Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 33–39
ISBN 978-83-60810-22-4

# CWJess: Implementation of an Expert System Shell for Computing with Words

Elham S. Khorasani, Shahram Rahimi, Purvag Patel, Daniel Houle
Department of Computer Science
Southern Illinois University Carbondale
Carbondale, IL, USA

*Abstract*—**Computing with Words (CW) is an emerging paradigm in knowledge representation and information processing. It provides a mathematical model to represent the meaning of imprecise words and phrases in natural language, and to perform reasoning on perceptual knowledge. This paper describes a preliminary extension to Jess, CWJess, which allows reasoning in the framework of Computing with Words (CW). The resulting inference shell significantly enhances the expressiveness and reasoning power of fuzzy expert systems and provides a Java API which allows users to express various types of fuzzy concepts, including: fuzzy graphs, fuzzy relations, fuzzy arithmetic expression, and fuzzy quantified propositions. CWJess is fully integrated with jess and utilizes jess Rete network to perform a chain of reasoning on fuzzy propositions**

*Index Terms*—**Computing with Words; fuzzy Logic; Expert systems; knowledge representation**

## I. Introduction

HUMAN mind has a limited capability for processing a huge amount of detailed information in his environment; thus, to compensate, the brain groups together the information it perceives by its similarity, proximity, or functionality and assigns to each group a name or a "word" in natural language. This classification of information allows human to perform complex tasks and make intelligent decisions in an inherently vague and imprecise environment without any measurements or computation. Inspired by this human capability, Zadeh introduced the machinery of CW as a tool to formulate human reasoning with perceptions drawn from natural language and argued that the addition of CW theory to the existing tools gives rise to the theories with enhanced capabilities to deal with real-world problems and makes it possible to design systems with higher level of machine intelligence [1][2]. To do this, CW offers two principal components, (1) a language for representing the meaning of words taken from natural language, this language is called the Generalized Constraint Language (GCL), and (2) a set of deduction rules for computing and reasoning with words instead of numbers. CW is rooted in fuzzy logic; however, it offers a much more general methodology for fusion of natural language propositions and computation with fuzzy variables. CW inference rules are drawn from various fuzzy domains, such as fuzzy logic, fuzzy arithmetic, fuzzy probability, and fuzzy syllogism. This paper reports a preliminary work on the implementation of a CW inference system on top of JESS expert system shell (CWJess) . The CW reasoning is fully integrated with JESS facts and inference engine and allows knowledge to be specified in terms of GCL assertions.

The current fuzzy logic expert system shells, such as: fuzzyclips [3], fuzzyjess [4], FLOPS [5], and Frill [6] are much devoted to implementing Mamdani inference system and have left out reasoning with other fuzzy concepts such as: fuzzy relations, fuzzy arithmetic, fuzzy quantifiers, and fuzzy probabilities. This paper presents a roadmap to implement a CW expert system shell capable of representing and reasoning with such concepts. The resulting CWJess expert system shell would allow users to express their knowledge in form of fuzzy quantified propositions, discrete fuzzy relations, and fuzzy arithmetic expressions as well as fuzzy if-then rules and enables them to perform advanced fuzzy reasoning.

## II. Preliminary: Computing with Words

This section provides a very brief introduction to computing with words, the generalized constraint language, and CW inference rules. More detailed information can be found in Zadeh's paper [7].

The core of CW is to represent the meaning of a proposition in form of a generalized constraint (GC). 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, appearance, and etc. 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 term R, and the small r shows the semantic modality of this constraint, i.e., how X is related to R. Various modalities are introduced in the literature of CW, among them are:

- possibility (*r=blank*)*:* where R is a fuzzy set which denotes the possibility distribution of X [8] , e.g., "X is large".
- fuzzy graph (*r=fg*) : where *X* is a function of another variable, say *Y,* and *R* is a fuzzy estimation (or granulation) of that function. This modality corresponds to a collection of fuzzy if-then rules that share the same variables in their premises and consequences. e.g., "X=f(y) isfg (small × large + medium × medium + large × small)", which is equivalent to three fuzzy rules: *if Y is small then X is large, if Y is medium then X is medium* and *if Y is large then X is small.*
- probability (*r=p)* : where *X* is a random variable and *R* is the probability distribution of *X*, e.g., "(X is large) isp likely ".
- usuality *(r=*u): where *X* is a random variable and *R* is the usual ( or typical) value of X, e.g., "X isu big"

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 1 lists instances of inference rules introduced for GCL. As shown in this table, each rule has a symbolic part and a computational part. The symbolic part shows the general abstract form of the GCs that appear in the premises and conclusion of a rule, while the computational part calculates the fuzzy value of the consequent of the rule based on its premises.

TABLE I. INSTANCES OF CW INFERENCE RULES

| Inference rule | Symbolic part | Computational part |
|---|---|---|
| Conjunction Rule | $\dfrac{x\ is\ A \quad y\ is\ B}{(x,y)\ is\ C}$ | $\mu_C(u,v)=\mu_A(u)*\mu_B(v)$ |
| Extension Principle | $\dfrac{\text{X is A}}{\text{f(X) is B}}$ | $\mu_B(z)=\sup_U(\mu_A(u))$ <br> subject to : $z=f(u)$ |
| Compositional Rule of Inference | $\dfrac{X\ is\ A \quad (Y,X)\ is\ B}{Y\ is\ C}$ | $\mu_C(v)=\sup_u(\mu_A(u_i)*\mu_B(v,u))$ |
| Fuzzy graph Interpolation | $\dfrac{\sum_i if\ X\ is\ A_i\ then\ Y\ is\ B_i \quad X\ is\ A}{Y\ is\ B}$ | $\mu_B(v)=\sup_i(m_i*B_i)$ <br> $m_i=\sup_u(\mu_A(u)*\mu_{A_i}(u))$ <br> $i=1...n$ |
| Fuzzy Syllogism | $\dfrac{Q_1A's\ are\ B's \quad Q_2(B)'s\ are\ C's}{Q_3A's\ are\ (C)'s}$ | $\mu_{Q_3}(z)=\sup(\mu_{Q1}(w_1)*\mu_{Q2}(w_2))$ <br> subject to : $z=w_1\times w_2$ <br> $w_1$,$w_2$, and z are the universes of discourse of $Q_1$, $Q_2$,and $Q_3$ , respectively |

## III. INCORPPRATING CW REASONING IN JESS

Figure 1 shows a schematic view of CWJess Inference System.

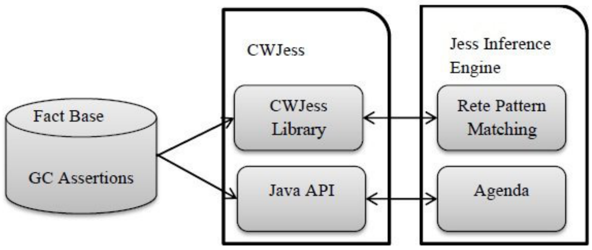



Figure 1. The schematic view of CWJess Inference System

CWJess consists of two components: the Java API and the CWJess library. The Java API is the core element which implements CW concepts, such as linguistic variables, various types of generalized constraints, and *the computational part* of CW inference rules. The CWJess library consists of a list of production rules which define the *symbolic part* of CW inference rules. When a user adds a GCL assertion to the fact base, Jess Rete algorithm checks the symbolic part of CW inference rules in CWJess library to find a production rule whose premises match with the facts and place the consequents of such rules into the agenda. When a CW inference rule is fired, its consequent calls the java method which performs the computational part of the rule and asserts the result back to the fact base.

### *A. GC Assersions in CWJess Fact Base*

The CWJess fact base consists of a set of GC assertions. A GC assertion may have one of the following forms:

(1) a simple generalized constraint,

(2) a fuzzy graph,

(3) a quantified generalized constraint

(4) an arithmetic generalized constraint.

Different forms of generalized constraints are implemented as java classes in the java API. The GC facts are then created as jess shadow facts connected to the corresponding java object.

A simple generalized constraint consists of four elements:

- An atomic or composite linguistic variable, such as: "age", "size", "weight", "price", "distance", etc.
- A list of objects of linguistic variables, for example object "Mary" for the linguistic variable "Age", or object "McDonald's", for the linguistic variables "service-quality".
- A semantic modality. The semantic modality may be: "possibistic", "probabilistic" "verisitc" . The semantic modality is by default possibilistic. At this point, we have only implemented the possibilistic semantic modality. The implementation of probabilistic and veristic modalities and their combination with possibilistic modality requires the availability of more complex theories and will be considered as future work of this study.

- A linguistic value associated with the linguistic variable and constraint its values.

For example, the generalized constraint: *service-quality(McDonald's) is good"* consists of the linguistic variable: *service quality,* object: *McDonald's*, semantic modality: *possibilistic,* and linguistic value: *good.*

Two types of linguistic variables are considered: *atomic* and *composite*. The atomic linguistic variable consists of a name ( e.g., oil-price), a unit ( e.g., $ per gallon), a range of values (e.g, 1.0-5.0), and a set of linguistic terms (e.g., "cheap", "mid-price", and "expensive"). The linguistic terms are defined using a term name and a membership function that is a fuzzy set over the range of values for the linguistic variable. The membership functions may be defined to have a standard shape, such as : triangular, trapezoid, Gaussian, Pi shape, S shape, Z shape, crisp interval, or it may be a piecewise linear, or a discrete function specified by a set of singletons. Figure 2 shows three linguistic terms with different type of membership function for the linguistic variable: "oil-price".

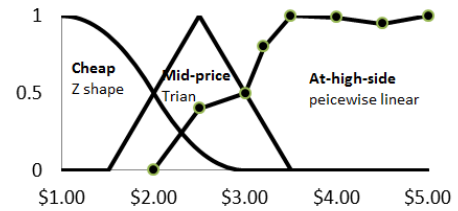



Figure 2. Examples of membership functions for the linguistic variable: oil_price

A composite linguistic variable is a fuzzy relation between two or more linguistic variables. To reduce the complexity of computation, the membership function of composite linguistic variable is assumed to be discrete. The inclusion of continuous fuzzy relation requires complex non-linear computation, and will be considered as the future developments of CWJess. Figure 3 shows the membership function for the linguistic term "petite" of the composite linguistic variable, "size", where "size" is a fuzzy relation of two atomic linguistic variables: "height" and "weight".

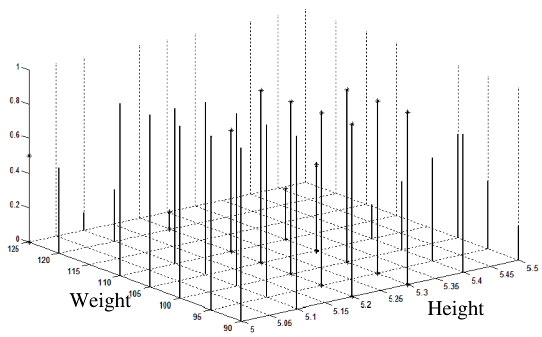



Figure 3. The membership function for th linguistic term: "petite" of the composite linguistic variable "size".

A linguistic value in a generalized constraint may be modified by a fuzzy modifier. The modifiers implemented in CWJess are: *not, more-or-less, a-little-more, slightly-more, somewhat, very, extremely, indeed,* and their combinations.

A Fuzzy graph is a collection of fuzzy if…then rules in which all the premises and the conclusion share the same linguistic variables. The general form of a fuzzy graph in CWJess is as follows:

$$\sum_{i=1\ldots n} if\ (GC_{i1} \wedge GC_{i2} \ldots \wedge GC_{in})\ then\ GC_{ik}$$

Where $GC_{i1}, GC_{i2}, \ldots, GC_{in}$ denote the generalized constraints in the premise of the rule, and $GC_{ik}$ is the generalized constraint in the consequent of the rule. For example:

*if age(x) is young $\wedge$ health(x) is good then insurance(x) is very low*
*if age(x) is middle-age $\wedge$ health(x) is good then insurance (x) is average*
*if age(x) is old $\wedge$ health(x) is moderate then insurance(x) is relatively high*
*if age(x) is old $\wedge$ health(x) is poor then insurance(x) is very high*

Although fuzzy graph may seem as the conjunction of a set of fuzzy if-then rules (or fuzzy implications), it has a very different meaning. From the mathematical point of view, a fuzzy graph expresses a functional dependency between two or more linguistic variables and provides a fuzzy description of such function when the point to point data is not available. Hence, fuzzy graph has a completely different semantics than the fuzzy implication and must be treated differently by the inference engine.

A quantified generalized constraint (QGC) consists of a fuzzy quantifier and a generalized constraint with an anonymous object (variable) which is bounded by the quantifier. For example: "most$_x$ price(x) is expensive" is a quantified generalized constraint in which *x* is an anonymous object bounded by the quantifier "most". Fuzzy quantifiers are defined as fuzzy sets and are assigned a membership function: $\mu : [0,1] \rightarrow [0,1]$ . In this version of CWJess we limited ourselves to monotone increasing quantifiers, such as : "most", "many", "several", "a few", etc., for applying Zadeh's syllogism reasoning [9].

We have implemented two types of QGC: unary and binary. A unary QGC puts constraint on the proportion of objects that satisfy a single generalized constraint, e.g., "most $_x$ (age(x) is young)" whereas a binary QGC imposes a constraint on the proportion of number of objects that satisfy two generalized constraints to the number of objects that satisfy one. For example: *"most$_x$ (* age(x) is young, health(x) is good)" is a binary QGC which states that the number of "x" s who are young and healthy over the number of "x"s who are young, is most.

An arithmetic generalized constraint (AGC) is a generalized constraint which has an arithmetic fuzzy expression as its linguistic value. The arithmetic expression may consist of linguistic terms or other linguistic variables. For example: "gas-price(Europe) is gas-price(US) + approximately $4 per gallon ", which states that the gas price in Europe is approximately $4 more per gallon than in the United States.

### *B. Implementation of CW Inference Rules*

The computational part of all CW rules is implemented in the java API as the following java methods. The result of the each method is a normalized fuzzy set.

- **Conjunction method**: The conjunction method takes a number of linguistic terms associated with a linguistic variable, and computes their minimum intersection. (The minimum operation can be easily replaced with a t-norm [10])
- **Disjunction method**: The disjunction method takes a number of terms associated with a linguistic variable and computes their maximum union. The maximum operation can be replaced with a t-conorm.
- **Compositional method**: The compositional method takes two linguistic terms: one associated with a composite linguistic variable and the other one associated with an atomic linguistic variable, which occurs in the composite linguistic variable, and calculates their max-min composition. As an example let us assume that the composite linguistic variable size of a woman consists of the atomic linguistic variables, height and weight, i.e.,

$$size(x) = (height(x), weight(x))$$

let's also assume that the membership function for the term "petite" associated with size, and the term "about 5.2" associated with the height are given. The compositional method calculates the membership function of the weight of a petite woman who is about 5.2 tall, as shown in the following figure.

petitie

| height \ weight | 90 | 100 | 115 | 120 | 130 |
|---|---|---|---|---|---|
| 5 | 1 | 1 | 1 | 0.5 | 0.2 |
| 5.1 | 1 | 1 | 0.9 | 0.3 | 0.1 |
| 5.2 | 1 | 1 | 0.7 | 0.1 | 0 |
| 5.3 | 1 | 1 | 0.5 | 0.3 | 0 |
| 5.4 | 0.8 | 0.6 | 0.4 | 0.2 | 0 |
| 5.5 | 0.6 | 0.4 | 0.2 | 0 | 0 |

about 5.2

| height | |
|---|---|
| 5 | 0.1 |
| 5.1 | 0.5 |
| 5.2 | 1 |
| 5.3 | 0.5 |
| 5.4 | 0.1 |
| 5.5 | 0 |

min

| weight | | | | | | |
|---|---|---|---|---|---|---|
| 90 | 0.1 | 0.5 | 1 | 0.5 | 0.1 | 0 |
| 100 | 0.1 | 0.5 | 1 | 0.5 | 0.1 | 0 |
| 115 | 0.1 | 0.5 | 0.7 | 0.5 | 0.1 | 0 |
| 120 | 0.1 | 0.3 | 0.1 | 0.3 | 0.1 | 0 |
| 130 | 0.1 | 0.1 | 0 | 0 | 0 | 0 |

max

The resulting fuzzy set for weight

| weight | |
|---|---|
| 90 | 1 |
| 100 | 1 |
| 115 | 0.7 |
| 120 | 0.3 |
| 130 | 0.1 |

Figure 4. Compositional rule of inference. Composition of the fuzzy relation "petite" with the fuzzy set "about 5.2". the

- **Fuzzy graph Interpolation method**: This method corresponds to Mamdani method of inference. It takes a fuzzy graph as well as a set of linguistic terms associated with the linguistic variables in the premises of a fuzzy graph and computes the corresponding fuzzy set for the linguistic variable in the conclusion. Unlike Mamdani inference method, where the input is a singleton, the fuzzy interpolation method accepts a linguistic term (fuzzy set) as input and, for each rule, computes the match score as the degree of similarity between the input fuzzy set and the ones in the premise of the rule. For instance, consider the fuzzy graph mentioned as an example in the previous subsection, and let us assume that the input it: "age is about 55" and "health is "excellent", then the fuzzy graph interpolation method would calculate the fuzzy set for the linguistic variable "insurance" as illustrated in figure 5.

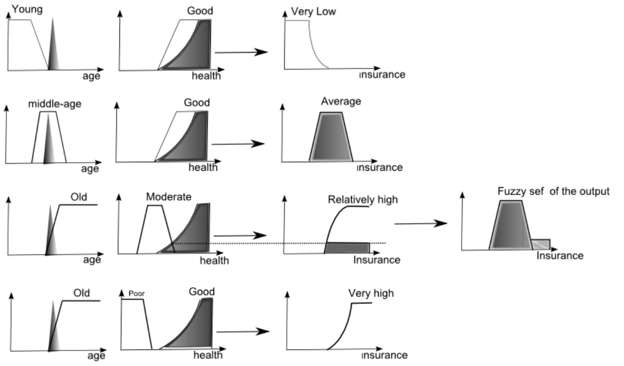
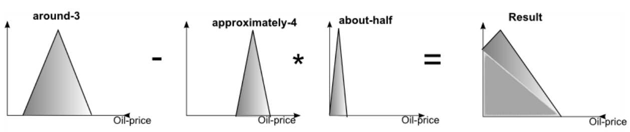



Figure 5. Example of fuzzy graph interpolation rule. The gray areas in the first and second column show the memebrship functions of "about 55", and "excellent" for the linguistic variables "age" and "health", respectively.

- **Fuzzy Extension method:** The fuzzy extension method, takes an arithmetic expression on a number of linguistic terms associated with a linguistic variable, and returns the fuzzy set resulted from performing the arithmetic operations. The linguistic terms appear in the arithmetic expressions are required to have a normal convex fuzzy set.

The implementation of the extension principle is not trivial and involves nonlinear optimization. Thus approximation methods are usually used to obtain the membership function of the resulting fuzzy set. A common practice is to discretize the membership interval [0,1] into a finite number of values and, for each value, take the $\alpha$-cut of all the operands. The arithmetic operations then may be performed on the resulting intervals, using the interval arithmetic, in order to come up with the $\alpha$-cut of the output fuzzy set. Finally, these $\alpha$-cuts are put together to obtain the output of the arithmetic operations. This approach can be efficiently implemented and provides a good approximation to the exact solution of the extension principle [11]. Figure 6 demonstrates the $\alpha$-cut method for computing the result of the arithmetic expression: "around3 + approximately4 * about-half", for the linguistic variable "oil_price".




| alpha | around 3 | approximately4 | about half | result |
|---|---|---|---|---|
| 0 | [1, 5] | [3, 4] | [0, 1] | [-3, 5] |
| .1 | [1.2, 4.8] | [3.1,4.9] | [.05, .95] | [-3.45, 4.65] |
| .2 | [1.4, 4.6] | [3.2, 4.8] | [.1, .9] | [-2.92, 4.28] |
| .3 | [1.6, 4.4] | [3.3, 4.7] | [.15, .85] | [-2.35, 3.91] |
| .4 | [1.8, 4.2] | [3.4, 4.6] | [.2, .8] | [-1.88, 3.52] |
| .5 | [2, 4] | [3.5, 4.5] | [.25, .75] | [-1.37, 3.13] |
| .6 | [2.2, 3.8] | [3.6, 4.4] | [.3, .7] | [-.88, 2.72] |
| .7 | [2.4, 3.6] | [3.7, 4.3] | [.35, .65] | [-.39, 2.31] |
| .8 | [2.6, 3.4] | [3.8, 4.2] | [.4, .6] | [.08, 1.88] |
| .9 | [2.8, 3.2] | [3.9, 4.1] | [.45, .55 | [.55, 1.45] |
| 1.0 | [3, 3] | [4, 4] | [.5, .5] | [1, 1] |

Figure 6. An example of alpha-cut implementation of the extension principle. The table shows the $\alpha$ -cuts of the operands and the output of the arithmetic expression

- **Fuzzy Syllogism:** The fuzzy syllogism method takes two monotone increasing fuzzy quantifiers, and calls the extension method to calculate and return their multiplication.

The symbolic parts of the above rules are implemented as production rules in the CWJess library. For example, the symbolic part of the conjunction rule may be defined as follows:

```
;;  The conjunction rule
(defrule conjunction
(declare (salience 100))
?c1<-(GC (linguistic_var ?x ) (object ?a)
   (linguistic_value ?w))
?c2<-(GC (linguistic_var ?x)  (object ?a)
   (linguistic_value ?z&:(<> ?z ?w)))
=>
(assert (GC (linguistic_var ?x) (object ?a)
   (value (CWRULEs.conjunction ?w ?z))))
(retract ?c1 ?c2))
```

This rule states that if there are two different generalized constraints in the fact base sharing the same linguistic variable and the same object, they are combined to one generalized constraint whose value is the conjunction of the linguistic values of the premises. GC is a template automatically created from the generalized constraint class in the java API. GCWRULES is a java class which implements the computational parts of CW rules as described earlier.

As another example, consider the compositional rule of inference, the symbolic part of this rule is defined in CWJess library as follows:

```
;; The compositional rule of inference.
(defrule composition
(GC (linguistic_var  ?u &:(instanceof  ?u
     COMPOSITE_LINGUISTIC_VARIABLE))
     (object ?o) (linguistic_value ?w))
(GC (linguistic_var  ?x &:(instanceof ?x
     ATOMIC_LINGUISTIC_VARIABLE))
     (object ?o) (linguistic_value ?z))
(test (occurs  ?x ?u )
 =>
(assert (GC (linguistic_var (CWRULES.composedVar ?u
    ?x)) (object ?o)(linguistic_value
(CWRULES.composedValue ?w ?z ))))
```

This rule states that if there are two generalized constraints in the fact base, one with a composite linguistic variable ?u, and another one with an atomic linguistic variable ?x , and ?x occurs in ?u, then assert a new generalized constraint which is the composition of the two generalized constraints in the premises. The methods composedVar, and composedValue are static methods defined in CWRULES and return, respectively, the linguistic variable and the fuzzy set resulted from the composition.

The symbolic parts of the fuzzy graph interpolation rule, fuzzy syllogism, and fuzzy extension principle, are defined in a similar way but are excluded from the paper due to their complexity and length.

## IV. An Example

To demonstrate the CWJess we present a simple example.

Suppose that we would like to encode the following information in a CWJess fact base:

- *The average chance that a woman is diagnosed by breast cancer depends directly on her age. The younger a woman is, the lower is her risk of developing a breast cancer.*
- *Many obese women have higher risk of developing a breast cancer and a Mammogram is strongly recommended for most women at high risk of breast cancer.*
- *Mary has a son who is about 15 . She gave birth to her son when she was in her 20's. Also She is few years younger than Ann who is in her mid-50.*

The dependency between the age and the risk of breast cancer should be represented as a fuzzy graph. To create a fuzzy graph, the related linguistic variables and their associated terms must be first defined. In the following Jess code, "trap-mf", "tri-mf", "z-mf", "p-imf", "s-mf", and "interval-mf" denote trapezoid, triangular, z shape, pi shape, s shape, and the crisp interval membership functions, respectively.

```
;; Defining the linguistic variables and terms that
;; appear in the fuzzy graph
(bind ?age (new ATOMIC_LINGUISTIC_VARIABLE "age"
"year" 0 120 ))
(?age addTerm " young"  "(0,1) (25,1) (50,0)" )
(?age addTerm "middle-age" "trap-mf" "30 40 50 60")
(?age addTerm  "old"(40, 0) (65 1) (120,1)" )

(bind ?riskbc  (new  ATOMIC_LINGUISTIC_VARIABLE
"risk-breast-cancer"  "percentage" 0 100)
(?riskbc addTerm " low"  "z-mf"  "5 10" )
(?riskbc addTerm  "average" "pi-mf"   "5 10 15 20")
(?riskbc addTerm  "high"  "s-mf"  "15 30")
```

Now the fuzzy graph may be created to show the dependency between the linguistic variables: ?age and ?riskbc, and fuzzy rules may be created and added to the fuzzy graph. A fuzzy rule consists of a list of generalized constraints that makes its premises and its conclusion. The parameter "?x" denotes an anonymous object which may be instantiated by the object of a matching generalized constraint in the fact base. The fuzzy graph is added as a shadow fact [12] to the working memory.

```
;; Creating the fuzzygraph
(bind ?fg  (new  FUZZY_GRAPH)

;;Defining the fuzzy rule: "if age(x) is young then
;;riskbc(x)  is low"
(bind  ?gc1  (new GENERALIZED_CONSTRAINT ?age  "?x"
   "young")
(bind  ?gc2  (new GENERALIZED_CONSTRAINT ?riskbc
   "?x"  "low"))
(bind ?rule1  (new FUZZY_RULE ?gc1 ?gc2))

;;Defining the fuzzy rule: "if age(x) is middle-age
;;then riskbc(x)"  is average"
(bind  ?gc3  (new GENERALIZED_CONSTRAINT ?age  "?x"
 "middle-age"))
(bind  ?gc4  (new GENERALIZED_CONSTRAINT ?riskbc
 "?x"  "average"))
(bind ?rule2  (new FUZZY_RULE ?gc3 ?gc4))

;;Defining the fuzzy rule: "if age(x) is old then
;;riskbc(x)"  is high"
(bind  ?gc5  (new GENERALIZED_CONSTRAINT ?age  "?x"
```

```
  "old"))
(bind  ?gc6  (new GENERALIZED_CONSTRAINT ?riskbc
  "?x"  "high"))
(bind ?rule3  (new FUZZY_RULE ?gc3 ?gc4))
(?fg addRule ?rule3)

;;Adding the rules to the fuzzy graph
(?fg addRule ?rule1)
(?fg addRule ?rule2)

;; adding the fuzzy graph to the working memory
(add ?fg)
```

The statement with fuzzy quantifiers may be represented as a binary quantified generalized constraint. In the following parameter "?x" is a variable bounded by the quantifier.

```
;; creating the generalized constraints that appear
;; in the quantified statements
(bind ?weight (new ATOMIC_LINGUISTIC_VARIABLE
  "weight" "Kilogram" 40 250))
(?weight addTerm "obese"  "s-mf"  "85 100")
(bind ?gc1 (new GENERALIZED_CONSTRAINT ?weight "?x"
  "obese" ) )
(bind ?gc2 (new GENERALIZED_CONSTRAINT ?riskbc "?x"
  "high"))
(bind ?recom (new ATOMIC_LINGUISTIC_VARIABLE
  "mammogram-recommendation" "percentage" 0 1))
(?recom addTerm "strongly-recommended"  "s-mf"
  ".85 1")
(bind ?gc3 (new GENERALIZED_CONSTRAINT ?recom "?x"
  "strongly-recommended" ))

;; Creating  the quantified generalized constraints
(bind ?qgc1  (new QUANTIFIED_GENERALIZED_CONSTRAINT
  "many" "?x"  ?gc1 ?gc2))
(bind ?qgc2  (new QUANTIFIED_GENERALIZED_CONSTRAINT
  "most" "?x"  ?gc2 ?gc3))
;; Adding the quantified generalized constraints to
;; the working memory
(add ?qgc1)
(add ?qgc2)
```

Given that the age of mother is equal to the age of her son plus the age that she gave birth to him, we have two pieces of information regarding Mary's age which we can be encoded as the following arithmetic generalized constraints.

```
;; Defining the linguistic terms that appear in the
;;arithmetic generalized constraints
(?age addTerm "in-20's "  "Interval-mf"  "20 30" )
(?age addTerm "about 15"  "tri-mf"  "13 15 17" )
(?age addTerm "few-years" "trap-mf" " 2 3 5 7")

;; Defining the arithmetic generalized constraints
;; and adding them to the working memory
(bind ?agc1  (new ARITHMETIC_GENERALIZED_CONSTRAINT
  ?age  "Mary" "in-20's + about 15" ))
(bind ?agc2 (new ARITHMETIC_GENERALIZED_CONSTRAINT
  ?age "Mary" "?age(Ann)  -  few-years"))

;; Adding the arithmetic generalized constraints to
;; the working memory
(add ?agc1)
(add ?agc2)
```

And finally, the information regarding Ann's age can be defined as a simple generalized constraint:

```
(?age addTerm "mid-50"  "tri-mf"  "50 55 60")
(bind ?gc8 (new GENERALIZED_CONSTRAINT ?age "Ann"
  "mid-50"))
(add ?gc8)
```

Once the above Jess program is run, the pair of simple and arithmetic facts: ?gc8 and ?agc2 fire the extension principle rule. The extension principle rule replaces "?age(Ann)" with "mid-50" in the arithmetic expression "?age(Ann) – few years" to achieve a fuzzy value for Mary's Age. The extension principle rule is also fired for the arithmetic fact ?agc1 to calculate the value "in-20's + about-15". At this point the fact base would consist of two generalized constraints regarding Mary's age with different fuzzy values which leads to firing the conjunction rule to combine the constraints to obtain one fuzzy value for Mary's age.

Now the resulting fact regarding Mary's age together with the fuzzy graph fact ?fg activate the fuzzygraph interpolation rule, which in turn calculates the value of ?riskbc (Mary) and adds it as a simple generalized constraint to the fact base.

The fuzzy graph interpolation rule is also fired by the pair of facts: ?fg and ?gc8 to calculate the value of ?riskbc(Ann).

Furthermore, the pair of quantified facts: ?qgc1 and ?qgc2 activate the fuzzy syllogism rule which calculates the proportion of obese women for whom the Mammogram is highly recommended. The result of this calculation is asserted to the fact base as a quantified generalized constraint.

In order to show the output of a jess program to the user, the CWJess library includes a jess function: "print-GC" . This function allows users to query the fact base to find the value of a given linguistic variable. For example in the above program if the user wishes to know Mary's age, or her risk of developing a breast cancer, she can add the following lines to the jess program:

```
;; Querying the fact-base
(print-GC ?age "Mary")
(print-GC ?riskbc "Mary")
```

The output is the two normalized fuzzy sets in figures 7 and 8, deduced by using the extension principle, the conjunction, and the fuzzy graph interpolation rules.

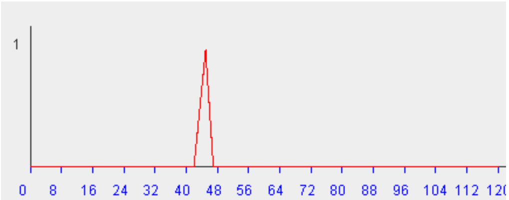



Figure 7. The value of the linguistic variable "age(Mary)"

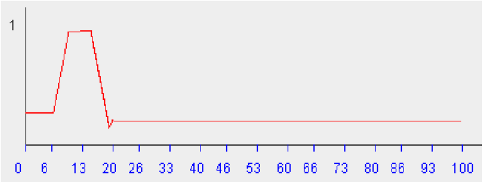



Figure 8. The value of the linguistic variable: "riskbc(Mary)"

## V. Summary and Future Work

The paper reports a preliminary work on the implementation of an expert system shell on top of Jess to perform CW reasoning. The resulting shell, CWJess, is a powerful tool which allows users to express fuzzy facts in form of generalized constraints of various types. It provides java classes which enable users to define atomic or composite linguistic variables, linguistic terms, fuzzy quantified statements, fuzzy rules, fuzzy arithmetic expressions, and fuzzy graphs. The CW inference rules, implemented in CWJess, along with Jess Rete algorithm render an inference engine which is able to perform forward reasoning over a set of generalized constraints.

We are working on making the CWJess tool available for download via www.purvag.com/cwjess. We will provide a complete set of instructions to help users to express their knowledge in terms of GCL. Nevertheless, to facilitate easier translation of linguistic knowledge, we are planning to develop a human readable intermediate language to hide the complexity of underlying GCL.

Several other developments to CWjess can be proposed to make it more expressive and to enhance its reasoning power, among which are:

- Developing necessary classes and methods to allow users to assert generalized constraints with probabilistic modality, such as: "(age(Mary) is young) isp likely" and perform reasoning on such generalized constraints.
- Implementing the compositional rule of inference for fuzzy relations with a continuous membership function.

## References

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

[2] L. A. Zadeh, "Computing With Words and Perceptions-A Paradigm-Shift," Proc IEEE IRI-2009.

[3] FuzzyCLIPS Version 6.04a User's Guide, Nat. Res. Council, Ottawa,ON, Canada, 1998

[4] R. Orchard, "Fuzzy reasoning in jess : The fuzzyj toolkit and fuzzyjess," Proc. in ICEIS 2001, 3 rd International Conference on Enterprise Information Systems, 2001, pp. 533-542.

[5] W. Siler, D. Tucker, and J. Buckley, "A parallel rule firing fuzzy production system with resolution of memory conflicts by weak fuzzy monotonicity, applied to the classification of multiple objects characterized by multiple uncertain features,"*International Journal of Man-Machine Studies*, vol. 26, no. 3, 1987, pp. 321-332; DOI 10.1016/s0020-7373(87)80066-4.

[6] J. Baldwin, T. Martin , and B. Pilsworth, "Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence", Research Studies Press, Taunton, Somerset, England, 1995.

[7] L. A. Zadeh, "Toward a generalized theory of uncertainty (GTU): an outline," *Inf. Sci. Inf. Comput. Sci.*, vol. 172, no. 1-2, 2005, pp. 1-40.

[8] D. Dubois and H. M. Prade, "Possibility theory : an approach to computerized processing of uncertainty", Plenum Press, 1988, p. xvi, 263 p.

[9] L. A. Zadeh, "Commonsense reasoning based on fuzzy logic," *Book Commonsense reasoning based on fuzzy logic*, Series Commonsense reasoning based on fuzzy logic, ed., Editor ed.^eds., ACM, 1986, pp. 445-447.

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

[11] W. M. Dong and F. S. Wong, "Fuzzy weighted averages and implementation of the extension principle," *Fuzzy Sets and Systems*, vol. 21, no. 2, 1987, pp. 183-199.

[12] E. F. Hill, *Jess in Action: Java Rule-Based Systems*, Manning Publications Co., 2003, p. 480.