "if φ_1 then φ_2" is a formula.
- If φ is a formula, and *V* a veristic granule value such as: "certainly true", "positive", "not true", and so forth, then "φ isv V" is a formula.

The above formation rules construct the syntax of GCL recursively. The notion of free and bound variables is defined similar to the classical logics. A variable y is *substitutable* for variable *x* in a formula φ if it has the same sort as *x*, and it does not change any free occurrence of *x* in φ into a bound occurrence of *y*. Some examples of formula of GCL are listed in table 6.

Table 6. Examples of GCL formula

Joe is handsome	Joe is handsome
Mary and Ann are friends	(Mary, Anna) is friend
Most Students are healthy	Most $_{x:STUDENT}$ (x is healthy)
Most Young students are healthy	Most $_{x:STUDENT}$ (x is healthy , age(x) is young)
Three students got a low grade in math	Three$_{x:\ STUDENT}$ (grade(x, math) is low)
A lot of IT employees get a high salary	A lot of$_{x:IT\ EMPLOYEE}$ (salary(x) is high)
The gas pressure has an inverse relation with its temperature	All$_{x:GAS}$ (pressure, temperature) isfg {(high, low), (medium, medium), (low, high)}
Most of the people, spend more, when they earn more money.	Many $_{x:PEOPLE}$ ((salary(x),spend(x)) isfg { (low, low), (medium, medium), (high, high)}
It is true that most Americans	(Most$_{x:PEOPLE}$ (

have a higher standard of living than most people from the Middle East.	Most $_{y:PEOPLE}$ ((livingStan(x), livingStan(y)) is higher , (x is American *and* y is Middle-Eastern)))) isv true

Remark 1. As in the case of predicate logic, the use of sorts (or typed variables) can be avoided by introducing unary crisp predicates signifying different types. For example instead of the GCL expression: "lot of$_{x:EMPLOYEE}$ (salary(x) is high)" one could say: "lot of$_x$ (salary(x) is high and x is employee)". However, many-sorted logics are used widely in computer science and linguistics, e.g. in definitions of programming languages, semantics verification, databases, and abstract data types 39. Defining different categories of object variables in GCL allows excluding malformed expressions such as: "lot of$_{x:Dog}$ (salary(x) is high)", by elucidating the fact that the function "salary" is defined on objects of sort "EMPLOYEE".

3.2 GCL Semantics

In this section we adapt Zadeh's informal description of test score semantics to define a formal semantics for GCL. We assume that the structure of truth values, denoted by L, is in form of a BL-Algebra. A closer look at Zadeh's description of test score semantics shows that it is closely related to model-theoretic semantics. Thus it is reasonable to start with explanatory database and express it in terms of structures in the model theory. Consequently, test score can be assessed by evaluating the truth value of each formula recursively from its constituents.

3.2.1 The Explanatory database as a structure

The explanatory database consists of a set of relations and functions that model the state of the world and provides interpretation for the formulas in GCL.
Let L be a BL-Algebra, an explanatory database, *D*, may be defined as the following structure:

$D = \left\langle D_s, f_f, r_P, r_Q, r_v, r_m \right\rangle$, where:

- D_s is a nonempty set that forms the domain of variables (objects) of sort *s*. This could be the universe of discourse of CITIES, PEOPLE, PROVINCES, and COUNTRIES in geography, or the universe of discourse of STUDENT, FACULTY, and STAFF in the school context.

- Each function symbol f_i of arity n, rank $(s_1,...,s_n)$, and sort s_f is interpreted as $f_i : D_{s_a} \times ... \times \mathrm{D}_{\mathrm{s_n}} \rightarrow D_{S_f}$ which returns the value of f_i for each n-tuple of variables of sorts $s_1, s_s, ..., s_n$. Note that Zadeh's linguistic variables 40 may be modeled as functions. For example the linguistic variable *"distance(x,y)"* is a function of rank $(CITY, CITY)$ and sort REAL, and is interpreted as: $f_{dis\tan ce} : D_{CITY} \times D_{CITY} \rightarrow \mathrm{D}_{\mathrm{REAL}}$ where D_{CITY} is the domain of variables of sort $CITY$, and $\mathrm{D}_{\mathrm{REAL}}$ is the domain of real numbers. Or the linguistic variable "*age(x)"* may be modeled as a function of rank *PEOPLE* and sort INT, and is interpreted as: $f_{Age} : D_{PEOPLE} \rightarrow D_{INT}$ that maps the individuals in the domain to an integer representing their age.
- Each possibilistic granule (or crisp) value of rank $(s_1, s_2, ..., s_n)$ is interpreted as a fuzzy (or crisp) relation, $r_P : D_{s_1} \times D_{s_2} \times ... \times D_{s_n} \rightarrow L$. For example the granule value "far" of rank (REAL), may be interpreted as $r_{Far} : D_{REAL} \rightarrow L$ that assigns to each real number denoting a distance between two cities, the degree that it is considered "far".
- Each veristic granule value is interpreted as a function: $r_v : L \rightarrow L$, which assigns to each test score its membership degree in *v*.
- Each <u>absolute quantifier</u> is interpreted as a function: $r_Q : Z^+ \rightarrow L$, where Z^+ is the set of non-negative integers denoting the cardinality of a set. In other words, each absolute fuzzy quantifier is interpreted as a fuzzy set that assigns to a number (cardinality of a set) the degree to which it satisfies the quantifier. Note that if the quantifier is crisp then L is reduced to Boolean algebra.
- Each <u>relative fuzzy quantifier</u> is interpreted as a function: $r_Q : [0,1] \rightarrow L$ which assigns to a proportion, the degree to which is satisfies the relative quantifier. When the relative quantifier is crisp then L is reduced to Boolean algebra.
- Each fuzzy modifier is interpreted as a function: $r_m : L \rightarrow L$. For example the fuzzy modifier "very" is interpreted as a function: $r_{very}(a) = a^2, a \in L$

3.2.2 Test Score as Truth Evaluation

Now that we have defined the explanatory database, the test score of a natural language statement can be determined by evaluating the truth value of its equivalent formula in GCL.

Let *D* be an explanatory database, a valuation (*v*) is a function that assigns to each term in GCL its corresponding interpretation in D. The valuation function gives a meaning to the formula with free variables, such as "*Age(x) is young*", or the formula "*(size(y),size(z)) is bigger*". The truth value of these formulas depend on the value of *x,* or *y* and *z*. The valuation of GCL terms is defined as follows:

- $v(x) = d_i \in D_s$, where *v* maps a variable of sort *s* to an element in its domain.
- $v(f_i(t_1, t_2, ..., t_n)) = f_i(v(t_1), v(t_2), ..., v(t_n))$, where f is a crisp function.

The *test-score* of a formula φ with respect to an explanatory database *D* and a valuation function v is denoted by $ts_v^D(\varphi)$ and shows the degree of compatibility of φ with respect to *D*. The quantity $ts_v^D(\varphi)$ is defined as follows:

- The test score $ts_v^D((t_1, t_2, ..., t_n)\ is\ A) = r_A(v(t_1), v(t_2), ..., v(t_n))$, where r_A is the interpretation of the possibilistic granule value A.
- The test score $ts_v^D((t_1, t_2, ..., t_n)\ is\ mA) = r_m(ts_v^D((t_1, t_2, ..., t_n)\ is\ A))$, where $\mathrm{r_m}$ is the interpretation of the fuzzy modifier *m*.
- The test score $ts^D(t_1 = t_2) = 1$ iff $v(t_1) = v(t_2)$, for all valuations *v*, otherwise it is zero. Notice that, for simplicity, the equality symbol is interpreted as a crisp identity rather than similarity or fuzzy equality.
- The test score
  $ts_v^D((t_1, t_2)\ isfg\ \{(A_1, B_1), (A_2, B_2), .., (A_n, B_n)\}) = \bigvee_i (ts_v^D(t_1\ is\ A_i) * ts_v^D(t_2\ is\ B_i))$,
  where $\vee$ and * are operations of L. This test score is defined based on the mathematical properties of fuzzy approximation of a function and Mamdani style fuzzy control [41,42].
- The test score $ts_v^D(Q_{x:s}\ \varphi) = r_Q(\sum_{v'} ts_{v'}^D \varphi)$, where Q is an absolute quantifier and v' is an evaluation that differs with *v* only in the assignment of *x*. In other words, the test score of $Q_x\ \varphi$ relates to the sum of the test scores of all formula obtained

by instantiating x in φ. This corresponds with the concept of sigma-count defined by Zadeh for measuring the cardinality of a fuzzy set. Note that there are other interpretations that relate the truth value of a quantifier to the subsets of the universe of discourse as in the case of classical generalized quantifiers 43. For simplicity, this paper follows Zadeh's approach that evaluates a quantified statement based on the cardinality of a fuzzy set and this should suffice for most applications. If the quantifier is relative (e.g. "several" , "many"), then

$ts(Q_{x:s}\ \varphi_1) = r_Q(\frac{\sum_{v'}(ts_{v'}^D(\varphi_1)}{|M_s|})$, where $|M_s|$ is the cardinality of the domain of variables of sort s.

- The test score $ts(Q_{x:s}\ (\varphi_1,\varphi_2)) = r_Q(\frac{\sum_{v'}(ts_{v'}^D(\varphi_1) * ts_{v'}^D(\varphi_2))}{\sum_{v'} ts_{v'}^D(\varphi_2)})$, which relates the truth value of a binary quantifier to the cardinality of crisp or fuzzy set, where the formula $Q_{x:s}\ (\varphi_1,\varphi_2)$ means: *"Q* objects of type *s* satisfying φ_2 also satisfy φ_1. Note that the truth values of the crisp quantifiers: ($\forall$ and $\exists$) may also be derived from the above definition, provided that $r_\forall$ and $r_\exists$ are the following singletons: $r_\forall(a \in L) = \begin{cases} 0 & a = 1 \\ 1 & a \neq 1 \end{cases}$, and $r_\exists(a \in L) = \begin{cases} 0 & a = 0 \\ 1 & a \neq 0 \end{cases}$.
- The test score $ts_v^D(\varphi\ isv\ V) = ts_v^D(r_V(\varphi))$, where r_V is the interpretation of the veristic granule value *V.*
- The test score $ts_v^D(\varphi_1\ and\ \varphi_2) = ts_v^D(\varphi_1) * ts_v^D(\varphi_2)$, where * is the t-norm operation of L.
- The test score $ts_v^D(\varphi_1\ or\ \varphi_2) = ts_v^D(\varphi_1) \oplus ts_v^D(\varphi_2)$, where $\oplus$ is the t-conorm.
- The test score $ts_v^D(if\ \varphi_1\ then\ \varphi_2) = ts_v^D(\varphi_1) \rightarrow ts_v^D(\varphi_2)$ where $\rightarrow$ is the residuation operation of L
- The test score $ts_v^D(not\ \varphi) = ts_v^D(\neg\varphi) = ts_v^D(\varphi) \rightarrow 0$

As an example, suppose that we want to assess the test score of the formula $\varphi =$ *Several* $_{x:\ STUDENT}$ *(grade(x, math) is low),* given the following explanatory database.

$D = \{M_{Student}, M_{Grade}, f_{math\_grade\ \ STUDENT \rightarrow GRADE}, r_{Low(GRADE) \rightarrow\ L}, r_{Several(STUDENT) \rightarrow\ L}\}$, where

- $M_{Student}$ *={"Ellie", "Alex", "Steve", "Tony", "Sarah", "Chet", "Christ"}* is the domain of the objects of sort STUDENT,
- $M_{Grade} = \{A^+, A, A^-, B^+, B, B^-, C^+, C, C^-, D^+, D, D^-, F\}$ is the domain of objects of sort Grade.
- $f_{math\_grade\ STUDENT \to GRADE}$ = *{(Ellie, A), (Alex, A⁺), (Steve, B⁺), (Tony, C⁺), (Sarah, B), (Chet, C⁻), (Christ, D)}* is a function of rank STUDENT and sort GRADE, which assigns to each student his/her grade in math
- $r_{Low(GRADE) \to L} = \{(A^+,0),(A,0), (A^-,0), (B^+, .1), (B,.2), (B^-, .3), (C^+, .5), (C,.6), (C^-, .7), (D^+, .9), (D,1), (D^-, 1),(F,1)\}$ is a fuzzy relation "Low" with rank GRADE that assigns to each grade its membership degree in "Low",
- $r_{several(STUDENT)}(x) = \begin{cases} 0 & x < .2 \\ 2.5x - .5 & .2 < .x < .6 \\ 1 & x > .6 \end{cases}$ is a function that assigns to $x \in [0,1]$ its degree of membership in fuzzy set "several".

Having the above explanatory database at hand, the test score of the formula: $\varphi =$ *Several* $_{x:\ STUDENT}$ *(grade(x, math) is low)* may be calculated as follows.

$$ts^D(\varphi) = r_{Several}\left(\frac{\sum_v ts_v^D(\text{grade(x, Math) is low})}{|M_{Student}|}\right)$$

Notice that this formula does not contain any free variable, thus its test score is independent of the valuation of variables.

$|M_{STUDENT}| = 7$ is the cardinality of the universe of discourse of students.

$$\sum_v ts_v^D(grade(x, Math)\ is\ low) = \sum_{u \in M_{STUDENT}} ts^D(grade(u, MATH)\ is\ low) =$$

$$r_{Low}(A) + r_{Low}(A^+) + r_{Low}(B^+) + r_{Low}(C^+) + r_{Low}(B) + r_{Low}(C^-) + r_{Low}(D) =$$

$$.1 + .5 + .2 + .6 + 1 = 2.4$$

Hence; $ts^D(\varphi) = r_{Several}(2.4/7) = .35$

Remark2. We shall point out here that although we made extensive use of the results and ideas of mathematical fuzzy logics 41, 44), our formalization of GCL is different from that of fuzzy predicate logics and their counterparts. Those logics achieved a remarkable success in providing a mathematical foundation for fuzzy logic in the spirit of classical logics, but they are not suitable for the machinery of computing with words, since fuzzy predicate logics do not account for Zadeh's different constrained modalities and fuzzy quantifiers. Moreover, the rules of inference in mathematical fuzzy logics, namely the modus ponens and generalization rules, are purely syntactical while Zadeh's rules of inference for CW are both syntactical and semantical. Figure 1 illustrates the general architecture of a CW inference engine. Given a knowledgebase, which is expressed in form of GCL, the CW inference engine is typically intended to answer questions of the form: "What is the value of a term t?". The answer to this question is in general a fuzzy set that interprets the constrained values of t. To arrive at such answer, the inference engine must apply a chain of CW inference rules, stored in the deduction database, to the related information in knowledgebase. Moreover, in order to apply the deduction rules, the inference engine would need to know the fuzzy subsets which interpret the granule values of the variables in knowledgebase, as shown in the figure.

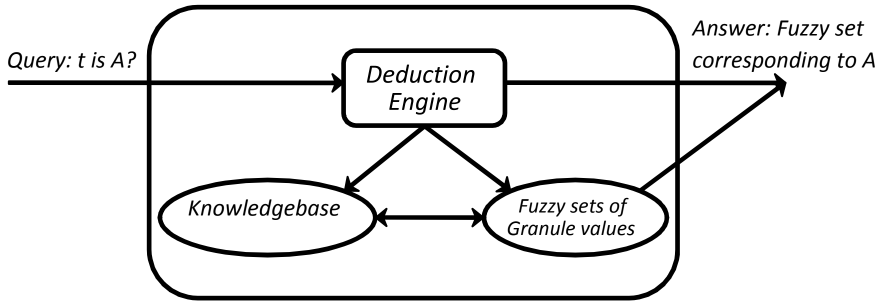

Figure 1. Schematic view of CW inference engine

Hence, reasoning in CW is clearly semantical and computes the answer to a query based on the interpretation of the words in an explanatory database. Formulating such deductions in a purely syntactical way relying solely on the rules of modus ponens and generalization is rather difficult and may not be practical. For example consider Zadeh's inference rules in table 2. In 41 it is stated that the semantic part of the compositional rule of inference can be expressed as the following formula in the basic many-valued predicate logic:

$$(\forall y)(C(y) \equiv (\exists x)(A(x) \,\&\, B(X,Y))$$

If we call the preceding formula *comp_sem*, then the compositional rule may be expressed as the following axiom:

$$comp\_sem \rightarrow ((A(x) \,\&\, B(X,Y) \rightarrow C(Y)) .$$

Where A,B,C are fuzzy predicates, and the universal and existential quantifiers in the basic predicate logic are defined as follows . (Hájek 1998):

$$\forall x\varphi(x) = \bigwedge_{u\in domain-of(x)} (\varphi(u)) \quad , \qquad \exists x\varphi(x) = \bigvee_{u\in domain \;\; of(x)} (\varphi(u))$$

While the semantic part of some of CW rules (such as conjunction rule, and interpolation rule) may be expressed in form of universal and existential quantifiers, this approach fails to express the semantic part of the rules that include fuzzy quantifiers and probabilistic and veristic modalities. For example consider the semantic part of the fuzzy syllogism rule in table 2. This rule computes the interpretation of the fuzzy quantifier, $Q_3(z)$, in the conclusion based on the ones in the premises, $Q_1(w_1)$ and $Q_2(w_2)$. Unlike the preceding rule, the semantic condition of the fuzzy syllogism cannot be expressed syntactically by means of the universal and existential quantifiers, because such quantifiers range over the domain of objects while z, w_1, w_2 are elements of a lattice.

In summary, as was pointed out by Zadeh 46 , CW has aspects, such as fuzzy syllogistic reasoning and constraint modalities, which cannot be modeled in the spirit of classical logics. Our aim in this work is to put forward a formal language that offers a logical syntax and semantics for GCL to support CW applications while relaxing the obligation to make purely syntactic inferences.

4. On Zadeh's Deduction Rules and Their Extensions

When Zadeh formulated CW, he listed a set of basic deduction rules from fuzzy control, fuzzy arithmetic, and syllogism. He also suggested that it would be necessary to extend the deduction database and develop new inference rules to cover more constraint modalities. The formal GCL language, proposed in the previous section, enables us to make such extensions. To this end, we first show that Zadeh's basic rules are consistent with the proposed GCL Language. Then we show, by an example, how new inference rules may be defined using the compositional semantics of GCL.

4.1 Soundness of CW Rules in GCL

In this subsection we describe some of the inference rules characterized by Zadeh for CW and prove that they are valid in GCL. Recall that, unlike the case of classical logic, the CW inference rules are semantical. Given an explanatory database that interprets a set of premises, a CW inference rule typically obtains a fuzzy relation that interprets the consequent of the rule (figure 1). Before proceeding, some definitions are needed.

Definition 1. (Tautology) : a formula φ in GCL is a tautology iff $ts_v^D(\varphi)=1$ for all D and v , where D is an explanatory database and v is a valuation of D.

Definition 2. (Soundness): an inference rule of GCL is written in the form:

$\dfrac{\varphi_1 \; and \; \varphi_2,...and \; \varphi_n}{\varphi_{n+1}}$, where $\varphi_1, \varphi_2, ..., \varphi_{n+1}$ are formula of GCL. such inference rule is sound if and only if the formula: $\varphi_1 \; and \; \varphi_2,...and \; \varphi_n \rightarrow \varphi$ is a tautology. This means $ts_v^D(\varphi_1 \; and \; \varphi_2,...and \; \varphi_n \rightarrow \varphi)=1$ and hence, based on table 4,

$ts_v^D(\varphi_1 \; and \; \varphi_2 ...and \; \varphi_n) \leq \; ts_v^D(\varphi_{n+1})$ for all explanatory databases D and valuations v.

The following lemmas demonstrate the soundness of Zadeh's deduction rules in GCL.

Lemma 1. (Soundness of Conjunction Rule) The conjunction rule is formulated as follows:

$$\dfrac{\begin{array}{c} x \; is \; A \\ y \; is \; B \end{array}}{(x,y) \; is \; C}$$

, where A, B, C are possibilistic granule values and for all explanatory databases: $r_C(u,v)=r_A(u)*r_B(v)$, where $u \in M_x$ and $v \in M_y$. M_x and M_y are domains of x and y, respectively.

To prove the soundness of the conjunction rule, we need to show that for all explanatory database D and valuation v, we have:

$ts_v^D((x \, is \; A) \; and \; (y \; is \; B)) \leq ts_v^D((x,y) \; is \; C))$. Let $v(x)=u$ and $v(y)=v$. Then

$ts_v^D((x \, is \; A) \; and \; (y \; is \; B)) = ts^D(u \, is \; A) * ts^D \; (v \; is \; B) = r_A(u)*r_B(v)$. But

$r_A(u) * r_B(v) = r_C(u,v) = ts^D((u,v)\ is\ C)$. Consequently:

$ts_v^D((x\ is\ A)\ and\ (y\ is\ B)) \le ts_v^D((x,y)\ is\ C))$ for all D and v, and the conjunction rule is sound. Notice that r_C is also the smallest interpretation (in the sense of fuzzy set inclusion) which makes the conjunction rule sound.

Lemma 2. (Soundness of Projection Rule)

The projection rule is stated as follows:

$\dfrac{(x,y)\ is\ A}{x\ is\ B}$, where $r_B(u) = \sup_v(r_A(u,v))$, for all $u \in M_x, v \in M_y$.

We now prove the soundness of projection rule. Notice that

$\forall\, u \in\ M_x, \forall\, v \in\ M_y\ (ts^D((u,v)\ is\ A) = r_A(u,v) \le \sup_v(r_B(u)) = ts^D(u\ is\ B))$

Consequently: $ts_v^D((x,y)\ is\ A) \le ts_v^D((x)\ is\ B))$ and the projection rule is sound.

The soundness of projection rule implies that

$$\forall\, u \in M_x, \forall\, v \in M_y\ (r_B(u) \ge\ r_A(u,v)) \Rightarrow \forall\, u \in M_x(r_B(u) \ge\ \forall\, v \in M_y(r_B(u,v)))$$
$$\Rightarrow \forall\, u \in M_x(r_B(u) \ge\ \sup_v(r_A(u,v))$$

Hence $r_B(u) = \sup_v(r_A(u,v))$ is the smallest interpretation which makes the projection rule sound.

Lemma 3. (Soundness of Extension Principle)

The extension principle is stated as follows:

$\dfrac{\text{x is A}}{\text{f(x) is B}}$, where $r_B(v) = \sup_{u \in M_x \mid f(u)=v}(r_A(u))$

To demonstrate the soundness of extension principle, note that

$\forall\, u \in M_x, \forall v = f(u)\ ts^D(u\ is\ A) = r_A(u) \le \sup_{u \mid f(u)=v}(r_A(u) = r_B(v) = ts^D(vis\ B)$.

Now $r_B(v) = \sup_{u \mid f(u)=v}(r_A(u))$ is the smallest interpretation which makes the extension principle sound. Indeed,

$$\forall v = f(u)\quad (ts^D(u\,is\ A) \le ts^D(f(u)\ is\ B)) \Rightarrow \forall v = f(u)\ (r_B(v) \ge\ r_A(u))$$
$$\Rightarrow r_B(v) \ge \sup_{u \mid f(u)=v} r_A(u)$$

Lemma 4. (Soundness of Compositional Rule of Inference)

The compositional rule of inference finds the image of a fuzzy set in a fuzzy relation. The rule is stated as follows:

$$\frac{\begin{array}{l} x\ is\ A \\ (x,y)\ is\ B \end{array}}{y\ is\ C}$$, where $r_C(v) = \sup_u (r_A(u) * r_B(u,v))$ for all $u \in M_x, v \in M_y$.

We now prove the soundness of compositional rule of inference. Observe that

$$\forall\, u \in M_x, v \in M_y : ts^D((x\ is\ A)\ and\ ((x,y)\ is\ B)) = r_A(u) * r_B(u,v) \le \sup_u (r_A(u) * r_B(u,v))$$

$$\le \sup_u (r_A(u) * r_B(u,v)) = r_C(u,v) = ts^D(y\ is\ C) \quad , for\ all\ D.$$

Now $r_C(v) = \sup_u (r_A(u) * r_B(u,v))$ is the smallest interpretation (or fuzzy set) which makes the compositional rule sound. Indeed, the soundness implies:

$\forall v \in M_y\ (r_c(v) \le \forall u \in M_x\ (r_B(u,v) * r_A(u)))$, hence

$\forall\, v \in M_y\ (\, r_c(v) \le \sup_{u \in M_x}\ (r_B(u,v) * r_A(u))\,)$

Lemma 5. (soundness of Fuzzy graph Interpolation Rule)

The fuzzy interpolation rule is the most used rule in the fuzzy control and it corresponds to the Mamdani system of inference:

$$\frac{\begin{array}{l} \sum_i if\ \ X\ is\ A_i\ \ then\ Y\ is\ B_i \\ X\ is\ A \end{array}}{Y\ is\ B}$$, where $\mu_B(v) = \sup_i (\sup_u ((r_A(u) * r_{A_i}(u)) * r_{B_i}(v))$

The rule is formulated in GCL as follows:

$$\frac{\begin{array}{l} x\ is\ A \\ (x,y)\ isfg\ \{(A_i, B_i)\} \end{array}}{y\ is\ B}$$, where $r_B(v) = \bigvee_i (\sup_{u \in M_X} (r_A(u) * r_{A_i}(u)) * r_{B_i}(v))$, for all

$u \in M_x, v \in M_y$

To prove the soundness of the fuzzy graph rule, we note that

$\forall\, u \in M_x, \forall\, v \in M_y \quad ts^D(u\ is\ A\ and\ (u,v)\ isfg\ \{(A_i, B_i)\}) =$

$r_A(u) * \bigvee_{i'} (r_{A_i}(u) * r_{B_i}(v)\,) = \bigvee_{i'} (r_A(u) * r_{A_i}(u) * r_{B_i}(v)\,) \bigvee_{i'} \le (\sup_{u \in M_X} (r_A(u) * r_{A_i}(u)) * r_{B_i}(v))$

$= \bigvee_i (\sup_{u \in M_X} (r_A(u) * r_{A_i}(u)) * r_{B_i}(v)) = r_B(v) = ts^D(y\ is\ B), \quad for\ all\ D$

Again, we see that that $r_B(v) = \bigvee_i (\sup_{u \in M_X} (r_A(u) * r_{A_i}(u)) * r_{B_i}(v))$ is the smallest fuzzy set for B which makes the fuzzy interpolation rule sound. Indeed,

$$ts^D(x\ is\ B) \geq ts^D(x\ is\ A\ and\ (x,y)\ isfg\ \{(A_i,B_i)\})$$
$$\Rightarrow \forall\, u \in M_x, \forall\, v \in M_y \quad r_B(v) \geq\ (r_A(u) * \bigvee_i (r_{A_i}(u) * r_{B_i}(v)\,)$$
$$\Rightarrow \forall\, v \in M_y\ r_B(v) \geq \forall\, u \in M_x\ (r_A(u) * \bigvee_i (r_{A_i}(u) * r_{B_i}(v)\,)$$
$$\Rightarrow \forall\, v \in M_y\ r_B(v) \geq \sup_u\ (r_A(u) * \bigvee_i (r_{A_i}(u) * r_{B_i}(v)\,) = \bigvee_i (\sup_{u \in M_X} (r_A(u) * r_{A_i}(u)) * r_{B_i}(v))$$

Lemma 6. (Soundness of Fuzzy Syllogism Rule)

The fuzzy syllogism rule, in its classical form, is stated as:

$$\frac{\begin{array}{l} Q_1 A's\ are\ \ B's \\ Q_2 (A \,\&\, B)'s\ are\ C's \end{array}}{Q_3 A's\ are\ (B \,\&\, C)'s} \qquad \text{where } \mathrm{Q}_3 = \mathrm{Q}_1 \times \mathrm{Q}_2$$

This rule can be formulated in GCL as follows*:*

$$\frac{\begin{array}{l} Q_{1\,x:s}(x\ is\ B \mid x\ is\ A) \\ Q_{2\,x:s}(x\ is\ C \mid x\ is\ A\ and\ x\ is\ B) \end{array}}{Q_{3\,x:s}(x\ is\ B\ and\ x\ is\ C \mid x\ is\ A)} \quad \text{, where}$$

$r_{Q3}(u) = \sup_{v_1,v_2}(r_{Q1}(v_1) * r_{Q2}(v_2)), \quad u = v_1 \times v_2$ and $u, v_1, v_2 \in [0{,}1]$. The soundness of the soundness of the fuzzy syllogism rule is demonstrated as follows:

$$ts^D(Q_1 x : s(x\ is\ B \mid x\ is\ A))\ and\ ts^D(Q_2 x : s(x\ is\ C \mid x\ is\ A\ and\ x\ is\ B)) =$$
$$r_{Q1}\Big(\frac{\sum_{u \in M_S} r_A(u) * r_B(u)}{\sum_{u \in M_S} r_A(u)}\Big) * r_{Q1}\Big(\frac{\sum_{u \in M_S} r_A(u) * r_B(u) * r_C(u)}{\sum_{u \in M_S} r_A(u) * r_B(u)}\Big)$$

$$ts^D(Q_{3\,x:s}(x\ is\ B\ and\ x\ is\ C \mid x\ is\ A) = r_{Q3}\Big(\frac{\sum_{u \in M_S} r_A(u) * r_B(u) * r_C(u)}{\sum_{u \in M_S} r_A(u)}\Big)\text{, let}$$

$$v_1 = \frac{\sum_{u \in M_S} r_A(u) * r_B(u)}{\sum_{u \in M_S} r_A(u)} \ \text{ and } \ v_2 = \frac{\sum_{u \in M_S} r_A(u) * r_B(u) * r_C(u)}{\sum_{u \in M_S} r_A(u) * r_B(u)}\text{, and}$$

$$u = \frac{\sum_{u \in M_S} r_A(u) * r_B(u) * r_C(u)}{\sum_{u \in M_S} r_A(u)} \ \text{ then } u = v_1 * v_2 \text{ , thus}$$

$$r_{Q1}(v_1) * r_{Q1}(v_2) \leq \sup_{v_1,v_2}(r_{Q1}(v_1) * r_{Q1}(v_2) = r_{Q3\ \ u = v_1 \times v_2}(u)\,)$$

Notice that r_{Q3} is also the smallest interpretation (or fuzzy set) that makes the fuzzy syllogism rule sound (the proof is similar to that of the extension principle).

4.2 Extending Zadeh's Deduction Rules

The compositional semantics of GCL along with the notion of soundness provides a basis to describe new inference rules that are semantically valid. This is illustrated via an example. Let us assume that we would like to find the average risk of developing a breast cancer for a person, knowing that the average risk is, in general, a fuzzy function of the person's age. Suppose:

> For most x, if age(x) is in 30s then average-riskbc(x) is about 0.4 %, if age(x) is in 40s then average-riskbc(x) is about 1.5 %, and if age(x) is in 50s or greater, then average-riskbc(x) is about 2.5%

Based on this information, we are interested to find the average risk of breast cancer for a young person. In other words, we would like to perform the following deduction:

$$\frac{\begin{array}{l} most_{x:people}((age(x), riskBC(x))\ isfg\ \{(30's, 0.\overset{*}{4}\%), (40's, 1.\overset{*}{5}\%), (\geq 50's, 2.\overset{*}{5}\%)\}) \\ age(y)\ is\ young \end{array}}{riskBc(y)\ is\ ?}$$

The abstract form (protoform) of this deduction is:

$$\frac{\begin{array}{l} Q_{x:s}(\ (f(x), g(x))\ isfg\ \{(A_i, B_i)\}) \\ f(y)\ is\ A \end{array}}{g(y)\ is\ B?}$$

Where the interpretations of A_is (r_{Ai}), B_is (r_{Bi}), and A (r_{Ai}) are given in some explanatory database and the inference rule would have to derive the interpretation of B (r_B). The above rule can be viewed as an instance of a quantified fuzzy graph interpolation rule. A variant of this rule (called the usuality-qualified fuzzy graph interpolation rule) is nominated by Zadeh as an instance of an inference rule that has to be developed and added to the deduction database.

Based on definition 2, this rule is sound in GCL if and only if:

$$ts_v^D(g(y)\ is\ B) \geq ts_v^D(\ Q_{x:s}(\ (f(x), g(x))\ isfg\ \{(A_i, B_i)\})\ ) * ts_v^D(f(y)\ is\ A)$$

Thus we need to find the smallest fuzzy set for B**,** r_B, which makes the above rule sound:

$$\forall v \; (\; r_B(v) \geq r_Q\Big(\frac{\sum_{w \in Ms} \bigvee_{i} (r_{A_i}(f(w)) * r_{B_i}(g(w)))}{\sum_u 1} \Big) * r_A(f(u)) \,,\; for\, all\; u \in M_s, v = g(u) \,)$$

Hence, $\forall v \;\; r_B(v) = r_Q\Big(\frac{\sum_{u \in M_S} \bigvee_{i} (r_{A_i}(f(u)) * r_{B_i}(g(u)))}{| M_S |} \Big) * \sup_{u \in M_s, v = g(u)} r_A(f(u))$

is the smallest interpretation for *B* which makes the inference rule sound.
Accordingly, the answer to the preceding example is calculated as follows:
$\forall v \in M_{RISK} \;\; r_B(v) = r_{most}(c) * \sup_{u \in M_{PEOPLE}, v = riskBC(u)} (age(u))$, where

$$c = \frac{\sum_{u \in PEOPLE} \bigvee_{i} (r_{A_i}(age(u)) * r_{.B_i}(riskbc(u)))}{| M_{People} |} \quad , \; A_i = \{20s, 30s, \geq 50s\} \text{ and}$$

$B_i = \{.4^*, 1.5^*, 2.5^*\}$

5. Summary and Future Work

The generalized constraint language is the essence of CW. In this paper, we argued that further theoretical and practical advancements of CW demand formalization of GCL. We defined recursive syntax for GCL and characterized the test score semantics in the spirit of model theory. We proved the soundness of Zadeh's basic inference rules in the proposed language and pointed out briefly how new sound rules may be developed and added to the deduction database.

The GCL language defined here includes formula with fuzzy quantifiers as well as possibilistic, veristic, and fuzzy graph semantics modalities.

Our future work must extend the language to take account of other modalities such as probability, rough sets, random sets, and bimodal (possibility/probability) distributions. The two major future objectives would be (1) to formulate a set of axioms that is both complete and consistent, and (2) to identify a set of canonical forms that every formula in GCL may be reduced to, and to develop an automated inference mechanism on top of GCL.

References

[1] L. A. Zadeh, "Fuzzy logic = computing with words," *Fuzzy Systems, IEEE Transactions on,* vol. 4, no. 2, pp. 103-111, 1996.

[2] J. Mendel, L. Zadeh, E. Trillas *et al.*, "What Computing with Words Means to Me [Discussion Forum]," *Computational Intelligence Magazine, IEEE,* vol. 5, no. 1, pp. 20-26, 2010.

[3] J. Kacprzyk, and S. Zadrozny, "FQUERY for Access: towards human consistent querying user interface," in Proceedings of the 1996 ACM symposium on Applied Computing, Philadelphia, Pennsylvania, United States, 1996, pp. 532-536.

[4] J. Kacprzyk, and S. Zadrozny, "Computing with words in intelligent database querying: standalone and Internet-based applications," *Information Sciences,* vol. 134, no. 1-4, pp. 71-109, 2001.

[5] J. Kacprzyk, and S. Zadrozny, "Computing With Words Is an Implementable Paradigm: Fuzzy Queries, Linguistic Data Summaries, and Natural-Language Generation," *Fuzzy Systems, IEEE Transactions on,* vol. 18, no. 3, pp. 461-472, 2010.

[6] F. Herrera, S. Alonso, F. Chiclana *et al.*, "Computing with words in decision making: foundations, trends and prospects," *Fuzzy Optimization and Decision Making,* vol. 8, no. 4, pp. 337-364, 2009.

[7] B. Juliano, "Cognitive sciences and computing with words.," *Computing with Words*, Wiley Series on Intelligent Systems P. p. Wang, ed., pp. 235-250: A Wiley-Interscience publication, 2001.

[8] R. R. Gudwin, and F. A. C. Gomide, "Object Networks: A Computational Framework to Compute with Words," *Computing with Wrods in Information/Intelligent Systems*, Studies in Fuzziness and Soft Computing L. A. Zadeh and J. Kacprzyk, eds., pp. 443-478: Springer-Verlag, 1999.

[9] M. Reformat, and C. Ly, "Ontological approach to development of computing with words based systems," *Int. J. Approx. Reasoning,* vol. 50, no. 1, pp. 72-91, 2009.

[10] V. Raskin, and J. M. Taylor, "The (not so) unbearable fuzziness of natural language: The ontological semantic way of computing with words." pp. 1-6.

[11] E. S. Khorasani, S. Rahimi, and B. Gupta, "A Reasoning Methodology for CW-Based Question Answering Systems," in Proceedings of the 8th International Workshop on Fuzzy Logic and Applications, Palermo, Italy, 2009, pp. 328-335.

[12] R. R. Yager, "Approximate Reasoning as a basis for Computing with Words," *Computing with Words in Information/Intelligent Systems 1 Foundations*, Studies in Fuzziness and Soft Computing 13, L. A. Zadeh and J. Kacprzyk, eds., pp. 50-77: Springer-Verlag, 1999.

[13] R. R. Yager, “Knowledge trees and protoforms in question-answering systems: Special Topic Section on Soft Approaches to Information Retrieval and Information Access on the Web,” *J. Am. Soc. Inf. Sci. Technol.,* vol. 57, no. 4, pp. 550-563, 2006.

[14] L. Feilong, and J. M. Mendel, “Encoding Words Into Interval Type-2 Fuzzy Sets Using an Interval Approach,” *Fuzzy Systems, IEEE Transactions on,* vol. 16, no. 6, pp. 1503-1521, 2008.

[15] J. M. Mendel, and W. Dongrui, “Perceptual Reasoning for Perceptual Computing,” Fuzzy Systems, IEEE Transactions on, vol. 16, no. 6, pp. 1550-1564, 2008.

[16] B. Türksen, “Type 2 representation and reasoning for CWW,” Fuzzy Sets Syst., vol. 127, no. 1, pp. 17-36, 2002.

[17] I. B. Türkşen, “Meta-Linguistic axioms as a foundation for computing with words,” Information Sciences, vol. 177, no. 2, pp. 332-359, 2007.

[18] R. R. Yager, “On the retranslation process in Zadeh's paradigm of computing with words,” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 34, no. 2, pp. 1184-1195, 2004.

[19] R. Kowalczyk, "On linguistic approximation of subnormal fuzzy sets," in Fuzzy Information Processing Society - NAFIPS, 1998 Conference of the North American, pp. 329-333, 1998.

[20] A. Dvořák, “On linguistic approximation in the frame of fuzzy logic deduction,” Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 3, no. 2, pp. 111-116, 1999.

[21] T. Whalen and B. Schott, "Empirical comparison of techniques for linguistic approximation," in IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, vol.1, pp. 93-97, 2001.

[22] R. Scott and T. Whalen, "Linguistic approximation of nonconvex membership functions using “...Except...” or “...Or...”," in Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American, pp. 388-391, 2000.

[23] D. Wu and J. M. Mendel, "A vector similarity measure for linguistic approximation: Interval type-2 and type-1 fuzzy sets," Inf. Sci., vol. 178, pp. 381-402, 2008.

[24] S. Zahan, "Neural approach to linguistic approximation of fuzzy sets," in IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, 2001, pp. 2693-2697 vol.5.

[25] F. Herrera and L. Martinez, "A 2-tuple fuzzy linguistic representation model for computing with words," Fuzzy Systems, IEEE Transactions on, vol. 8, pp. 746-752, 2000.

[26] F. Herrera, et al., "A Fuzzy Linguistic Methodology to Deal With Unbalanced Linguistic Term Sets," Fuzzy Systems, IEEE Transactions on, vol. 16, pp. 354-370, 2008.

[27] J. H. Wang and J. Hao, "An Approach to Computing With Words Based on Canonical Characteristic Values of Linguistic Labels," Fuzzy Systems, IEEE Transactions on, vol. 15, pp. 593-604, 2007.

[28] . Lawry, "An alternative approach to computing with words," Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol. 9, pp. 3-16, 2001.

[29] . H. Rubin, "Computing with words," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 29, pp. 518-524, 1999.

[30] H. Wang and D. Qiu, "Computing with words via Turing machines: a formal approach," Fuzzy Systems, IEEE Transactions on, vol. 11, pp. 742-753, 2003.

[31] M. Ying, "A formal model of computing with words," Fuzzy Systems, IEEE Transactions on, vol. 10, pp. 640-652, 2002.

[32] Y. Cao and M. Ying, "Retraction and Generalized Extension of Computing With Words," Fuzzy Systems, IEEE Transactions on, vol. 15, pp. 1238-1250, 2007.

[33] N. Fuchs, Ed., *Controlled Natural Language* (Lecture Notes in Computer Science. Springer Verlag, 2010.

[34] L. A. Zadeh, "Toward a generalized theory of uncertainty (GTU): an outline," *Inf. Sci. Inf. Comput. Sci.,* vol. 172, pp. 1-40, 2005.

[35] J. Kacprzyk, "Human-centric Computing Paradigms, Computing with Words, and Intelligent Decision Support Systems," in *EUROCON, 2007. The International Conference on "Computer as a Tool"*, 2007, pp. 2790-2790.

[36] L. A. Zadeh, "Test-score semantics as a basis for a computational approach to the representation of meaning," *Lit. Linguist. Comput.,* vol. 1, pp. 24-35, 1986.

[37] P. Dwinger, *Introduction to Boolean algebras*. Würzburg,: Physica-Verlag, 1961.

[38] N. Galatos, *Residuated lattices : an algebraic glimpse at substructural logics*, 1st ed. Amsterdam ; Boston: Elsevier, 2007.

[39] M. Manzano. Extensions of first order logic, Cambridge University Press, 1996.

[40] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I, II, III," *Information Sciences,* vol. 8, pp. 199-249, 1975.

[41] P. Hájek, *Metamathematics of Fuzzy Logic*: Dordrecht: Kluwer, 1998.

[42] V. Novák*, et al.*, *Mathematical principles of fuzzy logic*. Boston: Kluwer Academic, 1999.

[43] M. Ying, "Linguistic quantifiers modeled by Sugeno integrals," *Artif. Intell.,* vol. 170, pp. 581-606, 2006.

[44] S. Gottwald, "Mathematical fuzzy logic as a tool for the treatment of vague information," *Inf. Sci.,* vol. 172, pp. 41-71, 2005.

[45] P. Cintula*, et al.*, "Triangular norm based predicate fuzzy logics," *Fuzzy Sets Syst.,* vol. 161, pp. 311-346, 2010.

[46] A. L. Zadeh, "In Fuzzy Logic Technology and Applications," R. J. Marks-II, Ed., ed: IEEE Technical Activities Board, 1994.