

7-2009

Square Root of "Not": A Major Difference Between Fuzzy and Quantum Logics

Vladik Kreinovich

The University of Texas at El Paso, vladik@utep.edu

Ladislav J. Kohout

Eunjin Kim

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep



Part of the [Computer Engineering Commons](#)

Comments:

Technical Report: UTEP-CS-08-10c

Short version published in *Proceedings of the 27th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2008*, New York, New York, May 19-22, 2008; full paper to appear in *International Journal of General Systems*

Recommended Citation

Kreinovich, Vladik; Kohout, Ladislav J.; and Kim, Eunjin, "Square Root of "Not": A Major Difference Between Fuzzy and Quantum Logics" (2009). *Departmental Technical Reports (CS)*. 76.
https://scholarworks.utep.edu/cs_techrep/76

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.

RESEARCH ARTICLE

Square Root of “Not”: A Major Difference Between Fuzzy and Quantum Logics

Vladik Kreinovich^{a,*}, Ladislav Kohout^b, and Eunjin Kim^c

^a*Department of Computer Science, University of Texas, El Paso, TX 79968, USA*

^b*Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA*

^c*Dept. Computer Science, University of North Dakota, Grand Forks, ND 58202, USA*

(Received June 2008)

Many authors have mentioned the similarity between quantum logic and fuzzy logic. In this paper, we show that, in spite of this similarity, these logics are not identical. Specifically, we emphasize that while quantum logic has a special “square root of not” operation which is very useful in quantum computing, fuzzy logic lacks such an operation.

Keywords: fuzzy logic; quantum logic; quantum computing; square root of “not”

1. Similarity Between Quantum Logic and Fuzzy Logic

Both quantum logic and fuzzy logic describe uncertainty:

- quantum logic describes uncertainties of the real world (to be more precise, the uncertainty of our best theory of the world), while
- fuzzy logic described the uncertainty of our reasoning.

Due to this common origin, there is a lot of similarity between the two logics, similarities which have been mentioned in several papers on fuzzy logic; see, e.g., Kosko (1993), Seising (2008) and references therein.

2. What We Plan to Do

We plan to emphasize that, in spite of the similarity, quantum logic and fuzzy logic are not mathematically identical.

Specifically, in this paper, we show the difference on the example of one of the major features of quantum logic, a features that underlies most successful quantum computing algorithms: that in quantum logic, there is a “square root of not” operation.

*Corresponding author. Email: vladik@utep.edu

3. What Is a Square Root of Not?

In precise terms, the fact that an operation $s(x)$ is a square root of “not” means that if we apply this operation twice to a truth value a , we get $\neg a$ (“not a ”):

$$s(s(a)) = \neg a$$

for all a .

4. Negation (“Not”) in Classical (2-Valued) Logic

In the traditional (2-valued) logic, we have two possible truth values – “true” and “false”. In the computer, “true” is usually represented as 1, and “false” as 0.

In these terms, the negation operation has a very simple form: $\neg(0) = 1$ and $\neg(1) = 0$.

5. There Is No Square Root of Not in Classical Logic

In classical logic, a unary operation $s(a)$ can be described by listing its values $s(0)$ and $s(1)$. There are two possible values of $s(0)$ and two possible values of $s(1)$, so overall, we have $2 \times 2 = 4$ possible unary operations:

- when $s(0) = s(1) = 0$, then we get a constant function whose value is “false”;
- when $s(0) = s(1) = 1$, then we get a constant function whose value is “true”;
- when $s(0) = 0$ and $s(1) = 1$, we get the identity function;
- finally, when $s(0) = 1$ and $s(1) = 0$, we get the negation.

In all four cases, the composition $s(s(a))$ is different from the negation:

- for the “constant false” function s , we have $s(s(a)) = s(a)$,
i.e., the composition of s and s is also a constant false function;
- for the “constant true” function s , also $s(s(a)) = s(a)$,
i.e., the composition of s and s is also a constant true function;
- for the identity function s , we have $s(s(a)) = s(a)$,
i.e., the composition of s and s is also the identity function;
- finally, for the negation s , the composition of s and s is the identity function.

6. Quantum Mechanics

Since early 20th century, physicists have found out that our physical world is better described not by the classical Newtonian physics, but by the laws of quantum mechanics. The smaller the particles, the larger the deviation between the classical and quantum descriptions. So, for macro-size bodies, Newtonian mechanics provides a very accurate description. However, for molecules and atoms, it is important to take into account quantum effects.

One of the main features of quantum mechanics is the possibility of *superpositions*. Namely, each classical state a is also a quantum state – denoted by $|a\rangle$. However, in addition to this, for every n states a_1, \dots, a_n , and for every n complex numbers c_1, \dots, c_n for which

$$|c_1|^2 + \dots + |c_n|^2 = 1,$$

the following state is also possible:

$$c_1 \cdot |a_1\rangle + \dots + c_n \cdot |a_n\rangle.$$

If, in this state, we try to measure whether we are in the state a_1 , or in the state a_2 , etc., then:

- we will get the state a_1 with the probability $|c_1|^2$;
- ...
- we will get the state a_n with the probability $|c_n|^2$.

The above requirement $|c_1|^2 + \dots + |c_n|^2 = 1$ simply comes from the fact that the probabilities should add up to 1.

It is worth mentioning that if we multiply all the values c_i by the same constant $e^{i\alpha}$ (with real α) whose absolute value is 1, we get the same probabilities of all the states. In quantum mechanics, mathematically different states a and $e^{i\alpha} \cdot a$ are therefore considered to be corresponding to the same physical state.

7. Quantum Logic

Quantum Logic is an application of the general idea of quantum mechanics to logic. In the classical logic, there are two possible states: 0 and 1. In quantum logic, in addition to these states $|0\rangle$ and $|1\rangle$, we can have arbitrary superpositions

$$c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$$

for complex values c_0 and c_1 for which $|c_0|^2 + |c_1|^2 = 1$.

These superpositions are the “truth values” of quantum logic.

8. Negation in Quantum Logic

For “pure” (classical) states $|0\rangle$ and $|1\rangle$, negation can be defined in a standard way:

$$\neg(|0\rangle) = |1\rangle \text{ and } \neg(|1\rangle) = |0\rangle.$$

In general, in quantum mechanics, all operations are linear in terms of superpositions. By using this linearity, we can describe the negation of an arbitrary quantum state:

$$\neg(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle) = c_0 \cdot |1\rangle + c_1 \cdot |0\rangle.$$

9. Alternative Quantum Negation

Since $-|1\rangle$ and $|1\rangle$ is the same physical state, we can have the following alternative description of quantum negation:

$$\neg(|0\rangle) = -|1\rangle; \quad \neg(|1\rangle) = |0\rangle.$$

By using this linearity, we can describe the negation of an arbitrary quantum state:

$$\neg(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle) = -c_0 \cdot |1\rangle + c_1 \cdot |0\rangle.$$

Here,

$$\neg\neg(|0\rangle) = \neg(-|1\rangle) = -|0\rangle; \quad \neg\neg(|1\rangle) = \neg(|0\rangle) = -|1\rangle.$$

Due to linearity, we have

$$\neg\neg(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle) = -(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle).$$

In other words, $\neg\neg(a) = -a$, i.e., $\neg\neg(a)$ and a is the same physical state.

10. Square Root of Not: Case of Alternative Definition

Let us show that in quantum mechanics, there exists an operation s whose square is equal to negation. The alternative quantum negation operation has the form:

$$\neg(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle) = -c_0 \cdot |1\rangle + c_1 \cdot |0\rangle.$$

This formula has a natural geometric interpretation: possible pairs (c_0, c_1) can be geometrically described as points on a plane, and an alternative negation is rotation by 90 degrees in this plane.

Thus, there is a natural square root $s(a)$ of this alternative negation operation: rotation by 45 degrees. The resulting rotation-by-45-degrees formulas for the basic states $|0\rangle$ and $|1\rangle$ take the following form:

$$s(|0\rangle) = \frac{1}{\sqrt{2}} \cdot |0\rangle - \frac{1}{\sqrt{2}} \cdot |1\rangle; \quad s(|1\rangle) = \frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle.$$

Due to linearity, the resulting formulas for the general case are:

$$s(c_0 \cdot |0\rangle + c_1 \cdot |1\rangle) = c_0 \cdot \left(\frac{1}{\sqrt{2}} \cdot |0\rangle - \frac{1}{\sqrt{2}} \cdot |1\rangle \right) + c_1 \cdot \left(\frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle \right).$$

One can easily show that the composition of this operation with itself indeed leads to the (alternative) negation, i.e., that $s(s(a)) = \neg(a)$ for all a .

11. Square Root of “Not” Is An Important Part of Quantum Algorithms

The above-described square root of the alternative “not” operation is an important part of quantum algorithms: these algorithms usually start with applying this operation to a basic state and often end with this same “square root of negation” operation.

For example, without using quantum effects, we need – in the worst case – at least N computational steps to search for a desired element in an unsorted list of size N . A quantum computing algorithm proposed by Grover (see, e.g., Grover (1996, 1997), Nielsen and Chuang (2000)) can find this element much faster – in $O(\sqrt{N})$ time.

An even more impressive speedup is achieved in factoring large integers. This problem is extremely important for computer security. Most security features of online communications and e-commerce use RSA encryption algorithm originally invented by R. Rivest, A. Shamir, and L. Adleman; see, e.g., Cormen *et al.* (2001).

At present, this algorithm provides safe communication because to decrypt RSA-encrypted messages, one needs to factor large integers, and all known factoring algorithms require an unrealistically large computation time: namely, time which grows exponentially with the length of the integer. On the other hand, there exist quantum algorithms which enable us to perform this factorization in time which only grows polynomially with this size – and which is, thus, quite practically feasible Nielsen and Chuang (2000), Shor (1994). Thus, quantum computers will lead to breaking most existing encryption codes.

12. Comment: Square Root of “Not”: Case of Original Definition

For simplicity, quantum computing algorithms are usually described as using the square root of the alternative negation – since it enables us to perform all operations with real numbers, without the need to use imaginary complex numbers. However, in principle, we can also use the square root of the original negation operation

$$\neg(|0\rangle) = |1\rangle \text{ and } \neg(|1\rangle) = |0\rangle.$$

Let us show that in quantum mechanics, there indeed exists an operation s for which $s(s(a)) = \neg(a)$.

The negation operation is a linear operation described by the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

This matrix has two eigenvalues described by the equation

$$\det \begin{pmatrix} -\lambda & 1 \\ 1 & -\lambda \end{pmatrix} = (-\lambda)^2 - 1 = 0,$$

hence $\lambda^2 = 1$ and we have two eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -1$. In the basis formed by the corresponding eigenvectors, the transformation has a diagonal form

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

To get a square root of this transformation, it is therefore necessary to take the square roots of the corresponding eigenvalues. By taking $\sqrt{1} = 1$ and $\sqrt{-1} = i$, we get the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

By transforming to the original basis $|0\rangle$ and $|1\rangle$, we get the following expression for this “square root of not” operation $s(x)$:

$$s(|0\rangle) = \frac{1+i}{\sqrt{2}} \cdot |0\rangle + \frac{1-i}{\sqrt{2}} \cdot |1\rangle; \quad s(|1\rangle) = \frac{1-i}{\sqrt{2}} \cdot |0\rangle + \frac{1+i}{\sqrt{2}} \cdot |1\rangle.$$

One can easily show that this operation is indeed the square root of negation.

13. Fuzzy Logic

Let us now consider fuzzy logic; see, e.g., Klir and Yuan (1995), Nguyen and Walker (2006). In fuzzy logic, in addition to the classical values 0 and 1, we also allow intermediate truth values, i.e., arbitrary real numbers from the interval $[0, 1]$.

14. Negation in Fuzzy Logic

Usually, in fuzzy logic, negation is defined as

$$\neg(a) = 1 - a.$$

In principle, there exist other negation operations, but it is known (see, e.g., Klir and Yuan (1995), Nguyen and Walker (2006)) that they can be reduced to this standard negation by an appropriate re-scaling of the interval $[0, 1]$.

15. There Is No Continuous Square Root of Not in Fuzzy Logic

In fuzzy logic, usually, we only consider logical operations which are continuous functions of their inputs. This makes sense because the degree of uncertainty are only approximately known, and for operations to be meaningful, we need to require that similar values of the input degrees lead to similar values of the result of the logical operation.

It is known that, in contrast to quantum logic, in fuzzy logic, if we restrict ourselves to continuous operations $s : [0, 1] \rightarrow [0, 1]$, there is no square root of negation; see, e.g., Dalla Chiara *et al.* (2004) (p. 259) and Dalla Chiara *et al.* (2005).

Indeed, suppose, by contradiction, that there exists a continuous function $s : [0, 1] \rightarrow [0, 1]$ such that for any $x \in [0, 1]$, $s(s(x)) = x$. Clearly, $s(0) \neq 0$ and $s(0) \neq 1$. Since s is onto, there exist a and b in $[0, 1]$ such that $s(a) = 0$ and $s(b) = 1$. If $a < b$ then, since s is continuous on the interval $[0, 1]$ and takes on the values 0 and 1 over the interval, it must take on all intermediate values by Bolzano's Theorem, so that $s([a, b]) = [0, 1]$. Thus, $\exists c \in [a, b]$ such that $s(c) = s(0)$. Since s is one-to-one on $[0, 1]$, we must have $c = 0$, and therefore $a = 0$. But $s(0) \neq 0$, contradiction.

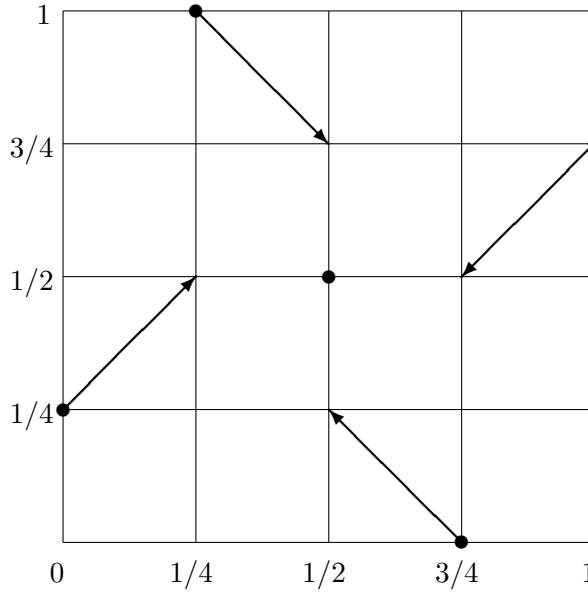
A similar argument shows that if $b < a$, then $b = 0$. Thus, $s(0) = 1$, contradiction. (We are thankful to the anonymous referee for providing this simplified proof.)

16. Discontinuous Square Roots of “Not” Are Possible in Fuzzy Logic

Let us show that if we do not require continuity, then a square root of not is possible in fuzzy logic. The first such example appeared in Dalla Chiara *et al.* (2004), pp. 259–260. Let us describe a similar example, with the following piecewise linear function $s(x)$:

- when $0 \leq x < \frac{1}{4}$, we set $s(x) = x + \frac{1}{4}$;
- when $\frac{1}{4} \leq x < \frac{1}{2}$, we set $s(x) = \frac{5}{4} - x$;
- when $x = \frac{1}{2}$, we set $s(x) = \frac{1}{2}$;

- when $\frac{1}{2} < x \leq \frac{3}{4}$, we set $s(x) = \frac{3}{4} - x$;
- finally, when $\frac{3}{4} < x \leq 1$, we set $s(x) = x - \frac{1}{4}$.



By considering all 5 cases, we can check that $s(s(x)) = 1 - x$ for all $x \in [0, 1]$. Indeed:

When $0 \leq x < \frac{1}{4}$, then for $s(x) = x + \frac{1}{4}$, we get $\frac{1}{4} \leq s(x) < \frac{1}{2}$. Thus, in this case, $s(s(x)) = \frac{5}{4} - s(x)$. Since $s(x) = x + \frac{1}{4}$, we get

$$s(s(x)) = \frac{5}{4} - s(x) = \frac{5}{4} - \left(x + \frac{1}{4}\right) = \frac{5}{4} - x - \frac{1}{4} = 1 - x.$$

When $\frac{1}{4} \leq x < \frac{1}{2}$, then for $s(x) = \frac{5}{4} - x$, we get $\frac{3}{4} < s(x) \leq 1$. Thus, in this case, $s(s(x)) = s(x) - \frac{1}{4}$. Since $s(x) = \frac{5}{4} - x$, we get

$$s(s(x)) = s(x) - \frac{1}{4} = \left(\frac{5}{4} - x\right) - \frac{1}{4} = \frac{5}{4} - x - \frac{1}{4} = 1 - x.$$

When $x = \frac{1}{2}$, then $s(x) = \frac{1}{2}$ and thus, $s(s(x)) = s\left(\frac{1}{2}\right) = \frac{1}{2} = 1 - x$.

When $\frac{1}{2} < x \leq \frac{3}{4}$, then for $s(x) = \frac{3}{4} - x$, we get $0 \leq s(x) < \frac{1}{4}$. Thus, in this case, $s(s(x)) = s(x) + \frac{1}{4}$. Since $s(x) = \frac{3}{4} - x$, we get

$$s(s(x)) = s(x) + \frac{1}{4} = \left(\frac{3}{4} - x\right) + \frac{1}{4} = \frac{3}{4} - x + \frac{1}{4} = 1 - x.$$

Finally, when $\frac{3}{4} < x \leq 1$, then for $s(x) = x - \frac{1}{4}$, we get $\frac{1}{2} < s(x) \leq \frac{3}{4}$. Thus, in this case, $s(s(x)) = \frac{3}{4} - s(x)$. Since $s(x) = x - \frac{1}{4}$, we get

$$s(s(x)) = \frac{3}{4} - s(x) = \frac{3}{4} - \left(x - \frac{1}{4}\right) = \frac{3}{4} - x + \frac{1}{4} = 1 - x.$$

So, this discontinuous function is indeed a square root of negation.

17. Commonsense Interpretation of the Above Example

Question. What is the commonsense meaning of the square root of negation?

Commonsense meaning of negation. To find this meaning, let us start by recalling the commonsense meaning of negation itself. Negation means that we change our opinion about a statement S from true to false and from false to true. Thus, negation can be viewed as an introduction of new strong evidence, new knowledge.

If we originally assumed that S is true, then *new* knowledge means knowledge that contradicts the original assumption S – because evidence in support of the original assumption S does not provide us with any new knowledge. Since we assumed that the new evidence is strong, we thus switch from believing in S into believing in $\neg S$.

If we originally assumed that S is false, then *new* knowledge means knowledge that contradicts the original assumption $\neg S$ – because evidence in support of the original assumption $\neg S$ does not provide us with any new knowledge. Since we assumed that the new evidence is strong, we thus switch from believing in $\neg S$ into believing in S .

Commonsense meaning of the square root of negation. In practice, new knowledge does not necessarily comes in a single piece, it can come in two sequential pieces. In this case, instead of an abrupt change in belief based on the whole new knowledge, we:

- first have a change of belief caused by the first piece of the new knowledge, and
- then a similar change in belief cause by the second piece of the new knowledge.

If we submit medium strength new evidence twice, the result is equivalent to introducing both pieces of new evidence – i.e., a strong evidence. Thus, the operation of a change in belief caused by the introduction of the medium strength evidence has the property that if we apply it twice, we get negation. So, this operation is a square root of negation.

Two scenarios: static and dynamic. How does this operation look like? Let us consider two possible scenarios,

- a *static* scenario in which we deal with reasonably static objects, whose rate of change is small, and
- a *dynamic* scenario in which we deal with objects which can change rapidly.

Case of the static scenario. In the static scenario, if we originally assumed (based on some reasoning or some evidence) that a statement S is true, and we are presented with medium-strength evidence that S is false, then we still retain our belief in S , but the strength of this belief decreases. In other words, we go from the

degree of belief T (strong belief that S is true) into a weak belief that S is true. The corresponding degree of belief will be denoted by t . When we are in the state t , and we are presented with the second part of evidence against S , then this new evidence prevails, and we end up believing in $\neg S$ (i.e., with the degree of belief F in S).

Similarly, if we originally assumed (based on some reasoning or some evidence) that a statement S is false, and we are presented with medium-strength evidence that S is true, then we still retain our belief in $\neg S$, but the strength of this belief decreases. In other words, we go from the degree of belief F (strong belief that S is false) into a weak belief that S is false. The corresponding degree of belief will be denoted by f . When we are in the state f , and we are presented with the second part of evidence for S , then this new evidence prevails, and we end up believing in S (i.e., with the degree of belief T in S).

In this case, we have 4 degrees of belief $F < f < t < T$, with a natural negation

$$\neg(F) = T; \quad \neg(f) = t; \quad \neg(t) = f; \quad \neg(T) = F,$$

and the following square root of negation operation:

$$F \rightarrow f \rightarrow T \rightarrow t \rightarrow F.$$

Case of the dynamic scenario. In the dynamic scenario, we are studying an object whose properties change fast. In this scenario, we put more weight on the new evidence, since the old knowledge was collected when the object was different.

So, if we originally assumed that a statement S is true, and we are presented with new medium-strength evidence that S is false, then we put more weight on the new evidence and thus conclude that S is false. However, since we are only presented with a piece of evidence, we are not completely convinced that S is false, we just have a weak belief that S is false.

In other words, in the dynamic scenario, the partial exposure function $s(x)$ maps the degree T into the degree f . When we get the remaining piece of the new evidence, we become fully confident in $\neg S$, i.e., we go from f to F .

Similarly, if we originally assumed that a statement S is false, and we are presented with new medium-strength evidence that S is true, then we put more weight on the new evidence and thus conclude that S is true. However, since we are only presented with a piece of evidence, we are not completely convinced that S is true, we just have a weak belief that S is true. So, we move from F to t . When we get the remaining piece of the new evidence, we become fully confident in S , i.e., we go from t to T .

In this case, we also have 4 degrees of belief $F < f < t < T$, with the following square root of negation operation:

$$F \rightarrow t \rightarrow T \rightarrow f \rightarrow F.$$

Why discontinuous? In this interpretation, discontinuity makes sense: at some point, we do need to switch from believing in S to believing in $\neg S$ and vice versa.

Similarity with the above discontinuous operation. In both static and dynamic cases, we have an operation which is described by 4 different formulas on four different zones. Thus, each of these operations is a natural analogue of the above discontinuous square-root-of-not operation.

18. How to Describe All Discontinuous Square Roots of Fuzzy Negation

General problem. In the previous section, we presented one example of a square root of the fuzzy negation. It is desirable to describe all such examples.

Finite versions of fuzzy logic. To get this description, let us start with a discrete version of fuzzy logic. In the 2-valued logic, we have only two possible truth values: 0 and 1, with 1 being “more true” than 0: $0 < 1$.

To describe intermediate degree of belief, we can add several intermediate truth values. In the standard fuzzy logic, we consider all possible real values from the interval $[0, 1]$. However, in practice, we encounter only finitely many different distinguishable degrees of belief, so it makes sense to consider only finitely many truth values $t_1 < t_2 < \dots < t_n$, where t_1 corresponds to “absolutely false” (0 in the standard fuzzy logic) and t_n corresponds to “absolutely true” (1 in the standard fuzzy logic).

Once we associated t_1 with 0 and t_n with 1, we need to associate $n - 1$ intermediate values t_2, \dots, t_{n-1} with intermediate values from the interval $[0, 1]$. For this association, it makes sense to select equally spaced values, i.e., select

$$t_1 = 0, \quad t_2 = \frac{1}{n-1}, \quad t_3 = \frac{2}{n-1}, \dots, t_k = \frac{k-1}{n-1}, \dots, t_{n-1} = \frac{n-2}{n-1}, \quad t_n = 1.$$

This association prompts a natural negation operation $\neg(a) = 1 - a$ to be described as $\neg(t_i) = t_{n+1-i}$.

For simplicity of notations, let us describe each degree t_i by simply listing the corresponding integer i . In this simplified notation, the set of possible truth values is a finite set $\{1, 2, \dots, n\}$ on which there is a negation operation $\neg(i) = n + 1 - i$.

Two natural questions. Now, we can ask two natural questions.

First, when does this operation have a square root, i.e., does there exist a function $s : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ for which $s(s(a)) = \neg(a)$?

Second, when such a square root exists, how to describe all such square roots?

First observation: $s(x)$ is a permutation. To answer these two questions, let us first notice the function s must be a permutation, i.e., a 1-1 onto mapping from the set of truth values onto itself. Indeed, if $i \neq j$, then we cannot have $s(i) = s(j)$: otherwise, we would have $s(s(i)) = s(s(j))$, but

$$s(s(i)) = n + 1 - i \neq n + 1 - j = s(s(j)).$$

So, two different value $i \neq j$ lead to two different values $s(i) \neq s(j)$. Thus, n different truth values $1, \dots, n$ lead to n different values $s(i)$. Since there are exactly n different truth values, every truth values must be represented as $s(i)$ for some i – i.e., $s(x)$ is indeed a permutation.

Analysis of the problem. Let us now start with an arbitrary truth value i and consequently apply the operation s . First, we get some value $j \stackrel{\text{def}}{=} s(i)$. Then, we get the value $s(j) = s(s(i)) = n + 1 - i$. Then, we get

$$s(n + 1 - i) = s(s(j)) = n + 1 - j.$$

Finally, we get

$$s(n + 1 - j) = s(s(n + 1 - i)) = \neg(n + 1 - i) = n + 1 - (n + 1 - i) = i,$$

and thus, i again. So, we get a cycle of four elements:

$$i \rightarrow j \rightarrow n + 1 - i \rightarrow n + 1 - j \rightarrow i.$$

Can this be a smaller cycle, i.e., can some elements from this cycle be equal to each other?

If i is equal to j , then we have $n + 1 - i = i$, i.e., $i = (n + 1)/2$. In general, if two consequent elements of this cycle are equal to each other, then all the elements coincide, and we have $i = (n + 1)/2$. So, this can only happen for one truth value – the element $i = (n + 1)/2$, and this truth value is only possible if n is odd.

If i is equal to $n + 1 - i$, then we also have $i = (n + 1)/2$. In this case, $j = s(i)$ must also be equal to $n + 1 - j = s(n + 1 - i) = s(i)$, so we must have $j = n + 1 - j$ and thus, $j = (n + 1)/2$ and $j = i$.

So, with the only possible exception of the element $i_0 = (n + 1)/2$ for which $s(i_0) = i_0$, all other truth values are divided – by the action of the function $s(x)$ – into cycles consisting of 4 elements each. So, if there is no fixed-point element $s(i_0) = i_0$, then the total number of truth values must be divisible by 4.

If there is a fixed-point truth value, then the number of truth values must be equal to $4k + 1$ for some k .

In general, it is sufficient to define $s(i)$ for values $i \leq n/2$: once we know $s(i)$ for all such i , we can then compute $s(n + 1 - i) = n + 1 - j$. For each $i \leq n/2$, either $j = s(i) \leq n/2$ or $n + 1 - j \leq n/2$. Thus, to each $i \leq n/2$, we can put into correspondence another value (j or $n + 1 - j$) which is $\leq n/2$ and which belongs to the same cycle. So, each function $s(x)$ divides the truth values $\leq n/2$ into pairs.

Once the division into pairs is fixed, for each pair (i, k) , we have either $s(i) = k$ or $s(i) = n + 1 - k$.

- When $s(i) = k$, we have $s(k) = \neg(i) = n + 1 - i$.
- When $s(i) = n + 1 - k$, we have

$$s(k) = s(\neg(s(i))) = s(s(s(s(i)))) = \neg(\neg(i)) = i.$$

So, once $n = 4k$ or $n = 4k + 1$, we can construct the desired square root of negation.

Resulting answers to the above two questions. Thus, we arrive at the following answer to the first question: a square root of negation is possible if and only if $n = 4k$ or $n = 4k + 1$ for some integer k . So, for $n \geq 2$, square roots of negation are only possible for $n = 4$, $n = 5$, $n = 8$, $n = 9$, etc. The simplest possible case is the case $n = 4$; this is the discrete analogue of the above 4-piece discontinuous square root of negation operation.

We have also shown how to describe all possible operations: we divide all the elements from 1 to $n/2$ into pairs, and then select, for each pair (i, k) , whether $s(i) = k$ or $s(i) = n + 1 - k$. For $i > n/2$, we can then take

$$s(i) = \neg(s(\neg(i))) = n + 1 - s(n + 1 - i).$$

Indeed, since $\neg(a) = s(s(a))$, we get

$$\neg(s(\neg(i))) = s(s(s(s(s(i))))) = \neg(\neg(s(i))) = s(i).$$

Example 1. For $n = 4$, we thus need to divide numbers 1 and 2 into pairs. Since

there are only two elements, there is only one possible pairs $(1, 2)$. Now, we can have $s(1) = 2$ and $s(1) = \neg(2) = 3$.

When $s(1) = 2$, then we have $s(2) = s(s(1)) = \neg(1) = 4$, $s(4) = s(s(2)) = \neg(2) = 3$, and $s(3) = s(s(4)) = \neg(4) = 1$. Similarly, we can describe the function $s(x)$ for the case when $s(1) = 3$. So, we get the following two operations s :

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1,$$

and

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1.$$

These examples correspond to the above static and dynamic scenarios in which we also had four different truth values: $F < f < t < T$.

Example 2. For $n = 5$, we also have only one subdivision, so we also get two operations:

$$1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 1; \quad 3 \rightarrow 3;$$

and

$$1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 1 \quad 3 \rightarrow 3.$$

These are discrete analogues of the above 4-piece discontinuous square root of negation operation.

Example 3. For $n = 8$, we need to divide elements 1, 2, 3, and 4 into pairs. Once we pick a companion for 1, the remaining two elements also form a pair. So, by associating 1 with 2, 3, and 4, we get three possible subdivision into pairs:

$$(1, 2), (3, 4); \quad (1, 3), (2, 4); \quad (1, 4), (2, 3).$$

For each subdivision, we have $2^2 = 4$ possible operations, so we have a total of 12 possible square roots of negation operations:

$$1 \rightarrow 2 \rightarrow 8 \rightarrow 7 \rightarrow 1; \quad 3 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 3;$$

$$1 \rightarrow 2 \rightarrow 8 \rightarrow 7 \rightarrow 1; \quad 3 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 3;$$

$$1 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 1; \quad 3 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 3;$$

$$1 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 1; \quad 3 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 3;$$

$$1 \rightarrow 3 \rightarrow 8 \rightarrow 6 \rightarrow 1; \quad 2 \rightarrow 4 \rightarrow 7 \rightarrow 5 \rightarrow 2;$$

$$1 \rightarrow 3 \rightarrow 8 \rightarrow 6 \rightarrow 1; \quad 2 \rightarrow 5 \rightarrow 7 \rightarrow 4 \rightarrow 2;$$

$$1 \rightarrow 6 \rightarrow 8 \rightarrow 3 \rightarrow 1; \quad 2 \rightarrow 4 \rightarrow 7 \rightarrow 5 \rightarrow 2;$$

$$1 \rightarrow 6 \rightarrow 8 \rightarrow 3 \rightarrow 1; \quad 2 \rightarrow 5 \rightarrow 7 \rightarrow 4 \rightarrow 2;$$

$$1 \rightarrow 4 \rightarrow 8 \rightarrow 5 \rightarrow 1; \quad 2 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 2;$$

$$1 \rightarrow 4 \rightarrow 8 \rightarrow 5 \rightarrow 1; \quad 2 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 2;$$

$$1 \rightarrow 5 \rightarrow 8 \rightarrow 4 \rightarrow 1; \quad 2 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 2;$$

$$1 \rightarrow 5 \rightarrow 8 \rightarrow 4 \rightarrow 1; \quad 2 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 2.$$

How many possible operations are there? We start with $k = \lfloor n/2 \rfloor$ elements. First, we find a pair for the element 1; there are $k - 1$ options for such pairing.

After this pairing, we have $k - 2$ un-paired elements remaining. We pick the first un-paired element; there are $k - 3$ possible pairs for this element, so we end up with $(k - 1) \cdot (k - 3)$ possible combinations.

After we have formed 2 pairs, we have $k - 4$ remaining elements. We pick the first un-paired one; for this element, there are $k - 5$ possible pairing, so we have $(k - 1) \cdot (k - 3) \cdot (k - 5)$.

Overall, we thus have $(k - 1) \cdot (k - 3) \cdot (k - 5) \cdot \dots$ possible pairings.

For each subdivision into $k/2$ pairs, within each pair, we have 2 options, to the total of $2^{k/2}$. Thus, overall, we have

$$(k - 1) \cdot (k - 3) \cdot (k - 5) \cdot \dots \cdot 2^{k/2}$$

different square roots of negation.

Examples. For $n = 4$, we have $k = 2$, so we have $(k - 1) \cdot 2^{k/2} = 1 \cdot 2^1 = 2$ possible square roots of negation. For $n = 5$, we also have $k = \lfloor n/2 \rfloor = 2$, so we also have two possible square roots of negation.

For $n = 8$, we have $k = 4$, so we have $(k - 1) \cdot (k - 3) \cdot 2^{k/2} = 3 \cdot 1 \cdot 2^2 = 12$ possible square roots of negation – which are listed above.

Asymptotic number of operations: an interesting observation. It is known that asymptotically, $n! = 1 \cdot 2 \cdot \dots \cdot n \approx \left(\frac{n}{e}\right)^n$. We would like to use this formula to estimate the number of square roots of negation. To do that, let us supplement the product $(k - 1) \cdot (k - 3) \cdot \dots$ by missing integers, i.e., write it down as

$$(k - 1) \cdot (k - 3) \cdot \dots = \frac{k \cdot (k - 1) \cdot (k - 2) \cdot \dots}{k \cdot (k - 2) \cdot (k - 4) \cdot \dots}.$$

The numerator of the resulting fraction is $k!$. Dividing each element of the denominator by 2, we can describe it as

$$k \cdot (k - 2) \cdot (k - 4) \cdot \dots = 2^{k/2} \cdot (k/2) \cdot (k/2 - 1) \cdot \dots = 2^{k/2} \cdot (k/2)!.$$

Thus,

$$(k-1) \cdot (k-3) \cdot \dots = \frac{k!}{2^{k/2} \cdot (k/2)!}.$$

By using the above asymptotic formula for the factorial function, we now conclude that

$$(k-1) \cdot (k-3) \cdot \dots \approx \frac{\left(\frac{k}{e}\right)^k}{2^{k/2} \cdot \left(\frac{k/2}{e}\right)^{k/2}} = \frac{k^k \cdot e^{k/2}}{e^k \cdot k^{k/2}} = \frac{k^{k/2}}{e^{k/2}} = \left(\frac{k}{e}\right)^{k/2} \approx \left(\frac{n}{2e}\right)^{n/4}.$$

The overall number of functions $s : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ is equal to n^n : we can assign n different values to each of n possible inputs. Thus, asymptotically, the total number of square-root-of-not operations is equal to the 4-th order root – square root of the square root – of the total number of all possible operations.

Limit $n \rightarrow \infty$. In the above example, we had 4 zones of continuity. We can use discrete operations of order $n > 4$ and get more such zones.

In the limit $n \rightarrow \infty$, we get operations which are everywhere discontinuous like fractals.

19. Geometric Similarity Between Quantum and Fuzzy Negations

We have already mentioned that in quantum mechanics, the most widely used negation operation is, in effect, rotation by 90 degrees. Thus, the square root of negation can be naturally defined as rotation by 45 degrees.

It is worth mentioning that the square root $s(x)$ of fuzzy negation can also be naturally described in terms of rotation; see, e.g., Guintini *et al.* (2007). Indeed, in mathematics, a function $s : [0, 1] \rightarrow [0, 1]$ is defined as a set $s \subseteq [0, 1] \times [0, 1]$ of all possible pairs $(x, s(x))$. The fact that a function is a square root of negation means that if $y = s(x)$, then $s(y) = 1 - x$. In other words, if $(x, y) \in s$, then $(y, 1 - x) \in s$. One can easily check that the transformation $(x, y) \rightarrow (y, 1 - x)$ is a rotation by 90 degrees around the central point $(1/2, 1/2)$. Indeed, if we use the central point as the origin of the new coordinate system, then in the new coordinates $x' \stackrel{\text{def}}{=} x - 1/2$ and $y' \stackrel{\text{def}}{=} y - 1/2$ the above transformation takes the form $(x', y') \rightarrow (y', -x')$, i.e., it is indeed rotation by 90 degrees.

In particular, this operation turns the vertices of the unit square $[0, 1] \times [0, 1]$ into each other:

$$(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (1, 0) \rightarrow (0, 0).$$

Thus, a fuzzy logic operation is a square root of negation if and only if it is invariant relative to this rotation.

Please notice that while both square root operations are related to rotation, the relation is different:

- in quantum logic, both negation and its square root *are* rotations;
- in fuzzy logic, a square root operation is *invariant* under rotation.

Comment. Please notice that fuzzy negation $\neg(x) : [0, 1] \rightarrow [0, 1]$ defined as $\neg(x) =$

$1 - x$ also has a geometric interpretation: it is a reflection relative to the point $x = 1/2$.

20. Analytical similarity between functions in quantum and fuzzy logics

Another similarity between fuzzy and quantum logics comes from the way functions are extended from real to fuzzy or quantum inputs.

In the fuzzy case, a classical function $f(x)$ is extended to a fuzzy input with membership function $\mu(x)$ by using Zadeh’s extension principle (see, e.g., Klir and Yuan (1995)):

$$\mu_f(y) = \max\{\mu(x) : f(x) = y\}.$$

In particular, if we have several values x with the same value $f(x)$, then the corresponding membership values $\mu(x)$ are combined by a maximum operation.

To describe how functions $f(x)$ can be extended to the quantum case (Nielsen and Chuang (2000)), we must take into account that in the computer, every object is represented as a sequence of 0s and 1s. Thus, without losing generality, we can assume that for every input x , the value $f(x)$ is a finite sequence of bits (0s and 1s). Let n denote the length of this sequence. A quantum analogue of the classical function $f(x)$ is then defined on the pairs (x, y) , where y is an auxiliary n -bit sequence, as $f(x, y) = (x, y \oplus f(x))$, where \oplus is a bit-wise exclusive “or” operation – i.e., addition modulo 2. For a general quantum state $\sum \alpha_i \cdot (x_i, y_i)$, the quantum analogue of $f(x)$ is defined as follows:

$$\sum \alpha_i \cdot (x_i, y_i) \rightarrow \sum \alpha_i \cdot (x_i, y_i \oplus f(x_i)).$$

So, if there are several different values x with the same value $f(x)$, the corresponding quantum values α_i are combined by addition.

Thus, the quantum extension of functions is similar to Zadeh’s (fuzzy) extension principle, with the main difference that

- in quantum logic, we use addition to combine different values; while
- in fuzzy logic, we use maximum for similar combination purposes.

21. Conclusions and Future Work

What do the results of the paper mean for both logics?

Our main point is that in spite of the seeming similarity between the two logics, they are different. Moreover, these logics are different in an important feature (square root of “not”) which is crucial for the most impressive applications of quantum logic – to the drastic computation speed-up.

This difference is not unexpected. After all, fuzzy logic is a human way of reasoning about the real-world phenomena. Most real-world phenomena are well described by classical physics, so it is not surprising that our way of reasoning about these phenomena is well-suited for classical physics, but not for the quantum physics.

Our auxiliary construction means that if we add some non-classical (quantum) features into fuzzy logic, then we can emulate very intuitively unusual quantum features such as the square root of “not”. Specifically, we show that in order to be able to represent a square root of “not” in fuzzy logic, it is sufficient to add discontinuity to this logic. Discontinuity is one of the original phenomena which

characterized quantum phenomena – where, e.g., an atom in an excited state, instead of continuously decreasing its energy, decreases it abruptly, by emitting a quantum of energy – a photon.

It is worth mentioning that discontinuity seems to be opposite to the main idea behind fuzzy logic: that everything is a matter of degree, and that every seemingly discontinuous transition is actually continuous.

This auxiliary construction raises the possibility that by combining such empirically clear quantum phenomena as discontinuity with the main intuitively clear ideas behind fuzzy logic, we can get a better explanation of very technical (and somewhat counter-intuitive) quantum phenomena such as the square root of “not”.

Acknowledgments

This work was supported in part by NSF grants HRD-0734825, EAR-0225670, and EIA-0080940, by Texas Department of Transportation contract No. 0-5453, by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and by the Max Planck Institut für Mathematik.

The authors are thankful to all the participants of the 27th International Conference of the North American Fuzzy Information Processing Society NAFIPS’2008 (New York, New York, May 19–22, 2008) for valuable suggestions, especially to Rudolf Seising and Mark Wierman. The authors are also grateful to the anonymous referees for their useful advice.

References

- Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, S., 2001. *Introduction to Algorithms*, Cambridge, MA:MIT Press.
- Dalla Chiara, M.L., Giuntini, G., and Greechie, R., 2004. *Reasoning in Quantum Theory: Sharp and Unsharp Quantum Logics*, Dordrecht:Kluwer.
- Dalla Chiara, M.L., Giuntini, G., and Leporini, R., 2005. Logics from Quantum Computation. *International Journal of Quantum Information*, 3, 293–337.
- Grover, L., 1996. A fast quantum mechanical algorithm for database search, *Proc. 28th ACM Symp. on Theory of Computing*, 212–219.
- Grover, L.K., 1997. Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.*, 79(2), 325–328.
- Grover, L.K., 1998. A framework for fast quantum mechanical algorithms, *Phys. Rev. Lett.* 80, 4329–4332.
- Giuntini, R., Ledda, A., and Paoli, F., 2007. Expanding Quasi-MV Algebras by a Quantum Operator, *Studia Logica*, 87, 99–128.
- Klir, G., and Yuan, B., 1995. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ:Prentice Hall.
- Kosko, B., 1993. *Fuzzy Thinking: The New Science of Fuzzy Logic*, New York:Hyperion.
- Nguyen, H.T., and Walker, E.A., 2006. *First Course in Fuzzy Logic*, Boca Raton, FL:CRC Press.
- Nielsen, M.A., and Chuang, I.L., 2000. *Quantum computation and quantum information*, Cambridge, U.K.:Cambridge University Press.
- Seising, R., 2008. From principles of mechanics to quantum mechanics – a survey on fuzziness in scientific theories. *Proceedings of the 27th International Conference of the North American Fuzzy Information Processing Society NAFIPS’2008*, New York, New York, May 19–22, 2008.

Shor, P., 1994. Algorithms for quantum computation: Discrete logarithms and factoring, *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 124–134.