12-2019

Finitely Generated Sets of Fuzzy Values: If "And" Is Exact, Then "Or" Is Almost Always Approximate, And Vice Versa -- A Theorem

Julio Urenda

Olga Kosheleva

Vladik Kreinovich

Comments:
Technical Report: UTEP-CS-19-109

Finitely Generated Sets of Fuzzy Values: If "And" Is Exact, Then "Or" Is Almost Always Approximate, And Vice Versa – A Theorem

Julio Urenda[1,2], Olga Kosheleva[3], and Vladik Kreinovich[2]
[1]Department of Mathematical Sciences
[2]Department of Computer Science
[3]Department of Teacher Education
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
jcurenda@utep.edu, olgak@utep.edu, vladik@utep.edu

Abstract

In the traditional fuzzy logic, experts' degrees of confidence are described by numbers from the interval $[0,1]$. Clearly, not all the numbers from this interval are needed: in the whole history of the Universe, there will be only countably many statements and thus, only countably many possible degree, while the interval $[0,1]$ is uncountable. It is therefore interesting to analyze what is the set S of actually used values. The answer depends on the choice of "and"-operations (t-norms) and "or"-operations (t-conorms). For the simplest pair of min and max, any finite set will do – as long as it is closed under negation $1-a$. For the next simplest pair – of algebraic product and algebraic sum – we prove that for a finitely generated set, if the "and"-operation is exact, then the "or"-operation is almost always approximate, and vice versa. For other "and"- and "or"-operations, the situation can be more complex.

1 Formulation of the Problem

Need for fuzzy degrees: a brief reminder. Computers are an important part of our lives. They help us understand the world, they help us make good decisions. It is desirable to make sure that these computers possess as much of our own knowledge as possible.

Some of this knowledge is precise; such a knowledge is relatively easy to describe in computer-understandable terms. However, a significant part of our knowledge is described by using imprecise ("fuzzy") words from natural language. For example, to design better self-driving cars, it sounds reasonable to ask experienced drivers what they do in different situations, and implement the corresponding rules in the car's computer. The problem with this idea is that expert drivers usually describe their rules by saying something like "If the car in front of you is close, and it slows down a little bit, then ...". Here, "close", "a little bit", etc. are imprecise words.

To translate such knowledge into computer-understandable terms, Lotfi Zadeh invented *fuzzy logic*, in which each imprecise terms like "close" is described by assigning, to each possible value x of the corresponding quantity (in this case, distance), a degree to which x satisfies the property under consideration (in this case, to what exact, x is close); see, e.g., [3, 6, 11, 14, 15, 19]. In the original formulation of fuzzy logic, the degrees are described by numbers from the interval $[0, 1]$, so that:

- 1 means that we are absolutely sure about the corresponding statement,
- 0 means that we are sure that this statement is false, and
- intermediate degrees corresponding to intermediate degrees of certainty.

Need for operations on fuzzy degrees, i.e., for fuzzy logic: a brief reminder. Our rules often use logical connectives like "and" and "or". In the above example of the car-related statement, the person used "and" and "if-then" (implication). Other statements use negation or "or".

In the ideal world, we should ask each expert to describe his or her degree of confidence in each such statement – e.g., in the statement that "the car in front of you is close, and it slows down a little bit". However, here is a problem: to describe each property like "close" for distance or "a little bit" for a change in speed, it is enough to list possible values of one variable. For a statement about two variables – as above – we already need to consider all possible pairs of values. So, if we consider N possible values of each variable, we need to ask the expert N^2 questions. If our statement involves three properties – which often happens – we need to consider N^3 possible combinations, etc. With a reasonably large N, this quickly becomes impossible to ask the expert all these thousands (and even millions) of questions. So, instead of explicitly asking all these questions about composite statements like $A \,\&\, B$, we need to be able to estimate the expert's degree of confidence in such a statement based on his or her known degrees of confidence a and b in the original statements A and B.

A procedure that transform these degrees a and b into the desired estimate for the degree of confidence in $A \,\&\, B$ is known as an *"and"-operation* (or, for historical reasons, a *t-norm*). We will denote this procedure by $f_\&(a, b)$. Similarly, a procedure corresponding to "or" is called an *"or"-operation* or a *t-conorm*; it will be denoted by $f_\vee(a, b)$. We can also have negation operations $f_\neg(a)$, implication operation $f_\rightarrow(a, b)$, etc.

Simple examples of operations on fuzzy degrees. In his very first paper on fuzzy logic, Zadeh considered the two simplest possible "and"-operations $\min(a, b)$ and $a \cdot b$, the simplest negation operation $f_\neg(a) = 1 - a$, and the simplest possible "or"-operations $\max(a, b)$ and $a + b - a \cdot b$.

It is easy to see that the corresponding "and"- and "or"-operations form two dual pairs, i.e., pairs for which $f_\vee(a,b) = f_\neg(f_\&(f_\neg(a), f_\neg(b))$ – this reflects the fact that in our reasoning, $a \vee b$ is indeed usually equivalent to $\neg(\neg a \,\&\, \neg b)$. Indeed, for example, what does it mean that a dish contains either pork or alcohol (or both)? It simply means that it is not true that this is an alcohol-free and pork-free dish.

Both pairs of operations can be derived from the requirement of the smallest sensitivity to changes in a and b (see, e.g., [13, 14, 18])– which makes sense, since experts can only mark their degree of confidence with some accuracy, and we do not want the result of, e.g., "and"-operation drastically change if we replace the original degree 0.5 with a practically indistinguishable degree 0.51. If we require that the worst-case change in the result of the operation be as small as possible, we get $\min(a,b)$ and $\max(a,b)$. If we require that the mean squares value of the change be as small as possible, we get $a \cdot b$ and $a + b - a \cdot b$.

Implication $A \to B$ means that, if we know A and we know that implication is true, then we can conclude B. In other words, implication is the weakest of all statements C for which $A \,\&\, C$ implies B. So, if we know the degrees of confidence a and b in the statements A and B, then a reasonable definition of the implication $f_\vee(a,b)$ is the smallest degree c for which $f_\&(a,c) \ge b$. In this sense, implication is, in some reasonable sense, an inverse to the "and"-operation. In particular, when the "and"-operation is multiplication $f_\&(a,b) = a \cdot b$, the implication operation is simply division: $f_\to(a,b) = b/a$ (if $b \le a$).

Not all values from the interval $[0,1]$ make sense. While it is reasonable to use numbers from the interval $[0,1]$ to describe the corresponding degrees, the inverse is not true – not every number from the interval $[0,1]$ makes sense as a degree. Indeed, whatever degree we use corresponds to some person's informal description of his or her degree of confidence. Whatever language we use, there are only countably many words, while, as is well known, the set of all real numbers from an interval is uncountable.

Usually, we have a finite set of basic degrees, and everything else is obtained by applying some logical operations. A natural question is: what can we say about the resulting – countable – sets of actually used values? This is a general question to which, in this paper, we provide a partial answer.

Simplest case: $\min$ and $\max$. The simplest case if when have $f_\&(a,b) = \min(a,b)$, $f_\neg(a) = 1 - a$, and $f_\vee(a,b) = \max(a,b)$. In this case, if we start with a finite set of degrees $a_1, \ldots, a_n$, then we add their negations $1 - a_1, \ldots, 1 - a_n$, and that, in effect, is it: $\min(a,b)$ and $\max(a,b)$ do not generate any new values, they just select one of the two given ones (a or b).

What about $a \cdot b$ and $a + b - a \cdot b$. What about the next simplest pair of operations? Since the product is the simplest of the two, let us start with the product. Again, we start with a finite set of degrees $a_1, \ldots, a_n$. We can also consider their negations $a_{n+1} \stackrel{\text{def}}{=} 1 - a_1, \ldots, a_{n+i} \stackrel{\text{def}}{=} 1 - a_i, \ldots, a_{2n} \stackrel{\text{def}}{=} 1 - a_n$.

If we apply "and"-operation to these values, we get products, i.e., values the type

$$a_1^{k_1} \cdot \ldots \cdot a_{2n}^{k_{2n}} \tag{1}$$

for integers $k_i \geq 0$. If we also allow implication – i.e., in this case, division – then we get values of the same type (1), but with integers k_i being possibly negative.

The set of all such values is generated based on the original finite set of values. Thus, we can say that this set is *finitely generated*.

Every real number can be approximated, with any given accuracy, by a rational number. Thus, without losing generality, we can assume that all the values a_i are rational numbers – i.e., ratios of two integers.

Since for dual operations, the result of applying the "or"-operation is the negation of the result of applying the "and"-operation – to negations of a and b – a natural question is: if we take values of type (1), how many of their negations are also of the same type? This is a question that we study in this paper.

2 Definitions and the Main Result

Definition 1. *By a* finitely generated set *of fuzzy degrees, we mean a set S of values of the type (1) from the interval* $[0, 1]$*, where $a_1, \ldots, a_n$ are given rational numbers, $a_{n+i} = 1 - a_i$, and $k_1, \ldots, k_{2n}$ are arbitrary integers.*

Examples. If we take $n = 1$ and $a_1 = 1/2$, then $a_2 = 1 - a_1 = 1/2$, so all the values of type (1) are $1/2$, $1/4$, $1/8$, etc. Here, only for one number $a_1 = 1/2$, the negation $1 - a_1$ belongs to the same set.

If we take $a_1 = 1/3$ and $a_2 = 2/3$, then we have more than one number a from the set S for which its negation $1 - s$ is also in S:

- we have $1/4 = a_1^2 \cdot a_2^{-2} \in S$ for which $1 - 1/4 = 3/4 = a_1 \cdot a_2^{-2} \in S$; and
- we have $1/9 = a_1^2 \in S$ and $1 - 1/9 = 8/9 = a_1^{-1} \cdot a_2^3 \in S$.

Proposition 1. *For each finitely generated set S of fuzzy degrees, there are only finitely many element $s \in S$ for which $1 - s \in S$.*

Proof is, in effect, contained in [2, 4, 5, 7, 8, 10, 12, 16, 17], where the values $s \in S$ are called *S-units* and the desired formula $s + s' = 1$ for $s, s' \in S$ is known as the *S-unit equation*.

Historical comment. The history of this mathematical result is unusual (see, e.g., [9]): the corresponding problem was first analyzed by Axel Thue in 1909, it was implicitly proven by Carl Lugwig Siegel in 1929, then another implicit proof was made by Kurt Mahler in 1933 – but only reasonably recently this result was explicitly formulated and explicitly proven.

Discussion. Proposition 1 says that for all but finitely many ("almost all") values $s \in S$, the negation $1 - s$ is outside the finitely generated set S.

Since, as we have mentioned, to get an "or"-operation out of "and" requires negation, this means that while for this set, "and"-operation is exact, the corresponding "or"-operation almost always leads us to a value outside S. So, if we restrict ourselves to the finitely generated set S, we can only represent the results of "or"-operation approximately.

In other words, if "and" is exact, then "or" is almost always approximate. Due to duality between "and"- and "or", we can also conclude that if "or" is exact, then "and" is almost always approximate.

Computational aspects. The formulation of our main result sounds like (too) abstract mathematics: there exists finitely many such values s; but how can we find them? Interesting, there exists a reasonably efficient algorithms for finding such values; see, e.g., [1].

Relation to probabilities. Our current interest is in fuzzy logic, but it should be mentioned that a similar results holds for the case of probabilistic uncertainty, when, instead of degrees of confidence, we consider possible probability values a_i. In this case:

- if an event has probability a, then its negation has probability $1 - a$;
- if two independent events have probabilities a and b, then the probably that both events will happen is $a \cdot b$; and
- if an event B is a particular case of an event A, then the conditional probability $P(B \mid A)$ is equal to b/a.

Thus, in the case of probabilistic uncertainty, it also makes sense to consider multiplication and division operations – and thus, to consider sets which are closed under these operations.

3 How General Is This Result?

Formulation of the problem. In the previous section, we considered the case when $f_\&(a, b) = a \cdot b$ and $f_\neg(a) = 1 - a$. What if we consider another pair of operations, will the result still be true?

For example, is it true for *Archimedean* "and"-operations?

Analysis of the problem. It is known that every Archimedean "and"-operation is equivalent to $f_\&(a, b) = a \cdot b$ – namely, we can reduce it to the product by applying an appropriate strictly increasing re-scaling $r : [0, 1] \to [0, 1]$; see, e.g., [6, 14].

Thus, without losing generality, we can assume that the "and"-operation is exactly the product $f_\&(a, b) = a \cdot b$, but the negation operation may be different – as long as $f_\neg(f_\neg(a)) = a$ for all a.

Result of this section. It turns out that there are some negation operations for which the above result does not hold.

Proposition 2. *For each finitely generated set S – with the only exception of the set generated by a single value 1/2 – there exists a negation operation $f_\neg(a)$ for which, for infinitely many $s \in S$, we have $f_\neg(s) \in S$.*

Proof. When at least one of the original values a_i is different from $1/2$, this means that the fractions a_i and $1 - a_i$ have different combinations of prime numbers in their numerators and denominators. In this case, for every $\varepsilon > 0$, there exists a number $s \in S$ for which $1 - \varepsilon < s < 1$.

We know that one of the original values a_i is different from $1/2$. Without losing generality, let us assume that this value is $1/2$. If $a_1 > 1/2$, then $1 - a_1 < 1/2$. So, again without losing any generality, we can assume that $a_1 < 1/2$.

Let us now define two monotonic sequences p_n and q_n. For the first sequence, we take the values

$$p_0 = 1/2 > p_1 = a_1 > p_2 = a_1^2 > p_3 = a_1^3 > \ldots$$

The second sequence is defined iteratively:

- As q_0, we take $q_0 = 1/2$.
- As q_1, let us select some number (smaller than 1) from the set S which is greater than or equal to $1 - p_1$.
- Once the values $q_1, \ldots, q_k$ have been selected, we select, as q_{k+1}, a number (smaller than 1) from the set S which is larger than q_k and larger than $1 - p_{k+1}$, etc.

For values $s \le 0.5$, we can then define the negation operation as follows:

- for each k, we have $f_\neg(p_k) = q_k$ and
- it is linear for $p_{k+1} < s < p_k$, i.e.

$$f_\neg(s) = q_{k+1} + (s - p_{k+1}) \cdot \frac{q_{k+1} - q_k}{p_{k+1} - p - k}.$$

The resulting function maps the interval $[0, 0.5]$ to the interval $[0.5, 1]$. For values $s \ge 0.5$, we can define $f_\neg(a)$ as the inverse function to this.

4 What If We Allow Unlimited Number of "And"-Operations and Negations: Case Study

Formulation of the problem. In the previous sections, we allowed an unlimited application of "and"-operation and implication. What if instead, we allow an unlimited application of "and"-operation and negation?

Here is our related result.

Proposition 3. *The set S of degrees that can be obtained from 0, 1/2, and 1 by using "and"-operation $f_\&(a, b) = a \cdot b$ and negation $f_\neg(a) = 1 - a$ is the* *set of all binary-rational numbers, i.e., all numbers of the type $p/2^k$ for natural numbers p and k for which $p \leq 2^k$.*

Proof. Clearly, the product of two binary-rational numbers is binary-rational, and 1 minus a binary-rational number is also a binary-rational number. So, all elements of the set S are binary-rational.

To complete the proof, we need to show that every binary-rational number $p/2^k$ belongs to the set S, i.e., can be obtained from $1/2$ by using multiplication and $1 - a$. We will prove this result by induction over k.

For $k = 1$, this means that 0, $1/2$, and 1 belong to the set S – and this is clearly true, since S consists of all numbers that can be obtained from these three, these three numbers included.

Let us assume that this property is proved for k. Then, for $p \leq 2^k$, each element $p/2^{k+1}$ is equal to the product $(1/2) \cdot (p/2^k)$ of two numbers from the set S and thus, also belongs to S. For $p > 2^k$, we have

$$p/2^{k+1} = 1 - (2^{k+1} - p)/2^{k+1}.$$

Since $p > 2^k$, we have $2^{k+1} - p < 2^k$ and thus, as we have just proved, $(2^{k+1} - p)/2^{k+1} \in S$. So, the ratio $p/2^{k+1}$ is obtained by applying the negation operation to a number from the set S and is, therefore, itself an element of the set S.

The induction step is proven, and so is the proposition.

Comment. If we also allow implication $f_{\&}(a, b) = b/a$, then we will get all possible rational numbers p/q from the interval $[0, 1]$. Indeed, if we pick k for which $q < 2^k$, then for $a = q/2^k$ and $b = p/2^k$, we get $b/a = p/q$.

Acknowledgments

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] A. Alvarado, A. Koutsianas, B. Malmskog, C. Rasmussen, D. Roe, C. Vincent, and M. West, *Solve S-unit equation x + y = 1 — Sage Reference Manual v8.7: Algebraic Numbers and Number Fields*, 2019, available at http://doc.sagemath.org/html/en/reference/number_fields/sage/rings/number_field/S_unit_solver.html

[2] A. Baker and G. Wüstholz, *Logarithmic Forms and Diophantine Geometry*, Cambridge University Press, Cambridge, UK, 2007.

[3] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[4] E. Bombieri and W. Gubler, *Heights in Diophantine Geometry*, Cambridge University Press, Cambridge, UK, 2006.

[5] G. Everest, A. van der Poorten, I. Shparlinski, and Th. Ward, *Recurrence Sequences*, American Mathematical Society, Providence, Rhode Island, 2003.

[6] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[7] S. Lang, *Elliptic Curves: Diophantine Analysis*, Springer Verlag, Berin, Heidelberg, New York, 1978.

[8] S. Lang, *Algebraic Number Theory*, Springer Verlag, Berin, Heidelberg, New York, 1986.

[9] D. Mackenzie, "Needles in an infinite haystack", In: D. Mackenzie, *What's Happening in the Mathematical Sciences, Vol. 11*, American Mathematical Society, Providence, Rhode Island, 2019, pp. 123–136.

[10] B. Malmskog and C. Rasmussen, "Picard curves over $\mathbb{Q}$ with good reduction away from 3", *London Mathematical Society (LMS) Journal of Computation and Mathematics*, 2016, Vol. 19, No. 2, pp. 382–408.

[11] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.

[12] J. Neukirch, *Class Field Theory*, Springer-Verlag, Berlin-Heidelberg-New York, 1986.

[13] H. T. Nguyen, V. Kreinovich, and D. Tolbert, "On robustness of fuzzy logics", *Proceedings of the 1993 IEEE International Conference on Fuzzy Systems FUZZ-IEEE'93*, San Francisco, California, March 1993, Vol. 1, pp. 543–547.

[14] H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2019.

[15] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[16] N. P. Smart, "The solution of triangularly connected decomposable form equations", *Mathematics of Computation*, 1995, Vol. 64, No. 210, pp. 819–840.

[17] N. P. Smart, *The Algorithmic Resolution of Diophantine Equations*, London Mathematical Society, London, UK, 1998.

[18] D. Tolbert, *Finding "and" and "or" operations that are least sensitive to change in intelligent control*, University of Texas at El Paso, Department of Computer Science, Master's Thesis, 1994.

[19] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.