On Efficient Representation of Expert Knowledge by Fuzzy Logic

Hung T. Nguyen[1] and Vladik Kreinovich[2]

[1] Department of Mathematical Sciences,
New Mexico State University, Las Cruces, NM 88003, USA,
hunguyen@nmsu.edu
[2] Computer Science, University of Texas at El Paso,
500 W. University, El Paso, TX 79968, USA,
vladik@cs.utep.edu

Abstract. A natural approach to designing an intelligent system is to incorporate expert knowledge into this system. One of the main approaches to translating this knowledge into computer-understandable terms is the approach of fuzzy logic. It has led to many successful applications, but in several aspects, the resulting computer representation is somewhat different from the original expert meaning. In this paper, we overview several such situations, and describe how to modify the fuzzy logic approach so that its results become closer to the original expert meaning.

1 Introduction

A natural approach to designing an intelligent system is to incorporate expert knowledge into this system.

One of the main approaches to translating this knowledge into computer-understandable terms is the approach of fuzzy logic [24, 52]. This approach has led to many successful applications, but in several aspects, the resulting computer representation is somewhat different from the original expert meaning. In this section, we overview several such situations, and describe how to modify the fuzzy logic approach so that its results become closer to the original expert meaning.

Some of these results first appeared in [17, 22, 26, 27, 47]

2 How to Make Fuzzy Arithmetic Closer to Common Sense: I

Intuitive property of commonsense arithmetic. To explain the problem that we try to solve in this section, let us start with a joke. A museum guide tells the visitors that a dinosaur that they are looking at is 14,000,005 years old. An impressed visitor asks how scientists can be so accurate in their predictions. "I don't know how they do it, – explains the guide – but 5 years ago, when I

started working here, I was told that this dinosaur is 14,000,000 years old, so now it must be 5 years older".

This is clearly a joke, because from the common sense viewpoint, a dinosaur which was approximately 14,000,000 years old 5 years ago is still 14,000,000 years old. In more precise terms, if we add 5 to a "fuzzy" number "approximately 14,000,000", we should get the answer "approximately 14,000,000".

Similarly, if a person weighs, say, approximately 100 kg, and he gains 1 kg, he still weighs approximately 100 kg. So, if we add 1 to a "fuzzy" number "approximately 100", we should get the answer "approximately 100".

In general, if a is much larger than b ($a \gg b$), and we add b to "approximately a", we should get "approximately a". It is therefore natural to expect formal systems which formalize commonsense reasoning to have this property.

Fuzzy arithmetic: a natural formalization of commonsense arithmetic. A natural way of dealing with approximately known values (such as "approximately a") is *fuzzy arithmetic*. In fuzzy arithmetic, each such value is represented by a membership function $\mu(x)$ describing, for each real number x, to what extent x matches the description (see, e.g., [24, 52]).

For example, if the value that we want to formalize is "approximately a" (for some given real number a), then the value $x = a$ matches the described property perfectly well ($\mu(a) = 1$), while the more distant the value x from a, the smaller the degree of matching. In other words, a natural way to represent a property "approximately a" is to have a membership function $\mu(x)$ which:

- attains its maximum value 1 for $x = a$,
- increase for $x < a$, and
- decreases for $x > a$.

In practical applications, researchers have used membership functions $\mu(x)$ of different shape to represent the property "approximately a": Gaussian, piecewise linear, etc.; all these shapes have a clear maximum at $x = a$.

Vice versa, if we have a membership function $\mu(x)$ which:

- has a clear maximum at some point $x = a$,
- is increasing for $x < a$, and
- is decreasing for $x > a$,

it is natural to interpret this function as describing a property "approximately a".

When several numbers A, B, etc., are described by membership functions, we can use the extension principle to describe the result of applying an arithmetic operation to these numbers. For example, if a number A is described by a membership function $\mu_A(x)$, and the number B is described by a membership function $\mu_B(x)$, then their sum $C = A + B$ is described by the following membership function:

$$\mu_C(x) = \max_{y,z:y+z=x} \min(\mu_A(y), \mu_B(z)). \tag{1}$$

We can also have a more general formula, if we use an arbitrary t-norm instead of the minimum.

Whether we use min or a more general t-norm, in the simple case when the number B is crisp ($B = b$), the resulting membership function is equal to $\mu_C(x) = \mu_A(x - b)$; in other words, it has the same shape as the membership function for A – but it is shifted by b.

Problem: traditional fuzzy arithmetic does not have the desired property. In many practical applications, the traditional fuzzy arithmetic works well. Unfortunately, the traditional fuzzy arithmetic does not satisfy the desired intuitive property.

Indeed, let A mean "approximately a" (e.g., "approximately 100"). Then, the corresponding membership function $\mu_A(x)$ has a maximum at $x = a$, is increasing for $x < a$ and decreasing for $x > a$. When we add, to A, a crisp number $B = b$ (e.g., 1), we get a shifted membership function which has a maximum at $x = a + b$, is increasing for $x < a + b$ and decreasing for $x > a + b$. In accordance with the above interpretation, we thus interpret the sum $A + B$ as "approximately $a + b$". Thus, the sum "≈ 100"+1 is equal not to ≈ 100 as we would intuitively expect, but to ≈ 101.

How can we modify fuzzy arithmetic to make sure that the desired property is satisfied, and the sum of "≈ 100" and 1 is equal to ≈ 100?

Main idea. When we only know a (crisp of fuzzy) interval of possible values of a certain quantity (or a more general *set* of possible values), it is desirable to characterize this interval by supplying the user with the "simplest" element from this interval, and by characterizing how far away from this value we can get. For example, if, for some unknown physical quantity x, measurements result in the interval $[1.95, 2.1]$ of possible values, then, most probably, the physicist will publish this result as $y \approx 2$. Similarly, a natural representation of the measurement result $x \in [3.141592, 3.141593]$ is $x \approx \pi$.

So, intuitively, if we know the membership functions for A and for B, we should:

- compute the membership function $\mu_C(x)$ for $C = A + B$;
- find the interval of possible values of C (e.g., as all the values for which $\mu_C(x) \geq d_0$ for some value d_0);
- pick the simplest value c on this interval, and then
- return "approximately c" as the result of adding A and B.

In particular, when A is "approximately 14,000,000" – meaning that the interval of possible values is probably [13,500,000; 14,500,000] – and B is a crisp value 5, then for $A + B$, the interval of possible values is [13,500,005; 14,500,005]. On this interval, 14,000,000 is probably still the simplest value, so we conclude that the sum of "approximately 14,000,000" and 5 is – as we expected – equal to "approximately 14,000,000".

Similarly, in this new definition, if we add 1 kg to a weight of approximately 100 kg, we still get approximately 100 kg as the result.

How to formalize this definition? In order to formalize the above definition, we must formalize what "simplest" means. Intuitively, the simpler the description of a real number, the simpler this number. Thus, to define relative complexity

of different real numbers, we fix some logical theory T in which we will describe real numbers.

We will consider languages in which the list of sorts $\mathcal{S}$ contains two symbols: "integer" and "real", and which contain standard arithmetic predicates and function symbols such as 0, 1, $+$, $-$, $\cdot$, $/$, $=$, $<$, $\leq$, both for integers and for reals. We will assume that this theory contains both the standard first order theory of integers (Peano arithmetic [4, 13, 56]) and a standard first order theory of real numbers [7, 12, 57, 60]. One of the possibilities is to consider, as the theory T, axiomatic set theory (e.g., ZF), together with explicit definitions of integers, real numbers, and standard operations and predicates in terms of set theory.

Once a theory T is fixed, we can define a *complexity* $D(x)$ of a real number x as the shortest length of a formula $F(y)$ in the language L which defines this particular number x, i.e., which is true for $y = x$ and false for $y \neq x$.

To clarify this definition, let us give examples of formulas which define different real numbers:

- A formula $(y \cdot y = 1 + 1) \,\&\, y \geq 0$ is true if and only if $y = \sqrt{2}$; thus, this formula defines the number $\sqrt{2}$.
- Similarly, a formula $\forall x\, (x \cdot y = x + x + x)$ defines a real number 3.
- If the language of the theory T contains the sine function sin, and if the corresponding theory contains the standard definition of the sine function, then the formula $\sin(y) = 0 \;\&\; 3 \leq y \leq 4$ defines a real number π.

Comment 1. This definition is similar to the so-called *Kolmogorov complexity* $C(x)$ (invented independently by Chaitin, Kolmogorov, and Solomonoff), which is defined as the smallest length of the program that *computes* x (for a current survey on Kolmogorov complexity, see, e.g., [32]). In our case, however, we do not care that much about how to compute: computing 3.141592 may be easier than computing π; we are more interested in how easy it is to *describe* x. Due to this difference, we cannot simply use the original Kolmogorov's definition: we have to modify it.

Comment 2. It is worth mentioning that not all real numbers are definable: indeed, there are only countably many formulas, so there can be no more than countably many definable real numbers, while the total cardinality of the set of all real numbers is known to be larger ($\aleph_1 > \aleph_0$).

This new definition solves the above problem, but – in full accordance with the saying "there is no free lunch" – it comes with drawbacks. We will see that these drawbacks do not mean that our solution is bad, they seem to be implied (surprisingly) by the very properties that we try to retain.

First drawback: addition is no longer always associative. This drawback is the easiest to describe and to explain. Both standard arithmetic and traditional fuzzy arithmetic are *associative*: if we add several numbers $A_1 + \ldots + A_n$, the resulting sum does not depend on the order in which we add them; in particular,

$$(\ldots((A_1 + A_2) + A_3) + \ldots) + A_n =$$
$$A_1 + (A_2 + (A_3 + (\ldots + A_n)\ldots)). \tag{2}$$

Let us show that for the newly defined addition, this formula is no longer always true.

Indeed, suppose now that we want to formalize the idea that, say "≈ 100" + 1 is equal to ≈ 100 (this is just an example, but any other example can be used to illustrate non-associativity). Let us take $n = 101$, "approximately 100" as A_1, and $A_2 = \ldots = A_n = 1$ (crisp numbers). In terms of the newly defined numbers A_i, the desired property takes the form $A_1 + A_2 = A_1$ (similarly, $A_1 + A_3 = A_1$, etc.). Thus, $A_1 + A_2 = A_1$, hence $(A_1 + A_2) + A_3 = A_1 + A_3 = A_1$, etc., and hence the left-hand side of the formula (2) is equal to "approximately 100":

$$(\ldots((A_1 + A_2) + A_3) + \ldots) + A_n = A_1.$$

On the other hand, since $A_2, \ldots, A_n$ are crisp numbers (equal to 1 each), their sum $A_2 + (A_3 + (\ldots + A_n)\ldots)$ is simply a crisp number $1 + \ldots + 1 = 100$. Thus, the right-hand side of the formula (2) is equal to

"approximately 100" + 100

which, intuitively, should be rather "approximately 200" than "approximately 100". Thus, the left-hand side of (2) is clearly different from its right-hand side. Hence, the newly defined addition is not associative.

Second drawback: addition is no longer always easily computable. Traditional fuzzy arithmetic – defined by the extension principle – provides an explicit formula for computing the sum $C = A + B$ of two fuzzy numbers A and B. So, we can still find the interval of possible values for C. Unfortunately, as we will now show, the next step – finding the simplest possible real number on this interval – is no longer easily computable.

Theorem 2.1 [27]**.** *No algorithm is possible that, given an interval with definable endpoints, would return the simplest real number from this interval.*

A similar result holds for computable real numbers. A similar result holds if we restrict ourselves to *computable* real numbers, i.e., real numbers that can be computed with an arbitrary accuracy (see, e.g., [5, 8, 9, 11]). To be more precise, a real number x is called computable if there exists an algorithm (program) that transforms an arbitrary integer k into a rational number x_k that is 2^{-k}–close to x. It is said that this algorithm *computes* the real number x.

Every computable real number is uniquely determined by the corresponding algorithm and is, therefore, definable.

Theorem 2.2 [27]**.** *No algorithm is possible that, given an interval with computable endpoints, returns the simplest computable real number from this interval.*

Conclusion. From the commonsense viewpoint, if 5 years ago, a dinosaur was approximately 14,000,000 years old, it is still approximately 14,000,000 years years old. Unfortunately, when we formalize the notion "approximately 14,000,000" in traditional fuzzy arithmetic, we do not get this property. In this section, we have described a natural modification of fuzzy arithmetic which does

have this property. This modification is closer to commonsense reasoning, but this closeness comes at a cost: addition is no longer always associative and no longer always easily computable.

3 How to Make Fuzzy Arithmetic Closer to Common Sense: II

Granularity Approach. People often need to make crude estimates of a quantity, e.g., estimating the size of a crowd or someone's salary. When people make these crude estimates, they usually feel reasonably comfortable choosing between alternatives which differ by a half order of magnitude (HOM). For example, a person can reasonably estimate whether the size of a crowd was closer to 100, or to 300, or to 1000. If we ask for an estimate on a more refined scale, e.g., 300 or 350, people will generally be unable to make it. If we ask for an estimate on a coarser scale, e.g., 100 or 1000, people may be able to answer, but they will feel their answer is uninformative.

A particularly striking case of the utility of HOMs is presented by coinage and currency. Most countries have, in addition to denominations for the powers of ten, one or two coins or bills between every two powers of ten. Thus, in the United States, in addition to coins or bills for \$.01, \$.10, \$1.00, \$10.00, and \$100.00, there are also coins or bills in common use for \$.05, \$.25, \$5.00, \$20,00, and \$50.00. These latter provide rough HOM measures for monetary amounts.

It is natural that people should categorize the sizes of physical objects in terms of how they must interact with them. When two objects are roughly of the same size, we manipulate them or navigate about them in roughly the same way. But when one object is about three times larger in linear dimension than another, it must be handled in a different manner. Thus, an orange can be held in one hand, whereas a basketball is more easily held with two, A carton is held in our arms rather than our hands, and carrying a table often requires a second person. For further arguments along these lines, see [21].

These observations lead naturally to the following question: If we are to have a rough logarithmic classification scheme for quantities, what is the *optimal* granularity for commonsense estimates?

There are three requirements we would like the classification scheme to have.

- The categories should be small enough that the types of our interactions with objects are predictable from their category; that HOMs accomplish this is argued above and in [21].
- The categories should be large enough that ordinary variation among objects in a class do not usually cross category boundaries and that aggregation operations have reasonably predictable results; we show that HOMs satisfy these requirements.

Thus we describe two different models for commonsense estimation and show that in both models the optimal granularity is in good accordance with observations about the utility of HOMs. We thus provide a theoretical explanation for the importance of half orders of magnitude in commonsense reasoning.

Main idea behind Gaussian model. We are interested in the situation where we estimate a quantity which can only take non-negative values. To estimate the values of this quantity, we select a sequence of positive numbers $\ldots < e_0 < e_1 < e_2 < \ldots$ (e.g., 1, 3, 10, etc.), and every actual value x of the estimated quantity is then estimated by one of these numbers. Each estimate is approximate: when the estimate is equal to e_i, the actual value x of the estimated quantity may differ from e_i; in other words, there may be an estimation error $\Delta x = e_i - x \neq 0$.

What is the probability distribution of this estimation error? This error is caused by many different factors. It is known that under certain reasonable conditions, an error caused by many different factors is distributed according to Gaussian (normal) distribution (see, e.g., [62]; this fact – called *central limit theorem* – is one of the reasons for the widespread use of Gaussian distribution in science and engineering applications). It is therefore reasonable to assume that Δx is normally distributed.

It is known that a normal distribution is uniquely determined by its two parameters: its average a and its standard deviation σ. Let us denote the average of the error Δx by Δe_i, and its standard deviation by σ_i. Thus, when the estimate is e_i, the actual value $x = e_i - \Delta x$ is distributed according to Gaussian distribution, with an average $e_i - \Delta e_i$ (which we will denote by $\widetilde{e}_i$), and the standard deviation σ_i.

For a Gaussian distribution with given a and σ, the probability density is everywhere positive, so theoretically, we can have values which are as far away from the average a as possible. In practice, however, the probabilities of large deviations from a are so small that the possibility of such deviations can be safely neglected. For example, it is known that the probability of having the value outside the "three sigma" interval $[a - 3\sigma, a + 3\sigma]$ is $\approx 0.1\%$ and therefore, in most engineering applications, it is assumed that values outside this interval are impossible.

There are some applications where we cannot make this assumption. For example, in designing computer chips, when we have millions of elements on the chip, allowing 0.1% of these elements to malfunction would mean that at any given time, thousands of elements malfunction and thus, the chip would malfunction as well. For such critical applications, we want the probability of deviation to be much smaller than 0.1%, e.g., $\leq 10^{-8}$. Such small probabilities (which practically exclude any possibility of an error) can be guaranteed if we use a "six sigma" interval $[a - 6\sigma, a + 6\sigma]$. For this interval, the probability for a normally distributed variable to be outside it is indeed $\approx 10^{-8}$.

Within this Gaussian model, what is the optimal granularity?

Optimal granularity: informal explanation. In accordance with the above idea, for each e_i, if the actual value x is within the "three sigma" range $I_i = [\widetilde{e}_i - 3\sigma_i, \widetilde{e}_i + 3\sigma_i]$, then it is reasonable to take e_i as the corresponding estimate.

We want a granulation which would cover all possible values, so each positive real number must be covered by one of these intervals. In other words, we want the union of all these intervals to coincide with the set of all positive real numbers.

We also want to makes sure that all values that we are covering are indeed non-negative, i.e., that for every i, even the extended "six sigma" interval

$$[\widetilde{e}_i - 3\sigma_i, \widetilde{e}_i + 3\sigma_i]$$

only contains non-negative values.

Finally, since one of the main purposes of granularity is to decrease the number of "labels" that we use to describe different quantities, we want to consider optimal (minimal) sets of intervals. Formally, we can interpret "minimal" in the sense that whichever finite subset we pick, we cannot enlarge their overall coverage by modifying one or several of these intervals. Let us formalize these ideas.

In the following definitions, we will use the fact that an arbitrary interval $[a^-, a^+]$ can be represented in the Gaussian-type form $[a - 3\sigma, a + 3\sigma]$: it is sufficient to take $a = (a^- + a^+)/2$ and $\sigma = (a^+ - a^-)/6$.

Definition 3.1.

- *We say that an interval $I = [a - 3\sigma, a + 3\sigma]$ is reliably non-negative if every real number from the interval $[a - 6\sigma, a + 6\sigma]$ is non-negative.*
- *A set $\{I_i\}$, $i = 1, 2, \ldots$, of reliably non-negative intervals I_i is called a granulation if every positive real number belongs to one of the intervals I_i.*
- *We say that a granulation can be improved if, for some finite set $\{i_1, \ldots, i_k\}$, we can replace intervals I_{i_j} with some other intervals I'_{i_j} for which*

$$\bigcup_{j=1}^{k} I_{i_j} \subset \bigcup_{j=1}^{k} I'_{i_j} \qquad \bigcup_{j=1}^{k} I_{i_j} \neq \bigcup_{j=1}^{k} I'_{i_j},$$

  and still get a granulation.
- *A granulation is called optimal if it cannot be improved.*

Theorem 3.1 [22]. *In an optimal granulation, $I_i = [a_i, a_{i+1}]$, where $a_{i+1} = 3a_i$.*

So, half-orders of magnitude are indeed optimal.

Uniform model: motivations. In the Gaussian model, we started with a 3σ bound, and we ended up with a sequence of granules $[a_i, a_{i+1}]$ in which the boundary points a_i form an arithmetic progression: $a_{i+1} = q \cdot a_i$ and $a_i = a_0 \cdot q^i$, with $q = 3$. We could start with a bound of 2.5σ, then we would have got a geometric progression with a different q. Which value of q is indeed optimal?

To find out, let us take into consideration the fact that a granulation is not just for *storing* values, it is also for *processing* these values. Of course, when we replace the actual value by the granule to which it belongs, we lose some information. The idea is to choose the q for which this loss is the smallest.

To estimate the loss, we will consider the simplest data processing operation possible: addition. If we know the exact values of two quantities A and B, then we can compute the exact value of their sum $A + B$. In the granulated case, we do not know the exact values of A and B, we only know the *granules* to

which A and B belong, and we want to find out to which of the granules the sum belongs. For example, in the above half-order granulation, we know that the first room has about 10 books, the second about 30, and we want to express the total number of books in the two rooms in similar terms.

The trouble with this problem is that the sum may belong to two different granules. Let us take an example in which we use granules $[1,3]$, $[3,9]$, $[9,27]$, etc. Let us assume that all we know about the first quantity A is that $A \in [1,3]$, and all we know about the second quantity B is that $B \in [3,9]$. In this case, the smallest possible values of $A+B$ is $1+3=4$, and the largest possible value of $A+B$ is $3+9=12$. In general, the sum $A+B$ can thus take any value from the interval $[4,12]$. So, it could happen that the sum is in the granule $[3,9]$, but it could also happen that the sum is in the granule $[9,27]$.

If we want the granulation to be useful, we must assign a certain granule to the sum $A+B$. Since in reality, the value $A+B$ may belong to two different granules, no matter which of the two granules we assign, there is always a probability that this assignment is erroneous. We would like to select q for which this error probability is the smallest possible.

In order to formulate this question in precise terms, we must describe the corresponding probabilities. A natural way to describe them is as follows: If all we know about A is that A belongs to a granule $\mathbf{a}_i = [a_i, a_{i+1}]$, then it is reasonable to consider all the values from this granule to be equally probable, i.e., to assume that we have a *uniform* distribution on the interval $\mathbf{a}_i = [a_i, a_{i+1}]$. Similarly, If all we know about B is that B belongs to a granule $\mathbf{a}_j = [a_j, a_{j+1}]$, then it is reasonable to consider all the values from this granule to be equally probable, i.e., to assume that we have a *uniform* distribution on the interval $\mathbf{a}_j = [a_j, a_{j+1}]$. Since we have no information about the possible dependence between A and B, it is natural to assume that A and B are independent random variables. We are now ready for the formal definitions.

Let $a_0 > 0$ and $q \geq 2$ be real numbers, and let $a_k \stackrel{\text{def}}{=} a_0 \cdot q^k$ and $\mathbf{a}_i \stackrel{\text{def}}{=} [a_i, a_{i+1}]$.

Definition 3.2. *For every three integers i, j, and k, we can define $P(\mathbf{a}_i + \mathbf{a}_j \in \mathbf{a}_k)$ as the probability that $A_i + A_j \in \mathbf{a}_k$, where A_i is uniformly distributed on the interval $\mathbf{a}_i$, A_j is uniformly distributed on the interval $\mathbf{a}_j$, and A_i and A_j are independent.*

If, as a result of adding $\mathbf{a}_i$ and $\mathbf{a}_j$, we select the granule $\mathbf{a}_k$, then the probability that this assignment is erroneous (i.e., that the actual value of $A_i + A_j$ is not in $\mathbf{a}_k$) is equal to $1 - P(\mathbf{a}_i + \mathbf{a}_j \in \mathbf{a}_k)$. For every i and j, we want to minimize this error, so we select the value k for which this error probability is the smallest:

Definition 3.3. *For every two integers i and j, we define the sum $\mathbf{a}_i + \mathbf{a}_j$ of granules $\mathbf{a}_i$ and $\mathbf{a}_j$ as a granule $\mathbf{a}_k$ for which the error probability $1 - P(\mathbf{a}_i + \mathbf{a}_j \in \mathbf{a}_k)$ is the smallest possible. The error probability E_{ij} related to this addition is then defined as this smallest probability, i.e., as $E_{ij} \stackrel{\text{def}}{=} \min\limits_k (1 - P(\mathbf{a}_i + \mathbf{a}_j \in \mathbf{a}_k))$.*

Theorem 3.2 [22]. *When* $q \geq \sqrt{2}+1 (\approx 2.41)$, *then*

$$\mathbf{a}_i + \mathbf{a}_i = \mathbf{a}_{i+1}, \text{ and } \mathbf{a}_i + \mathbf{a}_j = \mathbf{a}_{\max(i,j)} \text{ for } i \neq j.$$

When $2 \leq q < \sqrt{2}+1$, *then* $\mathbf{a}_i + \mathbf{a}_i = \mathbf{a}_{i+1}$,

$$\mathbf{a}_i + \mathbf{a}_{i+1} = \mathbf{a}_{i+1} + \mathbf{a}_i = \mathbf{a}_{i+2}, \text{ and}$$

$$\mathbf{a}_i + \mathbf{a}_j = \mathbf{a}_{\max(i,j)} \text{ for } |i-j| \geq 2.$$

It is worth mentioning that for every q, thus defined addition of granules is commutative but not associative. Indeed, for $q \geq \sqrt{2}+1$, we have:

$$(\mathbf{a}_0 + \mathbf{a}_0) + \mathbf{a}_1 = \mathbf{a}_1 + \mathbf{a}_1 = \mathbf{a}_2, \text{ while}$$

$$\mathbf{a}_0 + (\mathbf{a}_0 + \mathbf{a}_1) = \mathbf{a}_0 + \mathbf{a}_1 = \mathbf{a}_1 \neq \mathbf{a}_2.$$

For $q < \sqrt{2}+1$, we have:

$$(\mathbf{a}_0 + \mathbf{a}_0) + \mathbf{a}_2 = \mathbf{a}_1 + \mathbf{a}_2 = \mathbf{a}_3, \text{ while}$$

$$\mathbf{a}_0 + (\mathbf{a}_0 + \mathbf{a}_2) = \mathbf{a}_0 + \mathbf{a}_2 = \mathbf{a}_2 \neq \mathbf{a}_3.$$

Which q is the best? As a measure of quality of a given granulation, it is natural to take the *worst-case* error probability, i.e., the error probability corresponding to the worst-case pair (i,j) (i.e., to the pair with the largest E_{ij}):

Definition 3.4. *By an error probability of a granulation, we mean the value* $E(q) \stackrel{\text{def}}{=} \max\limits_{i,j} E_{ij}$. *The granulation with the smallest possible error probability is called optimal.*

Theorem 3.3 [22]. *The granulation is optimal when*

$$q^3 - 5q^2 + 4q + 1 = 0$$

(*i.e., when* $q \approx 3.9$).

Conclusion. When people make crude estimates, they feel comfortable choosing between alternatives which differ by a half-order of magnitude (e.g., were there 100, 300, or 1,000 people in the crowd), and less comfortable making a choice on a more detailed scale (like 100 or 110 or 120) or on a coarser scale (like 100 or 1,000). We have shown that for two natural models of choosing granularity in commonsense estimates, in the optimal granularity, the next estimate is 3-4 times larger than the previous one. Thus, we have explained the commonsense HOM granularity.

4 How to Make Fuzzy Logic Closer to Common Sense: I

In expert systems, we need estimates for the degree of certainty of $S_1 \,\&\, S_2$ and $S_1 \vee S_2$. In many areas (medicine, geophysics, military decision-making, etc.), top quality experts make good decisions, but they cannot handle all situations. It is therefore desirable to incorporate their knowledge into a decision-making computer system.

Experts describe their knowledge by statements $S_1, \ldots, S_n$ (e.g., by if-then rules). Experts are often not 100% sure about these statements S_i; this uncertainty is described by the *subjective probabilities* p_i (degrees of belief, etc.) which experts assign to their statements. The conclusion C of an expert system normally depends on several statements S_i. For example, if we can deduce C either from S_2 and S_3, or from S_4, then the validity of C is equivalent to the validity of a Boolean combination $(S_2 \,\&\, S_3) \vee S_4$. So, to estimate the reliability $p(C)$ of the conclusion, we must estimate the probability of Boolean combinations. In this section, we consider the simplest possible Boolean combinations are $S_1 \,\&\, S_2$ and $S_1 \vee S_2$.

In general, the probability $p(S_1 \,\&\, S_2)$ of a Boolean combination can take different values depending on whether S_1 and S_2 are independent or correlated. So, to get the precise estimates of probabilities of all possible conclusions, we must know not only the probabilities $p(S_i)$ of individual statements, but also the probabilities of all possible Boolean combinations. To get all such probabilities, it is sufficient to describe 2^n probabilities of the combinations $E_1^{\varepsilon_1} \,\&\, \ldots \,\&\, E_n^{\varepsilon_n}$, where $\varepsilon_i \in \{+, -\}$, E^+ means E, and E^- means $\neg E$. The only condition on these probabilities is that their sum should add up to 1, so we need to describe $2^n - 1$ different values. A typical knowledge base may contain hundreds of statements; in this case, the value $2^n - 1$ is astronomically large. We cannot ask experts about all 2^n such combinations, so in many cases, we must estimate $p(S_1 \,\&\, S_2)$ or $p(S_1 \vee S_2)$ based only on the values $p_1 = p(S_1)$ and $p_2 = p(S_2)$.

Interval estimates are possible, but sometimes, numerical estimates are needed. It is known that for given $p_1 = p(S_1)$ and $p_2 = p(S_2)$:

- possible values of $p(S_1 \,\&\, S_2)$ form an interval $\mathbf{p} = [p^-, p^+]$, where $p^- = \max(p_1 + p_2 - 1, 0)$ and $p^+ = \min(p_1, p_2)$; and
- possible values of $p(S_1 \vee S_2)$ form an interval $\mathbf{p} = [p^-, p^+]$, where $p^- = \max(p_1, p_2)$ and $p^+ = \min(p_1 + p_2, 1)$

(see, e.g., a survey [51] and references therein).

So, in principle, we can use such interval estimates and get an interval $\mathbf{p}(C)$ of possible values of $p(C)$. Sometimes, this idea leads to meaningful estimates, but often, it leads to a useless $\mathbf{p}(C) = [0, 1]$ (see, e.g., [51, 53]). In such situations, it is reasonable, instead of using the entire interval $\mathbf{p}$, to select a point within this interval as a reasonable estimate for $p(S_1 \,\&\, S_2)$ (or, correspondingly, for $p(S_1 \vee S_2)$).

Natural idea: selecting a midpoint as the desired estimate. Since the only information we have, say, about the unknown probability $p(S_1 \,\&\, S_2)$ is that

it belongs to the interval $[p^-, p^+]$, it is natural to select a *midpoint* of this interval as the desired estimate. In other words, if we know the probabilities p_1 and p_2 of the statements S_1 and S_2, then, as estimates for $p(S_1 \,\&\, S_2)$ and $p(S_1 \vee S_2)$, we can take the values $p_1 \,\&\, p_2$ and $p_1 \vee p_2$, where

$$p_1 \,\&\, p_2 \stackrel{\text{def}}{=} \frac{1}{2} \cdot \max(p_1 + p_2 - 1, 0) + \frac{1}{2} \cdot \min(p_1, p_2);$$

$$p_1 \vee p_2 \stackrel{\text{def}}{=} \frac{1}{2} \cdot \max(p_1, p_2) + \frac{1}{2} \cdot \min(p_1 + p_2, 1).$$

This midpoint selection is not only natural from a common sense viewpoint; it also has a deeper justification. Namely, in accordance of our above discussion, for $n = 2$ statements S_1 and S_2, to describe the probabilities of all possible Boolean combinations, we need to describe $2^2 = 4$ probabilities $x_1 = p(S_1 \,\&\, S_2)$, $x_2 = p(S_1 \,\&\, \neg S_2)$, $x_3 = p(\neg S_1 \,\&\, S_2)$, and $x_4 = p(\neg S_1 \,\&\, \neg S_2)$; these probabilities should add up to 1: $x_1 + x_2 + x_3 + x_4 = 1$. Thus, each probability distribution can be represented as a point $(x_1, \ldots, x_4)$ in a 3-D simplex

$$\mathcal{S} = \{(x_1, x_2, x_3, x_4) \,|\, x_i \geq 0 \,\&\, x_1 + \ldots + x_4 = 1\}.$$

We know the values of $p_1 = p(S_1) = x_1 + x_2$ and $p_2 = p(S_2) = x_1 + x_3$, and we are interested in the values of $p(S_1 \,\&\, S_2) = x_1$ and $p(S_1 \vee S_2) = x_1 + x_2 + x_3$. It is natural to assume that *a priori*, all probability distributions (i.e., all points in a simplex $\mathcal{S}$) are "equally possible", i.e., that there is a uniform distribution ("second-order probability") on this set of probability distributions. Then, as a natural estimate for the probability $p(S_1 \,\&\, S_2)$ of $S_1 \,\&\, S_2$, we can take the conditional mathematical expectation of this probability under the condition that the values $p(S_1) = p_1$ and $p(S_2) = p_2$:

$$E(p(S_1 \,\&\, S_2) \,|\, p(S_1) = p_1 \,\&\, p(S_2) = p_2) =$$

$$P(x_1 \,|\, x_1 + x_2 = p_1 \,\&\, x_1 + x_3 = p_2).$$

(This idea was proposed and described in [2, 16]; see also [6].)

From the geometric viewpoint, the two conditions $x_1 + x_2 = p_1$ and $x_1 + x_3 = p_2$ select a straight line segment within the simplex $\mathcal{S}$, a segment which can be parameterized by

$$x_1 \in [p^-, p^+] = [\max(p_1 + p_2 - 1, 0), \min(p_1, p_2)];$$

then, $x_2 = p_1 - x_1$, $x_3 = p_2 - x_1$, and $x_4 = 1 - (x_1 + x_2 + x_3)$. Since we start with a uniform distribution on $\mathcal{S}$, the conditional probability distribution on this segment is uniform, i.e., x_1 is uniformly distributed on the interval $[p^-, p^+]$. Thus, the conditional mathematical expectation of x_1 with respect to this distribution is equal to $(p^- + p^+)/2$, i.e., to the midpoint of this interval. Similarly, for an "or" operation, we can conclude that

$$E(p(S_1 \vee S_2) \,|\, p(S_1) = p_1 \,\&\, p(S_2) = p_2) =$$

$$\frac{1}{2} \cdot \max(p_1, p_2) + \frac{1}{2} \cdot \min(p_1 + p_2, 1).$$

Problem: midpoint operations are not associative. Any "and" operation $p_1 \,\&\, p_2$ enables us to produce an estimate for $P(S_1 \,\&\, S_2)$ provided that we know estimates p_1 for $p(S_1)$ and p_2 for $p(S_2)$. If we are interested in estimating the degree of belief in a conjunction of three statements $S_1 \,\&\, S_2 \,\&\, S_3$, then we can use the same operation twice:

- first, we apply the "and" operation to p_1 and p_2 and get an estimate $p_1 \,\&\, p_2$ for the probability of $S_1 \,\&\, S_2$;
- then, we apply the "and" operation to this estimate $p_1 \,\&\, p_2$ and p_3, and get an estimate $(p_1 \,\&\, p_2) \,\&\, p_3$ for the probability of $(S_1 \,\&\, S_2) \,\&\, S_3$.

Alternatively, we can get start by combining S_2 and S_3, and get an estimate $p_1 \,\&\, (p_2 \,\&\, p_3)$ for the same probability $p(S_1 \,\&\, S_2 \,\&\, S_3)$. Intuitively, we would expect these two estimates to coincide: $(p_1 \,\&\, p_2) \,\&\, p_3 = p_1 \,\&\, (p_2 \,\&\, p_3)$, i.e., in algebraic terms, we expect the operation $\&$ to be associative. Unfortunately, midpoint operations are *not* associative [6]: e.g., $(0.4 \,\&\, 0.6) \,\&\, 0.8 = 0.2 \,\&\, 0.8 = 0.1$, while $0.4 \,\&\, (0.6 \,\&\, 0.8) = 0.4 \,\&\, 0.5 = 0.2 \neq 0.1$.

By itself, a small non-associativity may not be so bad:

- associativity comes from the requirement that our reasoning be rational, while
- it is well known that our actual handling of uncertainty is not exactly following rationality requirements; see, e.g., [59].

So, it is desirable to find out how non-associative can these operations be.

How non-associative are natural (midpoint) operations? Main results and their psychological interpretation

We know that the midpoint operations are non-associative, i.e., that sometimes, $(a \,\&\, b) \,\&\, c \neq a \,\&\, (b \,\&\, c)$. We want to know how big can the difference $(a \,\&\, b) \,\&\, c - a \,\&\, (b \,\&\, c)$ can be.

Theorem 4.1 [17]**.** $\max\limits_{a,b,c} |(a \,\&\, b) \,\&\, c - a \,\&\, (b \,\&\, c)| = 1/9.$

Theorem 4.2 [17]**.** $\max\limits_{a,b,c} |(a \vee b) \vee c - a \vee (b \vee c)| = 1/9.$

Human experts do not use all the numbers from the interval $[0, 1]$ to describe their possible degrees of belief; they use a few words like "very probable", "mildly probable", etc. Each of words is a "granule" covering the entire sub-interval of values. Since the largest possible non-associativity degree $|(a \,\&\, b) \,\&\, c - a \,\&\, (b \,\&\, c)|$ is equal to $1/9$, this non-associativity is negligible if the corresponding realistic "granular" degree of belief have granules of width $\geq 1/9$. One can fit no more than 9 granules of such width in the interval $[0, 1]$. This may explain why humans are most comfortable with ≤ 9 items to choose from – the famous "7 plus minus 2" law; see, e.g., [37, 38].

This general psychological law has also been confirmed in our specific area of formalizing expert knowledge: namely, in [15], it was shown that this law

explains why in intelligent control, experts normally use ≤ 9 different degrees (such as "small", "medium", etc.) to describe the value of each characteristic.

Auxiliary results: alternatives to midpoint. Instead of selecting a midpoint, we can make a more general selection of a value in the interval $\mathbf{p}$.

By a *choice function*, we mean a function s that maps every interval $\mathbf{u} = [u^-, u^+]$ into a point $s(\mathbf{u}) \in \mathbf{u}$ so that for every c and $\lambda > 0$:

- $s([u^- + c, u^+ + c]) = s([u^-, u^+]) + c$ (*shift-invariance*);
- $s([\lambda \cdot u^-, \lambda \cdot u^+]) = \lambda \cdot s([u^-, u^+])$ (*unit-invariance*).

Proposition [44]**.** *Every choice function has the form* $s([u^-, u^+]) = \alpha \cdot u^- + (1-\alpha) \cdot u^+$ *for some* $\alpha \in [0, 1]$.

The combination $p = \alpha \cdot p^- + (1-\alpha) \cdot p^+$ (first proposed by Hurwicz [23]) has been successfully used in areas ranging from submarine detection to petroleum engineering [44]; in [63], this approach is applied to second-order probabilities.)

With this approach, we get the following formulas which generalize the above definitions:

$$p_1 \,\&\, p_2 \stackrel{\text{def}}{=} \alpha \cdot \max(p_1 + p_2 - 1, 0) + (1-\alpha) \cdot \min(p_1, p_2);$$

$$p_1 \vee p_2 \stackrel{\text{def}}{=} \alpha \cdot \max(p_1, p_2) + (1-\alpha) \cdot \min(p_1 + p_2, 1).$$

Theorem 4.3 [17]**.**

$$\max_{a,b,c} |(a \,\&\, b) \,\&\, c - a \,\&\, (b \,\&\, c)| = \frac{\alpha \cdot (1-\alpha)}{2 + \alpha \cdot (1-\alpha)}.$$

$$\max_{a,b,c} |(a \vee b) \vee c - a \vee (b \vee c)| = \frac{\alpha \cdot (1-\alpha)}{2 + \alpha \cdot (1-\alpha)}.$$

Comment. This non-associativity degree is the smallest ($= 0$) when $\alpha = 0$ or $\alpha = 1$, and the largest ($= 1/9$) for midpoint operations ($\alpha = 0.5$).

5 How to Make Fuzzy Logic Closer to Common Sense: II

Second order descriptions: the main idea. Experts are often not 100% certain in the statements they make; therefore, in the design of knowledge-based systems, it is desirable to take this uncertainty into consideration. Usually, this uncertainty is described by a number from the interval $[0, 1]$; this number is called *subjective probability*, *degree of certainty*, etc. (see, e.g., [58]).

One of the main problems with this approach is that we must use *exact* numbers from the interval $[0, 1]$ to represent experts' degrees of certainty; an expert may be able to tell whether his degree of certainty is closer to 0.9 or

to 0.5, but it is hardly possible that an expert would be able to meaningfully distinguish between degrees of certainty, say, 0.7 and 0.701. If you ask the expert whether his degree of certainty about a certain statement A can be described by a certain number d (e.g., $d = 0.701$), the expert will, sometimes, not be able to give a definite answer, she will be uncertain about it. This uncertainty can be, in its turn, described by a number from the interval $[0, 1]$. It is, therefore, natural to represent our degree of certainty in a statement A not by a *single* (crisp) number $d(A) \in [0, 1]$ (as in the $[0, 1]$-based description), but rather by a *function* $\mu_{\mathbf{d}(A)}$ which assigns, to each possible real number $d \in [0, 1]$, a *degree* $\mu_{\mathbf{d}(A)}(d)$ with which this number d can be the (desired) degree of certainty of A. This is called a *second-order* description of uncertainty.

Third and higher order descriptions. In second-order description, to describe a degree with which a given number $d \in [0, 1]$ can be a degree of certainty of a statement A, we use a *real number* $\mu_{\mathbf{d}(A)}(d)$. As we have already mentioned, it is difficult to describe our degree of certainty by a single number. Therefore, to make this description even more realistic, we can represent each degree of certainty $d(P(x))$ not by a (more traditional) $[0, 1]$-based description, but by a *second order* description. As a result, we get the *third order* description.

Similarly, to make our description even more realistic, we can use the third order descriptions to describe degrees of certainty; then, we get *fourth order* uncertainty, etc.

Third order descriptions are not used: why? Theoretically, we can define third, fourth order, etc., descriptions, but in practical applications, only second order descriptions were used so far (see, e.g., [36, 39, 43, 51]). Based on this empirical fact, it is natural to conclude that third and higher order descriptions are not really necessary. We will show that this conclusion can be theoretically justified.

First step in describing uncertainty: set of uncertainty-describing words. Let us first describe the problem formally. An expert uses words from a natural language to describe his degrees of certainty. In every language, there are only finitely many words, so we have a finite set of words that needs to be interpreted. We will denote this set of words by W.

Second step: a fuzzy property described by a word-valued "membership function". If we have any property P on a universe of discourse U, an expert can describe, for each element $x \in U$, his degree of certainty $d(x) \in W$ that the element x has the property P.

Traditional fuzzy logic as a first approximation: numbers assigned to words describing uncertainty. Our ultimate goal is to provide a computer representation for each word $w \in W$. In the traditional $[0, 1]$-based description, this computer representation assigns, to every word, a *real number* from the interval $[0, 1]$; in general, we may have some other computer representations (examples will be given later). Let us denote the set of all possible computer representations by S.

In the first approximation, i.e., in the first order description, we represent each word $w \in W$, which describes a degree of uncertainty, by an element $s \in S$

(e.g., by a real number from the interval $[0,1]$). In this section, we will denote this first-approximation computer representation of a word w by $s = \|w\|$.

If the set S is too small, then it may not contain enough elements to distinguish between different expert's degree of belief: this was exactly the problem with classical $\{0,1\}$-based description, in which we only have two possible computer representations – "true" and "false" – that are not enough to adequately describe the different degrees of certainty. We will therefore assume that the set S is rich enough to represent different degrees of certainty.

In particular, the set $[0,1]$ contains infinitely many points, so it should be sufficient; even if we only consider computer-representable real numbers, there are still much more of them (millions and billions) than words in a language (which is usually in hundreds of thousands at most), so we can safely make this "richness" assumption. In mathematical terms, it means that two different degrees of belief are represented by different computer terms, i.e., that if $w_1 \neq w_2$, then $\|w_1\| \neq \|w_2\|$.

First approximation is not absolutely adequate. The problem with the first-order representation is that the relation between words $w \in W$ and computer representation $s \in S$ is, in reality, also imprecise. Typically, when we have a word $w \in W$, we cannot pick a single corresponding representative $s \in S$; instead, we may have *several* possible representatives, with different degrees of adequacy.

Actual description of expert uncertainty: word-valued degree to which a word describes uncertainty. In other words, instead of a *single* value $s = \|w\|$ assigned to a word w, we have *several* values $s \in S$, each with its own degree of adequacy; this degree of adequacy can also be described by an expert, who uses an appropriate word $w \in W$ from the natural language.

In other words, for every word $w \in W$ and for ever representation $s \in S$, we have a degree $w' \in W$ describing to what extent s is adequate in representing w. Let us represent this degree of adequacy by $a(w,s)$; the symbol a represents a function $a : W \times S \to W$, i.e., a function that maps every pair (w,s) into a new word $a(w,s)$.

Second-order description of uncertainty as a second approximation to actual uncertainty. So, the meaning of a word $w \in W$ is represented by a *function* a which assigns, to every element $s \in S$, a degree of adequacy $a(w,s) \in W$. We want to represent this degree of adequacy in a computer; therefore, instead of using the word $a(w,s)$ itself, we will use the computer representation $\|a(w,s)\|$ of this word. Hence, we get a *second-order* representation, in which a degree of certainty corresponding to a word $w \in W$ is represented not by a *single* element $\|w\| \in S$, but by a *function* $\mu_w : S \to S$, a function which is defined as $\mu_w(s) = \|a(w,s)\|$.

Second-order description is not 100% adequate either; third-, fourth-order descriptions, etc. The second-order representation is also not absolutely adequate, because, to represent the degree $a(w,s)$, we used a single number $\|a(w,s)\|$. To get a more adequate representation, instead of this single value, we can use, for each element $s' \in S$, a degree of adequacy with which the

element s' represents the word $a(w,s)$. This degree of adequacy is also a word $a(a(w,s),s')$, so we can represent it by an appropriate element $\|a(a(w,s),s')\|$. Thus, we get a *third-order* representation, in which to every element s, we assign a second-order representation. To get an even more adequate representation, we can use fourth- and higher order representations.

Let us express this scheme formally.

Definition 5.1.

- *Let W be a finite set; element of this set will be called words.*
- *Let U be a set called a universe of discourse. By a fuzzy property P, we mean a mapping which maps each element $x \in U$ into a word $P(x) \in W$; we say that this word described the degree of certainty that x satisfies the property P.*
- *By a first-approximation uncertainty representation, we mean a pair $\langle S, \|.\|\rangle$, where:*
  - *S is a set; elements of this set will be called computer representations; and*
  - *$\|.\|$ is a function from W to S; we say that an element $\|w\| \in S$ represents the word w.*
- *We say that an uncertainty representation is sufficiently rich if for every two words $w_1, w_2 \in W$, $w_1 \neq w_2$ implies $\|w_1\| \neq \|w_2\|$.*

Definition 5.2. *Let W be a set of words, and let S be a set of computer representations. By an adequacy function, we mean a function $a : W \times S \to W$; for each word $w \in W$, and for each representation $s \in S$, we say that $a(w,s)$ describes the degree to which the element s adequately describes the word w.*

Definition 5.3. *Let U be a universe of discourse, and let S be a set of computer representations. For each $n = 1, 2, \ldots$, we define the notions of n-th order degree of certainty and of a n-th order fuzzy set, by the following induction over n:*

- *By a first-order degree of certainty, we mean an element $s \in S$ (i.e., the set S_1 of all first-order degrees of certainty is exactly S).*
- *For every n, by a n-th order fuzzy set, we mean a function $\mu : U \to S_n$ from the universe of discourse U to the set S_n of all n-th order degrees of certainty.*
- *For every $n > 1$, by a n-th order degree of certainty, we mean a function s_n which maps every value $s \in S$ into an $(n-1)$-th order degree of certainty (i.e., a function $s_n : S \to S_{n-1}$).*

Definition 5.4. *Let W be a set of words, let $\langle S, \|.\|\rangle$ be an uncertainty representation, and let a be an adequacy function. For every $n > 1$, and for every word $w \in W$, we define the n-th order degree of uncertainty $\|w\|_{a,n} \in S_n$ corresponding to the word w as follows:*

- *As a first order degree of uncertainty $\|w\|_{a,1}$ corresponding to the word w, we simply take $\|w\|_{a,1} = \|w\|$.*

– *If we have already defined degrees of orders* $1, \ldots, n-1$*, then, as an* n*-th order degree of uncertainty* $\|w\|_{a,n} \in S_n$ *corresponding to the word* w*, we take a function* s_n *which maps every value* $s \in S$ *into a* $(n-1)$*-th order degree* $\|a(w,s)\|_{a,n-1}$.

Definition 5.5. *Let* W *be a set of words, let* $\langle S, \|.\| \rangle$ *be an uncertainty representation, let* a *be an adequacy function, and let* P *be a fuzzy property on a universe of discourse* P*. Then, by a* n*-th order fuzzy set (or a* n*-th order membership function)* $\mu^{(n)}_{P,a}(x)$ *corresponding to* P*, we mean a function which maps every value* $x \in U$ *into an* n*-th order degree of certainty* $\|P(x)\|_{a,n}$ *which corresponds to the word* $P(x) \in W$.

We will prove that for properties which are *non-degenerate* in some reasonable sense, it is sufficient to know the *first* and *second* order membership functions, and then the others can be uniquely reconstructed. Moreover, if we know the membership functions of first two orders for a non-degenerate *class* of fuzzy properties, then we will be able to reconstruct the higher order membership functions for *all* fuzzy properties from this class.

Definition 5.6.

– *We say that a fuzzy property* P *on a universe of discourse* U *is non-degenerate if for every* $w \in W$*, there exists an element* $x \in U$ *for which* $P(x) = w$.
– *We say that a class* $\mathcal{P}$ *of fuzzy properties* P *on a universe of discourse* U *is non-degenerate if for every* $w \in W$*, there exists a property* $P \in \mathcal{P}$ *and an element* $x \in U$ *for which* $P(x) = w$.

Comment. For example, if $W \neq \{0,1\}$, then every crisp property, i.e., every property for which $P(x) \in \{0,1\}$ for all x, is *not* non-degenerate (i.e., degenerate).

Theorem 5.1 [26, 46]**.** *Let* W *be a set of words, let* $\langle S, \|.\| \rangle$ *be a sufficiently rich uncertainty representation, let* U *be a universe of discourse. Let* P *and* P' *be fuzzy properties, so that* P *is non-degenerate, and let* a *and* a' *be adequacy functions. Then, from* $\mu^{(1)}_{P,a} = \mu^{(1)}_{P',a'}$ *and* $\mu^{(2)}_{P,a} = \mu^{(2)}_{P',a'}$*, we can conclude that* $\mu^{(n)}_{P,a} = \mu^{(n)}_{P',a'}$ *for all* n.

Comments.

– In other words, under reasonable assumptions, for each property, the information contained in the first and second order fuzzy sets is sufficient to reconstruct all higher order fuzzy sets as well; therefore, in a computer representation, it is sufficient to keep only first and second order fuzzy sets.
– This result is somewhat similar to the well-known result that a Gaussian distribution can be uniquely determined by its moments of first and second orders, and all higher order moments can be uniquely reconstructed from the moments of the first two orders.
– It is possible to show that the non-degeneracy condition is needed, because if a property P is not non-degenerate, then there exist adequacy functions

$a \neq a'$ for which $\mu^{(1)}_{P,a} = \mu^{(1)}_{P,a'}$ and $\mu^{(2)}_{P,a} = \mu^{(2)}_{P,a'}$, but $\mu^{(3)}_{P,a} \neq \mu^{(3)}_{P,a'}$ already for $n = 3$.

Theorem 5.2 [26]. *Let W be a set of words, let $\langle S, \|.\| \rangle$ be a sufficiently rich uncertainty representation, let U be a universe of discourse. Let $\mathcal{P}$ and $\mathcal{P}'$ be classes of fuzzy properties, so that the class $\mathcal{P}$ is non-degenerate, and let $\varphi : \mathcal{P} \to \mathcal{P}'$ be a 1-1-transformation, and let a and a' be adequacy functions. Then, if for every $P \in \mathcal{P}$, we have $\mu^{(1)}_{P,a} = \mu^{(1)}_{\varphi(P),a'}$ and $\mu^{(2)}_{P,a} = \mu^{(2)}_{\varphi(P),a'}$, we can conclude that $\mu^{(n)}_{P,a} = \mu^{(n)}_{\varphi(P),a'}$ for all n.*

Comment. So, even if we do not know the adequacy function (and we do not know the corresponding fuzzy properties $P \in \mathcal{P}$), we can still uniquely reconstruct fuzzy sets of all orders which correspond to all fuzzy properties P.

6 How to Make Fuzzy Logic Closer to Common Sense: III

Why only unary and binary operations? Traditionally, in logic, only unary and binary operations are used as basic ones – e.g., "not", "and", "or" – while the only ternary (and higher order) operations are the operations which come from a combination of unary and binary ones.

A natural question is: are such combinations sufficient? I.e., to be more precise, *can an arbitrary logical operation be represented as a combination of unary and binary ones?*

For the classical logic, with the binary set of truth values $V = \{0, 1\}$ (=$\{\text{false}, \text{true}\}$), the positive answer to this question is well known. Indeed, it is known that an arbitrary logical operation $f : V^n \to V$ can be represented, e.g., in DNF form and thus, it can indeed be represented as a combination of unary ("not") and binary ("and" and "or") operations.

We are interested in explaining why unary and binary logical operations are the only basic ones. If we assume that the logic of human reasoning is the two-valued (classical) logic, then the possibility to transform every logical function into a DNF form explains this empirical fact.

However, classical logic is not a perfect description of human reasoning: for example, it does not take into consideration *fuzziness* and uncertainty of human reasoning. This uncertainty is taken into consideration in *fuzzy logic* [24, 52, 65]. In the traditional fuzzy logic, the set of truth values is the entire interval $V = [0, 1]$. This interval has a natural notion of continuity, so it is natural to restrict ourselves to *continuous* unary and binary operations.

With this restriction in place, a natural question is: *can an arbitrary continuous function $f : [0, 1]^n \to [0, 1]$ be represented as a composition of continuous unary and binary operations?* The positive answer to this question was obtained in our papers [45, 49].

In $[0, 1]$-based fuzzy logic, an arbitrary logical operation can be represented as a composition of unary and binary ones. However, the $[0, 1]$-based fuzzy logic is, by itself, only an approximation to the actual human reasoning about uncertainty.

Indeed, how can we describe the expert's degree of confidence $d(S)$ in a certain statement S? A natural way to determine this degree is, e.g., to ask an expert to estimate his degree of confidence on a scale from 0 to 10. If he selects 8, then we take $d(S) = 8/10$.

To get a more accurate result, we can then ask the same expert to estimate his degree of confidence on a finer scale, e.g., from 0 to 100, etc. For example, if an expert selects 81, we will take $d(S) = 81/100 = 0.81$. If we want an even more accurate estimate, we can ask the expert to estimate his degree of confidence on an even finer scale, etc.

The problem with this approach is that experts cannot describe their degrees of too fine scales. For example, an expert can point to 8 on a scale from 0 to 10, but this same expert will hardly be able to pinpoint a value on a scale from 0 to 100.

So, to attain a more adequate description of human reasoning, we must modify the traditional $[0, 1]$-based fuzzy logic. Two types of modifications have been proposed.

One possibility is to take the finest (finite) scale which an expert can still use, and take the values on this scale as the desired degrees of confidence. This approach leads to a *finite-valued* fuzzy logic, in which the set of truth values V is finite.

This approach has been successfully used in practice; see, e.g., [1, 15, 46, 55]. It is therefore desirable to check whether in a finite logic, every operation can be represented as a composition of unary and binary operations.

The problem with finite-valued logics is that the set V of resulting truth values depends on which scale we use.

Instead of fixing a finite set, we can describe the expert's degree of confidence by an *interval* from $[0, 1]$. For example, if an expert estimates his degree of confidence by a value 8 on a 0 to 10 scale, then the only thing that we know about the expert's degree of confidence is that it is closer to 0.8 (8/10) than to 0.7 or to 0.9, i.e., that it belongs to the interval $[0.75, 0.85]$.

So, a natural way of describing degrees of confidence more adequately is to use *intervals* $\mathbf{a} = [a^-, a^+]$ instead of real numbers. In this representation, real numbers can be viewed as particular – degenerate – cases of intervals $[a, a]$. The idea of using intervals have been originally proposed by Zadeh himself and further developed by Bandler and Kohout [3], Türkşen [61], and others; for a recent survey, see, e.g., [51].

In interval-valued fuzzy approach, to describe each degree of confidence, we must describe two real numbers: the lower endpoint and the upper endpoint of the corresponding "confidence interval".

We can go one step further and take into consideration that the endpoints of the corresponding interval are also not precisely known. Thus, each of these endpoints is, in actuality, an interval itself. So, to describe a degree of confidence, we now need *four* real numbers: two to describe the lower endpoint, and two to describe the upper one.

In general, we get a *multi-D* fuzzy logic. A natural question is: can every (continuous) operation on a multi-D fuzzy logic be represented as a composition of (continuous) unary and and binary operations?

Uncertainty of expert estimates is only one reason why we may want to go beyond the traditional $[0,1]$-valued logic; there are also other reasons:

- A 1-D value is a reasonable way of describing the uncertainty of a single expert. However, the confidence strongly depends on the *consensus* between different experts. We may want to use additional dimensions to describe how many expert share the original expert's opinion, and to what degree; see, e.g., [30, 50].
- Different experts may strongly disagree. To describe the degree of this disagreement, we also need additional numerical characteristics, which make the resulting logic multi-D; see, e.g., [48].

In all these cases, we need a multi-D logic to adequately describe expert's degree of confidence.

In this section, we show that both for finite-valued logics and for multi-D logics, every logical operation can be represented as a composition of unary and binary operations. Thus, we give a general explanation for the above empirical fact.

Theorem 6.1 [27]**.** *For every finite set V, and for every positive integer n, every n-ary operation $f : V^n \to V$ can be represented as a composition of unary and binary operations.*

Theorem 6.2 [27]**.** *For every multi-D set of truth values V, and for every positive integer n, every continuous n-ary operation $f : V^n \to V$ can be represented as a composition of continuous unary and binary operations.*

This result is based on the following known result:

Theorem (Kolmogorov)**.** *Every continuous function of three or more variables can be represented as a composition of continuous functions of one or two variables.*

This result was proven by A. Kolmogorov [25] as a solution to the conjecture of Hilbert, formulated as the thirteenth problem [20]: one of 22 problems that Hilbert has proposed in 1900 as a challenge to the 20 century mathematics.

This problem can be traced to the Babylonians, who found (see, e.g., [10]) that the solutions x of quadratic equations $ax^2 + bx + c = 0$ (viewed as function of three variables a, b, and c) can be represented as superpositions of functions of one and two variables, namely, arithmetic operations and square roots. Much later, similar results were obtained for functions of five variables a, b, c, d, e, that represent the solution of quartic equations $ax^4 + bx^3 + cx^2 + dx + e = 0$. But then, Galois proved in 1830 that for higher order equations, we cannot have such a representation. This negative result has caused Hilbert to conjecture that not all functions of several variables can be represented by functions of two or fewer variables. Hilbert's conjecture was refuted by Kolmogorov (see, e.g., [33], Chapter 11) and his student V. Arnold.

It is worth mentioning that Kolmogorov's result is not only of theoretical value: it was used to speed up actual computations (see, e.g., [14, 18, 28, 29, 40, 41]).

It turns out that one can generalize Kolmogorov's theorem and prove that a similar representation holds for multi-D logics as well.

Let m be a positive integer, and let V be a closure of a simply connected bounded open set in R^m (e.g., of a convex set). Such a set V will be called a *multi-D set of truth values*. For example, for interval-valued fuzzy sets,

$$V = \{(a, b) \mid 0 \leq a \leq b \leq 1\}.$$

Conclusion. Traditionally, in logic, only unary and binary operations are used as basic ones. In traditional (2-valued) logic, the use of only unary and binary operations is justified by the known possibility to represent an arbitrary n-ary logical operation as a composition of unary and binary ones. A similar representation result is true for the $[0, 1]$-based fuzzy logic. However, the $[0, 1]$-based fuzzy logic is only an approximation to the actual human reasoning about uncertainty. A more accurate description of human reasoning requires that we take into consideration the uncertainty with which we know the values from the interval $[0, 1]$. This additional uncertainty leads to two modifications of the $[0, 1]$-based fuzzy logic: finite-valued logic and multi-D logic.

We show that for both modifications, an arbitrary n-ary logical operation can be represented as a composition of unary and binary ones. Thus, the above justification for using only unary and binary logical operation as basic ones is still valid if we take interval uncertainty into consideration.

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209, by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-00-1-0365, and by Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

References

1. Agustíi, J., et al.: Structured local fuzzy logics in MILORD", In: Zadeh, L., Kacrpzyk, J., eds.: Fuzzy Logic for the Management of Uncertainty, Wiley, N.Y., 1992, 523–551.
2. Bamber, D.: Entailment with near surety of scaled assertions of high conditional probability, Journal of Philosophical Logic, 2001 (to appear).
3. Bandler, W., Kohout, L.J.: Unified theory of multi-valued logical operations in the light of the checklist paradigm, Proc. of IEEE Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, Oct. 1984.

4. Barwise, J., ed.: Handbook of Mathematical Logic, North-Holland, Amsterdam, 1977.
5. Beeson, M.J.: Foundations of computable mathematics, Springer-Verlag, N.Y., 1985.
6. Bennett, A.D.C., Paris, J.B., Vencovská, A.: A new criterion for comparing fuzzy logics for uncertain reasoning, Journal of Logic, Language, and Information **6** (2000) 31–63.
7. Ben-Or, N., Kozen, D., and Reif, J. The complexity of elementary algebra and geometry, Journal of Computer and System Sciences **32** (1986) 251–264.
8. Bishop, E.: Foundations of Computable Analysis, McGraw-Hill, 1967.
9. Bishop, E., Bridges, D.S.: Computable Analysis, Springer, N.Y., 1985.
10. Boyer, C.B., Merzbach, U.C.: A History of Mathematics, Wiley, N.Y., 1991.
11. Bridges, D.S.: Computable Functional Analysis, Pitman, London, 1979.
12. Canny, J.: Improved algorithms for sign determination and existential quantifier elimination. The Computer Journal **36**(5) (1993) 409–418.
13. Enderton, H.B.: A mathematical introduction to logic, Academic Press, N.Y., 1972.
14. Frisch, H.L., Borzi, C., Ord, G., Percus, J.K., Williams, G.O.: Approximate Representation of Functions of Several Variables in Terms of Functions of One Variable, Physical Review Lett. **63**(9) (1989) 927–929.
15. Godo, L., Lopez de Mantaras, R., Sierra, C., Verdaguer, A.: MILORD: The Architecture and management of Linguistically expressed Uncertainty, International Journal of Intelligent Systems **4** (1989) 471–501.
16. Goodman, I.R., Nguyen, H.T.: Probability updating using second-order probabilities and conditional event algebra, Information Sciences, 2000.
17. Goodman, I.R., Trejo, R.A., Kreinovich, V., Martinez, J., Gonzalez, R.: An even more realistic (non-associative) interval logic and its relation to psychology of human reasoning, Proc. IFSA/NAFIPS'2001, Vancouver, Canada, July 25-28, 2001, 1586–1591.
18. Hecht-Nielsen, R.: Kolmogorov's Mapping Neural Network Existence Theorem, IEEE Int'l Conf. on Neural Networks, San Diego **2** (1987) 11–14.
19. Heindl, G., Kreinovich, V., Rifqi, M.: In case of interval (or more general) uncertainty, no algorithm can choose the simplest representative, Reliable Computing, 2002 (to appear).
20. Hilbert, D.: Mathematical Problems, lecture delivered before the Int'l Congress of Mathematics in Paris in 1900, translated in Bull. Amer. Math, Soc. **8** (1902) 437–479.
21. Hobbs, J.R.: Half orders of magnitude, In: Obrst, L., Mani, I., eds.: Proc. of KR'2000 Workshop on Semantic Approximation, Granularity, and Vagueness, Breckenridge, Colorado, April 11, 2000, pp. 28–38.
22. Hobbs, J.R., Kreinovich, V.: Optimal Choice of Granularity In Commonsense Estimation: Why Half-Orders of Magnitude, Proc. IFSA/NAFIPS'2001, Vancouver, Canada, July 25-28, 2001, 1343–1348.
23. Hurwicz, L.: A criterion for decision-making under uncertainty, Technical Report 355, Cowles Commission, 1952.
24. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, Upper Saddle River, NJ, 1995.
25. Kolmogorov, A.N.: On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition, Dokl. Akad. Nauk SSSR **114** (1957) 369–373.

26. Kreinovich, V., Nguyen, H.T.: 1st Order, 2nd Order, What Next? Do We Really Need Third-Order Descriptions: A View From A Realistic (Granular) Viewpoint, Proc. IFSA/NAFIPS'2001, Vancouver, Canada, July 25-28, 2001, 1908–1913.
27. Kreinovich, V., Nguyen, H.T., Pedrycz, W.: How to Make Sure That $\approx 100 + 1$ Is ≈ 100 in Fuzzy Arithmetic: Solution and Its (Inevitable) Drawbacks, Proc. IFSA/NAFIPS'2001, Vancouver, Canada, July 25-28, 2001, 1653–1658.
28. Kurkova, V.: Kolmogorov's Theorem Is Relevant, Neural Computation **3** (1991) 617–622.
29. Kurkova, V.: Kolmogorov's Theorem and Multilayer Neural Networks, Neural Networks **5** (1991) 501–506.
30. Langrand, G., Kreinovich, V., Nguyen, H.T.: Two-dimensional fuzzy logic for expert systems, Sixth International Fuzzy Systems Association World Congress, San Paulo, Brazil, July 22–28, 1995, **1** 221–224.
31. Lewis, L.R., Papadimitriou, C.H.: Elements of the theory of computation, Prentice-Hall, Englewood Cliffs, NJ, 1981.
32. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications, Springer-Verlag, N.Y., 1997.
33. Lorentz, G.G.: Approximation of functions, Halt, Reinhart, and Winston, N.Y., 1966.
34. Martin, J.C.: Introduction to languages and the theory of computation, McGraw-Hill, N.Y., 1991.
35. McCarty, G.: Topology, Dover, New York, 1988.
36. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper Saddle River, NJ, 2001.
37. Miller, G.A.: The magical number seven plus or minus two: some limits on our capacity for processing information, Psychological Review **63** (1956) 81–97.
38. Milner, P.M.: Physiological psychology, Holt, NY, 1970.
39. Proc. NAFIPS/IFIS/NASA'94, San Antonio, December 18–21, 1994.
40. Nakamura, M., Mines, R., Kreinovich, V.: Guaranteed intervals for Kolmogorov's theorem (and their possible relation to neural networks), Interval Computations No. 3 (1993) 183–199.
41. Ness, M.: Approximative versions of Kolmogorov's superposition theorem, proved constructively, J. Comput. Appl. Math. (1993).
42. Nesterov, V.M.: Interval analogues of Hilbert's 13th problem, Abstracts of the Int'l Conference Interval'94, St. Petersburg, Russia, March 7–10, 1994, 185–186.
43. Nguyen, H.T., Kreinovich, V.: Towards theoretical foundations of soft computing applications, Int'l J. on Uncertainty, Fuzziness, and Knowledge-Based Systems **3**(3) (1995) 341–373.
44. Nguyen, H.T., Kreinovich, V.: Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications, In: Kearfott, R.B. et al., eds.: Applications of Interval Computations, Kluwer, Dordrecht, 1996, 245-290.
45. Nguyen, H.T., and Kreinovich, V.: "Kolmogorov's Theorem and its impact on soft computing", In: Yager, R.E., Kacprzyk, J.: The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer, Boston, MA, 1997, 3–17.
46. Nguyen, H.T., Kreinovich, V.: Possible new directions in mathematical foundations of fuzzy technology: a contribution to the mathematics of fuzzy theory, In: Nguyen Hoang Phuong and A. Ohsato, eds.: Proceedings of the Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications VJFUZZY'98, HaLong Bay, Vietnam, 30th September–2nd October, 1998, 9–32.

47. Nguyen, H.T., Kreinovich, V., Goodman, I.R.: Why Unary and Binary Operations in Logic: General Result Motivated by Interval-Valued Logics, Proc. IFSA/NAFIPS'2001, Vancouver, Canada, July 25-28, 2001, 1991–1996.
48. Nguyen, H.T., Kreinovich, V., Shekhter, V.: On the Possibility of Using Complex Values in Fuzzy Logic For Representing Inconsistencies, International Journal of Intelligent Systems **13**(8) 1998 683–714.
49. Nguyen, H.T., Kreinovich, V., Sprecher, D.: Normal forms for fuzzy logic – an application of Kolmogorov's theorem, International Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems **4**(4) (1996) 331–349.
50. Nguyen, H.T., Kreinovich, V., Wu, B.: Fuzzy/probability $\sim$ fractal/smooth, International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS) **7**(4) (1999) 363–370.
51. Nguyen, H.T., Kreinovich, V., Zuo, Q.: Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control, International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS) **5**(3) (1997) 317–358.
52. Nguyen, H.T., Walker, E.A.: First Course in Fuzzy Logic, CRC Press, Boca Raton, FL, 1999.
53. Nilsson, N.J.: Probabilistic logic, Artificial Intelligence **28** (1986) 71–87.
54. Papadimitriou, C.H.: Computational Complexity, Addison Wesley, San Diego, 1994.
55. Puyol-Gruart, J. Godo, L., Sierra, C.: A specialization calculus to improve expert systems communication, Research Report IIIA 92/8, Institut d'Investigació en Intelligènicia Artificial, Spain, May 1992.
56. Schoenfield, J.R.: Mathematical logic,Addison-Wesley, 1967.
57. Seidenberg, A.: A new decision method for elementary algebra. Annals of Math. **60** (1954) 365–374.
58. Shafer, G., Pearl, J., eds: Readings in Uncertain Reasoning, M. Kaufmann, San Mateo, CA, 1990.
59. Suppes, P., Krantz, D.M., Luce, R.D., Tversky, A.: Foundations of measurement, Vol. I-III, Academic Press, San Diego, CA, 1989.
60. Tarski, A.: A Decision Method for Elementary Algebra and Geometry, University of California Press, Berkeley, 1948.
61. Türkşen, I.B.: Interval valued fuzzy sets based on normal forms, Fuzzy Sets and Systems **20** (1986) 191–210.
62. Wadsworth, H.M., ed.: Handbook of statistical methods for engineers and scientists, McGraw-Hill Publishing Co., N.Y., 1990.
63. Whalen, T.: Interval probabilities induced by decision problems, In: Yager, R.R., Kacprzyk, J., Pedrizzi, M., eds.: Advances in the Dempster-Shafer Theory of Evidence, Wiley, N.Y., 1994, 353–374.
64. Yamakawa, T., Kreinovich, V.: Why Fundamental Physical Equations Are of Second Order?, International Journal of Theoretical Physics **38**(6) (1999) 1763–1770.
65. Zadeh, L.A.: Fuzzy Sets, Information and Control **8** (1965) 338–353.
66. Zemanian, A.H.: Distribution theory and transform analysis, Dover, New York, 1987.