Towards an Automated Reasoning for Computing with Words

Elham S.Khorasani and Shahram Rahimi

Abstract—This paper paves the path for the design and implementation of a practical automated reasoning mechanism for computing with words. The presented methodology aims at filling the gap between the raw theory and the implementation of CW paradigm. As part of this methodology, a portion of a formal grammar for Generalized Constraint Language (GCL) as well as a systematic approach for applying the deduction rules are introduced.

I. INTRODUCTION

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 making complex reasoning and human-like decisions [1]. 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, fuzzy relations, fuzzy arithmetic, fuzzy quantifiers and fuzzy probability. The phrase “computing with words” was first introduced in [2] as:

“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 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:

\[ X \text{ isr } R \]

Where \( X \) is a linguistic (or constrained) variable whose values are constrained by the linguistic value \( R \), 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 but the most important ones are: (1) possibility \((r=\text{blank})\) where \( R \) denotes the possibility distribution of \( X \), (2) probability \((r=p)\) where \( X \) is a random event and \( R \) is the probability distribution of \( X \), (3) verity \((r=v)\) where \( X \) is a conjunctive variable that can take more than one value and \( R \) denotes the truth distribution of \( X \).

Table I shows some examples of natural language propositions and their meaning representation in GCL. As can be seen in this table, a linguistic variable takes various forms; it can be in form of a generalized constraint, a function, a relation, and so forth. Each proposition in natural language may constrain one or more variables.

<table>
<thead>
<tr>
<th>NL proposition</th>
<th>Generalized Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>The price of oil has inverse relation with the oil production.</td>
<td>Relation(price(oil), production(oil)) is inverse R X</td>
</tr>
<tr>
<td>Most Indian foods are spicy</td>
<td>Count(taste(Indian-food) is spicy) is most X1 R1 R2 X2</td>
</tr>
<tr>
<td>It is likely to rain today</td>
<td>(weather(today) is rainy) is likely R X1 X2 R1 R2</td>
</tr>
</tbody>
</table>

After representing the natural language propositions in form of GCs, CW defines a set of inference rules to perform reasoning among GCs. Each rule has a syntactic part and a semantic part. The syntactic part shows the general abstract form (So called the protoform) of the GCs of the premises and the conclusion of the rule, while the semantic part shows how to compute the truth degree of the conclusion of a rule based on its premises. As opposed to the classical bivalent logic, such as predicate logic, CW allows premises that are only partially true. Thus the truth degree of the conclusion of a rule should be determined from the truth degree of its premises. 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. Some instances of these rules [2]-[4] are listed in table II.

Figure 1 shows the overall view of a CW machine. The heart of this machine, which is the focus of this paper, is the inference engine which performs reasoning on a set of given generalized constraints via the inference rules that reside in the deduction database. In order to carry out the computational part of the rules, the inference engine needs to
know the fuzzy subsets which correspond to various granule values of a linguistic variable. This information is stored in the word-fuzzy dictionary which contains all the linguistic variables in the problem domain as well as their granule values and the fuzzy subsets associated with each granule value. The main task of the inference engine is to match the domain knowledge with the deduction rules in a systematic approach to draw new conclusions.

### Table II

<table>
<thead>
<tr>
<th>Inference rule</th>
<th>Symbolic part</th>
<th>Computational part</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Extension Principle</td>
<td>$g(X_1, ..., X_n) \text{ is } A$ [ f(X_1, ..., X_n) \text{ is } B ]</td>
<td>$\mu_i(v) = \sup_{u_1,...,u_n} (\mu_i(g(u_1,...,u_n)))$ ( v = f(u_1,...,u_n) )</td>
</tr>
<tr>
<td>(2) Compositional Rule of Inference</td>
<td>$\sum_{Y \text{ is } C} X_1 \text{ is } A$, $X_2 \text{ is } B$</td>
<td>$\mu_i(v) = \sup_{u_1,...,u_n} (\mu_i(u)) \land \mu_i(v,u_1,...,u_n)$</td>
</tr>
<tr>
<td>(3) Fuzzy Interpolation</td>
<td>$\sum_{X \text{ is } A}$ ( \text{if } X \text{ is } A, \text{then } Y \text{ is } B )</td>
<td>$\mu_i(v) = \sup_{u_1,...,u_n} (m_i \land B_i)$</td>
</tr>
<tr>
<td>(4) Fuzzy Syllogism</td>
<td>$Q, A \text{ are } B$’s ( Q \text{ is the } C \text{’s } Q, A \text{ are } (B \land C)’s )</td>
<td>$Q_i = Q \times Q_i$</td>
</tr>
</tbody>
</table>

As shown in the figure, the input to the inference engine is a set of generalized constraints. However, in some application (e.g., a text summarization or a question answering system), the input to a CW system may be expressed in form of the natural language. In this case, a translation element is required to convert the natural language sentences into the GCL. This translation may or may not be done with human intervention; however, performing an unsupervised translation is not a trivial task and requires a deep understanding of syntax, semantics, lexicon, morphology and even the pragmatics [5]. One difficulty in such translation is to explicate the implicit constrained variables because most of the times the constraints in the natural language are implicit rather than explicit. One promising direction in this regard is the use of ontology [6].

The rules in the deduction database do not provide a linguistic label for the value of the constrained variables of the conclusion, but they merely compute the fuzzy subsets that represent these values. However, in some application, it might be desired to specify the output in terms of linguistic terms rather than fuzzy subsets. Thus a retranslation element is needed to assign the most appropriate word from the domain vocabulary to each fuzzy subset obtained from the inference engine. The main issue in the retranslation process is to minimize the loss of information due to the linguistic approximation. [7] introduces a number of criteria for evaluating a retranslation method.

It is important not to confuse CW with natural language processing. CW does not claim that it is able to fully model complex natural language propositions nor does it argue that it can perform reasoning on such statements. But it rather offers a system of computation that is superior to the traditional bivalent computing systems because of its capability to reason and compute with linguistic words hence modeling human reasoning [7]. As an example assume that we have the following information:

- “it is usually hard to find a parking spot in the campus during the day”
- “Bob is planning to drive to campus in the afternoon.”

Suppose that we would like to know” the probability that Bob finds a parking spot in the campus”? A conventional bivalent reasoning system, such as the probabilistic inference, fails to give an answer in this situation unless the exact numeric probability is provided for the word “usually”, while CW offers additional methods to represent and compute with such imprecise probabilities.

![Fig1. The machinery of CW](image)

The theory of CW, as currently introduced, is considered raw and needs intensive work and research before it can be applied to the practical use. Since its introduction, quite a few researches have been undertaken in this area but the ultimate goal of building an actual CW computational engine has thus far proven to be elusive. Two important reasons for this incapability are:

1. **Lack of a standard formal definition for the GCL.**
   There is no formal definition of GCL that determines the domain of the language as well as its set of the well-formed-formula. This formalization is essential in two aspects: first, it helps to standardize the notion of a valid argument or a deduction rule in CW, and allows automation of the reasoning process. Second, it describes the grammar of GCL and thus facilitates the translation of natural language statements into GCL.
2. **Lack of an efficient deduction methodology.** The unrestrained list of deduction rules established by
CW makes the automated manipulation of knowledge inefficient. Thus a competent deduction methodology needs to be defined to automate the reasoning process. This methodology should identify a complete and sound set of deduction rules that can subsume the current list of rules in CW and define a systematic mechanism for applying these rules.

This paper focuses on addressing the two issues above and proposes a methodology for developing a formal tool which allows the computers to perform CW reasoning automatically or nearly automatically.

II. STATE OF THE ART

After Zadeh’s seminal paper, there have been quite a few authors that picked CW framework and tried to develop it in a theoretical and a practical sense. The articles published by these authors reflect very different point of views and research directions regarding CW, but generally they can be classified as follows:

- **The articles that developed a formal model of computation for CW**: These articles ([9]-[11]) 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.

- **The articles that focused on the problem of loss of information**: These articles mostly focused on minimizing the loss of information in the retranslation process or proposed alternative computational methods to compute with linguistic labels. ([12]-[19])

- **The articles that dealt with the linguistic aspects of CW**: These articles ([6], [20], [21]) are the starting points for addressing the non-trivial problem of automated translation of information from natural language to GCL. One promising direction in this regard is the use of ontology to enhance the semantics of GCL.

- **The articles that captured the uncertainty underlying the words**: These articles ([22]-[27]) mostly make use of the interval type II fuzzy sets to model the uncertainty embedded in the membership function of the linguistic terms.

In summary, most of the studies in the literature are either concerned about developing alternative representation and/or computational methods to facilitate CW or they focused on the retranslation process. However, despite its great importance, there have not been many studies on developing an automated reasoning tool for CW using Zadeh’s generalized constraints and deduction rules. We are aware of only two studies ([28],[29]) that point to this direction. These studies adopt approximate reasoning (AR) as a formal reasoning basis for CW and define the canonical forms of sentences within the AR framework as well as a set of basic operations and deduction rules. Although pointing at a promising direction, this work still lacks a systematic methodology to mechanize the reasoning process.

III. PROPOSED METHODOLOGY

We propose a three step approach for constructing a formal mechanism that facilitates the automation of the reasoning process in CW. In the first step, we define a set of well formed formula (WFF) for GCL. Then we establish a set of canonical forms and provide a number of simplifying rules to convert the GCL expressions to their equivalent canonical forms. Finally, in the last step, we present a backward inference mechanism for answering a query posed to the system.

It is important to note that the formalization presented in this paper is different from the branch of literature that studies the mathematics of fuzzy logic based on the traditional multi-valued logic [30][31]. As Zadeh made the distinction in [32]:

“fuzzy logic has two different meanings-wide and narrow.In a narrow sense fuzzy logic is a logical system which aims at formalization of the approximate reasoning. As such it is rooted in multi-valued logic but its agenda is quite different from that of traditional multi-valued logical systems, e.g., Łukasiewicz logic. In this connection it should be noted that many of the concepts which account for the effectiveness of fuzzy logic as a logic of approximate reasoning are not a part of traditional multi-valued logical systems. Among these are the concepts of a linguistic variable, canonical form, fuzzy rule, fuzzy graph, fuzzy quantifiers and such mode of reasoning as interpolative reasoning, syllogistic reasoning and dispositional reasoning.”

Hence, the research presented in this paper is more concerned about the fuzzy logic in a broader sense and aims at mechanizing the reasoning process in CW.

A. GCL Formalization

GCL, as currently introduced, does not have a formal and standard definition. The operations are not clearly defined and the syntax structure of the language is not determined. A formal definition of GCL is essential for automated reasoning to determine a set of well-formed formula (WFF) that can appear in the deduction rules and to standardize the notion of a valid formula. Below is an example of a set of WFFs that may be defined for a portion of GCL that includes the simple-structured possibilistic expressions. Notice that this grammar also includes the WFFs for the predicate logic and
the universal and existential quantifiers are generalized to fuzzy quantifiers.

**Symbols:**
- An infinite set of simple variables, denoted by the lowercase letters at the end of the alphabet: \(X, Y, Z, \ldots\)
  The subscripts are used to distinguish the variables: \(X_1, X_2, X_3, \ldots\)
- An infinite set of linguistic variables with arity greater or equal to 0, denoted by the uppercase letters at the end of the alphabet: \(X, Y, Z, \ldots\). Such as: "age(x)", "price(x)", "size(x)", "distance(x, y)", etc.
- An infinite set of linguistic value constants (granule values), denoted by the uppercase letters at the beginning of the alphabet: \(A, B, C, \ldots\). The linguistic value may be fuzzy (such as: "tall", "slim", "tasty", "about 6") or crisp ("6", (3,6), or so forth)
- An infinite set of predicates with arity greater than or equal to 0, denoted by the uppercase letters at the end of the alphabet: \(P, Q, \ldots\), such as: "student(x)", "married(x, y)", "man(y)", and so forth.
- An infinite number of crisp function symbols with arity greater than 0, denoted by lowercase letters at the end of the alphabet: \(f, g, h, \ldots\).
  Function symbols of valence 0 are called **constant symbols**, and are often denoted by lowercase letters at the beginning of the alphabet: \(a, b, c\).
- Fuzzy or crisp quantifiers, denoted by \(Q, Q, Q, \ldots\) (such as: "many", "most", "almost all", "all", "some",...)
- The possibility distribution symbol, denoted by: "ise".
- The equality constraint symbol denoted by: "ise".
- Parentheses, brackets, and other punctuation symbols.
- Connectives and operations: \(\land\) for conjunction, \(\lor\) for disjunction, \(\neg\) for negation, and "if ... then" for implication.
- Fuzzy modifiers, denoted by \(m_1, m_2, m_3, \ldots\), such as: "very", "more or less", "much", "somewhat" "to some extent", so forth.

**Formation rules**

The formation rules define terms and valid formula of GCL. These rules can be mapped into a context-free grammar with infinite set of symbols and possibly many start symbols.

(1) **Terms:**

We have two different types of term set in this grammar:
(1) **TERMII**, which consists of the simple variables and their functions (the term set of predicate logic), and (2) **TERMI**, which consists of the linguistic variables, their relation and functions. These term sets may be recursively defined as follows:

**TERMI:**
- If \(x\) is a simple variable then \(x \in \text{TERMI}\)
- If \(t_1, t_2, \ldots, t_n \in \text{TERMI}\) and \(f\) is a function then \(f(t_1, t_2, \ldots, t_n) \in \text{TERMI}\)

**TERMI:**
- If \(X\) is a linguistic variable then \(X \in \text{TERMI}\)
- If \(X_1, X_2, \ldots, X_n\) are linguistic variables then \((X_1, X_2, \ldots, X_n) \in \text{TERMI}\) and denotes a joint variable. For example: \((\text{Age}(x), \text{Age}(y))\)
- If \(t_1, t_2, \ldots, t_n \in \text{TERMI}\) are terms and \(f\) is a crisp function then \(f(t_1, t_2, \ldots, t_n) \in \text{TERMI}\).

(2) **Formula:**

- If \(t_1 \in \text{TERMI}\) and \(A\) is a granule value, then \("t is A" is a formula.
- If \(t_1, t_2 \in \text{TERMI}\), then \("t_1 ise t_2" is a formula.
  \((t_1\) and \(t_2\) have identical possibility distributions).
- If \(t \in \text{TERMI}\) and \(A\) is a granule value, then \("t is mA" is a formula, where \(m\) is a fuzzy modifier.
- If \(P\) is an \(n\)-ary predicate symbol and \(t_1, t_2, \ldots, t_n \in \text{TERMI}\), then \(P(t_1, t_2, \ldots, t_n)\) is a formula.
- If \(Q\) is a quantifier and \(\phi\) is a formula then "\(Q\), \(\phi\) is a formula. Where \(x\) is a free variable in \(\phi\).
- If \(\phi\) is a formula then \(\neg \phi\) is a formula.
- If \(\phi_1\) and \(\phi_2\) are formulas then \(\phi_1 \land \phi_2\) is a formula.
- If \(\phi_1\) and \(\phi_2\) are formulas then \(\phi_1 \lor \phi_2\) is a formula.
- If \(\phi_1\) and \(\phi_2\) are formulas then "if \(\phi_1\) then \(\phi_2\)" is a formula.

The above grammar serves as an example and can be used to represent the GCL expressions with the possibilistic variables. For example this grammar can represent the sentence: "most students are younger than their teachers" by generating the wff:

\[
\text{most}_s (\text{student}(x) \land \text{teacher}(x, y) \land (\text{Age}(x), \text{Age}(y)) \text{is younger})
\]

A formal grammar of GCL is a prerequisite to the next step of automated reasoning which is the canonicalization.

**B. Canonicalization**

As it was pointed out by Zadeh [4], the list of deduction
rules currently introduced for CW is not comprehensive. One can find thousands of examples of inferences in natural language that cannot be exactly matched against any existing deduction rule. This is because the number of syntactic forms that a proposition takes in natural language and GCL accordingly is infinite. Thus it is very inefficient (if not impossible) to define a comprehensive deduction database that can be matched against all syntactic extensions in GCL. However, many of these rules can be derived from the others. Consequently, it is exceptionally cost-effective to subsume the potentially large deduction database with a set of primary deduction rules and define appropriate simplifying rules for putting the knowledge into a form accepted by these rules. This approach would make it much easier to build an automated inference engine.

In this phase a set of canonical forms for the generalized constraints are defined. That is, the various structures that X can take in the expression “X is R” are classified and their premises and their conclusion (impossible) to define a comprehensive deduction database. For example, rules 1 and 2 in table II are in canonical form while rules 3 and 4 are not. However rules 3 and 4 can be derived from the other rules by using canonicalization as follows:

- Derivation of the fuzzy interpolation rule:

\[
\sum_{i} \text{if } X = A \text{ then } Y = B \quad \text{canonical form} \quad (X,Y) = C \\
X = A \quad \text{Y is B} 
\]

Where, \( \mu_{A}(u,v) = \text{sup}(\mu_{A}(u) \wedge \mu_{B}(v)) \)

So after canonicalization, this rule matches with the compositional rule of inference, hence:

- Derivation of the fuzzy syllogism rule:

\[
Q_{A's are B's} \quad \text{canonical form} \quad f(t) = Q \\
Q_{A's are (B & C)'s} \quad g(t) = Q \\
Q_{A's are (B & C)'s} \quad h(t) = Q
\]

By using the conjunction and quantification rule, we have:

\[
f(t) = \frac{\sum (\mu_{A}(t) \wedge \mu_{B}(t))}{\sum \mu_{A}(t)} \\
g(t) = \frac{\sum (\mu_{B}(t) \wedge \mu_{C}(t))}{\sum \mu_{B}(t)} \\
h(t) = \frac{\sum (\mu_{C}(t) \wedge \mu_{D}(t))}{\sum \mu_{C}(t)}
\]

Since \( h(t) = f(t) \times g(t) \), the above canonical form becomes:

\[
f(t) = Q_{f(t)} \\
g(t) = Q_{g(t)} \\
h(t) = Q_{h(t)}
\]

which matches the extension principle, so the possibility distribution of \( Q_{f(t)} \) is computed as:

\[
\mu_{Q_{f(t)}}(v) = \text{sup}_{v}(\mu_{Q_{f(t)}}(u) \wedge \mu_{Q_{g(t)}}(u)), \text{subject to } v = u_{1} \times u_{2}
\]

Which is the same as the computational part of rule 4, table II.

C. Development of an Inference Mechanism

The inference mechanism is a method of applying the deduction rules to the knowledgebase to derive new conclusions. Depending on the application, the inference mechanism may be forward (suitable for data-driven application such as text summarization) or backward.
(suitable for goal driven applications such as question/answering systems). In this paper we focus on the backward reasoning methods for answering a query posed to the system. A query may be of various types but generally it may be viewed as seeking a value for one or more variables [29]. This view of the query includes a wide range of query types such as: factual questions, list questions, and truth qualified questions. In [33], we have developed an inference mechanism for CW which consists of two components:

1. Information extraction and organization: In this part, we extract the knowledge that is relevant to the query and organize it in a tree data structure to perform backward reasoning. In particular we:
   a. use pattern matching and statistical methods to extract the information that is directly relevant to the query from the knowledgebase.
   b. convert this information to its canonical form by using the simplification rules described in the previous section.
   c. use the symbolic part of the primary inference rules to create new sub queries and continue the same process until no new sub queries can be created.

2. Evaluation and data fusion. In this part, we traverse the tree from bottom to the top applying the computational part of the inference rules to propagate the constraints from the relevant data to the query variable. In addition, we fuse the constraints attained for each sub query as well as the top level query to find the most appropriate answer.

To demonstrate the above procedure we present a simple example. Suppose that a website contains the following information for the breast cancer risk assessment:

The average chance that a woman is diagnosed by breast cancer depends on her age. From age 30 through age 39, it is about 0.4 %; from age 40 through age 49, it is about 1.5 %; from age 50 through age 59, it is about 2.5 %, and from age 60 through age 69 it is about 3.5 %. There are also some other factors that affects the average risk of breast cancer. Alcohol increases the average risk of breast cancer significantly; pregnancy in the age of 30 or before reduces the average risk of breast cancer by about 3 %, and in older women being overweight can increase the average risk of breast cancer slightly.

Suppose also that we know the following facts about Mary and we are interested to know Mary’s chances of developing a breast cancer:

Mary has a son who is about 20. She gave birth to her son when she was in her 20s. Mary is few years younger than Ann who is in her mid 50. Mary consumes about 1400 to 2000 calories a day. And she drinks moderately.

As a commonsense knowledge, we also know that:

Overeating causes one to be overweight. The age of mother is equal to the age of her son plus the age that she gave birth to him.

Assume that the above information is translated to GCL as follows:

- 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 % if age(x) is in 50s then average-riskbc(x) is about 2.5 % if age(x) is in 60s then average-riskbc(x) is about 3.5 %
- if drinking-habit(x) is regularly then alcohol-factor(riskbc(x)) is significant
- if age-pregnancy(x) is about 30 or before then pregnancy-factor(riskbc(x)) is about 3 %
- if age(x) is old and weight(x) is overweight then weight-factor(riskbc(x)) is slightly
- riskbc(x) ise average-riskbc(x) + alcohol-factor(riskbc(x)) + weight-factor(riskbc(x)) - pregnancy-factor(riskbc(x))
- age(son-off(Mary)) is about 20
- age(Mary) = Age(Ann) - few years
- age(Ann) is mid-50
- age(pregnancy(Mary)) is in 20s
- eating-habit(Mary) is about 1400 to 2000 calories per day
- if eating-habit(x) is overeat then weight(x) is overweight
- age(x) = age(son(x)) + age(pregnancy(x))
- drinking-habit(Mary) is moderate

We are now ready to initiate the reasoning process. The first step is to convert the above GCL expressions to their canonical forms. Assuming the canonical set in the previous section, we can canonicalize the above expression as follows:

- (Age(x), average-riskbc(x)) is R1 , where
  \[ \mu_{\alpha_1}(u, v) = \sup_i (\mu_{\alpha_1}(u) \land \mu_{\alpha_1}(v)) \]
  \[ A \in \{ in - 30^\prime s, in - 40^\prime s, in - 0^\prime s, in - 60^\prime s \}, \]
  \[ B \in \{ about : 0.4\%, 1.5\%, 2.5\%, 3.5\% \} \]
- (drinking-habit(x), alcohol-factor(riskbc(x))) is R2 , where
  \[ \mu_{\alpha_2}(u, v) = (\mu_{regular}(u) \land \mu_{significant}(v)) \]
- (age-pregnancy(x), pregnancy-factor(riskbc(x))) is R3 , where
  \[ \mu_{\alpha_3}(u, v) = (\mu_{overweight}(u) \land \mu_{significant}(v)) \]
- (age(x), weight(x), weight-factor(x)) is R4 , where
  \[ \mu_{\alpha_4}(u, v) = (\mu_{overweight}(u) \land \mu_{significant}(u) \land \mu_{regular}(v)) \]
- riskbc(x) ise average-riskbc(x) + alcohol-factor(riskbc(x)) + weight-factor(riskbc(x)) - pregnancy-factor(riskbc(x))
- age(son-off(Mary)) is about 20
- age(Mary) = Age(Ann) - few years
- age(Ann) is mid-50
- age(pregnancy(Mary)) is in 20s
- eating-habit(Mary) is about 1400 to 2000 calories per day
- eating-habit(x), weight(x)) is R5 , where
  \[ \mu_{\alpha_5}(u, v) = (\mu_{overweight}(u) \land \mu_{significant}(v)) \]
- age(x) = age(son(x)) + age(pregnancy(x))
- drinking-habit(Mary) is moderate

The knowledge relevant to the query is organized in a tree structure as illustrated in figure 2. The root node in this tree is the query and other nodes with a question mark are sub queries. Each (sub) query is connected to its immediate successors with a rule that is applied to the successors to get the value of the sub-query. (“E” denotes the extension principle and “C” denotes the compositional rule of inference). After constructing the tree, the answer to the query is found by traversing the tree from the bottom to the top and apply each rule to its related nodes [33]. By organizing the knowledge in a tree, we are able to identify the missing knowledge with respect to a query and improve
the answer by acquiring this knowledge from the user.

IV. SUMMARY AND FUTURE WORK

In this paper we proposed a methodology for developing a preliminary automated reasoning tool for CW. To demonstrate the approach, we developed a formal grammar for a portion of GCL that contains possibility distribution. We classified the GCL formula to a set of canonical forms and introduced a number of simplification rules to convert a valid GCL formula to its canonical form. We showed how canonicalization can facilitate the reasoning process by subsuming a potentially large deduction database with a set of primary rules. We believe that the methodology introduced in this paper points to a promising direction for future implementation of CW. Moreover by formalizing GCL, we place the necessary ground work for building a translation tool for converting the natural language expressions into GCL. A future work of this research would need to develop a comprehensive grammar for GCL that include other semantic modalities such as probability, verity, fuzzy graph, random set, and so forth. In addition to the WFFs, the notion of the free and bound variables as well as the unification should be clearly defined. Furthermore, the performance of the inference mechanism must be improved by designing appropriate heuristics to prune the branches of the tree that induce a lower information value. One should also investigate off-line techniques to store and update the value of the intermediate nodes from the previously generated trees for using in the later queries.

Fig. 2. The reasoning tree for the query: “what is the risk of developing a breast cancer for Mary?”

REFERENCES


