Normal forms and truth tables for fuzzy logics

Mai Gehrke Carol L. Walker Elbert A. Walker

{mgehrke, hardy, elbert}@nmsu.edu

Abstract

In this paper, we examine and compare De Morgan-, Kleene-, and Boolean-disjunctive and conjunctive normal forms and consider their role in fuzzy settings. In particular, we describe normal forms and truth tables for classical fuzzy propositional logic and interval-valued fuzzy propositional logic that are completely analogous to those for Boolean propositional logic.

Thus, determining logical equivalence of two expressions in fuzzy propositional logic is a finite problem, and similarly for the interval-valued case. Turksen's work on interval-valued fuzzy sets is reviewed in light of these results.

Keywords: fuzzy connectives, non-classical logics, fuzzy switching functions, continuous t-norm, algebra

1 Introduction

This paper is expository—the principal mathematical results appear already in the literature. See, for example, [2, 3, 4, 5, 8, 10, 11, 14, 15, 16]. However, some of this mathematics does not seem to be well known within the fuzzy community. These results can be gotten in short order by calling upon some standard theorems in universal algebra. We make a strong effort here to give a presentation that is as elementary as possible, with a leisurely and detailed exposition, to make the results readily accessible to a general audience. Our treatment also sheds some mathematical light on some work in [18, 19, 20, 21, 22, 23, 24, 25].

The notion of a normal form, or canonical form, pervades mathematics. This is the identification of a unique selection from each equivalence class of an equivalence relation together with a method of finding that selection, given an arbitrary element of an equivalence class. A mundane example is writing a rational number in lowest form—that is, with numerator and denominator relatively prime. Different equivalence relations on matrices lead to some well-known examples such as the rational and Jordan canonical forms. A normal form allows us, in particular, to check whether two expressions are equivalent.

We are interested in normal forms for algebras arising in classical and fuzzy logics, including Boolean, Kleene, and De Morgan algebras. The classical situation is this. Let $\mathbb{B} = (B, \wedge, \vee, ', 0, 1)$ be a Boolean algebra. We are given two Boolean polynomials such as

$$p(x, y, z) = (x \vee y) \wedge z'$$

and

$$q(x, y, z) = (x \wedge y \wedge z) \vee \left((x \wedge y)' \wedge z'\right) \vee (x \wedge y \wedge z')$$

where the operations $\vee$, $\wedge$, and $'$ are the usual Boolean operations of join, meet, and complement, and we want to determine whether or not they are equivalent over $\mathbb{B}$. The question is the following. For *any* elements $a, b, c \in B$, is it true that $p(a, b, c) = q(a, b, c)$? That is what equivalence of the two expressions over $\mathbb{B}$ means. There are two standard ways to answer this. One way is to write each polynomial in its disjunctive normal form, and note whether or not these two forms are identical (similarly for conjunctive normal forms). Another is to test the equality of the two polynomials on just the elements from the two element Boolean algebra. This is usually done using *truth tables* for the two-element Boolean algebra. Equality for that Boolean algebra implies equality for *any* Boolean algebra, a rather startling fact, but quite well known. And this holds for any finite number of variables x, y, $z, \ldots$ and any such expressions. In asking about the equivalence of two expressions for Boolean algebras, one has only to deal with the two-element Boolean algebra, which in particular is finite.

In the (classical) fuzzy situation, one deals with the algebra

$$\mathbb{I} = ([0, 1], \wedge, \vee, ', 0, 1).$$

where $\wedge$ and $\vee$ are min and max respectively, and $'$ is given by $x' = 1 - x$ for $x \in [0, 1]$. Consider the polynomials p and q above in this situation. Are they equivalent when considered as expressions with variables and operations from this algebra? That is, is $p(a, b, c) = q(a, b, c)$ for elements $a, b, c \in [0, 1]$? How can we tell? We might find some values of a, b, c for which they are not equal and that would settle it, but failing that, offhand we have an infinite number of values to check.

In the interval-valued fuzzy situation, the algebra of concern is the set $\{(a, b) : a, b \in [0, 1], a \leq b\}$ with component-wise operations $\wedge$ and $\vee$ of min and max, respectively, and negation $'$ given by $(a, b)' = (1 - b, 1 - a)$. Thus we have the algebra

$$\mathbb{I}^{[2]} = (\{(a, b) : a, b \in [0, 1], a \leq b\}, \wedge, \vee, ', 0, 1).$$

Determining the equivalence of the polynomials p and q in this situation might seem even more complicated.

The main purpose of this paper is to show that in both the classical fuzzy case and in the interval-valued fuzzy case, these questions are easily handled. In each case, one may as well work over a finite algebra, just as in the Boolean case. In the Boolean case, it sufficed to consider only the two-element Boolean algebra. It turns out that in the fuzzy case, there is an analogous three-element algebra, and in the interval-valued fuzzy case, there is an analogous four-element algebra. This at least reduces the problem of determining the equivalence of polynomials such as those above to a finite procedure. But there are totally analogous normal forms as well as truth tables in each case.

Further, this says something about the algebra of all fuzzy subsets of a set. Let X be any set and $\mathcal{F}(X)$ be the algebra of all mappings of X into $[0,1]$. Operations $\wedge$, $\vee$, and $'$ on $\mathcal{F}(X)$ are given by $(A \wedge B)(x) = \min\{(A(x), B(x)\}$, $(A \vee B)(x) = \max\{(A(x), B(x)\}$, and $A'(x) = 1 - A(x)$. So we have the algebra $\mathcal{F}(X) = ([0,1]^X, \wedge, \vee, ', 0, 1)$, and ask the same question. Are the polynomials p and q above equivalent when considered as expressions with variables and operations from this algebra? This question reduces immediately to the fuzzy situation. For fuzzy sets A, B, and C, $p(A,B,C) = q(A,B,C)$ means that $p(A,B,C)(x) = q(A,B,C)(x)$ for all $x \in X$, which means that $p(A(x), B(x), C(x)) = q(A(x), B(x), C(x))$. This in turn means that $p(a,b,c) = q(a,b,c)$ for all $a, b, c \in [0,1]$, which reduces to determining equality when a, b, and c are elements of a certain three-element algebra. So deciding whether two expressions with fuzzy sets as variables are equivalent is a finite problem, and can be handled algorithmically via normal forms or truth tables.

These facts can be put in terms of propositional logic. Classical Boolean propositional logic is based on the two-element Boolean algebra, the *algebra of truth values* for this logic. The same propositional logic results if the algebra of truth values is taken to be *any* Boolean algebra with two or more elements. Fuzzy propositional logic results when the algebra of truth values is taken to be $\mathbb{I}$. But the same propositional logic results when the truth value algebra is the three-element chain with the usual lattice operations, and the obvious $'$. Analogous statements hold for interval-valued fuzzy logic and a certain four-element algebra. These are the principal facts we will develop.

2 Some propositional logics

First we spell out exactly the mathematical framework for our discussion. We will deal with algebras $\mathbb{A} = (A, \wedge, \vee, ', 0, 1)$, where $\wedge$ is a binary operation on the set A, typically called "meet", or "intersection", or "conjunction", $\vee$ is a binary operation on A typically called "join", or "union", or "disjunction", $'$ is a unary operation, which will be referred to as "not", and 0 and 1 are constants, or nullary operations. So the algebra $\mathbb{A}$ is of *type* $(2,2,1,0,0)$. We now build a certain such algebra, called the *term algebra,* which is basic in all that follows.

Let V be a non-empty set, the set of *variables*. Using the operation symbols $\wedge, \vee, ', 0, 1$, form polynomials in the usual manner using the variables, and parentheses. Such a polynomial will be called a **term**. The set T of all these polynomials, or terms, is also called the set of *well-formed formulas*. For example, if x, y, z are variables, then $x \vee y$, $(x \wedge z') \vee x$, and y' are terms, but not $xy' \vee \wedge' z$. Also 0 and 1 are terms, as are $0'$, $1'$, $x \vee 0'$ and so forth. The set T with its connectives is an algebra. For example, if $s, t \in T$, then $s \vee t$ is a term, and similarly for the other operations. This yields the algebra

$$\mathbb{T} = (T, \wedge, \vee, ', 0, 1),$$

called the *term algebra* with variables V and operations $\wedge, \vee, ', 0, 1$. It is again of type $(2, 2, 1, 0, 0)$. The term algebra is infinite, even when V is finite.

Now let $\mathbb{A} = (A, \wedge, \vee, ', 0, 1)$ be any algebra of the same type. (We will consistently use the same symbols $\wedge, \vee, ', 0, 1$ for our operations since all our algebras will be of the same type.) For any mapping $v : V \to A$, there is a unique homomorphism $\widetilde{v} : \mathbb{T} \to \mathbb{A}$ agreeing with v on V. For example, for the term $(x \wedge z') \vee x$, $\widetilde{v}((x \wedge z') \vee x) = (v(x) \wedge (v(z))') \vee v(x)$. In building a logic, we talk of $\mathbb{A}$ as the *truth value* algebra, and call the maps $\widetilde{v}$ *truth evaluations*.

In the set T, we say that two terms s and t are *logically equivalent* if $\widetilde{v}(s) = \widetilde{v}(t)$ for all truth evaluations $\widetilde{v}$. This is the same thing as saying that $\varphi(s) = \varphi(t)$ for all homomorphisms $\varphi : \mathbb{T} \to \mathbb{A}$. If this is the case, we write $s \equiv_{\mathbb{A}} t$. The relation $\equiv_{\mathbb{A}}$ is an equivalence relation, but in fact is also a *congruence*. This means that the set of equivalence classes of this equivalence relation forms in a natural way an algebra of type $(2, 2, 1, 0, 0)$, the same type as $\mathbb{T}$. If $[s]$ and $[t]$ denote the equivalence classes containing the terms s and t, respectively, then $[s] \wedge [t] = [s \wedge t]$ is well defined, and similarly for the other operations. The resulting quotient algebra $\mathbb{T}/\equiv_{\mathbb{A}}$, which we will denote by $\mathbb{L}_{\mathbb{A}}$, is the *propositional logic* in the variables V with truth values in the algebra $\mathbb{A}$. It is also called the Tarski-Lindenbaum algebra.

In all of the cases we consider, the algebra $\mathbb{A}$ is a *De Morgan algebra*—that is, a distributive lattice $(A, \vee, \wedge, 0, 1, ')$ of the type $(2, 2, 0, 0, 1)$ satisfying identities

$$\begin{aligned} (x \vee y)' &= x' \wedge y' \\ (x \wedge y)' &= x' \vee y' \\ x'' &= x \end{aligned}$$

Some are also *Kleene algebras*—that is, De Morgan algebras satisfying

$$x \wedge x' \leq y \vee y'$$

for all x, y. And some are *Boolean algebras*—De Morgan algebras satisfying identities

$$x \wedge x' = 0 \text{ and } x \vee x' = 1$$

The logics $\mathbb{L}_{\mathbb{A}}$ are again De Morgan, Kleene, or Boolean algebras, respectively, according to the properties of $\mathbb{A}$. In fact, $\mathbb{L}_{\mathbb{A}}$ is the free De Morgan, Kleene, or Boolean algebra, respectively, on the set V of variables.

There are six principal cases of interest to us:

1. $\mathbb{B} = (B, \wedge, \vee, ', 0, 1)$, any Boolean algebra.

2. $\mathbf{2} = (\{\mathbf{0}, \mathbf{1}\}, \wedge, \vee, ', 0, 1)$, the two element Boolean algebra.

3. $\mathbb{I} = ([0, 1], \wedge, \vee, ', 0, 1)$, the algebra of truth values for classical fuzzy logic.

4. $\mathbf{3} = (\{0, u, 1\}, \wedge, \vee, ', 0, 1)$, where $\{0, u, 1\}$ is the lattice

$$\begin{array}{c} 1 \\ \uparrow \\ u \\ \uparrow \\ 0 \end{array}$$

   and $\wedge, \vee$ are the usual lattice operations, and $'$ fixes u and interchanges 0 and 1.

5. $\mathbb{I}^{[2]} = (\{(a, b) : a, b \in [0, 1], a \leq b\}, \wedge, \vee, ', 0, 1)$, the algebra of truth values for interval-valued fuzzy logic.

6. $\mathbb{D} = (D, \wedge, \vee, ', 0, 1)$, where D is the lattice pictured below.

$$\begin{array}{ccccc} & & 1 & & \\ & \nearrow & & \nwarrow & \\ u & & & & v \\ & \nwarrow & & \nearrow & \\ & & 0 & & \end{array}$$

   and $\wedge$ and $\vee$ are the lattice operations, and $'$ interchanges 0 and 1 and *fixes* u and v. This algebra is called *diamond*.

Forming $\mathbb{L}_{\mathbb{A}}$ for various algebras of truth values $\mathbb{A}$ gives various algebras, or propositional logics $\mathbb{L}_{\mathbb{A}}$, but *different truth value algebras can give the same propositional logic*. In fact, in the list above

$$\begin{aligned} \mathbb{L}_{\mathbb{B}} &= \mathbb{L}_{\mathbf{2}} \\ \mathbb{L}_{\mathbb{I}} &= \mathbb{L}_{\mathbf{3}} \\ \mathbb{L}_{\mathbb{I}^{[2]}} &= \mathbb{L}_{\mathbb{D}} \end{aligned}$$

Again, exactly what does this mean? It means that two polynomials, or expressions, or terms, that is, two elements in T, are logically equivalent over a Boolean algebra $\mathbb{B}$ if and only if they are logically equivalent over the two element Boolean algebra. They are logically equivalent where the algebra of truth values is the unit interval with the standard operations of fuzzy logic if and only if they are logically equivalent over the three element Kleene algebra **3.** The analogous statement holds for interval-valued fuzzy logic and the logic given by the four element algebra $\mathbb{D}$ of truth values.

And as pointed out, it means that it is a finite procedure to determine the logical equivalence of two expressions in fuzzy sets, with the usual connectives.

The first step in establishing these equalities is the following quite easy, but general, proposition.

Definition 1 *Let $\mathbb{G}$ and $\mathbb{H}$ be algebras of the same type. If for every pair of distinct elements x and y of the algebra $\mathbb{G}$, there is a homomorphism $\beta : \mathbb{G} \rightarrow \mathbb{H}$ such that $\beta(x) \neq \beta(y)$, then $\mathbb{H}$* ***separates points of*** *$\mathbb{G}$.*

Proposition 2 *If $\mathbb{H}$ separates points of $\mathbb{G}$, then $s \equiv_{\mathbb{H}} t$ implies that $s \equiv_{\mathbb{G}} t$.*

Proof. Suppose that $\mathbb{H}$ separates points of $\mathbb{G}$ and that $s \equiv_{\mathbb{G}} t$ does not hold. Then there is a homomorphism $\varphi : \mathbb{T} \rightarrow \mathbb{G}$ such that $\varphi(s) \neq \varphi(t)$ and a homomorphism $\beta : \mathbb{G} \rightarrow \mathbb{H}$ such that $\beta\varphi(x) \neq \beta\varphi(y)$. The homomorphism $\beta\varphi : \mathbb{T} \rightarrow \mathbb{H}$ says that $s \equiv_{\mathbb{H}} t$ does not hold. ■

Corollary 3 *If $\mathbb{H}$ separates points of $\mathbb{G}$ and $\mathbb{G}$ separates points of $\mathbb{H}$, then $\mathbb{L}_{\mathbb{G}} = \mathbb{L}_{\mathbb{H}}$.*

Corollary 4 *Let $\mathbb{B} = \prod_{x \in X} \mathbb{A}$ be a product of copies of the algebra $\mathbb{A}$. Then $\mathbb{L}_{\mathbb{A}} = \mathbb{L}_{\mathbb{B}}$.*

Proof. Mapping $\mathbb{A}$ to the diagonal of $\prod_{x \in X} \mathbb{A}$ says that $\prod_{x \in X} \mathbb{A}$ separates points of $\mathbb{A}$. If $x, y \in \prod_{x \in X} \mathbb{A}$**,** and $x \neq y$, then some component of x is not equal to that component of y. Project $\prod_{x \in X} \mathbb{A}$ to $\mathbb{A}$ along that component. Thus $\mathbb{A}$ separates points of $\prod_{x \in X} \mathbb{A}$**.** ■

Corollary 5 *Let $\mathcal{F}(X) = \left([0,1]^X, \wedge, \vee, \prime, 0, 1\right)$ be the algebra of all fuzzy subsets of the set X. Then $\mathbb{L}_{\mathcal{F}(X)} = \mathbb{L}_{\mathbb{I}}$.*

Proof. $\mathcal{F}(X)$ is the product of copies of the algebra $\mathbb{I}$**,** one copy for each element of X. ■

An analogous corollary may be stated for interval-valued fuzzy sets.

Note that if $\mathbb{H}$ is isomorphic to a subalgebra of $\mathbb{G}$, then $\mathbb{G}$ separates points of $\mathbb{H}$. There are theorems spelling out exactly when $\mathbb{L}_{\mathbb{G}} = \mathbb{L}_{\mathbb{H}}$, but they go beyond what we need here.

Every Boolean algebra $\mathbb{B}$ contains the two element Boolean algebra, so to show that $\mathbb{L}_{\mathbb{B}} = \mathbb{L}_{\mathbf{2}}$, it suffices to show that for x and y different elements in B, there is a homomorphism $\mathbb{B} \rightarrow \mathbf{2}$ taking x and y to different elements in **2,** that is, that **2** separates points in $\mathbb{B}$. This follows from M. H. Stone's prime filter theorem, since homomorphisms to **2** are in one-to-one correspondence to prime filters of the Boolean algebra. A proof of the prime filter theorem can be found in [17].

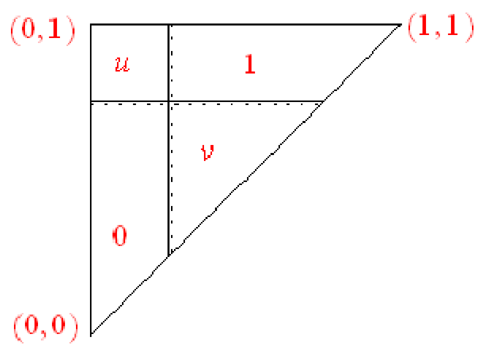
Consider now the claim that $\mathbb{L}_{\mathbb{I}} = \mathbb{L}_{\mathbf{3}}$. The algebra $\mathbb{I}$ clearly contains a copy of **3,** namely the three element subalgebra $\{0, 1/2, 1\}$. So we need to show that **3** separates points of $\mathbb{I}$. Let $a \in [0,1]$ with $0 < a \leq 1/2$. Then $\mathbb{I} \rightarrow \mathbf{3}$ with $[0, a)$ going to 0, $[a, 1-a]$ going to u, and $(1-a, 1]$ going to 1 is a homomorphism. Now, given $x, y \in [0,1]$ with $x \neq y$, the element a can be picked so that x and y go to different places in **3.** So we have

Theorem 6 *Fuzzy propositional logic* $\mathbb{L}_{\mathbb{I}}$ *is the same as* $\mathbb{L}_{\mathbf{3}}$*, the three-valued Kleene propositional logic.*

The algebra $\mathbb{I}^{[2]}$ contains a copy of $\mathbb{D}$, namely the subalgebra

$$\{(0,0),(0,1),(1/2,1/2),(1,1)\}$$

So we need to show that $\mathbb{D}$ separates points of $\mathbb{I}^{[2]}$. Consider the pictures in Figure 2.



These pictures describe a family of homomorphisms from $\mathbb{I}^{[2]}$ to $\mathbb{D}$ as follows For each point on the line $1-x$, $0 \leq x \leq 1$, draw the horizontal and vertical lines through the point. This will divide the triangle into three or four regions, depending on whether the point is below or above the diagonal. Map the points in these regions as indicated. It is an easy exercise to check that each of these maps is an algebra homomorphism, that is, preserves meets, joins, negations, 0 and 1. These homomorphisms separate points because, given two different points in the triangle you can pick a point on the line $1-x$ that gives horizontal and vertical lines separating the points. Thus we have

Theorem 7 *Fuzzy interval-valued propositional logic* $\mathbb{L}_{\mathbb{I}}$ *is the same as* $\mathbb{L}_{\mathbb{D}}$*, the four-valued diamond propositional logic.*

Let V be the set of variables and $\mathbb{I}$ the algebra of truth values. Given two elements s and t in $\mathbb{T}$, how can we tell whether or not they are logically equivalent in fuzzy propositional logic? For every map $v : V \to [0,1]$, we must determine whether or not the extension $\widetilde{v}$ of v to a homomorphism $\mathbb{T} \to \mathbb{I}$ satisfies $\widetilde{v}(s) = \widetilde{v}(t)$. There are infinitely many such maps v. But by Theorem 1, it suffices to make this determination when the algebra of truth values is **3.** Since any element of $\mathbb{T}$ involves only finitely many variables, this is a finite problem, and indeed may be accomplished using truth tables for **3**.

For the case of two variables $\{x, y\}$, for example, for $s = (x \vee y) \wedge x'$ and $t = x' \wedge \left((x' \vee y)' \vee y\right)$, the truth values shown in Table 1 are showing that s and t are

x	y	$(x \vee y) \wedge x'$	$x' \wedge \left((x' \vee y)' \vee y\right)$
0	0	0	0
0	u	u	u
0	1	1	1
u	0	u	u
u	u	u	u
u	1	u	u
1	0	0	0
1	u	0	0
1	1	0	0

Table 1:

logically equivalent in fuzzy propositional logic. Note that for n variables, a truth table for **3** has 3^n rows.

For interval-valued propositional logic, it suffices to determine whether or not for every map $v : V \to \mathbb{D}$, the extension $\widetilde{v}$ of v to a homomorphism $\mathbb{T} \to \mathbb{D}$ satisfies $\widetilde{v}(s) = \widetilde{v}(t)$. Again, since any element of $\mathbb{T}$ involves only finitely many variables, this is a finite problem that may be accomplished using truth tables for $\mathbb{D}$. For n variables, a truth table for $\mathbb{D}$ has 4^n rows.

One can take for the algebra of truth values the algebra $\mathcal{F}(X) = \left([0,1]^X, \wedge, \vee, '\right)$ of all fuzzy subsets of a set X and conclude things about expressions in fuzzy sets themselves. This algebra is also a Kleene algebra, and the propositional logic $\mathbb{L}_{\mathcal{F}(X)}$ is the same as $\mathbb{L}_{\mathbf{3}}$.

3 Normal forms

When describing normal forms and procedures for obtaining them, we will work as above with the term algebra $\mathbb{T}$ of type $(2, 2, 1, 0, 0)$, with operation symbols $\wedge, \vee$, $', 0, 1$. We will assume that the set V of variables is finite. This makes the discussion a little easier. So a polynomial, or a term, in these variables is an element in $\mathbb{T}$. We will interpret these forms in the propositional logics of interest to us. First some definitions.

Definition 8 *A* ***De Morgan algebra*** *is a bounded distributive lattice*

$$(A, \wedge, \vee, ', 0, 1)$$

with an involution $'$ that satisfies De Morgan's laws. A ***Kleene algebra*** *is a De Morgan algebra that satisfies $x \wedge x' \leq y \vee y'$. A* ***Boolean algebra*** *is a Kleene algebra that satisfies $x \wedge x' = 0$ and $y \vee y' = 1$*

We state the definitions this way to emphasize that they are successively more restrictive. We note the following.

- $\mathbb{I}^{[2]}$ and $\mathbb{D}$ are De Morgan algebras and are not Kleene algebras.
- $\mathbb{I}$ and 3 are Kleene algebras and are not Boolean algebras.

Only slightly more difficult are the following.

Proposition 9 *$\mathbb{L}_{\mathbb{D}}$ is a De Morgan algebra and is not a Kleene algebra.*

Proof. Let x and y be distinct variables. We show that $[x \wedge x'] \leq [y \vee y']$ is false. The square brackets mean the equivalence class containing the term within. We need to show that $[x \wedge x'] \wedge [y \vee y']$ is not $[x \wedge x']$, or that $x \wedge x'$ and $(x \wedge x') \wedge (y \vee y')$ are not logically equivalent. There is a homomorphism φ taking x to $u \in D$ and taking y to $v \in D$. Then $\varphi(x \wedge x') = u$ and $\varphi((x \wedge x') \wedge (y \vee y')) = \varphi(x \wedge x') \wedge \varphi(y \vee y') = u \wedge v = 0$. That does it. ■

The following proposition is similar.

Proposition 10 *$\mathbb{L}_{\mathbf{3}}$ is a Kleene algebra and is not a Boolean algebra.*

We first deal with normal forms for diamond propositional logic, or equivalently for fuzzy interval-valued propositional logic. The problem is this. The elements of $\mathbb{L}_{\mathbb{D}}$ are equivalence classes, with logical equivalence being the equivalence relation. Pick from each equivalence class an element (its "normal form") so that its form alone tells whether or not it is a chosen element. Further, given any element $t \in T$, give an algorithm for transforming t to that normal form. So a normal form is a representative of each equivalence class and a prescription for getting that particular representative. The basis for getting normal forms for the De Morgan, Kleene, and Boolean propositional logics is the notion of join-irreducible elements in lattices, and determining those elements in those logics.

Definition 11 *An element x of a finite lattice $(L, \wedge, \vee)$ is **join-irreducible** if it is distinct from 0 and is not the join of strictly smaller elements. Dually, an element is **meet-irreducible** if it is distinct from 1 and is not the meet of strictly larger elements.*

For example, every nonzero element in **3** is join-irreducible, and in $\mathbb{D}$, every element except 0 and 1 is join-irreducible. The pertinent fact for us is the following proposition.

Proposition 12 *In a finite distributive lattice, every element is uniquely the join of pairwise incomparable join-irreducible elements.*

Proof. If an element x is not join-irreducible, then $x = x_1 \vee x_2$, with each x_i strictly smaller than x. If x_1 is not join-irreducible, write it as the join of strictly smaller elements, and so on. By the finiteness of the lattice this process stops. So $x = y_1 \vee y_2 \vee ... \vee y_n$, with the y_i join-irreducible. Throw out any y_i that is strictly smaller than some other. The result is x written as the join of pairwise incomparable join-irreducible elements.

Now to the uniqueness part. If an element x is itself join-irreducible, there is no problem Suppose that $x = x_1 \vee x_2 \vee \cdots \vee x_m = y_1 \vee y_2 \vee \cdots \vee y_n$ with the x_i pairwise incomparable join-irreducible and strictly smaller than x, and similarly for the y_i. Then

$$\begin{aligned} x_i &= x_i \wedge (y_1 \vee y_2 \vee \cdots \vee y_n) \\ &= (x_i \wedge y_1) \vee (x_i \wedge y_2) \vee \cdots \vee (x_i \wedge y_n) \end{aligned}$$

and since x_i is join-irreducible, $x_i = x_i \wedge y_j$ for some j. Thus $x_i \leq y_j$. Similarly, $y_j \leq x_k$ for some k, whence $x_i \leq x_k$. The x's are incomparable, so $i = k$ and so $x_i = y_j$. Thus each x_i is some y_j. Similarly, each y_j is some x_i, and uniqueness follows. ■

What does this have to do with normal forms for $\mathbb{L}_{\mathbb{D}}$? We are assuming that the set V of variables is finite. Every element of $t \in T$, that is, every term, is an expression in those finite number of elements of V. Now, use the axioms of a De Morgan algebra to transform t into a join of meets of the variables and their negations. The variables and their negations are called **literals**. Thus, the result is an element of T of the form $x_1 \vee x_2 \vee \cdots \vee x_n$, with the x_i distinct and each x_i a meet of distinct literals. This transformation of t does not change the element of $\mathbb{L}_{\mathbb{D}}$ that it represents. That is, t and $x_1 \vee x_2 \vee \cdots \vee x_n$ are logically equivalent. But there are only finitely many elements in T of this form, there being only finitely many variables. Thus

Proposition 13 *If the set of variables is finite, then $\mathbb{L}_{\mathbb{D}}$ is finite.*

We have observed that every element of $\mathbb{L}_{\mathbb{D}}$ is a join of meets of literals. So for an element of $\mathbb{L}_{\mathbb{D}}$ to be join irreducible, it must be a meet of literals.

Corollary 14 *Every element of $\mathbb{L}_{\mathbb{D}}$ is uniquely the join of pairwise incomparable join-irreducibles, and every join-irreducible is a meet of literals.*

We still need to identify the meets of literals that are join irreducible, and the ordering between these join irreducibles. This is the real heart of the matter, and the hardest part. For this, we also need the following.

Theorem 15 *For any truth value algebra $\mathbb{A}$, and map $\varphi : V \to A$, there is exactly one homomorphism*

$$\beta : \mathbb{L}_{\mathbb{A}} \to \mathbb{A}$$

such that $\beta([v]) = \varphi(v)$ for all $v \in V$.

Proof. For $t \in T$, there is a unique homomorphism $\tilde{\varphi} : T \to \mathbb{A}$ extending φ. Now $[t]_{\mathbb{A}} = [s]_{\mathbb{A}}$ in $\mathbb{L}_{\mathbb{A}}$ implies that $\tilde{\varphi}(t) = \tilde{\varphi}(s)$. Thus for $[t] \in \mathbb{L}_{\mathbb{A}}$, we can let $\beta([t]) = \tilde{\varphi}(t)$. It is straightforward to show that β is a homomorphism. Moreover, β is unique because $\tilde{\varphi}$ is unique. ■

Theorem 16 *Let a and b be conjunctions of literals in $\mathbb{L}_{\mathbb{D}}$ with $b \leq a$. Then*

1. *Every literal that occurs in a also occurs in b.*
2. *$a = b$ if and only if they are the conjunctions of exactly the same literals.*
3. *a is join irreducible.*

Proof. (1) Suppose c is a literal that occurs in a and not in b. Define a function $\varphi : \{v_1, ..., v_n\} \to 3$ by

$$\varphi(v_i) = \begin{cases} 0 & \text{if } v_i = c \\ 1 & \text{if } v_i = c' \\ u & \text{otherwise} \end{cases}$$

Then φ extends to a homomorphism satisfying $\varphi(a) = 0$ and $\varphi(b) = u$. Thus b does not lie below a.

(2) This follows from (1).

(3) Define a function $\varphi : \{v_1, ..., v_n\} \to D$ by

$$\varphi(v_i) = \begin{cases} 1 & \text{if } v_i \text{ occurs in } a \text{ and } v_i' \text{ does not occur in } a \\ 0 & \text{if } v_i' \text{ occurs in } a \text{ and } v_i \text{ does not occur in } a \\ u & \text{if } v_i \text{ and } v_i' \text{ both occur in } a \\ v & \text{if neither } v_i \text{ nor } v_i' \text{ occur in } a \end{cases}$$

Then φ extends to a homomorphism satisfying $\varphi(a) = u$ or 1 and for $b < a$, $\varphi(b) = 0$ or v. Thus the join of all conjunctions b strictly below a cannot equal a. ■

Theorem 17 *Let a and b be conjunctions of literals in $\mathbb{L}_{\mathbf{3}}$ with $b \leq a$. Then*

1. *Every literal that occurs in a also occurs in b.*
2. *If no more than one literal occurs in a for any variable, then a is join irreducible.*
3. *If at least one literal occurs in a for every variable, then a is join irreducible.*
4. *If both literals occur in a for at least one variable, but for at least one variable no literal occurs in a, then a is join reducible.*

Proof. (1) The proof of this is the same as for $\mathbb{D}$.

(2) If no more than one literal occurs in a for any variable, define a function $\varphi : \{v_1, ..., v_n\} \to 3$ by

$$\varphi(v_i) = \begin{cases} 1 & \text{if } v_i \text{ occurs in } a \\ 0 & \text{if } v_i' \text{ occurs in } a \\ u & \text{otherwise} \end{cases}$$

Then φ extends to a homomorphism satisfying $\varphi(a) = 1$ and for every b such that $b < a$, $\varphi(b) = 0$ or u. Since the join of 0's and u's cannot equal 1, the join of all such b's does not equal a. It follows that a is join irreducible.

(3) Suppose at least one literal occurs in a for every variable. Define a function $\varphi : \{v_1, ..., v_n\} \to 3$ by

$$\varphi(v_i) = \begin{cases} 1 & \text{if } v_i \text{ occurs in } a \text{ and } v_i' \text{ does not occur in } a \\ 0 & \text{if } v_i' \text{ occurs in } a \text{ and } v_i \text{ does not occur in } a \\ u & \text{if } v_i \text{ and } v_i' \text{ both occur in } a \end{cases}$$

Then φ extends to a homomorphism satisfying $\varphi(a) = u$ or 1 and for every b such that $b < a$, $\varphi(b) = 0$. Since the join of 0's and u's cannot equal 1, the join of all such b's does not equal a. It follows that a is join irreducible.

(4) If both literals occur in a for at least one variable, and for $k \geq 1$ variables no literal occurs in a, let

$$b = \bigvee (a \wedge c_1 \wedge \cdots \wedge c_k)$$

where $c_1, ..., c_k$ range over all combinations of literals such that neither c_i nor c_i' occurs in a. We prove that $b = a$ by showing that there is no homomorphism $\varphi : \mathbb{A} \to 3$ that separates them. Let v_i be a variable such that both v_i and v_i' occur in a. If $\varphi(v_i) = 0$ or 1 then $\varphi(a) \leq \varphi(v_i) \wedge (\varphi(v_i))' = 0$ and $\varphi(b) \leq \varphi(a) = 0$. Thus we may assume that $\varphi(v_i) = u$ for all variables that occur twice in a. For the same reason, we may assume that each literal that occurs in a, whose negation does not occur in a, maps to u or 1. Thus $\varphi(a) = u$. Since $b \leq a$, either $\varphi(b) = \varphi(a)$ or $\varphi(b) = 0$. Now $\varphi(b) \geq \varphi(a \wedge c_1 \wedge \cdots \wedge c_k) = \varphi(a) \wedge \varphi(c_1) \wedge \cdots \wedge \varphi(c_k)$ for all literals $c_1, ..., c_k$ such that neither c_i nor c_i' are in a. Pick the list such that $\varphi(c_i)$ is either u or 1. For this choice, $u \geq \varphi(b) \geq \varphi(a) \wedge \varphi(c_1) \wedge \cdots \wedge \varphi(c_k) = u$. Thus no homomorphism into 3 separates a and b, proving that they are equal. ■

These last two theorems tell us what the join irreducibles are in the De Morgan and in the Kleene case, and tell us what the ordering is between these join irreducibles. It is clear that in both cases, these join irreducibles form lattices. They are not sublattices of $\mathbb{L}_{\mathbb{D}}$ and $\mathbb{L}_{\mathbf{3}}$ but do get lattice structures from the partial ordering of these algebras.

We remark at this point that in $\mathbb{L}_{\mathbf{2}}$, the join irreducibles are the conjunctions of literals with exactly one literal for each variable. This is easy to see, just using the Boolean algebra axioms.

3.1 Normal forms for fuzzy interval-valued logic

The normal form for the De Morgan algebra $\mathbb{L}_{\mathbb{D}}$ now follows readily. The join irreducible elements are exactly the conjunctions of distinct literals, and the element 1. Since $\mathbb{L}_{\mathbb{D}}$ is a finite distributive lattice, every element can be written uniquely as a conjunction of pairwise incomparable join irreducibles.

Following is a precise procedure for putting an arbitrary term w in the variables $x_1, ..., x_n$ in this De Morgan disjunctive normal form:

1. Given a term in T, first use De Morgan's laws to move all the negations in, so that the term is rewritten as a term w_1 which is of lattice type in the literals, 0, and 1.

2. Next use the distributive law to obtain a new term w_2 from w_1 which is a disjunction of conjunctions involving the literals, 0, and 1. At this point, discard any conjunction in which 0 or $1'$ appears as one of the conjuncts. Also discard any repetition of literals from any conjunction, as well as 1 and $0'$ from any conjunction in which they do not appear alone (if a conjunction consists entirely of 1's and $0'$'s, then replace the whole conjunction by 1). If no conjunctions are left, then you have the empty join—the normal form for 0. This yields a term w_3.

3. Now discard all non-maximal conjunctions among the conjunctions that w_3 is a disjunction of. The type of conjunctions we now are dealing with are either conjunctions of literals or 1 by itself. This process yields a term w_4.

The term thus obtained is now in De Morgan disjunctive normal form, and represents the same element as w when interpreted in $\mathbb{L}_{\mathbb{D}}$.

3.2 Normal forms for fuzzy logic

Following is a procedure for putting an arbitrary term $w \in \mathcal{T}(x_1, x_2, \ldots, x_n)$ in Kleene normal form:

1. Follow the steps in the previous section to put the term in De Morgan disjunctive normal form w_4.

2. At this point, replace any conjunction of literals, c, which contains both literals for at least one variable by the disjunction of all the conjunctions of literals below c that contain exactly one of the literals for each variable not occurring in c. This process yields a term w_5.

3. Finally, again discard all non-maximal conjunctions among the conjunctions that are left. If no conjunctions are left, then you have the empty join—the normal form for 0. This process yields a term w_6.

The term thus obtained is now in Kleene disjunctive normal form. Note that when interpreting these terms in $\mathbb{L}_{\mathbf{3}}$, the element obtained at each step is equal to the original element w interpreted in $\mathbb{L}_{\mathbf{3}}$.

3.3 Normal forms for Boolean logic

Here is the procedure for finding the Boolean disjunctive normal form.

1. Follow the steps in the previous two sections to put a term w in Kleene disjunctive normal form w_6.

2. If $w_6 = 1$, replace w_6 by the disjunction of all the conjunctions that contain exactly one literal for each variable. Otherwise, discard any conjunction of literals, c, which contains both literals for at least one variable; and replace any conjunction of literals, c, that does not contain both literals for any variable by the disjunction of all the conjunctions of literals below c that contain exactly one of the literals for each variable not occurring in c. This process yields a term w_7.

3. Finally, again discard all non-maximal conjunctions among the conjunctions that are left. If no conjunctions are left, then you have the empty join—the normal form for 0. This process yields a term w_8, the Boolean disjunctive normal form for $\mathbb{L}_{\mathbf{2}}$.

4 Comparison of normal forms

The dual forms to these various disjunctive normal forms are called *conjunctive normal forms*. For example, in the fuzzy interval-valued case, every element is uniquely the conjunction of pairwise incomparable disjunctions of literals, and any disjunction of literals is meet irreducible. We do not go through the details for these conjunctive normal forms. We just remark that applying De Morgan laws to the negation of the disjunctive normal form gives the conjunctive normal form of that negation. For any term $w \in \mathbb{T}$, let $\mathbf{D}_{\mathcal{B}}(w)$ denote the Boolean disjunctive normal form, $\mathbf{D}_{\mathcal{K}}(w)$ the Kleene disjunctive normal form, and $\mathbf{D}_{\mathcal{M}}(w)$ the De Morgan disjunctive normal form of that term. Let $\mathbf{C}_{\mathcal{B}}(w)$, $\mathbf{C}_{\mathcal{K}}(w)$, and $\mathbf{C}_{\mathcal{M}}(w)$ denote the corresponding conjunctive normal forms. The procedures for constructing the disjunctive normal forms, and the dual procedures for constructing the conjunctive normal forms, make it clear that interpreted as elements of $\mathbb{L}_{\mathbb{D}}$ the following equalities and inequalities hold:

$$\mathbf{D}_{\mathcal{B}}(w) \leq \mathbf{D}_{\mathcal{K}}(w) \leq \mathbf{D}_{\mathcal{M}}(w) = \mathbf{C}_{\mathcal{M}}(w) \leq \mathbf{C}_{\mathcal{K}}(w) \leq \mathbf{C}_{\mathcal{B}}(w)$$

Interpreting these elements in $\mathbb{L}_{\mathbf{3}}$ yields

$$\mathbf{D}_{\mathcal{B}}(w) \leq \mathbf{D}_{\mathcal{K}}(w) = \mathbf{D}_{\mathcal{M}}(w) = \mathbf{C}_{\mathcal{M}}(w) = \mathbf{C}_{\mathcal{K}}(w) \leq \mathbf{C}_{\mathcal{B}}(w)$$

and interpreting these elements in $\mathbb{L}_\mathbf{2}$ yields

$$\mathbf{D}_{\mathcal{B}}(w) = \mathbf{D}_{\mathcal{K}}(w) = \mathbf{D}_{\mathcal{M}}(w) = \mathbf{C}_{\mathcal{M}}(w) = \mathbf{C}_{\mathcal{K}}(w) = \mathbf{C}_{\mathcal{B}}(w)$$

But more is true. Each of the congruences $\equiv_\mathbb{D}$, $\equiv_\mathbf{3}$, and $\equiv_\mathbf{2}$ on the term algebra $\mathbb{T}$ induces a partition of $\mathbb{T}$, and these partitions are successively coarser. It is easier to be equivalent under $\equiv_\mathbf{2}$ than under $\equiv_\mathbf{3}$, for example. This means that for our three propositional logics, which are $\mathbb{L}_\mathbb{D}$, $\mathbb{L}_\mathbf{3}$, and $\mathbb{L}_\mathbf{2}$, we have homomorphisms

$$\mathbb{T} \to \mathbb{L}_\mathbb{D} \xrightarrow{\kappa} \mathbb{L}_\mathbf{3} \xrightarrow{\beta} \mathbb{L}_\mathbf{2}$$

and of course each is onto. Let $t \in \mathbb{T}$. Now t determines equivalence classes $[t]_{\equiv_\mathbb{D}}$, $[t]_{\equiv_3}$, and $[t]_{\equiv_\mathbf{2}}$ which are elements of $\mathbb{L}_\mathbb{D}$, $\mathbb{L}_\mathbf{3}$, and $\mathbb{L}_\mathbf{2}$, respectively. We have of course, $\kappa([t]_{\equiv_\mathbb{D}}) = [t]_{\equiv_3}$, and $\beta([t]_{\equiv_3}) = [t]_{\equiv_\mathbf{2}}$. But β is not one-to-one: it takes many elements onto $[t]_{\equiv_\mathbf{2}}$. The elements of $\mathbb{T}$ that are in $[t]_{\equiv_\mathbf{2}}$have the same Boolean disjunctive normal form d, and have the same Boolean conjunctive normal form c. These forms are representatives of the element $[t]_{\equiv_\mathbf{2}}$. As noted in the discussion of normal forms, $[d]_{\equiv_3} \leq [c]_{\equiv_3}$. The elements of these two equivalence classes all represent the same element $[t]_{\equiv_\mathbf{2}}$ in $\mathbb{L}_\mathbf{2}$, but distinct elements in $\mathbb{L}_\mathbf{3}$. The elements of $\mathbb{L}_\mathbf{3}$ that β takes onto the element $[t]_{\equiv_\mathbf{2}}$ form a sublattice of $\mathbb{L}_\mathbf{3}$ with smallest element $[d]_{\equiv_3}$ and largest $[c]_{\equiv_3}$. This is because in a lattice, the set of elements in a congruence class forms a convex set, in this case the congruence class being all those elements of $\mathbb{L}_\mathbf{3}$ that go onto $[t]_{\equiv_\mathbf{2}}$. These facts follow from first principles of algebra homomorphisms and the normal form discussion above.

Similar remarks hold for the homomorphism $\kappa : \mathbb{L}_\mathbb{D} \to \mathbb{L}_\mathbf{3}$, and in fact for the composition $\beta\kappa$.

5 Truth tables and normal forms

It is a simple exercise to derive the disjunctive normal form and the conjunctive normal form for Boolean algebras from "truth tables". The truth tables are constructed from the relevant algebra of truth values. For the Boolean case, the algebra of truth values is the Boolean algebra with two elements shown in Table 2.

x	y	$x \wedge y$	$x \vee y$	x'
0	0	0	0	1
1	0	0	1	0
0	1	0	1	1
1	1	1	1	0

Table 2:

To find the disjunctive normal for the expression $(a \vee b) \wedge c$ in three variables, one writes down Table 3. In those rows that have value 1 in the column of the expression

a	b	c	$(a \vee b) \wedge c$	a	b	c	$(a \vee b) \wedge c$
0	0	0	0	0	1	1	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	0
1	0	0	0	1	1	1	1

Table 3:

to be put into disjunctive normal form, form the conjunction $x \wedge y \wedge z$ where x is a or a' according as to whether the value in column a is 1 or 0, and similarly for the other two variables. The disjunction of these conjunctions is the **Boolean-disjunctive normal form**. There are three conjunctions to be taken in this case, and we get

$$(a \vee b) \wedge c = (a' \wedge b \wedge c) \vee (a \wedge b' \wedge c) \vee (a \wedge b \wedge c)\,.$$

To find the conjunctive normal for the expression $(a \vee b) \wedge c'$ in three variables, one writes down Table 4. In those rows that have value 0 in the column of the expression

a	b	c	$(a \vee b) \wedge c'$	a	b	c	$(a \vee b) \wedge c'$
0	0	0	0	0	1	1	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	1
1	0	0	1	1	1	1	0

Table 4:

to be put into conjunctive normal form, form the disjunction $x \vee y \vee z$ where x is a or a' according as to whether the value in column a is 0 or 1, and similarly for the other two variables. The conjunction of these disjunctions is the **Boolean-conjunctive normal form**. There are five disjunctions to be taken in this case, and we get

$$(a \vee b) \wedge c' = (a \vee b \vee c) \wedge (a \vee b \vee c') \wedge (a \vee b' \vee c') \wedge (a' \vee b \vee c') \wedge (a' \vee b' \vee c')\,.$$

The disjunctive and conjunctive normal forms for the fuzzy case can also be recovered from truth tables. The table of truth values for the fuzzy case is depicted in Table 5.

The disjunctive normal form of an element t for the fuzzy case is gotten from its truth table as follows:

- For those rows that have value 1 in the column of the expression to be put into disjunctive normal form, form the conjunction of the variables with truth value equal to 1 with the negations of the variables with truth value equal to 0 (and leave out variables with truth value equal to u).

x	y	$x \wedge y$	$x \vee y$	x'
0	0	0	0	1
0	1	0	1	0
1	0	0	1	0
1	1	1	1	0

x	y	$x \wedge y$	$x \vee y$	x'
u	u	u	u	u
u	0	0	u	u
u	1	u	1	u
0	u	0	u	1
1	u	u	1	0

Table 5:

- For those rows that have value u in the column of the expression to be put into disjunctive normal form, form the conjunction of the variables with truth value equal to 1 with the negations of the variables with truth value equal to 0, and with both the variables and the negated variables of the variables with truth value equal to u.

The irredundant Kleene-disjunctive normal form for the expression is then obtained by discarding redundant conjunctions—that is, any conjunction that contains the same, and possibly more, literals as another conjunction in the term.

Theorem 18 *The scheme described above yields the disjunctive normal forms for (classical) fuzzy logic.*

Proof. To see that this scheme works for a term t, we show that for every join irreducible $p \leq t$ there is a join irreducible p' picked up by the scheme satisfying $p \leq p' \leq t$. Then we show that every join irreducible p picked up by the scheme lies below t.

Let t be a formula, and let p be a join irreducible with $p \leq t$.

Case 1 p contains each variable at most once.

In this case consider the row in which each variable occurring in p (not negated) is assigned a 1, each variable occurring negated is assigned a 0 and each variable not occurring is assigned a u. This valuation clearly sends p to 1, and since $p \leq t$ it also sends t to 1. Further, the scheme applied to this row now picks up p.

Case 2 p contains each variable at least once.

In this case, consider the row in which each variable only occurring unnegated is assigned 1, each variable occurring only negated is assigned 0, each variable occurring twice is assigned u. In this row, p is assigned value u, and so t is assigned either u or 1. Either way the scheme will pick up a p' that is join irreducible and $p \leq p' \leq t$.

Therefore the scheme picks up all the maximal join irreducibles below t (and perhaps some that are not maximal).

For the converse, let p be a join irreducible that is picked up by the scheme from a row where t is assigned a 1. We want to show that $p \leq t$. Since $\tilde{v}(t) = 1$ for the v given by that row, there is some p' with p' join irreducible, $p' \leq t$ and $\tilde{v}(p') = 1$. If p' contains any literal not contained in p, then $\tilde{v}(p') \leq u$, and thus all the literals in p' are in p, that is, $p \leq p'$. So $p \leq t$ as desired.

Now assume p is picked up by the scheme from a row where t is assigned a u. Then again, there is a p', join irreducible, with $p' \leq t$ and $\tilde{v}(p') = u$. Suppose p' contains a literal not contained in p. Then this is a literal assigned 0 by $\tilde{v}$, which would imply $v(p') = 0$—a contradiction. So $p \leq p' \leq t$ as desired. ■

Table 6 shows an example for two rows of a table with n variables. Notice that in the first row shown, no literal is picked up for x_2 and in the second row shown, both literals are picked up for x_2.

x_1	x_2	x_3	$\cdots$	x_n	t	pick up
1	u	0	$\cdots$	0	1	$x_1 \wedge x_3' \wedge \cdots \wedge x_n'$
1	u	0	$\cdots$	1	u	$x_1 \wedge x_2 \wedge x_2' \wedge x_3' \wedge \cdots \wedge x_n$

Table 6:

Table 7 gives an example for a table with 2 variables.

x	y	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y')$	pick up
0	0	0	
0	1	0	
1	0	1	$x \wedge y'$
1	1	0	
u	u	u	$x \wedge x' \wedge y \wedge y'$
u	0	u	$x \wedge x' \wedge y'$
u	1	u	$x \wedge x' \wedge y$
0	u	u	$x' \wedge y \wedge y'$
1	u	u	$x \wedge y \wedge y'$

Table 7:

After discarding redundant conjunctions you get the normal form

$$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y') = (x' \wedge y \wedge y') \vee (x \wedge x' \wedge y) \vee (x \wedge y')$$

You follow a dual procedure (reverse the roles of 0 and 1 and reverse the roles of $\wedge$ and $\vee$) to obtain Kleene-conjunctive normal forms.

x	y	$x \wedge y$	$x \vee y$	x'	x	y	$x \wedge y$	$x \vee y$	x'
0	0	0	0	1	1	u	u	1	0
0	1	0	1	1	v	v	v	v	v
1	0	0	1	0	v	0	0	v	v
1	1	1	1	0	v	1	v	1	v
u	u	u	u	u	0	v	0	v	1
u	0	0	u	u	1	v	v	1	0
u	1	u	1	u	u	v	0	1	u
0	u	0	u	u	v	u	0	1	v

Table 8:

Also the same can be done for interval-valued fuzzy sets. There, the algebra of truth values $\mathbb{D}$ has the truth values shown in Table 8.

The disjunctive normal forms for the fuzzy case are gotten from a truth table as follows:

- For those rows that have value 1 in the column of the expression to be put into disjunctive normal form, and truth values 0 or 1 and possibly u or v (but not both) for the variables, form the conjunction of the variables with truth value equal to 1 with the negations of the variables with truth value equal to 0 (and leave out variables with truth value equal to u or v). If the row has values all u's (or all v's) pick up $p = 1$.
- For those rows that have value 1 in the column of the expression to be put into disjunctive normal form, and both u and v occur as truth values for the variables, form two conjunctions: one using the variables with truth value equal to 1 and the negations of the variables with truth value equal to 0 and using both the variables and the negated variables of the variables with truth value equal to u, the other using the variables with truth value equal to 1 and the negations of the variables with truth value equal to 0 and using both the variables and the negated variables of the variables with truth value equal to v.
- For those rows that have value u in the column of the expression to be put into disjunctive normal form, form the conjunction using the variables with truth value equal to 1, the negations of the variables with truth value equal to 0, and using both the variables and the negated variables of the variables with truth value equal to u (and leave out variables with truth value equal to v).
- For those rows that have value v in the column of the expression to be put into disjunctive normal form, form the conjunction using the variables with truth value equal to 1, the negations of the variables with truth value equal to 0, and using both the variables and the negated variables of the variables with truth value equal to v (and leave out variables with truth value equal to u).

The irredundant De Morgan-disjunctive normal form for the expression is then obtained by discarding redundant conjunctions—that is, any conjunction that contains the same, and possibly more, literals as another conjunction in the term.

Table 9 gives some sample rows with four variables.

x_1	x_2	x_3	x_4	t	pick up
1	u	0	0	1	$x_1 \wedge x_3' \wedge x_4'$
0	u	1	v	u	$x_1' \wedge x_2 \wedge x_2' \wedge x_3$
1	v	0	u	1	$x_1 \wedge x_2 \wedge x_2' \wedge x_3'$ and $x_1 \wedge x_3' \wedge x_4 \wedge x_4'$

Table 9:

Theorem 19 *The scheme described above yields De Morgan-disjunctive normal forms.*

Proof. Let t be a formula, and let p be a join irreducible with $p \leq t$. We need to show that the scheme either picks up p or some join irreducible p' with $p \leq p' \leq t$.

Case 1 p contains each variable at most once.

In this case consider the row in which each variable occurring in p (not negated) is assigned 1, each variable occurring negated is assigned 0 and each variable not occurring is assigned u. This valuation clearly sends p to 1, and since $p \leq t$ it also sends t to 1. Further, the scheme applied to this row now picks up p.

Case 2 p contains each variable at least once.

In this case, consider the row in which each variable only occurring unnegated is assigned 1, each variable occurring only negated is assigned 0, each variable occurring twice is assigned u. In this row, p is assigned value u, and so t is assigned either u or 1. If t is assigned u, the scheme picks up p. If t is assigned 1, the scheme will pick up a p' that is join irreducible with $p \leq p' \leq t$.

Case 3 p contains at least one variable together with its negation, and p does not contain a literal for at least one variable.

In this case, consider the row in which each variable only occurring unnegated is assigned 1, each variable occurring only negated is assigned 0, each variable occurring twice is assigned u and each variable not occurring is assigned v. In this row, p is assigned value u, and so t is assigned either u or 1. If t is assigned u, the scheme picks up p. If t is assigned v, the scheme picks up both p and a p' that is join irreducible and assigned the value v.

Therefore the scheme picks up all the maximal join irreducibles below t (and very likely some that are not maximal).

For the converse, let p be a join irreducible that is picked up by the scheme. We want to show that $p \leq t$. Now t is the join of join irreducibles that lie below t, say

$t = p_1 \vee p_2 \vee \cdots \vee p_k$ with each p_i join irreducible. If t is assigned one of the values 0, u or v, then at least one of the join irreducibles below t is assigned the same value as t. If t is assigned the value 1, then either one of the join irreducibles below t is also assigned 1 or there exist two join irreducibles below t, one assigned the value u and the other assigned the value v.

Case 1 Assume p is picked up by the scheme from a row where t is assigned u. Then there is a join irreducible $p' \leq t$ with the valuation of p' equal to u. Suppose p' contains a literal not contained in p. Then this is a literal assigned v by the valuation for that row, which would imply the valuation of p' is 0—a contradiction. So $p \leq p' \leq t$ as desired. If p is picked up by the scheme from a row where t is assigned a v, the argument is the same (with the roles of u and v reversed).

Case 2 Assume p is picked up by the scheme from a row where t is assigned 1 and assume there is some p' join irreducible with $p' \leq t$ and the value assigned to p' from the values in this row is 1. Now if p' contains a literal not contained in p, this literal must be assigned value u or v. But this contradicts the assumption that p' has value 1. Thus $p \leq p' \leq t$ as desired.

Case 3 Finally, assume p is picked up by the scheme from a row where t is assigned 1 and assume that every join irreducible $p' \leq t$ has value less than 1. Then there are join irreducibles p' and p'' below t with values u and v, respectively. If p were assigned the value 1, the row would not contain both values u and v and there could not be join irreducibles p' and p'' below t with values u and v, respectively. Thus p is assigned the value u or v. Suppose p is assigned the value u. If p' contains a literal not in p, this literal would have to be assigned the value v, contradicting the fact that p' is assigned the value u. Thus in this case, $p \leq p' \leq t$. Similarly, if p is assigned the value v, $p \leq p'' \leq t$. So $p \leq t$ as desired.

Therefore all the join irreducibles picked up by the scheme lie below t. ■

Table 10 is an example for a table with 2 variables. After discarding redundant conjunctions you get the normal form

$$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y') = (x \wedge x') \vee (x \wedge y') \vee (y \wedge y')$$

Checking this normal form in Table 11 verifies this equality:

6 The two variable case

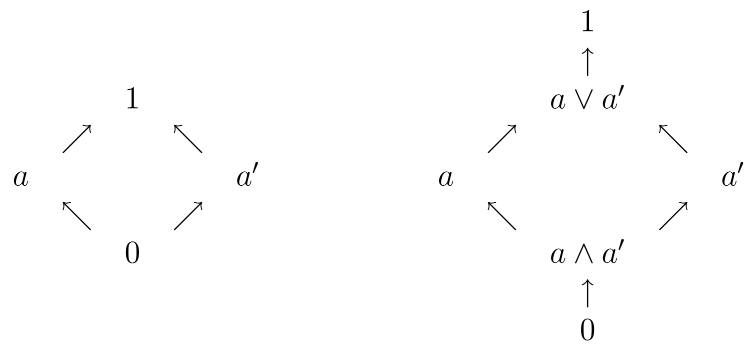
As stated, the three propositional logics above are finite, but get large very quickly as n increases. For the Boolean case, the algebra has 2^{2^n} elements. However, for the Boolean case and the Kleene case, the algebras can be examined by hand for $n \leq 2$.

x	y	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y')$	pick up
0	0	0	
0	1	0	
1	0	1	$x \wedge y'$
1	1	0	
u	u	u	$x \wedge x' \wedge y \wedge y'$
u	0	u	$x \wedge x' \wedge y'$
u	1	u	$x \wedge x' \wedge y$
0	u	u	$x' \wedge y \wedge y'$
1	u	u	$x \wedge y \wedge y'$
v	v	v	$x \wedge x' \wedge y \wedge y'$
v	0	v	$x \wedge x' \wedge y'$
v	1	v	$x \wedge x' \wedge y$
0	v	v	$x' \wedge y \wedge y'$
1	v	v	$x \wedge y \wedge y'$
u	v	1	$x \wedge x'$ and $y \wedge y'$
v	u	1	$x \wedge x'$ and $y \wedge y'$

Table 10:

x	y	$x \wedge x'$	$x \wedge y'$	$y \wedge y'$	$(x \wedge x') \vee (x \wedge y') \vee (y \wedge y')$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	1	0	1
1	1	0	0	0	0
u	u	u	u	u	u
u	0	u	u	0	u
u	1	u	0	0	u
0	u	0	0	u	u
1	u	0	u	u	u
v	v	v	v	v	v
v	0	v	v	0	v
v	1	v	0	0	v
0	v	0	0	v	v
1	v	0	v	v	v
u	v	u	0	v	1
v	u	v	0	u	1

Table 11:



Boolean case for $n = 1$ Kleene case for $n = 1$

The case $n = 1$ is not very challenging. In the Boolean case with $P = \{a\}$, it is a four-element lattice, and in the Kleene case a six-element lattice. For $n = 2$, the Boolean logic is the Boolean algebra with 16 elements, and in the Kleene case, the logic has 84 elements. We will examine those algebras explicitly.[1]

The Boolean propositional logic on two variables is pretty easy to figure out. Let the variables be a and b. The set of join irreducibles is $\{a \wedge b, a \wedge b', a' \wedge b, a' \wedge b'\}$. This is an anti-chain—that is, the elements are pairwise incomparable, and every element in $\mathbb{L}_\mathbf{2}$ is uniquely the join of a subset of this four element set. The logic $\mathbb{L}_\mathbf{2}$ has 16 elements. This lattice is pictured in Figure 1.

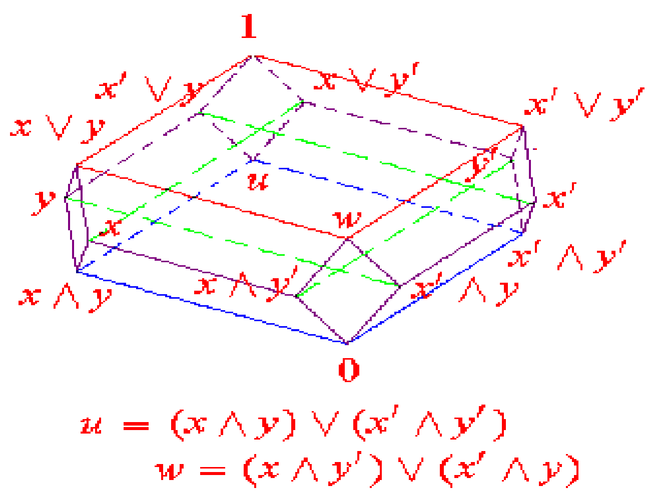


Figure 1: Boolean logic with two variables

The elements of the logic $\mathbb{L}_\mathbf{2}$ are listed in Table 12 in three forms, the short, disjunctive normal, and conjunctive normal forms.

[1]For $n = 2$ the enumeration in the Kleene case was done in [16] and [2]. For $n = 3$, there are 43918 truth tables [10, 11, 3] and for $n = 4$, there are 160,297,985,276 [4, 5].

Short form	Boolean disjunctive normal form	Boolean conjunctive normal form
1	$(a\wedge b)\vee(a'\wedge b)\vee(a\wedge b')\vee(a'\wedge b')$	1
a	$(a\wedge b)\vee(a\wedge b')$	$(a\vee b)\wedge(a\vee b')$
b	$(a\wedge b)\vee(a'\wedge b)$	$(a\vee b)\wedge(a'\vee b)$
a'	$(a'\wedge b)\vee(a'\wedge b')$	$(a\vee b)\wedge(a\vee b')$
b'	$(a\wedge b')\vee(a'\wedge b')$	$(a\vee b')\wedge(a'\vee b')$
$a\wedge b$	$a\wedge b$	$(a\vee b)\wedge(a\vee b')\wedge(a'\vee b)$
$a\wedge b'$	$a\wedge b'$	$(a\vee b)\wedge(a\vee b')\wedge(a'\vee b')$
$a'\wedge b$	$a'\wedge b$	$(a\vee b)\wedge(a'\vee b)\wedge(a'\vee b')$
$a'\wedge b'$	$a'\wedge b'$	$(a\vee b')\wedge(a'\vee b')\wedge(a'\vee b)$
$a\vee b$	$(a\wedge b)\vee(a\wedge b')\vee(a'\wedge b)$	$a\vee b$
$a\vee b'$	$(a\wedge b)\vee(a\wedge b')\vee(a'\wedge b')$	$a\vee b'$
$a'\vee b$	$(a\wedge b)\vee(a'\wedge b')\vee(a'\wedge b)$	$a'\vee b$
$a'\vee b'$	$(a'\wedge b)\vee(a\wedge b')\vee(a'\wedge b')$	$a'\vee b'$
$a\Leftrightarrow b$	$(a\wedge b)\vee(a'\wedge b')$	$(a\vee b')\wedge(a'\vee b)$
$a'\Leftrightarrow b$	$(a\wedge b')\vee(a'\wedge b)$	$(a\vee b)\wedge(a'\vee b')$
0	0	$(a\vee b)\wedge(a'\vee b)\wedge(a\vee b')\wedge(a'\vee b')$

Table 12:

The key to getting disjunctive normal forms was to figure out what the join irreducibles are. Table 13 lists them for the De Morgan, or interval-valued case, and for the Kleene, or fuzzy case. The 84 element Kleene counterpart is a bit too big to depict here. However, its join irreducible elements, shown in Figure 2, form a subposet of $\mathbb{L}_{\mathbb{K}}$.

This set of join irreducibles is a partially ordered set, getting its partial order from that of $\mathbb{L}_{\mathbb{K}}$. This partial order makes the join irreducibles into a lattice, as pictured in Figure 1. But it is not a sublattice of $\mathbb{L}_{\mathbb{K}}$.

Table 14 lists the number of elements in the Kleene two-variable propositional logic $\mathbb{L}_{\mathbf{3}}$ that collapse to each of the elements in the Boolean two-variable propositional logic $\mathbb{L}_{\mathbf{2}}$. This count shows that there are 84 elements in $\mathbb{L}_{\mathbf{3}}$. Each of the sets of 4 forms a diamond shaped sublattice (not, of course, a sub-bounded lattice). They are not subalgebras, as they are not closed under negation. The 17 element sets form a 16 element lattice isomorphic to $\mathbb{L}_{\mathbf{2}}$ as a lattice, with an extra top (or bottom) element.

There is another pertinent fact about Kleene propositional logic. In the two-variable case, it has 84 elements, each one being a function $\{0,u,1\}\times\{0,u,1\}\to\{0,u,1\}$. But there are 3^9 such functions, so not every one of them is represented by an element of the logic. Thus **3** is not *primal.* The two-element Boolean algebra is, and $\mathbb{D}$, like **3**, is not. One point about this: In fuzzy theory, sometimes implication $a\to b$ is taken to be $\sup\{c : a\wedge c\leq b\}$. If we do this in fuzzy propositional logic, there is no formula for it in terms of a and b. This function is not represented by an element of the logic, reflecting its non-primality.

Kleene (classical fuzzy logic)	De Morgan (interval-valued fuzzy logic)
1	1
a	a
b	b
a'	a'
b'	b'
$a \wedge b$	$a \wedge b$
$a' \wedge b$	$a' \wedge b$
$a \wedge b'$	$a \wedge b'$
$a' \wedge b'$	$a' \wedge b'$
$a \wedge a' \wedge b$	$a \wedge a' \wedge b$
$a \wedge a' \wedge b'$	$a \wedge a' \wedge b'$
$a \wedge b \wedge b'$	$a \wedge b \wedge b'$
$a' \wedge b \wedge b'$	$a' \wedge b \wedge b'$
$a \wedge a' \wedge b \wedge b'$	$a \wedge a' \wedge b \wedge b'$
	$a \wedge a'$
	$b \wedge b'$

Table 13: Join Irreducibles

Boolean short form	# Kleene elements	Boolean short form	# Kleene elements
1	17	0	17
$x \vee y$	4	x	4
$x \vee y'$	4	x'	4
$x' \vee y$	4	y	4
$x' \vee y'$	4	y'	4
$x \wedge y$	4	$(x \wedge y) \vee (x' \wedge y')$	1
$x \wedge y'$	4	$(x \vee y') \wedge (x' \vee y)$	1
$x' \wedge y$	4		
$x' \wedge y'$	4		
		TOTAL	84

Table 14:

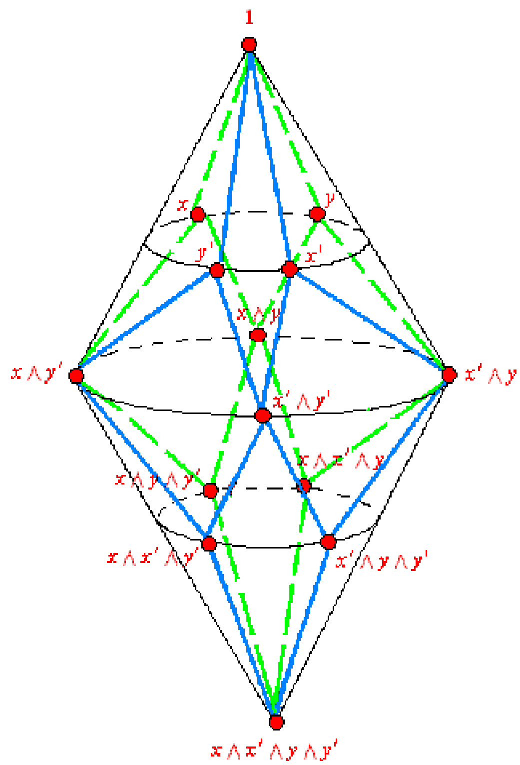


Figure 2: Join irreducibles in Kleene two-variable logic

7 Generalizations of Boolean terms

Turksen has written several papers concerned with normal forms in two variables [18, 19, 20, 21, 22, 23, 24, 25]. He uses these to discover alternative fuzzy generalizations of Boolean terms. We begin with an example of his having to do with the logical connective "AND". In Boolean logic, A "AND" B has $A \wedge B$ as its propositional counterpart, where $\wedge$ is the usual Boolean connective. What is the proper propositional counterpart in fuzzy logic? In Boolean propositional logic, we are operating in the Boolean algebra $\mathbb{L}_{\mathbf{2}}$, and in fuzzy propositional logic in the algebra $\mathbb{L}_{\mathbb{I}}$, where the algebra of truth values is $\mathbb{I}$, the unit interval with the operations $\wedge = \min$, $\vee = \max$, $x' = 1 - x$, and the usual constants 0 and 1. Turksen argues that in fuzzy logic, the "meta-linguistic" connective "AND" should not necessarily become simply $A \wedge B$ in the Kleene algebra $\mathbb{L}_{\mathbb{I}}$. There is merit to this skepticism. Consider the two variable case, with the variables being A and B, which is the only case Turksen considers. In the natural homomorphism

$$\mathbb{L}_{\mathbb{I}} \to \mathbb{L}_{\mathbf{2}}$$

there are exactly four elements that go onto $A \wedge B$ in $\mathbb{L}_{\mathbf{2}}$, namely the elements listed just below, all considered as elements of $\mathbb{L}_{\mathbb{I}}$.

$$A \wedge B$$
$$A \wedge (A' \vee B)$$
$$B \wedge (A \vee B')$$
$$(A \vee B) \wedge (A \vee B') \wedge (A' \vee B)$$

One could argue that any one of these is a reasonable candidate for the fuzzy counterpart to the Boolean "AND". They all collapse to the same element in Boolean logic. The elements in the same row in Table 12 are logically equivalent in the Boolean case. So perhaps there is some justifying to do to settle on $A \wedge B$ as the natural propositional counterpart of A "AND" B in fuzzy logic. Or perhaps each has its merits, or demerits as a candidate for "AND" in the fuzzy realm. This point of view is more convincing if one considers the notion of "IMPLIES". In Boolean logic, it is usually taken to be $A' \vee B$, namely "material implication." Again, in the homomorphism above, the four elements

$$A' \vee B$$
$$B \vee (A' \wedge B')$$
$$A' \vee (A \wedge B)$$
$$(A \wedge B) \vee (A' \wedge B') \vee (A' \wedge B)$$

of $\mathbb{L}_{\mathbb{I}}$ all map to $A' \vee B$ in $\mathbb{L}_{\mathbf{2}}$. So any one of these is a potential candidate to be the propositional counterpart of the linguistic connective "IMPLIES" in fuzzy logic. Indeed, in fuzzy logic, the form $A' \vee (A \wedge B)$ is sometimes used for implication. It is called Q-implication, and is used in other logics as well [15].

One could make these considerations for all sixteen elements in the Boolean algebra $\mathbb{L}_{\mathbf{2}}$. But there are difficulties. In this two variable case, $\mathbb{L}_{\mathbb{I}}$ has 84 elements, and 17 of them go onto 1 in $\mathbb{L}_{\mathbf{2}}$, giving 17 candidates in the fuzzy case for what Turksen calls "complete affirmation." Turksen takes another tack. He observes in the two variable case that for any term $t \in \mathbb{T}$, the Boolean disjunctive normal $\mathbf{D}_{\mathcal{B}}(t)$ for t is contained in its conjunctive normal form $\mathbf{C}_{\mathcal{B}}(t)$ when viewed as elements of $\mathbb{L}_{\mathbb{I}}$. This is one of the several inequalities we stated above (see Section 4), and which holds for any finite number of variables. Turksen proposes that each one of the sixteen "linguistic combinations" X (the sixteen elements X of $\mathbb{L}_{\mathbf{2}}$), be represented by the interval $(\mathbf{D}_{\mathcal{B}}(X), \mathbf{C}_{\mathcal{B}}(X))$. This interval actually contains all the elements in $\mathbb{L}_{\mathbb{I}}$ that go onto $\mathbf{D}_{\mathcal{B}}(X)$ in $\mathbb{L}_{\mathbf{2}}$, so if one is to choose an interval to represent the corresponding linguistic connective, this is the appropriate one. These sixteen elements X are listed in their short form, Boolean disjunctive, and Boolean conjunctive normal forms in Table 12. For each such X, and for each evaluation, that is, each assignment of the variables A and B to elements of the set of truth values $[0, 1]$, this gives a subinterval of $[0, 1]$.

Turksen's proof that $\mathbf{D}_{\mathcal{B}}(X) \leq \mathbf{C}_{\mathcal{B}}(X)$ in $\mathbb{L}_{\mathbb{I}}$ is on a case by case basis, for the sixteen cases. Our proof appeals to a basic general fact about the various normal forms, and gets the result all at once, not only for all sixteen cases, but for any number of variables. It is not easily seen how Turksen's method of proof would extend to the case of an arbitrary finite number of variables. A proof similar to ours was given in [26].

Here are some examples for "AND". Let $A = .3$ and $B = .4$. Now $A \wedge B = .3$ and Turksen's interval is $(.3 \wedge .4, (.3 \vee .4) \wedge ((1 - .3) \vee .4) \wedge (.3 \vee (1 - .4)) = (.3, .4)$. Turksen objects to taking A "AND" B to be $A \wedge B = .3$ because it does not reflect "any effect" of B ([18], page 204). But of course it does. The minimum of two numbers depends on both those numbers. If we take $A = .3$ and $B = .8$, then the interval is $(.3, .3)$.

There are some basic objections to taking the intervals Turksen proposes. One could argue that A "AND" B should be increasing in both variables. Now

$$(A \wedge B, (A \vee B) \wedge (A \vee B') \wedge (A' \vee B))$$

does not necessarily give a larger interval with increased A and B, as easy examples show. Further, it is not clear how these intervals should be manipulated. Are they to be manipulated, and if so, how, exactly? What is $(A$"AND"$B)$ "OR" A?

What has been done is, in the two variable case, to associate with each element of $\mathbb{L}_{\mathbb{I}}$ a pair of elements of $\mathbb{L}_{\mathbb{I}}$. This gets sixteen elements of $(\mathbb{L}_{\mathbb{I}})^{[2]}$. This image gets the structure of a lattice from the partial order on $\mathbb{L}_{\mathbb{I}}$, and in fact is a Boolean algebra with its negation inherited from $(\mathbb{L}_{\mathbb{I}})^{[2]}$. This Boolean algebra is isomorphic to $\mathbb{L}_{\mathbf{2}}$. One could work in this algebra, but it is Boolean, and not appropriate for some fuzzy considerations.

For fuzzy considerations we consider the following development more appropriate. We are dealing with propositional logic with truth values in the unit interval $[0, 1]$ with the usual operations. In the two variable case, this gives us the 84 element propositional logic $\mathbb{L}_{\mathbb{I}}$. The sixteen "meta-linguistic" expressions Turksen refers to are the sixteen elements of $\mathbb{L}_{\mathbf{2}}$, and he starts there. The proper place to start is in $\mathbb{L}_{\mathbb{I}}$. Now there are 84 "meta-linguistic" expressions to consider, providing ample opportunity to philosophize. For example, there are seventeen elements that correspond to 1 in $\mathbb{L}_{\mathbf{2}}$, so these seventeen are all different elements that correspond to "complete affirmation". Each element of $\mathbb{L}_{\mathbb{I}}$ represents a fuzzy proposition, and an evaluation of the two variables gives a truth evaluation of that proposition. Limiting oneself to Turksen's sixteen intervals loses much of the two variable fuzzy propositional logic. And the restriction to the two variable case causes pause.

If it is felt that there is not enough fuzziness in ordinary fuzzy logic, one should consider interval-valued fuzzy logic. There the truth values are the intervals $[0, 1]^{[2]} = \{(a, b) : a, b \in [0, 1],\ a \leq b\}$, with the operations we have indicated above. So an interval-valued fuzzy set is a mapping $X \to [0, 1]^{[2]}$. This introduces another level of fuzziness at the very beginning: a truth value is not a number between 0 and 1, but an interval. The logical theory is straightforward, yielding the De Morgan algebra

$\mathbb{L}_{\mathbb{I}^{[2]}} = \mathbb{L}_{\mathbb{D}}$ with normal forms and truth tables just as we have it in the Boolean or Kleene case. This algebra in the two variable case has 168 elements. This De Morgan algebra is not a Kleene algebra as is $\mathbb{L}_{\mathbb{I}}$, in particular, it does not satisfy $x \wedge x' \leq y \vee y'$, so has somewhat different algebraic properties. However, much of t-norm theory carries over from $\mathbb{I}$ to $\mathbb{I}^{[2]}$ [7].

8 Fuzzy logic with t-norms and t-conorms

The operations $\vee$ and $\wedge$ on $[0,1]$ are sometimes replaced by a t-norm $\triangle$ and a t-conorm $\bigtriangledown$, and the involution $1-x$ by an involution η on $[0,1]$ with respect to which the t-norm and t-conorm are dual. But norms and conorms are defined using the order relation on $[0,1]$, or equivalently, the lattice operations $\vee$ and $\wedge$ on $[0,1]$. Thus, as an algebra this setup is $(\mathbb{I}, \bigtriangledown, \triangle, \eta) = ([0,1], \vee, \wedge, \bigtriangledown, \triangle, \eta, 0, 1)$, an algebra with arity $(2,2,2,2,1,0,0)$. Thus to develop a logic based on norms and conorms, one has this as its algebra of truth values. We know a little bit here: For strict t-norms, one may as well take $\triangle$ to be multiplication, and then $\bigtriangledown$ is whatever η forces it to be. Such a setup is called a *De Morgan triple* in fuzzy theory. Fixing the t-norm to be multiplication, two negations η and β give isomorphic triple systems if and only if $\eta(x) = (\beta(x^r))^{1/r}$ for some positive real number r. In fact, one may as well require that the negations fix $1/2$, in which case multiplication and two negations η and β give isomorphic De Morgan triples if and only if $\eta = \beta$. And two De Morgan triples with strict t-norms give the same logic if and only if they are isomorphic. So we get distinct propositional logics for distinct negations that fix $1/2$. This is standard t-norm theory [9]. Properties of the logics whose algebras of truth values are a De Morgan triple will be the subject of a future paper.

References

[1] Berman, J. and W. J. Blok, "Stipulations, multivalued logic, and De Morgan algebras," *MultipleValued Logic*, 2000.

[2] J. Berman and Ph. Dwinger, "De Morgan algebras: free products and free algebras," manuscript 1973.

[3] J. Berman and P. Kohler, Cardinalities of finite distributive lattices, Mitt. Math. Sem. Geissen **121** (1976) 103-124.

[4] J. Berman and M. Mukaidono, The number of fuzzy switching functions in four variables, *Abstracts Amer. Math. Soc.* **2** (1981) 25-35.

[5] J. Berman and M. Mukaidono, Enumerating fuzzy switching functions and free Kleene algebras, *Comp. & Maths. with Appls.* **10**(1) (1984) 25-35.

[6] M. Gehrke, C. Walker, and E. Walker, De Morgan systems on the unit interval, *Int. J. Intel. Sys.* **11** (1996) 733-750.

[7] M. Gehrke, C. Walker, and E. Walker, Some comments on interval-valued fuzzy sets, *Int. J. Intel. Sys.* **11** (1996) 751-759.

[8] M. Gehrke, C. Walker, and E. Walker, A mathematical setting for fuzzy logic, *Int. J. Uncertainty, Fuzziness, and Knowledge-based Systems*, **5** (3) (1997) 223-238.

[9] M. Gehrke, C. Walker, and E. Walker, Fuzzy logics arising from strict De Morgan systems, to appear in *Linz '99 Conference Proceedings.*

[10] M. Mukaidono, A necessary and sufficient condition for a fuzzy function to be a fuzzy logic function, (in Japanese). Papers of technical group on automata and languages. IECE Japan **53** (1978) 1-9.

[11] M. Mukaidono, A necessary and sufficient condition for fuzzy logic functions, *Proc. 9th Int. Symp. on Multiple-Valued Logic, IEEE, Bath England* (1979) 159-166.

[12] M. Mukaidono and H. Kikuchi, Proposal on fuzzy interval logic, *Japan. J. Fuzzy Theo. Sys.* **2** (2) (1990) 99-117.

[13] D. E. Muller, Treatment of Transition Signals in Electronic Switching Circuits by Algebraic Methods, *IRE Transactions on Electronic Computers*, EC-**8** (3) (September 1959).

[14] H. T. Nguyen, O. M. Kosheleva, and V. Kreinovich, Is the success of fuzzy logic really paradoxical? or What is the actual logic behind expert systems?, *International Journal of Intelligent Systems*, **11** (1996).

[15] H. T. Nguyen and E. Walker, *A First Course in Fuzzy Logic*, 2-nd edition, Chapman and Hall/CRC, Boca Raton, 1999.

[16] F. Preparata and R. Yeh, Continuously valued logic, *J. Comput. System Sci.* **6** (1972) 397-418.

[17] M. H. Stone, The theory of representations for Boolean algebras, *Trans. Amer. Math. Soc.* **40** (1936) 37-111.

[18] I. B. Turksen, Interval-valued fuzzy sets based on normal forms, *Fuzzy Sets and Systems* **20** (1986) 191-210.

[19] I. B. Turksen, Inference regions for fuzzy proposition, in: P. P. Wang, Ed., *Fuzzy Set Theory and Applications*, Plenum Press, New York (1983) 137-148.

[20] I. B. Turksen, Fuzzy representations and inference with normal forms, in: J. Kacprzyk and R. R. Yager, Eds., *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory*, Verlag, 1984.

[21] I. B. Turksen, Theories of set and logic with crisp or fuzzy information granules, *Journal of Advanced Computational Intelligence,* to appear, 1999.

[22] I. B. Turksen, Interval-valued fuzzy sets and 'compensatory and', *Fuzzy Sets and Systems* **51** (1992) 295-307.

[23] I. B. Turksen, Non-specificity and interval-valued fuzzy sets, *Fuzzy Sets and Systems* **80** (1996) 87-100.

[24] I. B. Turksen, Fuzzy normal forms, Fuzzy Sets and Systems **69** (1995), 319-346.

[25] I. B. Turksen, Universal truth tables and normal forms, *IEEE Trans. on Fuzzy Systems*, **6** (2) (1998) 295-303.

[26] I. Q. Zuo, I. B. Turksen, H. T. Nguyen, and V. Kreinovich, In expert systems, even if we fix AND/OR operations, a natural answer to a composite query is the interval of possible degrees of belief, *Reliable Computing*, Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, February 1995, 236-240.