Expert Systems with Applications xx (0000) xxx–xxx

Expert Systems with Applications

www.elsevier.com/locate/eswa

Comparison of first order predicate logic, fuzzy logic and non-monotonic logic as knowledge representation methodology

Kyung Hoon Yang[a], David Olson[b,*], Jaekyung Kim[b]

[a]Department of Information Systems, College of Industrial Science, Chung Ang University, 221 Huksuk-dong, Dongjak-ku, Seoul 156 756, South Korea

[b]Department of Management, College of Business Administration, University of Nebraska, Lincoln, NE 68588-0491 United States

Abstract

The aim of this paper is to compare first order predicate logic, fuzzy logic and non-monotonic logic as knowledge representation methods. First, we define five properties of knowledge; conceptualization, transfer, modification, integration and decomposition. We also evaluate first order predicate logic, fuzzy logic and non-monotonic logic for the above properties, in the view of accuracy, complexity, and completeness. We then prove that the complexities of the three methods are NP-complete. We use this information to design a heuristic algorithm tested on probabilistic input to evaluate accuracy and completeness. With the results, we compare weaknesses and strengths of each method.

Keywords: Knowledge management; Knowledge representation; Predicate logic; Fuzzy logic; Non-monotonic logic

1. Introduction

Knowledge management is one of the most critical issues for advanced information systems. It is necessary for virtual organizations, agent based systems and intelligent database management systems, as well as many other advanced intelligent information systems. There are several research issues in knowledge management, to include knowledge acquisition, knowledge storage, knowledge inference, knowledge retrieval speed, learning, ease of use, and so forth. One of the key factors in these issues is knowledge representation.

Knowledge representation can be defined as the representation of knowledge in a structured manner. Among the many applications of automated knowledge representation are real-time knowledge-based control systems (Bhattacharyya & Koehler, 1998; Grabowski & Sanborn, 1992). Knowledge representation is critical because accuracy, speed of handling knowledge storage, inference, and knowledge retrieval all depend on its accuracy. Therefore, a good knowledge representation should have the capability to store and retrieve knowledge accurately and quickly. Many techniques have been suggested for that purpose (McCarthy, 1977, 1980; Mitchell, Keller, & Kedar-Cabelli, 1986; Moore, 1982; Wong, 2001).

* Corresponding author. Tel.: +402-472-4521; fax: +402-472-5855.
E-mail addresses: dolson3@unl.edu (D. Olson), yangkh@cau.ac.kr (K.H. Yang), jkim6@unl.edu (J. Kim).

In this paper, we will compare predicate logic, fuzzy logic, and non-monotonic logic in terms of accuracy, complexity and completeness. We first briefly summarize the logic related to knowledge representation because there are many books and articles about these topics (Ginsberg, 1993; Helft, 1989; Levesque, 1984; Reichgelt, 1991; Smith, 1985). Then, we define the properties of knowledge. We use knowledge for the following five usages: conceptualization, transfer, modification, integration and decomposition (Bacchus, Grove, Halpern, & Koller, 1996; Engelen, 1997; Greiner, Darken, & Santoso, 2001; Wellman, 1990, 1994). We will examine how well the suggested methods aid in attaining good performance in these topic areas. We then define the criteria of evaluation of knowledge representation methods. There are several of these criteria, such as ease of use, psychological aspects, technical aspects and so on. Here, we will only consider theoretical aspects: accuracy, complexity and completeness. Next we compare the complexity of each method with respect to the five usage criteria. We will show that the complexity of knowledge representation by logic is NP-complete. That means that we cannot get the exact meaning from the facts in a reasonable amount of time and we need a heuristic method that provides accuracy in a reasonable time. Then we will compare the accuracy of each method. Because we find that extracting exact knowledge from the set of facts is NP-complete, we suggest a heuristic method. For that purpose, we also suggest reasonable assumptions and

0957-4174/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2004.05.012

ESWA 1212—15/6/2004—09:43—BELLA—107298— MODEL 5 – pp. 1–19

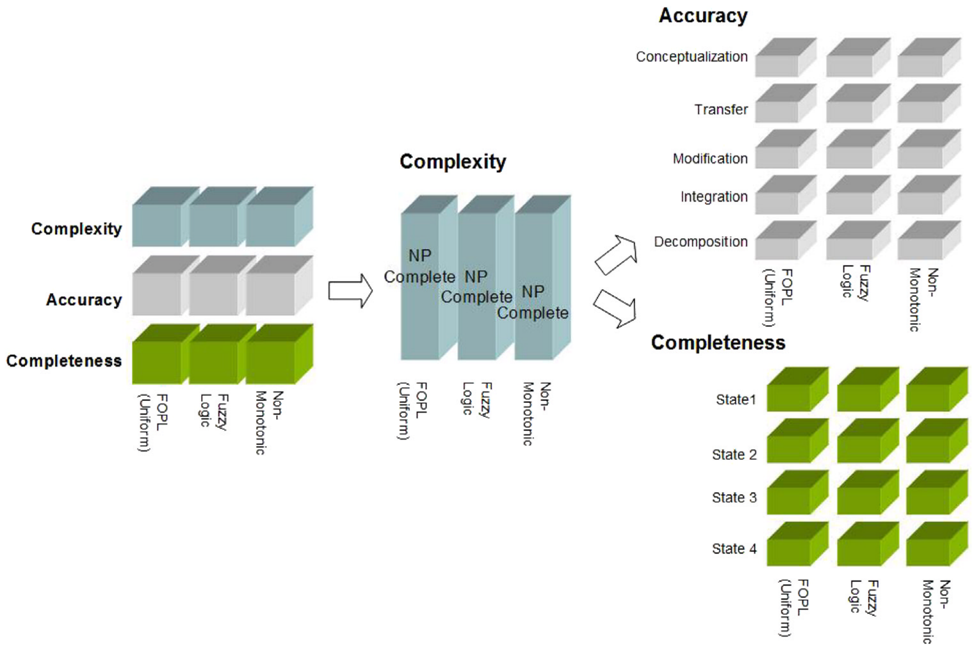

Fig. 1. Framework of research.

compared each method by probabilistic heuristics. We will also compare the completeness of each method by changing the ranges. Fig. 1 shows the framework of research. The last section of the paper is the conclusion and suggestions for further research. This includes discussion of the limitations of this research and further research possibilities.

2. Logic as a knowledge representation method

Previous researchers have suggested many knowledge representation methods, to include belief networks, frames, scripts, truth maintenance systems, and so forth (Patterson, 1990; Russell & Norvig, 2003; Torsun, 1995). Of course, each method has its strengths and weaknesses. However, logic is the most popular method because it is familiar to many people due to its long history and also because of its solid mathematical background. For these reasons, many different types of logic have evolved from traditional logic.

The use of logic to represent knowledge is not new to knowledge management. Even so, the application of logic as a practical means of representing and manipulating knowledge in a computer was not demonstrated until the early 1960s. Since that time, numerous methods have been implemented with varying degrees of success (Baldwin, 1981; Bouchon, 1988; Engelen, 1997; Patterson, 1990).

The objective of logic is production of a structured expression that the receivers or interpreters cannot interpret differently from the source or speaker's intention. Unfortunately, we have not found such a method. The most successful of all methods is first order predicate logic (FOPL) (Chang & Lee, 1973; Russell & Norvig, 2003). In this method, we only consider the quantifiers 'every', 'all' or 'some'. But in the real world, we need the use of more general quantifiers such as 'more or less', 'very', 'few' and so on, in order to increase the range of the usage of logic. Zadeh suggested fuzzy logic. But fuzzy logic does not have the solid completeness that predicate logic has (Bolloju, 1996; Zadeh, 1975, 1978, 1979, 1983; Zadeh & Kacprzyk, 1992). A defect of FOPL is the handling of incomplete knowledge. One-way humans deal with this problem is by making plausible default assumptions; that is, we make assumptions, which typically hold but may have to be retracted if new information is obtained to the contrary (Engelfriet & Treur, 2000; Ginsberg, 1993). For example, you can believe that birds can fly, until you find that the bird type penguins cannot fly. The basic idea of non-monotonic logic is that a new bit of knowledge can be derived from generally accepted premises unless a counter instance is explicitly proved (Engelfriet, 1998; Helft, 1989).

Today, predicate logic is one of the most important techniques for the representation of knowledge. A familiarity with predicate logic is important for the following reasons. First of all, logic is a formal method for reasoning. Many concepts that can be verbalized can also be translated into symbolic representations that closely approximate the meaning of these concepts. These symbolic structures can then be manipulated in programs to deduce various facts to carry out a form of automated reasoning. Second, logic offers the only formal approach to reasoning that has a sound theoretical foundation. This is especially important in order to mechanize or automate the reasoning process in that inferences should be correct and logically sound (Torsun, 1995).

Predicate logic is very solid and accurate, but its scope is too narrow for practical use. The reason is that the structure of predicate logic is not flexible enough to permit the accurate representation of natural language reasonably well. Therefore, several pseudo logics such as modal logic, temporal logic, fuzzy logic, non-monotonic logic, default logic and a closed world assumption have been suggested (Patterson, 1990; Torsun, 1995). In this paper we will classify logics into three sub-groups: (1) classical logic including prepositional logic and first order predicate logic (2) fuzzy logic that includes fuzzy logic and rough set logic and (3) non-monotonic logic including default logic, defeasible logic, truth maintenance system and the closed world assumption systems. However, we will consider only default reasoning to represent non-monotonic logic in this paper, because the basic ideas behind these kinds of logic are similar.

(1) *Classical logic*. There are two types of classical logic: propositional logic and first order predicate logic (Mendelson, 2001). However, since propositional logic is a simplified form of a first order predicate logic approach, we will only consider predicate logic.

This is a classical and traditional method used to express human knowledge in a structured manner. In predicate logic, statements from a natural language like English are translated into symbolic structures comprised of predicates, functions, variables, constants, quantifiers, and logical connectives (Mendelson, 2001). The symbols form the basic building blocks for knowledge, and their combination into valid structures is accomplished by using the syntax (rules of combination) for predicate logic. Once structures have been created to represent basic facts or procedures or other types of knowledge, inference rules may then be applied to compare, combine and transform these 'assumed' structures into new 'deduced' structures. This is how automated reasoning or inference is performed.

(2) *Fuzzy logic*. While there are many variants of non-traditional logic (Ross & Ross, 1995; Torsun, 1995; Zadeh & Kacprzyk, 1992), only fuzzy logic is considered.

Fuzzy logic has been suggested as a means to overcome the weak points of predicate logic (Baldwin, 1981; Zadeh, 1979). Vague expressions of a human expert can be interpreted by using the fuzzy logic theory developed by Zadeh (1975). His basic idea holds that, in the real world, we see ambiguous classes of objects, for example, a set of 'tall' people, a set of 'red' objects, and a set of 'stable' systems. However, human reasoning is often imprecise, most of it not amenable to a formulation within frameworks of classical logic and probability theory (Bouchon, 1987; Bouchon & Yao, 1990). The theory of classical logic permits a proposition to conclude one of two values: true or false. This kind of logic cannot represent vague concepts. In fuzzy logic theory, membership of an object in a set is represented by a real number between '0' and '1,' with '0' denoting no real membership and '1' denoting full membership. Thus, in fuzzy logic a proposition need not be simply true or false, but may signify some degree of truth or falsity (Zadeh, 1975).

The fuzzy logic approach is helpful in the interpretation of an ambiguous natural language (Zadeh, 1983). A fuzzy quantifier represents a value in natural language by using the certainty factor. It is convenient to express the membership function of a fuzzy set as a standard function whose parameters may be adjusted to fit a given membership function in an approximate fashion. This concept has proven useful in real applications as diverse as aluminum smelting control (Warren & Nicholls, 1999) and auditing (Lenard, Alam, & Booth, 2000).

(3) *Non-monotonic logic*. Non-monotonic logic makes assumptions with regard to incomplete knowledge that is more global in nature than single defaults. This type of assumption is useful in applications where most of the facts are known, and it is, therefore, reasonable to assume that if a proposition cannot be proven, it is false. This means that in a knowledge base if the fact $P(a)$ is not provable, then $\tilde{}P(a)$ is assumed to hold.

By augmenting a knowledge base with an assumption which states that if the fact $P(a)$ cannot be proved, assume its negation $\tilde{}P(a)$, non-monotonic logic completes the theory with respect to knowledge base. While a FOPL is complete if and only if every fact or its negation is in the system. Augmenting a knowledge base with the negation of all facts which are not derivable gives us a complete theory.

3. Properties of knowledge

We evaluate the above methods as knowledge representation methodologies by testing how much they satisfy the following five properties of knowledge. Plato defined knowledge as 'justified true belief' requiring three conditions: That something is true, that someone believes it is true, and that the particular person's belief is, indeed, justified (Russell & Norvig, 2003). Based on his definition, we defined five properties of knowledge: conceptualization, transfer, modification, integration and decomposition.

3.1. Conceptualization of knowledge

In knowledge management or management information systems, data or fact is defined as a primitive level of knowledge (Alavi & Leidner, 2001). Therefore, conceptualization is defined as the process of mapping from the facts (data or information) to a concept (Pearl, 1988; Shafer & Pearl, 1990). For example, mapping from the fact that 'the temperature is 100 °F' to the concept 'hot' is an example of conceptualization. The purpose of conceptualization is to simplify and summarize facts and eventually conceptualize and categorize facts and convert them into knowledge. We assume that everyone has the ability to conceptualize fact and also assume that the mechanism of conceptualization is different for everyone. For example, for some, the concept 'hot' means temperatures between 80 and 110 °F, while for

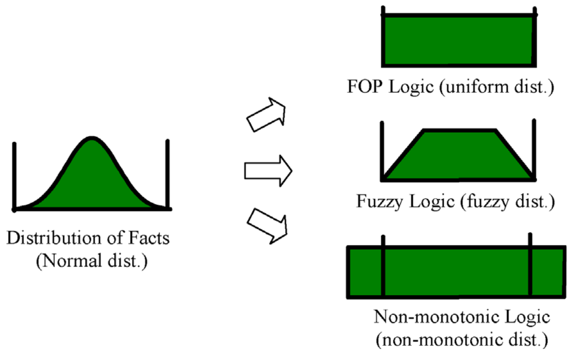

Fig. 2. Conceptualization.

the others, the concept 'hot' means temperatures between 90 and 120 °F.

Conceptualization can be considered as a grouping problem that classifies facts $X_1, X_2, \ldots, X_i, \ldots, X_n$ to knowledge groups $Y_1, Y_2, \ldots, Y_j \ldots, Y_m$ such as $Y_1 = \{x_1, \ldots, x_i\}, Y_2 = \{X_{i+1}, X_j\} \ldots, Y_m = \{X_{n-k}, \ldots, X_n\}$. In this paper, we will compare three logic-based methods, as shown in Fig. 2, and find out which one is more efficient in the sense of complexity, accuracy and completeness.

3.2. Transfer of knowledge

Knowledge generated by the conceptualization process will be transferred from an agent (human beings or computer, etc.) to an agent. However, an agent does not transfer fact itself, it transfers knowledge that is conceptualized. Then the agent which receives knowledge conceptualizes the received knowledge again with it's own conceptualization mechanism. For example, if knowledge were passed from agent 'A' to agent 'B', and 'B' to 'C' and 'C' to 'D', then knowledge is transferred from 'A' to 'D', and the meaning of knowledge may also be changed in each stage of transfer due to the conceptualization mechanism. Therefore, knowledge transfer is defined as the transmission process of knowledge, and can be expressed as the process $(Y_1, Y_2), (Y_2, Y_3), \ldots, (Y_i, Y_j), (Y_j, Y_k)$ and used to compare the meaning between Y_1, and Y_k. Fig. 3 is the pictorial representation of knowledge transfer. In this paper, we would like to know which method is more efficient in terms of accuracy, complexity and completeness.

3.3. Modification of knowledge

The application of knowledge to different but related domains is a common phenomenon. This kind of application is sometimes called an analogy or a guess. Therefore, knowledge modification can be defined as the modification of information or the application of the obtained knowledge to the different but related domains. Fig. 4 is the pictorial representation of knowledge modification.

For example, the fact 'temperature of 20 °F' will be interpreted 'very cold' in 'Florida' and this concept 'very cold' will be converted to ' − 20 °F' in 'Alaska'. To modify knowledge, the domains should be related to one another, or else the accuracy of modification will be decreased. This can be formalized as the problem of checking whether there is a mapping from one knowledge domain (Y_i) to another knowledge domain (Y_j). In this paper, we will test which one is better for this purpose.

3.4. Integration of knowledge

Sometimes we have to make our own knowledge by gathering information from several sources to include experiment. But the information from each source may be different, and we should adjust and integrate conclusions integrating the content of each piece of information. Hence, we define the integration of knowledge as a combination of the pieces of knowledge from different sources. There are two types of knowledge integration. The first is the combination of two different pieces of knowledge from different sources and the second type is the combination of prior knowledge with additional objective data. An example of the first case is that we can obtain different knowledge from two different people and make our own new knowledge by combining the two different pieces of knowledge. An example of the second case is that our knowledge is modified by new facts. In this paper, we will consider only the first case. This problem can be formalized as the problem of merging knowledge $Y_1, Y_2, \ldots,$ and knowledge Y_j

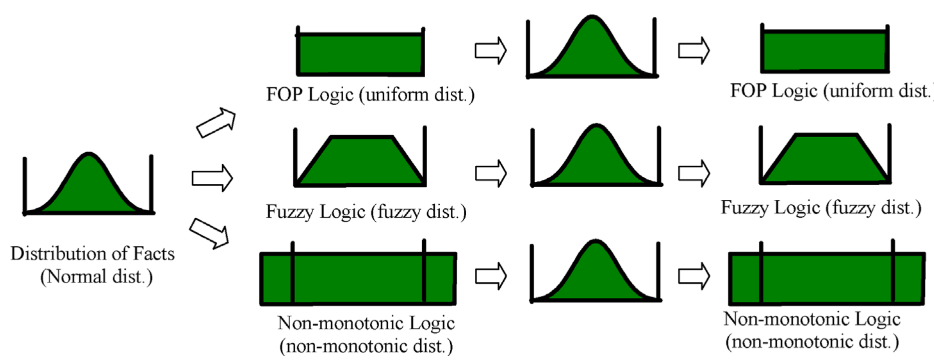

Fig. 3. Transfer.

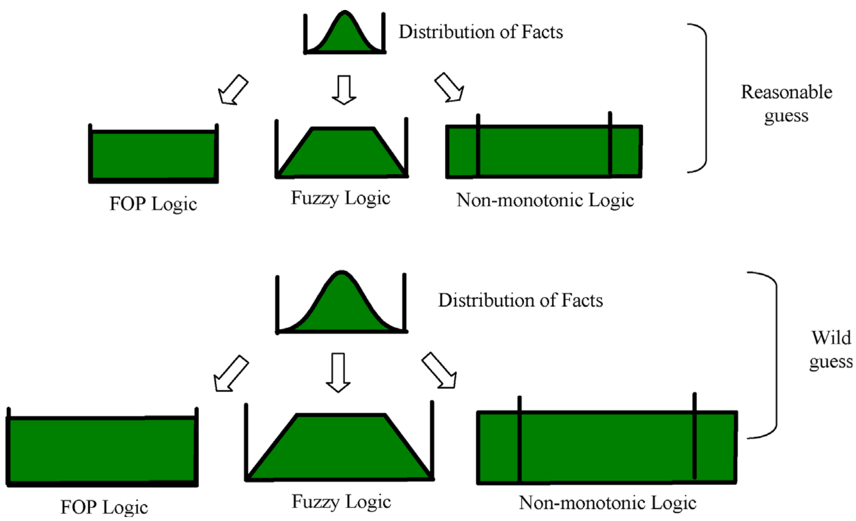

Fig. 4. Modification.

into the knowledge Y_{int}. Fig. 5 is the pictorial representation of knowledge.

3.5. Decomposition of knowledge

The last case is the analysis of complicated knowledge. Complicated knowledge is defined as knowledge that contains several concepts of sub-domains of knowledge. Segregation of complicated knowledge is defined as the decomposition. For example, if the concept of 'handsome' can be assumed as the complicated concept of 'tall' and the concept of 'slim', then we will infer height (the domain 'tall') and weight (the domain 'slim') from the concept 'handsome'. This problem may be considered as a multivariate problem. Therefore, this problem is classified into small groups from the combined concept. This problem can be formalized as the problem of how knowledge Y_k is classified into $Y_1, Y_2, \ldots, Y_j$. Fig. 6 is the pictorial representation of knowledge decomposition.

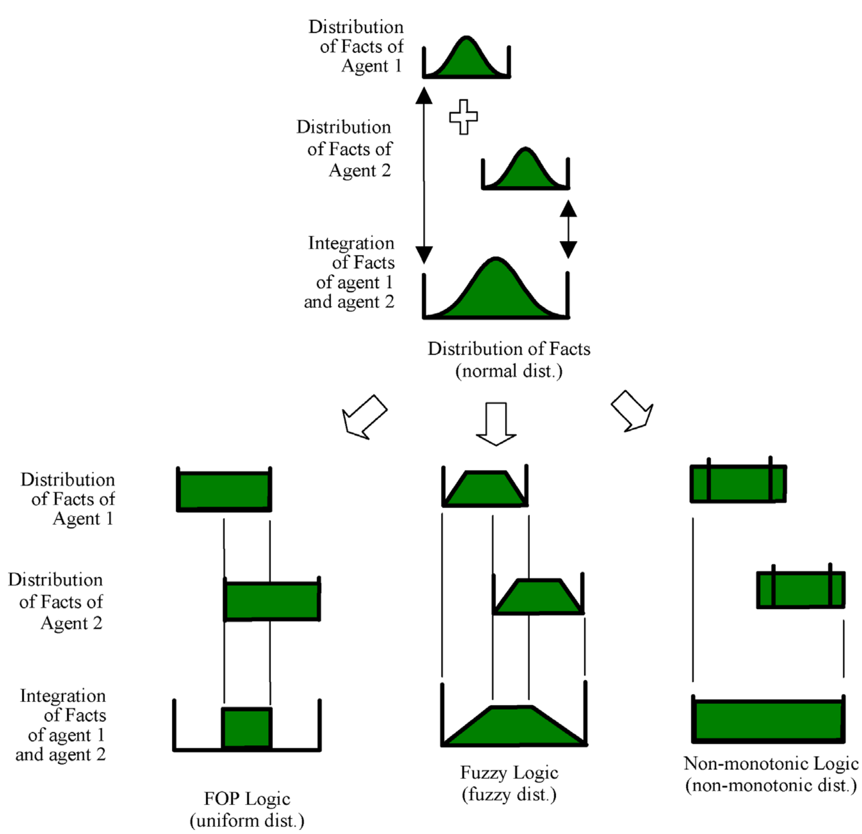

Fig. 5. Integration.

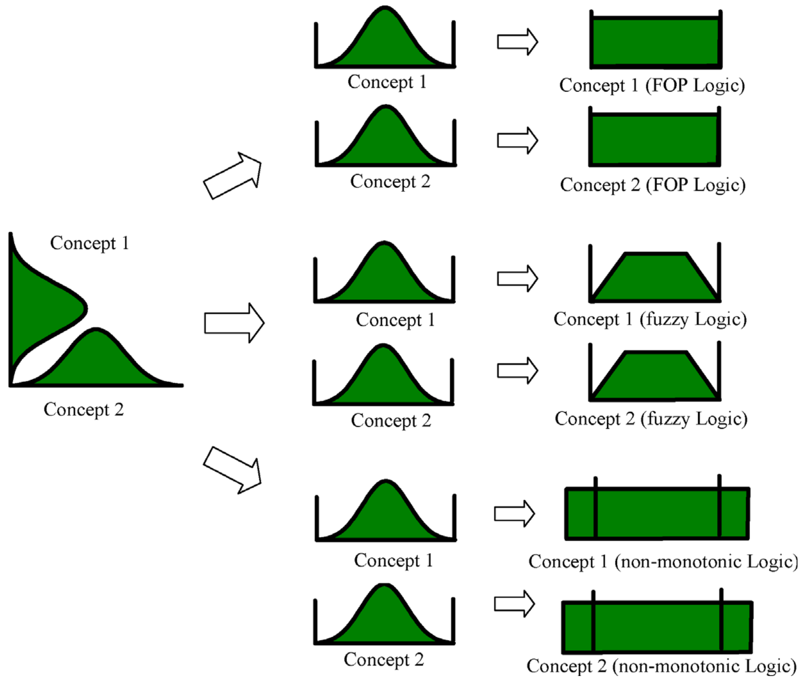

Fig. 6. Decomposition.

4. Evaluation of knowledge representation methods

The three knowledge representation methods are compared in this section. We will explain the criteria we used, and establish the relationship on complexity analytically. We will then use simulation to compare the methods on accuracy.

4.1. Criteria

Knowledge representation methods could be evaluated from different perspectives such as ease of use, psychological, or technological perspectives and so on. Within technological perspectives, completeness, consistency, complexity and accuracy can be considered. Here, we evaluate the five properties over the three criteria of accuracy, complexity and completeness.

Definitions for interpretation, validity, complexity, and completeness follow Change and Lee (1973), Mendelson (2001) and Torsun (1995). The knowledge representation method is defined as accurate if and only if there exists an interpretation such that knowledge is evaluated as true in the interpretation and the knowledge representation method is valid if and only if every interpretation of the knowledge is true. Complexity is a measurement of the time needed to decide the truth of the knowledge. If this time is exponential, then heuristics should be considered. Completeness measures whether the accuracy is consistent at any interpretations. If the accuracy of knowledge is consistent at any interpretations, then the knowledge is assumed complete. Based on these definitions of accuracy, complexity and completeness, we define the degree of accuracy, the degree of complexity and the degree of completeness as follows.

Definition. The degree of accuracy is defined as the percentage of the correct conceptualization in a given knowledge representation method. If the number of mis-mappings—from the fact to the concept increases, then the degree of accuracy decreases and vice versa. Because other properties of knowledge include the process of conceptualization, the degree of accuracy is related to every property of knowledge.

Definition. The degree of complexity is defined as the order of the complexity function for the operations of knowledge properties. Therefore, for example, the degree of complexity of $O(N^i)$ is higher than the degree of complexity of $O(N^j)$ if $i > j$. It means that if the degree of complexity is increases, the time of mapping increases and vice versa.

Definition. The degree of completeness is related to the range of the conceptualization. If the rate of accuracy is not changed, regardless the size of the range, then the knowledge representation method is complete. Therefore, even though the range of the application is changed, the rate of acceptance of the accuracy is not changed then the degree of completeness is higher and vice versa.

4.2. Comparison of complexity

We would like to show that all of three methods have an exponential time complexity. This means that it is impossible to manage the knowledge to obtain optimal solutions. We also find that accuracy and the complexity conflict, therefore, we need to trade off between accuracy and complexity.

Definition. A group is defined as a set of facts that have a similar property. That means that a group is a group of elements that have same concept.

Example. The concept 'teenager' is the group of facts of people whose ages are $\{13, 14, 15, 16, 17, 18, 19\}$.

Lemma. *As the order of the group increases, accuracy decreases.*

Proof. Let us define sub-concepts of the group as the member and the order of the group to be the number of members in the group. Let F be a set of facts and G a set of groups. If a group contains only one fact, then there is a one to one mapping from F to G. On the other hand, if every fact is mapped to one group, then it is a many to one mapping and there is no power of discrimination. Therefore, the greater the number of groups, the more accurate the mapping but more groups take more time to discriminate. □

Example. Let F be a set of persons aged $\{12, 25, 45, 56, 71, 93\}$, C_1 be a set of persons of all ages, C_2 be a set of {young, middle age, old}, C_3 be a set of each person. Then $C_1 = \{12, 25, 45, 56, 71, 93\}$, $C_2 = \{(12, 25), (45, 56), (71, 93)\}$, and $C_3 = \{(12), (25), (45), (56), (71), (93)\}$. Here C_2 has 3 members; (young), (middle age), (old). And each member has two facts. C_3 has 6 members and the order is 6. Each member has one fact. In this example, the order of accuracy is C_3, C_2, C_1 because the order of C_1 is 1, the order of C_2 is 3 and the order of C_3 is 6.

Theorem. *As accuracy increases, complexity also increases and as accuracy decreases, complexity decreases.*

Proof. If we form groups and categorize their facts we can group the given facts. Then, as we make the size of the group larger, the more facts will fit a given group, and the number of mappings will decrease, while the accuracy will also decrease. On the other hand, if we decrease the size of the groups, the number of the groups will increase and accuracy and time will increase. Let the order of group be j, the number of facts be n, mapping time be m and classification time be c. Then the worst case mapping time from F to C_i is $n^*(j^*c + m)$. It means that as the order increases, the time also increases. For accuracy, the mean value of members approaches to the real mean of the facts as the order of the group increases. Therefore, as the order of the group increases, accuracy and complexity increase and as the order of the group decreases, accuracy and complexity decrease. □

Example. Let mapping time be m and classification time be c. Then, in the above example, the time of mapping from F to C_1 is $6^*(1c + m)$, the time of mapping F to C_2 is $6^*(3c + m)$, C3 is $6^*(6c + m)$.

Corollary. *One to one mapping group (knowledge) and fact (information) is the most accurate knowledge representation.*

Proof. If a group is made for just one fact, then there is a one to one mapping from fact to group. This means that we don't group facts but rather simply list them. Then the total time of mapping is the product of the number of facts and the unit time of one mapping. This means the worst-case time complexity is $n^*(n^*c + m)$. Because it is a one to one mapping, the degree of accuracy is the highest. □

Theorem. *The complexity of conceptualization of knowledge representation by predicate logic is NP-complete.*

Definition. The conceptualization of knowledge representation (CKR) is to map the facts into groups. Therefore, facts $X_1, X_2, \ldots X_i, \ldots X_n$ are grouped into $Y_1, Y_2, \ldots Y_i, \ldots Y_m$ where $Y_1 = (X_1, X_2, \ldots X_j)$, $Y_2 = (X_{j+1}, X_{j+2}, \ldots X_{2j}), \ldots$ $Y_i = (X_{(i-1)j+1}, X_{(I-1)j+2}, \ldots X_{in}), \ldots$ $Y_m = (X_{(m-1)j+1}, \ldots, X_{mj})$.

Proof. Deciding the truth-value of fact $(X_1, X_2, \ldots X_j, \ldots X_n)$ using predicate logic is called a SATISFIABILITY problem and it is known to be NP-complete (Cormen, Leiserson, Rivest, & Stein, 2001; Papadimitriou & Steiglitz, 1998). The problem of deciding the truth value of the knowledge set $(Y_1, Y_2, \ldots Y_i, \ldots Y_m)$ is obviously NP, when m goes to infinity. To transform a SATISFIABILITY problem to a CKR problem, we need an algorithm that changes a SATISFIABILITY problem to a CKR problem in polynomial time. The time to classify X_i into group m takes time M and we repeat this procedure N times, the time complexity is M^*N, i.e. polynomial time, $O(N^2)$. Therefore the CKR problem is NP-complete. □

Corollary. *Other properties of knowledge are NP-complete.*

Proof. Because the other properties of knowledge involve the conceptualization of knowledge, all of the properties are NP-complete. Transfer can be defined as the concatenation of conceptualization, therefore transfer is NP-complete. Modification is defined as the concatenation of conceptualization and the shift of range, therefore modification is

NP-complete. Integration is defined as the concatenation of the combination of ranges and conceptualization, therefore integration is NP-complete. Decomposition is the concatenation of the separation of facts and conceptualization. Therefore, it is NP-complete. Finally, every property is also NP-complete. □

Theorem. *Complexity of knowledge representation by fuzzy logic is NP-complete.*

Definition. Conceptualization of fuzzy knowledge representation (CFKR) is to group facts into categories. Therefore, fact $X_1, X_2, \ldots X_i, \ldots X_n$ is grouped into $Y_1, Y_2, \ldots Y_i, \ldots Y_m$ where $Y_1 = (X_1, X_2, \ldots X_j)$, $Y_2 = (X_{j+1}, X_{j+2}, \ldots X_{2j}), \ldots$ $Y_i = (X_{(i-1)j+1}, X_{(I-1)j+2}, \ldots X_{ij}), \ldots$ $Y_m = (X_{(m-1)j+1}, \ldots, X_{mj})$. The only difference is that every element is weighted by a membership function whose value is between 0 and 1.

Proof. Deciding the truth-value of fact $(X_1, X_2, \ldots X_j, \ldots X_n)$ using predicate logic is called a SATISFIABILITY problem and it is known as NP-complete (Cormen et al., 2001). The problem of deciding the truth value of the knowledge $(Y_1, Y_2, \ldots Y_i, \ldots Y_m)$ is obviously NP, when m goes to infinity. To transform a SATISFIABILITY problem to a CKR problem, we need an algorithm that changes the SATISFIABILITY problem to a CKR problem in polynomial time. The time to classify X_i into m group takes time M and we repeat this procedure N times, therefore time complexity is M^*N, i.e. polynomial time, $O(N^2)$. Therefore, the CKR problem is NP-complete. The CFKR problem consists of two stages. The first stage is exactly the same as a CKR problem and the second stage is to weigh the each elements of the groups. Therefore, the time complexity is $M^*N + N$, i.e. polynomial time, $O(N^2)$. □

Corollary. *Other properties of knowledge using fuzzy logic are NP-complete.*

Proof. Other properties here are transfer, modification, integration and decomposition. These properties are the concatenation of conceptualization and other procedures. We already have found that the complexity of conceptualization is NP-complete, therefore, other properties are also NP-complete. □

Theorem. *Complexity of knowledge representation by non-monotonic logic is NP-complete.*

Definition. Conceptualization of non-monotonic knowledge representation (NMKR) is to group facts into categories. Therefore, fact $X_1, X_2, \ldots X_i, \ldots X_n$ is grouped into $Y_1, Y_2, \ldots Y_i, \ldots Y_m$ where $Y_1 = (X_1, X_2, \ldots X_j)$, $Y_2 = (X_{j+1}, X_{j+2}, \ldots X_{2j}), \ldots$ $Y_i = (X_{(i-1)j+1}, X_{(I-1)j+2}, \ldots X_{ij}), \ldots$ $Y_m = (X_{(m-1)j+1}, \ldots, X_{mj})$. If X_k is beyond the range of Y_m, then rearrange the range of the group Y_m and/or generate another group Y_{m+1}.

Proof. Deciding the truth-value of fact $(X_1, X_2, \ldots X_j, \ldots X_n)$ using predicate logic is called a SATISFIABILITY problem and it is known as NP-complete (Cormen et al., 2001). The problem of deciding the truth value of the knowledge $(Y_1, Y_2, \ldots Y_i, \ldots Y_m)$ is obviously NP, when m goes to infinity. To transform a SATISFIABILITY problem to a CKR problem, we need an algorithm that changes the SATISFIABILITY problem to a CKR problem in polynomial time. The time to classify X_i into m group takes time M and we repeat this procedure N times, therefore time complexity is M^*N, i.e. polynomial time, $O(N^2)$. Therefore, the CKR problem is NP-complete. The NMKR problem consists of two stages. The first stage is to find the facts that are beyond the scope and rearrange the group if needed. The second stage is exactly the same as a CKR problem. Therefore, the time complexity is $M^*N + (K^*M)$, where K is the number of outlier. i.e. polynomial time, $O(N^2)$. □

Corollary. *Other properties of knowledge by using non-monotonic logic are NP-complete.*

Proof. Other properties mean transfer, modification, integration and decomposition. These properties are the concatenation of conceptualization and other procedures. We already have found that the complexity of conceptualization is NP-complete, therefore, other properties are also NP-complete.
The complexity of each method is NP-complete. Therefore, it is not possible to obtain the optimal solution in a given time and we cannot say which method is better in terms of complexity. However, heuristic methods can be designed to obtain satisfying solutions for reasonably sized problems. □

4.3. Comparison of accuracy and completeness

To compare accuracy and completeness, we will develop a heuristic method. For that purpose, we will make some assumptions.

1. A fact is represented as a random number.
2. A group is considered as a concept of a certain domain and the distribution of a group is different for each knowledge representation method.
3. Facts related to a concept are normally distributed. This is considered to be a reasonable assumption because a concept has a range of facts and there are facts many people think of as a concept. For example, the temperature 60 °F is a fact. In a certain place, if people think a 'mild temperature' is somewhere between 50 and 80 °F, then we think the concept of a 'mild temperature' has a normal distribution with a mean of 65 °F and a certain variance.

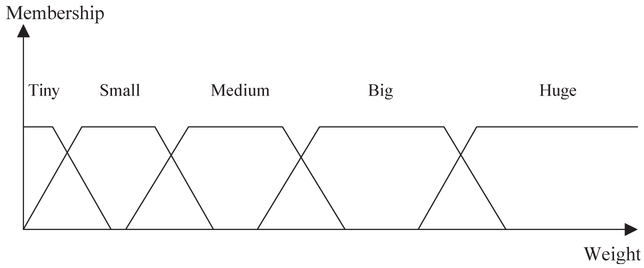

Fig. 7. Fuzzy distribution of concept ‘weight’.

4. The truth value of predicate logic is binary; true or false. This means that if a fact satisfies the meaning of a concept, it is definitely true and if not, it is definitely false. Without loss of generality, we can assume the fact of a concept in predicate logic follows the uniform distribution. For example, if a concept ‘young’ is defined as the age 1–20, then every age in this range is true and any age outside of this range is false in predicate logic.

   Expression of fuzzy logic assumes a fuzzy distribution. The following is an example of a fuzzy distribution of an ordered list of the domain ‘weight’ = {tiny, small, medium, big, huge}. Fig. 7 is the pictorial representation of fuzzy distribution of concept ‘weight’.
5. Expression of non-monotonic logic is assumed to have a uniform distribution like FOPL, the only difference from FOPL is that the range of the category is extended for the value of the outliers. For example, if a concept ‘young’ is defined as the age 1–20, then every age in this range is true and any age outside of this range is false in predicate logic. But age 25 can be considered as ‘young’ in a certain environments. It means the range of truth is flexible in non-monotonic logic.

Based on the above assumptions, we develop the heuristic algorithm.

5. Heuristic algorithm for accuracy

The heuristic algorithm used in the simulation is described over the five measures we use in this study.

5.1. Conceptualization of knowledge

Conceptualization is considered as a mapping from the set of facts to concepts. The ability of predicate logic to accurately reflect conceptualization is measured by the rate of match between the original normally distributed data and data drawn from a uniform distribution. The ability of fuzzy logic to accurately reflect conceptualization is measured by the rate of match between the original normally distributed data and data drawn from the fuzzy distribution. The ability of non-monotonic logic is the expansion of category whenever the outliers are founded.

```
Begin
  Do k = 1 to p
    Do j = 1 to m
      Do i = 1 to n
        Generate a random number from normal
        distribution.
      End Do /*Generate 'n' random numbers from
      normal distribution and map to the group*/
      Count the number (x) of random numbers within
      the original range.
      Calculate and keep the mean and the variance of
      the normal distribution
      Do i = 1 to q
        Generate the random numbers from the
        groups of knowledge representation distri-
        bution and uniform distribution as many
        times as the counting number
        If found outliers, put into the one of the end-
        side groups in the case of non-monotonic
        logic
      End Do /*Generate x random numbers from
      knowledge representation distribution*/
      Calculate and keep the mean and the variance of
      the knowledge representation distribution (uni-
      form, fuzzy, and non-monotonic distribution)
    End Do /*Generate "m" means and variances*/
    Test the mean and variance of the knowledge
    representation distribution and normal distribution
    to compare whether the mean and variance is the
    same.
    Summarize the result that is the percentages of m
    tests.
  End Do /*Repeat above procedures "p" times*/
  Summarize the results
End /*Result is the percentage of acceptance rate*/
```

5.2. Transfer of knowledge

Transfer is measured as accuracy after repeated conceptualization.

```
Begin
  Do k = 1 to p
    Do i = 1 to n
      Generate a random number from normal
      distribution /*This first group will be the basis
      of T-tests and F-test with other different phases
      of each distribution*/
    End Do /*Generate n random numbers from
    normal distribution and map to the group*/
    Count the number (x) of random numbers within
    the original range.
    Calculate and keep the mean and the variance of
    the normal distribution
    Do j = 1 to m
      Do i = 1 to q
```

```
            Generate the random numbers from the groups of knowledge representation distribution and uniform distribution as many times as the counting number
            If found outliers, put into the one of the end-side groups in the case of non-monotonic logic.
          End Do /*Generate x random numbers from knowledge representation distribution*/
          Calculate and keep the mean and the variance of the knowledge representation distribution (uniform, fuzzy, and non-monotonic distribution)
        End Do /*Generate m means and variances*/
          Test the mean and variance of the knowledge representation distribution (uniform, fuzzy, and non-monotonic distribution)
          and the first generate normal distribution to compare whether the mean and variance is the same.
          Summarize the result that is the percentages of m tests.
        End Do /*Repeat the above procedures p times*/
        Summarize the results
    End /*Result is the percentage of acceptance rate*/
```

5.3. Modification of knowledge

Modification of data is measured as the range or expansion of the range.

```
    Begin
      Do k = 1 to p
        Do l = 1 to q
          Do j = 1 to m
            Do i = 1 to n
              Generate a random number from a normal distribution.
            End Do /*Generate n random numbers from normal distribution and map to the
            group*/
              Count the number (x) of random numbers within the original range.
              Calculate and keep the mean and the variance of the normal distribution
            Do i = 1 to r
              Generate the random numbers from the groups of knowledge representation distribution and uniform distribution as many times as the counting number
              If found outliers, put into the one of the end-side groups in the case of non-monotonic logic
            End Do /*Generate x random numbers from knowledge representation distribution*/
            Calculate and keep the mean and the variance of the knowledge representation distribution (uniform, fuzzy, and non-monotonic distribution).
          End Do/*Generate m means and variances*/
            Test the mean and variance of the knowledge representation distribution and the first generate normal distribution to compare whether the mean and variance is the same.
            Summarize the percentages of m tests.
          Change the range of knowledge representation distribution
        End Do /*Do q times by changing range*/
        Summarize the results
      End Do /*Repeat the above procedures p times*/
      Summarize the results
    End /*Result is the percentage of acceptance rate*/
```

5.4. Integration of knowledge

Integration is measured as the ability to accurately combine data from normal distributions with different means.

```
    Begin
      Do k = 1 to p
        Decide the order of the groups of normal distributions
        Do j = 1 to m
          Do i = 1 to n
            Do f = 1 to r
              Generate a random number from normal distribution i.
              Map the random number to the corresponding group of the synthesized distribution
            End Do /*Generate r random numbers from normal distribution i*/
          End Do /*Generate "r*q" random numbers from normal distribution and map to the group*/
          Count the number (q) of random numbers within the original range.
          Calculate and keep the mean and the variance of the synthesized normal distribution
          Do i = 1 to (r*q)
            Generate random numbers from the group of the knowledge representation and uniform distribution by the counting number
            If found outliers, put into the one of the end-side groups in the case of non-monotonic logic.
          End Do/*Generate "r*q" random numbers from knowledge representation distribution*/
          Calculate and keep the mean and the variance
        End Do /*Generate m means and variances*/
          Test the mean and variance of the knowledge representation distribution (uniform, fuzzy, and non-monotonic distribution) and the first
```

```
            generate normal distribution to compare
            whether the mean and variance is the same.
        Summarize the result as the percentages of m tests.
    End Do /*Repeat the above procedures p times*/
    Summarize the results
End/*Result is the percentage of acceptance rate*/
```

5.5. Decomposition of knowledge

Decomposition is the segregation of multivariate distribution to several univariate distributions. It is measured by the ability of the predicate logic, fuzzy logic and non-monotonic logic to capture data composed from two sources.

```
Begin
    CRV[1,0] = 1
    Do k = 1 to p
        Decide the order of the group
        Do j = 1 to m
            Do f = 1 to r
                Do i = 1 to n
                    Generate a random number from normal
                    distribution i and call it rv.
                    RV[i] = rv
                    CRV[f, i] = CRV[f, i - 1]* RV[i]
                    Keep CRV[f, i]
                End Do /*Generate n random number and con-
                catenate r times, consider as r properties
                multivariate random variables*/
            End Do /*Generate "n* r" random numbers from
            normal distribution*/
            Do j = 1 to n* r
                Normalize the range of CRV[f, i] by
                CRV[f, i]/(range of rv)^r
            End Do /*Normalize n* r variables*/
            Map the random number to the corresponding
            group of distribution
            Count the number of random numbers in each
            group and call it a counting number
            Calculate and keep the mean and the variance of
            normal distribution
            Do i = 1 to (n* r)
                Generate a random number from the group of
                knowledge representation distribution by the
                counting number
                If found outliers, put into the one of the end-
                side groups in the case of non-monotonic logic.
            End Do/*Generate "n* r" random numbers from
            knowledge representation distribution*/
            Calculate and keep the mean and the variance of
            the knowledge representation distribution
            (uniform, fuzzy, and non-monotonic distribution)
        End Do /*Generate m means and variances*/
            Test the mean and variance of the knowledge
            representation distribution (uniform, fuzzy,
            and non-monotonic distribution) and the first
            generate normal distribution to compare whether
            the mean and variance is the same.
        Summarize the result that is the percentages of m
        tests.
    End Do /*Repeat the above procedures p times*/
    Summarize the results
End /*Result is the percentage of acceptance rate*/
```

6. Heuristic algorithm for completeness of knowledge

```
Begin
    Do k = 1 to p
        Decide (change) the size of range and the order of
        the group
            Do j = 1 to m
            Do i = 1 to n
                Generate a random number from normal
                distribution.
            End Do /*Generate 'n' random numbers from
            normal distribution and map to the group*/
            Count the number of random numbers within
            the original range.
            Call it a counting number x.
            Calculate and keep the mean and the variance of
            the normal distribution
            Do i = 1 to q
                Generate the random numbers from the groups
                of knowledge representation distribution and
                uniform distribution as many times as the
                counting number
                If found outliers, put into the one of the end-
                side groups in the case of non-monotonic logic.
            End Do /*Generate x random numbers from
            knowledge representation distribution*/
            Calculate and keep the mean and the variance of
            the knowledge representation distribution (uni-
            form, fuzzy, and non-monotonic distribution)
        End Do /*Generate "m" means and variances*/
            Test the mean and variance of the knowledge
            representation distribution (uniform, fuzzy, and
            non-monotonic distribution) and the first generate
            normal distribution to compare whether the mean
            and variance is the same.
        Summarize the result that is the percentages of m
        tests.
    End Do /*Repeat "p" times whenever we change the
    range of distribution (e.g. 100 → 50, 150, and 200)*/
    Summarize the results
End /*Result is the percentage of acceptance rate*/
```

7. Implementation of accuracy

We simulated the above algorithms as follows. For implementation, we made some additional assumptions in addition to the previous assumptions.

1. The range of random numbers 100–199 is the generally accepted range of a certain concept.
2. Some people extend the range in both directions.
3. We posit the null hypothesis that the mean of actual knowledge distribution and the knowledge representation method distribution (predicate logic, fuzzy logic and non-monotonic logic) is the same. If this null hypothesis is accepted, we interpret that the meaning of the knowledge has not changed.
4. We posit the null hypothesis that the variance of actual knowledge distribution and the knowledge representation method distribution (predicate logic, fuzzy logic and non-monotonic logic) is the same. If this null hypothesis is accepted, we interpret that the generally accepted perception of a concept or the norm is not changed.

7.1. Conceptualization of knowledge

7.1.1. Procedures

1. Generate 100 numbers the fuzzy distribution by normal random number generator at the range of 100–199.
2. These numbers are classified into the predetermined range. In this experiment, region is from 100 to 199
3. Count the random numbers that belong to the region. This number is used to generate the same number of random numbers from the uniform distribution and the normal distribution. The amount of random numbers in the region is called as the counting number.
4. Generate as many random numbers from the uniform distribution as the counting number.
5. Generate as many random numbers from the fuzzy distribution as the counting number.
6. Generate 100 random numbers from non-monotonic distribution.
7. Calculate the mean and the variance for each set of 100 numbers.
8. Do the *t*-test and *F*-test to compare the mean and the variance of numbers from the uniform distribution and the numbers from the normal distribution at the 95% significance level.
9. Do the *t*-test and *F*-test to compare the mean and the variance of numbers from the fuzzy distribution and the numbers from the normal distribution at the 95% significance level.
10. Do the *t*-test and *F*-test to compare the mean and the variance of numbers from the non-monotonic distribution and the numbers from the normal distribution at the 95% significance level.
11. Repeat procedures from steps 1 to 12 for 100 times and obtain results of 100 *t*-tests and 100 *F*-tests. We will call it one batch.
12. Repeat the 'batch' 100 times and obtain the 100 sets of 100 *t*-tests and *F*-tests.

7.1.2. Results

Table 1 shows abilities for numbers drawn from the fuzzy distribution (representing fuzzy logic), uniform distribution (representing predicate logic), and non-monotonic distribution (representing non-monotonic logic) to reflect the normally distributed numbers (reflecting true data). Using a 70% acceptance rate, we found that there are no differences between the variances of the normal distribution (actual knowledge distribution) and the fuzzy distribution (knowledge representation distribution) in 93 cases out of hundred. There are greater differences for the uniform distribution, and non-monotonic distribution. In the uniform distribution, more than half of variances (52 out of 100) fall in the range between 30 and 69% acceptance rate and only one quarter of the variances fall into the range of 'over 70%'. The non-monotonic distribution shows a little better results than uniform distribution. In the non-monotonic distribution, 46 out of 100 fall in the range between 30 and 69% acceptance rate and a third of results fall in the range of 'over 70%'.

Means obtained from all three methods were quite accurate. The non-monotonic distribution and uniform distribution had slightly better results than the fuzzy distribution. However, the superior performance on variance implies that the fuzzy representation more consistently reflects the true data than does the predicate logic method, and non-monotonic method. More detailed results are summarized in Tables 1 and 2.

7.2. Transfer of knowledge

7.2.1. Procedures

Transfer is the repeat of conceptualization. Therefore, this procedure is almost the same as for conceptualization except that the conceptualization procedure is repeated as many times as the number of information transfers. In this

Table 1
Result of conceptualization (variance)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	0	8	8
10–19	0	9	9
20–29	0	6	4
30–39	0	10	9
40–49	0	14	11
50–59	1	13	16
60–69	6	15	10
70–79	15	9	16
80–89	25	8	10
90–100	53	8	7
Total	100	100	100

Table 2
Result of conceptualization (mean)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	0	0	0
10–19	0	0	0
20–29	0	0	0
30–39	0	0	0
40–49	2	0	2
50–59	6	3	0
60–69	6	1	3
70–79	13	6	3
80–89	20	11	12
90–100	53	79	80
Total	100	100	100

Table 3
Result of transfer

First occurrence of reject	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–3	13	36	38
4–6	9	7	6
7–9	2	1	0
10–12	5	0	1
13–15	4	2	1
16–60	17	4	4
Total	50	50	50

experiment, steps 3–12 of procedure for 'conceptualization' were repeated 60 times. This means that information was transferred from one agent to another agent sequentially 60 times. We compared the meaning of the knowledge among the first agent, the second agent and so on until the 60th agent.

7.2.2. Results

The range 0–9 in Table 3 shows that the frequency of the first occurrence of rejection for the fuzzy distribution, which is 24, is smaller than that of the uniform distribution, which is 44, and that of non-monotonic distribution, which is also 44. This means that the first rejection occurred later for the fuzzy distribution than for the uniform distribution and non-monotonic distribution. This means that the fuzzy distribution retains variance similarity toward normal numbers further than the uniform distribution and non-monotonic distribution does. This can be interpreted as implying that the fuzzy distribution is more robust than predicate logic and non-monotonic logic in knowledge transfer. However, the accuracy of the fuzzy distribution decreased rapidly after 10 transfers. This means that all three methods are limited in transferring knowledge.

7.3. Modification of knowledge

7.3.1. Procedures

In modification of knowledge, we expanded the range of 'conceptualization'. By expanding the original range by five percent each time, we compared the variance and mean

Table 4
Acceptance rate by expansion (fuzzy distribution (variance))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	0	0	0	5	9	32	43	51	50	66	66
20–39	2	4	4	8	21	17	15	17	18	13	12
40–59	11	2	13	20	27	18	15	16	14	12	9
60–79	20	25	33	38	28	18	10	5	13	8	8
80–100	67	69	50	29	15	15	17	11	5	1	5
Total	100	100	100	100	100	100	100	100	100	100	100

Table 5
Acceptance rate by expansion (uniform distribution (variance))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	30	23	34	40	61	72	72	87	90	94	88
20–39	15	12	21	26	26	17	17	7	4	3	7
40–59	7	13	22	23	9	6	8	6	3	3	2
60–79	9	14	10	8	3	2	3	0	1	0	3
80–100	39	38	13	3	1	3	0	0	2	0	0
Total	100	100	100	100	100	100	100	100	100	100	100

Table 6
Acceptance rate by expansion (non-monotonic distribution (variance))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	25	30	23	34	40	61	72	72	87	90	94
20–39	18	15	12	21	26	26	17	17	7	4	3
40–59	9	7	13	22	23	9	6	8	6	3	3
60–79	8	9	14	10	8	3	2	3	0	1	0
80–100	40	39	38	13	3	1	3	0	0	2	0
Total	100	100	100	100	100	100	100	100	100	100	100

Table 7
Acceptance rate by expansion (fuzzy distribution (mean))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	0	0	0	0	7	29	44	53	54	64	74
20–39	2	1	4	9	17	16	13	12	19	16	11
40–59	10	7	7	25	27	12	18	14	10	14	6
60–79	15	21	35	31	20	18	12	10	11	4	6
80–100	73	71	54	35	29	25	13	11	6	2	3
Total		100	100	100	100	100	100	100	100	100	100

Table 8
Acceptance rate by expansion (uniform distribution (mean))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	27	23	22	35	53	59	76	85	88	96	93
20–39	8	15	20	25	23	18	14	9	7	3	5
40–59	7	14	25	20	18	13	5	5	3	1	1
60–79	18	8	15	10	5	9	4	1	1	0	1
80–100	40	40	18	10	1	1	1	0	1	0	0
Total	100	100	100	100	100	100	100	100	100	100	100

of two groups; the original and the expanded. We expanded ten times. For example, a five percent expansion of the original range changed the range from 100 to 95 for the lower limit and from 199 to 204 for the upper limit. Therefore, a 50% expansion of the original range became [75,224] from [100,199] and a 100% expansion became [50,249].

7.3.2. Results

Tables 4–6 show the results for the 100 rounds of the 100 F-tests for the variance of the fuzzy distribution, the uniform distribution and the non-monotonic distribution. The acceptance rate for the fuzzy distribution decreases gradually without radical change on the whole, while the acceptance

Table 9
Acceptance rate by expansion (non-monotonic distribution (mean))

Range of acceptance rate	Expansion										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0–19	26	22	26	37	56	63	79	84	89	96	96
20–39	8	15	17	25	23	14	10	9	6	3	2
40–59	8	13	20	17	14	13	5	6	2	1	1
60–79	13	10	19	10	5	8	4	1	2	0	1
80–100	45	40	18	11	2	2	2	0	1	0	0
Total	100	100	100	100	100	100	100	100	100	100	100

Table 10
Result of integration (variance)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	0	0	0
10–19	0	1	0
20–29	0	3	0
30–39	0	9	0
40–49	0	12	0
50–59	0	21	5
60–69	0	27	8
70–79	2	18	16
80–89	13	8	22
90–100	85	1	49
Total	100	100	100

rate of the uniform distribution and non-monotonic distribution for variance decrease radically at 20% and at 30% of expansion rate, respectively. Fuzzy logic has much higher average acceptance rate based on variance at the 10% expansion rate than uniform or non-monotonic distributions in the range of 80–100 (69, 38 and 39, respectively). This result makes sense, because predicate logic or non-monotonic logic is stricter for measuring truth than the fuzzy logic. This means that fuzzy logic is more robust than the uniform and non-monotonic distribution. In modification of knowledge, the acceptance rates of three logics decrease as the expansion rate increases. More detailed results are summarized in Tables 4–6. The analysis of mean in Tables 7–9 shows the similar results to the analysis of variance.

7.4. Integration of knowledge

7.4.1. Procedures

To test integration, we expanded the range of the 'conceptualization' by merging two sets of the 100 normal distribution numbers (one set with range from 100 to 199 and the other set with range from 150 to 249) into one normal distribution of 200 numbers with a range from 100 to 249. We also generated 200 random numbers from a uniform distribution and non-monotonic distribution with a range of [100,249]. Then we tested to compare the variance and the mean of the fuzzy distribution and the uniform distribution and non-monotonic distribution.

Table 11
Result of integration (mean)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	0	0	0
10–19	0	0	1
20–29	0	0	1
30–39	1	0	1
40–49	0	1	5
50–59	9	1	10
60–69	9	10	9
70–79	22	14	23
80–89	31	33	17
90–100	28	41	33
Total	100	100	100

Table 12
Result of decomposition (variance)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	0	35	37
10–19	6	9	8
20–29	11	5	7
30–39	16	5	4
40–49	6	5	6
50–59	11	9	6
60–69	13	4	2
70–79	15	8	10
80–89	13	9	9
90–100	9	11	11
Total	100	100	100

7.4.2. Results

In analyzing variances in Table 10, the fuzzy distribution shows a high acceptance rate (98 out of 100 reside within 80% or above acceptance rate), while the uniform distribution shows a relatively low acceptance rate (78 out of 100 reside between 40 and 79%). Non-monotonic distribution shows a relatively high acceptance rate compare to uniform distribution (71 out of 100 reside within 0% or above acceptance rate). These results show that the non-monotonic distribution has higher acceptance rate than fuzzy distribution, which has a higher acceptance rate than uniform distribution in the case of knowledge integration. This is interpreted as saying that the non-monotonic distribution is the most accurate when combining several pieces of knowledge. In analyzing means, all three distributions show no difference one another. Results are summarized in Tables 10 and 11.

Table 13
Result of decomposition (mean)

Frequency of acceptance rate	Distribution		
	Fuzzy dist	Uniform dist	Non-monotonic
0–9	16	25	84
10–19	18	14	4
20–29	10	10	5
30–39	21	19	2
40–49	21	19	0
50–59	14	12	2
60–69	0	1	1
70–79	0	0	0
80–89	0	0	1
90–100	0	0	1
Total	100	100	100

Table 14
Result of completeness (variance)

Frequency of acceptance rate	Range of distribution			
	50	100	150	200
0–9	4	0	0	0
10–19	3	0	0	0
20–29	0	0	1	0
30–39	0	0	0	0
40–49	7	0	1	1
50–59	13	0	1	1
60–69	5	4	5	0
70–79	4	13	4	1
80–89	9	27	11	3
90–100	55	56	77	94
Total	100	100	100	100

7.5. Decomposition of knowledge

In decomposition, we generated two sets of 20 normal numbers with a range from 0 to 99 and applied the Cartesian product to make 400 numbers. We also repeated the above procedure to make 400 numbers from the uniform distribution, the fuzzy distribution and non-monotonic distribution. We conducted a series of tests to compare the fuzzy distribution, the uniform distribution and the non-monotonic distribution.

7.5.1. Procedures

1. Generate two sets of 20 random numbers from the normal distribution with a range from 0 to 99.
2. Form Cartesian products with two sets of random numbers and obtain 400 pairs of random numbers
3. Add these two numbers pair-wise and obtain 400 numbers with a range of 0–198.
4. Count the numbers in the region of 0–198.
5. Normalize the counting number by dividing the number by 20.
6. Generate two sets of 20 uniformly distributed random numbers as many times as the counting numbers of the region with a range from 0 to 99.
7. Form Cartesian products with two sets of uniformly distributed numbers and obtain 400 pairs of random numbers.
8. Add these two numbers of the pair and obtain 400 numbers with a range of 0–198.
9. Repeat steps 7–9 for the fuzzy distribution.
10. Repeat steps 7–9 for the non-monotonic distribution
11. Repeat procedures from steps 1 to 10 for 100 times. There are the results of 100 *t*-tests and 100 *F*-tests. We will call it a batch.
12. Repeat the batch for 50 times and obtain the 50 sets of 100 *t*-tests and 100 *F*-tests.

Table 15
Result of completeness (mean)

Frequency of acceptance rate	Range of distribution			
	50	100	150	200
0–9	20	0	0	2
10–19	12	0	0	4
20–29	6	0	0	3
30–39	5	0	0	6
40–49	9	1	0	15
50–59	3	2	1	7
60–69	3	1	0	8
70–79	3	9	12	9
80–89	7	14	19	10
90–100	32	73	68	36
Total	100	100	100	100

Table 16
Result of completeness (variance)

Frequency of acceptance rate	Range and Num. in distribution			
	50	100	150	200
0–9	17	3	8	7
10–19	3	8	13	2
20–29	11	11	5	6
30–39	12	15	13	4
40–49	8	10	23	13
50–59	12	13	8	13
60–69	12	9	7	11
70–79	5	12	9	15
80–89	11	10	8	6
90–100	9	9	6	23
Total	100	100	100	100

7.5.2. Results

Analysis of the variances shown in Table 12 indicates that the fuzzy distribution shows a relatively high acceptance rate (60–100%) of 50 out of 100 and the uniform distribution and non-monotonic distribution show a relatively low acceptance rate (60–100%) of 32 out of 100.

Table 17
Result of completeness (mean)

Frequency of acceptance rate	Range and Num. in distribution			
	50	100	150	200
0–9	27	0	8	7
10–19	1	0	9	6
20–29	5	0	7	6
30–39	9	0	2	8
40–49	2	0	6	10
50–59	5	0	7	5
60–69	4	1	3	7
70–79	11	1	4	7
80–89	12	3	4	6
90–100	24	95	50	38
Total	100	100	100	100

Table 18
Result of completeness (variance)

Frequency of acceptance rate	Range and num. in distribution			
	50	100	150	200
0–9	16	7	9	36
10–19	4	5	13	16
20–29	5	9	5	13
30–39	9	14	13	6
40–49	10	9	20	12
50–59	12	10	8	7
60–69	3	10	10	4
70–79	22	12	8	1
80–89	14	16	10	2
90–100	5	8	4	2
Total	100	100	100	100

Table 19
Result of completeness (mean)

Frequency of acceptance rate	Range and num. in distribution			
	50	100	150	200
0–9	25	0	12	96
10–19	3	0	5	1
20–29	2	0	7	1
30–39	6	0	1	0
40–49	8	0	6	1
50–59	2	0	8	1
60–69	0	1	2	0
70–79	14	0	4	0
80–89	13	2	5	0
90–100	27	97	50	0
Total	100	100	99	100

Analysis of means in Table 13 shows a low acceptance rate, implying that a complicated concept cannot be separated efficiently by any kind of method. The acceptance rates of all three distributions are very low while fuzzy and uniform distributions show similar results and the non-monotonic distribution shows very low acceptance rate. The overall acceptance rate is very low. Therefore, it is hard to conclude that knowledge decomposition is more accurate by using fuzzy logic or predicate logic than by the non-monotonic logic.

8. Implementation of completeness

The procedure of testing completeness is the iterations of conceptualization. Therefore, this procedure is almost the same as for conceptualization except that the range of each distribution and the numbers in the each distribution is deliberately modified. To measure the completeness of knowledge for each distribution, we set four stages by increasing the numbers in distribution and the range simultaneously by 50 starting with range 50. The four stages have range of 50, 100, 150 and 200, respectively with the number of member of 50, 100, 150 and 200, respectively.

We believe if one distribution shows better result against other distributions in all four stages consistently, we believe the distribution has positive completeness. However, if the result of distribution fluctuates among four stages, we believe the distribution does not support completeness.

8.1. Fuzzy distribution

8.1.1. Result

In the variance analysis, the fuzzy distribution shows the acceptance frequency of 68, 96, 92 and 98 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. In the mean analysis, fuzzy distribution shows the acceptance frequency of 42, 96, 99 and 55 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. Detailed results are shown in Tables 14 and 15.

8.2. Uniform distribution

8.2.1. Result

In the variance analysis, uniform distribution shows the acceptance frequency of 25, 31, 23 and 44 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. In the mean analysis, uniform distribution shows the acceptance frequency of 47, 99, 58 and 51 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. Detailed results are shown in Tables 16 and 17.

Table 20
Overall result of completeness (variance)

Frequency of acceptance rate	50			100			150			200		
	F	U	NM	F	U	NM	F	U	NM	F	U	NM
70–79	4	5	22	13	12	12	4	9	8	1	15	1
80–89	9	11	14	27	10	16	11	8	10	3	6	2
90–100	55	9	5	56	6	8	77	6	4	94	23	2
Total	68	25	41	96	31	36	92	23	22	98	44	5

Table 21
Overall result of completeness (mean)

Frequency of acceptance rate	50			100			150			200		
	F	U	NM	F	U	NM	F	U	NM	F	U	NM
70–79	3	11	14	9	1	0	12	4	4	12	7	0
80–89	7	12	13	14	3	2	19	4	5	19	6	0
90–100	32	24	27	73	95	97	68	50	50	68	38	0
Total	42	47	54	96	99	99	99	58	59	99	51	0

Table 22
Summary of results for accuracy and completeness

Accuracy	Fuzzy dist		Uniform dist		Non-monotonic	
	Var.	Mean	Var.	Mean	Var.	Mean
Conceptualization[a]	93	86	25	96	33	95
Transfer[b]	Better		Similar each other			
Modification[c]	40%	30%	20%	20%	30%	20%
Integration[d]	100	81	27	88	87	73
Decomposition[e]	37	0	28	0	30	2
Complexity	NP-complete					
Completeness	Fuzzy dist	Uniform dist	Non-monotonic			
Variance of variances	144.8	67.2	195.5			
Variance of means	589.5	429.7	1240.5			

[a] Because the figures are the frequency of tests that fallen into the range of 70–100% of acceptance rate, the larger figure means more accuracy of knowledge conceptualization.
[b] The first occurrence of rejection is measured when 24 for fuzzy distribution, 44 for the uniform distribution, and 43 for non-monotonic distribution are the frequency at the range of 1–10th rejection occurred.
[c] The figures in table are the expansion rate when the major movement of acceptance rate noticed. The larger expansion rate means the modification of knowledge will be deferred and better knowledge accuracy.
[d] The way to understand the result of integration is the same as computerization.
[e] The way to understand the result of decomposition is the same as computerization.

8.3. Non-monotonic distribution

8.3.1. Result

In the variance analysis, non-monotonic distribution shows the acceptance frequency of 41, 36, 22 and 5 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. In the mean analysis, non-monotonic distribution shows the acceptance frequency of 54, 99, 59 and 0 in the interval of acceptance rates of 70–100% when the range of distribution is at 50, 100, 150 and 200, respectively. Detailed results are shown in Tables 18 and 19.

8.3.2. Overall result

In Tables 20 and 21, we summarize the completeness of three distributions. Based upon Tables 20 and 21, we calculate variance of variances and variance of means (reported in Table 22). In terms of variance of variances, the uniform distribution shows the lowest value as 67.2, while fuzzy and non-monotonic distribution shows 144.8 and 195.5, respectively. In terms of variance of means, the uniform distribution also shows the lowest value as 429.7, while fuzzy and non-monotonic distributions show 589.5 and 1240.5, respectively. The uniform distribution shows much better result against fuzzy and non-monotonic distributions in all four stages consistently.

The summarized results for accuracy, complexity and completeness are shown in Table 22.

9. Conclusions and further research

We proved that knowledge representations of all of three types of logic are NP-complete. For accuracy, fuzzy logic seems to be better than predicate logic and non-monotonic logic. Between predicate logic and non-monotonic logic, the later shows slightly better performance on knowledge representation on accuracy. For completeness, predicate logic is better than fuzzy logic and non-monotonic logic. Between fuzzy logic and non-monotonic logic, the former shows slightly better performance on knowledge representation as reflected by completeness.

For further research, we will compare the accuracy, complexity and completeness with modal logic and its various extensions such as temporal logic, epistemic logic, dynamic logic, and action logic.

References

Alavi, M., & Leidner, D. (2001). Knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly*, *25*(1), 107–136.

Bacchus, F., Grove, A., Halpern, J. Y., & Koller, D. (1996). From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, *87*, 75–143.

Baldwin, J. F. (1981). Fuzzy logic and fuzzy reasoning. In E. H. Mamdani, & B. R. Gaines (Eds.), *Fuzzy reasoning and its applications*. New York: Academic Press.

Bhattacharyya, S., & Koehler, G. J. (1998). Learning by objectives for adaptive shop-floor scheduling. *Decision Sciences*, *29*(2), 347–375.

Bolloju, N. (1996). Formulation of qualitative models using fuzzy logic. *Decision Support Systems*, *17*, 275–298.

Bouchon, B. (1987). Linguistic variables in the knowledge base of an expert system. In J. Rose (Ed.), *Cybernetics and systems: present and future*. London: Thales.

Bouchon, B. (1988). *Stability of linguistic modifiers compatible with a fuzzy logic*. *International conference on information processing and management of uncertainty in knowledge-based systems, Urbino, Italy*, pp. 63–70.

Bouchon, B., & Yao, J. (1990). *Gradual change of linguistic category by means of modifiers*. *International conference on information processing and management of uncertainty in knowledge-based systems, Paris*, pp. 242–244.

Chang, C. L., & Lee, C. T. (1973). *Symbolic logic and mechanical theorem proving*. New York: Academic Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms*. Cambridge, MA: MIT Press.

Engelen, R. A. V. (1997). Approximating Bayesian belief networks by arc removal. *IEEE PAMI*, *19*, 916–920.

Engelfriet, J. (1998). Monotonicity and persistence in preferential logics. *Journal of Artificial Intelligence Research*, *8*, 1–12.

Engelfriet, J., & Treur, J. (2000). Specification of nonmonotonic reasoning. *Journal of Applied Non-Classical Logics*, *10*, 7–27.

Ginsberg, M. L. (1993). *Essentials of artificial intelligence*. San Mateo, CA: Morgan Kaufmann.

Grabowski, M., & Sanborn, S. (1992). Knowledge representation and reasoning in a real-time operational control system: the shipboard piloting expert system (SPES). *Decision Sciences*, *23*(6), 1277–1296.

Greiner, R., Darken, C., & Santos o, N. I. (2001). Efficient reasoning. *ACM Computing Surveys*, *33*, 1–30.

Helft, N. (1989). *Induction as nonmonotonic inference*. *First international conference on knowledge representation and reasoning, Toronto, Canada*, pp. 149–156.

Lenard, M. J., Alam, P., & Booth, D. (2000). An analysis of fuzzy clustering and a hybrid model for the auditor's going concern assessment. *Decision Sciences*, *31*(4), 861–884.

Levesque, H. J. (1984). Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, *23*, 155–212.

McCarthy, J. (1977). *Epistemological problems in artificial intelligence*. *International joint conference on artificial intelligence, Cambridge, MA*, pp. 1038–1044.

McCarthy, J. (1980). Circumscription: a form of non-monotonic reasoning. *Artificial Intelligence*, *13*, 27–39.

Mendelson, E. (2001). *Introduction to mathematical logic*. London: Chapman & Hall.

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Example-based generalization: a unifying view. *Machine Learning*, *1*, 47–80.

Moore, R. C. (1982). *The role of logic in knowledge representation and commonsense reasoning*, *AAAI*, 428–433.

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. New York: Dover.

Patterson, D. W. (1990). *Introduction to artificial intelligence and expert systems*. Englewood Cliffs, NJ: Prentice Hall.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.

Reichgelt, H. (1991). *Knowledge representation: an AI perspective*. Norwood, NJ: Intellect.

Ross, T. J., & Ross, T. (1995). *Fuzzy logic with engineering applications*. New York: McGraw-Hill.

Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: a modern approach*. Upper Saddle River, NJ: Prentice Hall.

Shafer, G., & Pearl, J. (1990). *Readings in uncertain reasoning*. Los Altos, CA: Morgan Kaufmann.

Smith, B. C. (1985). Prologue to reflection and semantics in a procedural language. In R. J. Brachman, & H. J. Levesque (Eds.), *Readings in knowledge representation, Los Altos, CA*.

Torsun, I. S. (1995). *Foundations of intelligent knowledge-based systems*. London: Academic Press.

Warren, L. H., & Nicholls, M. G. (1999). Dynamic benchmarks for operations evaluation using soft computing methods. *Decision Sciences*, *30*(1), 19–45.

Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, *44*, 257–303.

Wellman, M. P. (1994). *Abstraction in probabilistic reasoning*. Corvallis, OR: Tutorial, Summer Institute on Probability in AI.

Wong, M. L. (2001). A flexible knowledge discovery system using genetic programming and logic grammars. *Decision Support Systems*, *31*, 405–428.

Zadeh, L. A. (1975). The concept of linguistic variable and its application to approximate reasoning. *Information Sciences*, *8*, 303–357.

Zadeh, L. A. (1978). PRUF: a meaning representation language for natural languages. *International Journal of Man–Machine Studies*, *10*, 395–460.

Zadeh, L. A. (1979). A theory of approximate reasoning. In J. Hayes, D. Michie, & L. I. Mikulich (Eds.), *Machine intelligence*. New York: Halstead Press.

Zadeh, L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, *11*, 199–227.

Zadeh, L. A., & Kacprzyk, J. (Eds.), (1992). *Fuzzy logic for the management of uncertainty*. New York: Wiley.