Introduction to Fuzzy Sets and Fuzzy Logic

Luca Spada

Department of Mathematics and Computer Science
University of Salerno

www.logica.dmi.unisa.it/lucaspada

REASONPARK. Foligno, 17 - 19 September 2009

Contents of the Course

PART I

Contents of the Course

PART II

Contents of the Course

PART III

Contents of the Course

PART IV

Part I

Fuzzy sets

Contents of part I

There are no whole truths; all truths are half- truths. It is trying to treat them as whole truths that plays the devil.

- Alfred North Whitehead

What Fuzzy logic is?

Fuzzy logic studies reasoning systems in which the notions of truth and falsehood are considered in a graded fashion, in contrast with classical mathematics where only absolutly true statements are considered.

From the *Stanford Encyclopedia of Philosophy*:

> *The study of fuzzy logic can be considered in two different points of view: in a narrow and in a broad sense.*

Fuzzy logic in broad sense

Fuzzy logic in broad sense serves mainly as apparatus for fuzzy control, analysis of vagueness in natural language and several other application domains.
It is one of the techniques of soft-computing, i.e. computational methods tolerant to suboptimality and impreciseness (vagueness) and giving quick, simple and sufficiently good solutions.

- Klir, G.J. and Yuan, B. Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall (1994)
- Nguyen, H.T. and Walker, E. A first course in fuzzy logic. CRC Press (2006)
- Novak, V. and Novbak, V. Fuzzy sets and their applications. Hilger (1989)
- Zimmermann, H.J. Fuzzy set theory–and its applications. Kluwer Academic Pub (2001)

Fuzzy logic in the narrow sense

Fuzzy logic in the narrow sense is symbolic logic with a comparative notion of truth developed fully in the spirit of classical logic (syntax, semantics, axiomatization, truth-preserving deduction, completeness, etc.; both propositional and predicate logic).
It is a branch of many-valued logic based on the paradigm of inference under vagueness.

- Cignoli, R. and D'Ottaviano, I.M.L. and Mundici, D. Algebraic foundations of many-valued reasoning. Kluwer Academic Pub (2000)
- Gottwald, S. A treatise on many-valued logics. Research Studies Press (2001)
- Hajek, P. Metamathematics of fuzzy logic. Kluwer Academic Pub (2001)
- Turunen, E. Mathematics behind fuzzy logic. Physica-Verlag Heidelberg (1999)

Fuzzy sets and crisp sets

In classical mathematics one deals with collections of objects called (crisp) sets.
Sometimes it is convenient to fix some *universe* U in which every set is assumed to be included. It is also useful to think of a set A as a function from U which takes value 1 on objects which belong to A and 0 on all the rest.
Such functions is called the characteristic function of A, χ_A:

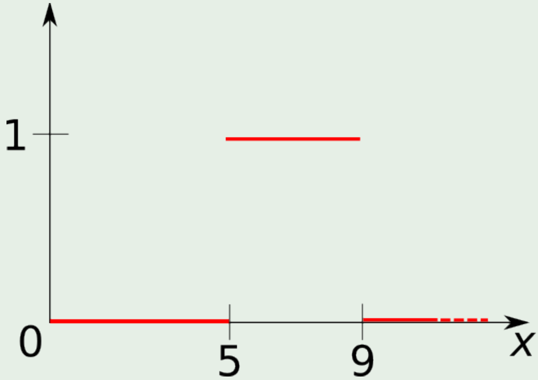
$$\chi_A(x) =_{def} \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

So there exists a bijective correspondence between characteristic functions and sets.

Crisp sets

Example

Let X be the set of all real numbers between 0 and 10 and let $A = [5, 9]$ be the subset of X of real numbers between 5 and 9. This results in the following figure:



Fuzzy sets

Fuzzy sets generalise this definition, allowing elements to belong to a given set with a certain degree.
Instead of considering characteristic functions with value in $\{0, 1\}$ we consider now functions valued in $[0, 1]$.

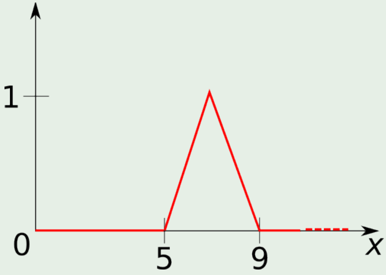
A fuzzy subset F of a set X is a function $\mu_F(x)$ assigning to every element x of X the degree of membership of x to F:

$$x \in X \mapsto \mu_F(x) \in [0, 1].$$

Fuzzy set

Example (Cont.d)

Let, as above, X be the set of real numbers between 1 and 10. A description of the fuzzy set of real numbers close to 7 could be given by the following figure:



Operations between sets

In classical set theory there are some basic operations defined over sets.

Let X be a set and $\mathscr{P}(X)$ be the set of all subsets of X or, equivalently, the set of all functions between X and $\{0,1\}$.

The operation of union, intersection and complement are defined in the following ways:

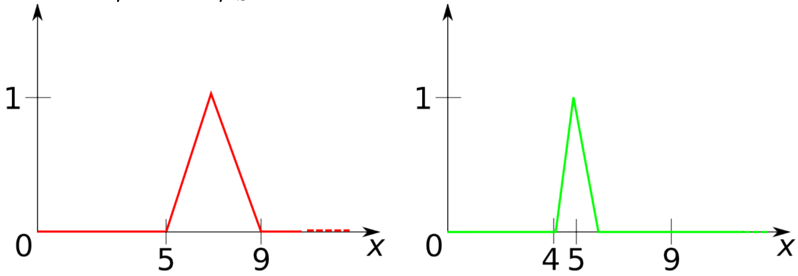
$$\begin{aligned} A \cup B &= \{x \mid x \in A \text{ or } x \in B\} \text{ i.e. } & \chi_{A\cup B}(x) &= \max\{\chi_A(x), \chi_B(x)\} \\ A \cap B &= \{x \mid x \in A \text{ and } x \in B\} \text{ i.e. } & \chi_{A\cap B}(x) &= \min\{\chi_A(x), \chi_B(x)\} \\ A' &= \{x \mid x \notin A\} \text{ i.e. } & \chi_{A'}(x) &= 1 - \chi_A(x) \end{aligned}$$

Operations between fuzzy sets: union

The law

$$\chi_{A\cup B}(x) = \max\{\chi_A(x), \chi_B(x)\}.$$

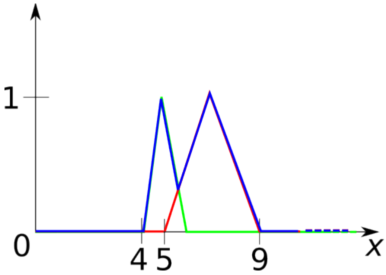
gives us an obvious way to generalise union to fuzzy sets.
Let F and S be fuzzy subsets of X given by membership functions μ_F and μ_S:



Operations between fuzzy sets: union

We set

$$\mu_{F\cup S}(x) = \max\{\mu_F(x), \mu_S(x)\}$$



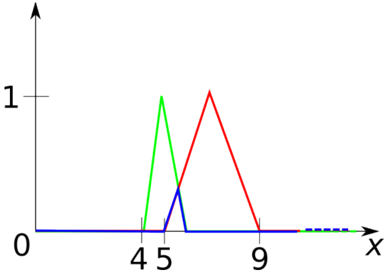
Operations between fuzzy sets: intersection

Analogously for intersection:

$$\chi_{A\cap B}(x) = \min\{\chi_A(x), \chi_B(x)\}.$$

We set

$$\mu_{F\cap S}(x) = \min\{\mu_F(x), \mu_S(x)\}$$



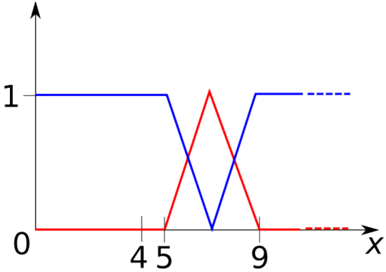
Operations between fuzzy sets: complement

Finally the complement for characteristic functions is defined by,

$$\chi_{A'}(x) = 1 - \chi_A(x).$$

We set

$$\mu_{F'}(x) = 1 - \mu_F(x).$$



Operations between fuzzy sets 2

Let's go back for a while to operations between sets and focus on intersection.
We defined operations between sets inspired by the operations on characteristic functions.
Since characteristic functions take values over $\{0, 1\}$ we had to choose an extension to the full set $[0, 1]$.

It should be noted, though, that also the product would do the job, since on $\{0, 1\}$ they coincide:

$$\chi_{A \cap B}(x) = \min\{\chi_A(x), \chi_B(x)\} = \chi_A(x) \cdot \chi_B(x).$$

Operations between fuzzy sets 2

So our choice for the interpretation of the intersection between fuzzy sets was a little illegitimate.
Further we have

$$\chi_{A\cap B}(x) = \min\{\chi_A(x), \chi_B(x)\} = \max\{0, \chi_A(x) + \chi_B(x) - 1\}$$

It turns out that there is an infinity of functions which have the same values as the minimum on the set $\{0, 1\}$.
This leads to isolate some basic property that the our functions must enjoy in order to be good candidate to interpret the intersection between fuzzy sets.

t-norms

In order to single out these properties we look again back at the crisp case: It is quite reasonable for instance to require the fuzzy intersection to be commutative, i.e.

$$\mu_F(x) \cap \mu_S(x) = \mu_S(x) \cap \mu_F(x),$$

or associative:

$$\mu_F(x) \cap [\mu_S(x) \cap \mu_T(x)] = [\mu_F(x) \cap \mu_S(x)] \cap \mu_T(x).$$

Finally it is natural to ask that if we take a set μ_F *bigger* than μ_S than the intersection $\mu_F \cap \mu_T$ should be bigger or equal than $\mu_S \cap \mu_T$:

If for all $x \in X \mu_F(x) \geq \mu_S(x)$ then $\mu_F(x) \cap \mu_T(x) \geq \mu_S(x) \cap \mu_T(x)$

.

t-norms

Summing up the few basic requirements that we make on a function $*$ that candidates to interpret intersection are:

- To extend the $\{0,1\}$ case, i.e. for all $x \in [0,1]$.

$$1 * x = x \text{ and } 0 * x = 0$$

- Commutativity, i.e., for all $x, y, z \in [0,1]$,

$$x * y = y * x$$

- Associativity, i.e., for all $x, y, z \in [0,1]$,

$$(x * y) * z = x * (y * z),$$

- To be non-decreasing, i.e., for all $x_1, x_2, y \in [0,1]$,

$$x_1 \leq x_2 \text{ implies } x_1 * y \leq x_2 * y.$$

t-norms

Objects with such properties are already known in mathematics and are called t-norms.

Example

(i) Łukasiewicz t-norm: $x \odot y = \max(0, x + y - 1)$.

(ii) Product t-norm: $x \cdot y$ usual product between real numbers.

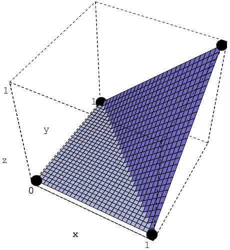
(iii) Gödel t-norm: $x \wedge y = \min(x, y)$.

(iv) Drastic t-norm: $x *_D y = \begin{cases} 0 & \text{if } (x,y) \in [0,1[^2 \\ \min(x,y) & \text{otherwise.} \end{cases}$

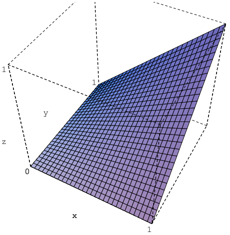
(v) The family of Frank t-norms is given by:

$$x *_F^{\lambda} y = \begin{cases} x \odot y & \text{if } \lambda = 0 \\ x \cdot y & \text{if } \lambda = 1 \\ \min(x,y) & \text{if } \lambda = \infty \\ \log_\lambda\left(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1}\right) & \text{otherwise.} \end{cases}$$

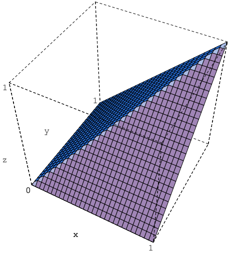
Examples



Łukasiewicz



Product



Minimum

Mostert and Shields' Theorem

An element $x \in [0,1]$ is idempotent with respect to a t-norm $*$, if $x * x = x$.
For each continuous t-norm $*$, the set E of all idempotents is a closed subset of $[0,1]$ and hence its complement is a union of a set $\mathcal{I}_{\text{open}}(E)$ of countably many non-overlapping open intervals. Let $[a,b] \in \mathcal{I}(E)$ if and only if $(a,b) \in \mathcal{I}_{\text{open}}(E)$. For $I \in \mathcal{I}(E)$ let $*|I$ the restriction of $*$ to I^2.

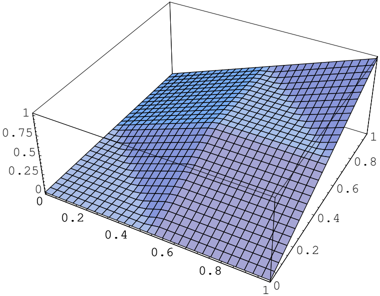
Theorem (Mostert and Shields, '57)

If $*, E, \mathcal{I}(E)$ are as above, then

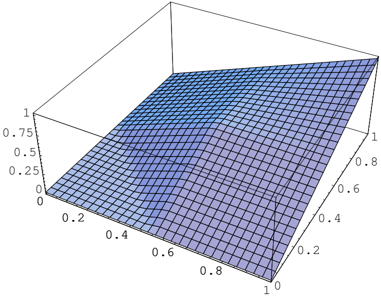
(i) *for each $I \in \mathcal{I}(E)$, $*|I$ is isomorphic either to the Product t-norm or to Łukasiewicz t-norm.*

(ii) *If $x, y \in [0,1]$ are such that there is no $I \in \mathcal{I}(E)$ with $x, y \in I$, then $x * y = \min(x,y)$.*

Examples



Two copies of Łukasiewicz



Łukasiewicz plus Product

Summing up

We have seen that it is possible to generalise the classic crisp sets to objects which naturally admits a notion of graded membership.

Also the fundamental operations between sets can be generalised to act on those new objects.

...but there is not just one of such generalisations.

A few natural requirements drove us to isolate the concept of t-norm as a good candidate for intersection.

There is a plenty of t-norms to choose from, but all of them can be reduced to a combination of three basic t-norms.

next aim: we have fuzzy properties and we can combine them, let us try to reason about them.

Part II

Mathematical logic

Contents of part II

What is a logic?

In mathematics a logic is a formal system which describes some set of rules for building new objects form existing ones.

Example

- Given the two words ab and bc is it possible to build new ones by substituting any b with ac or by substituting any c with a. So the words $aac, aaa, acc, aca, ..$ are deducible from the two given ones.
- The rules of chess allow to build new configurations of the pieces on the board starting from the initial one.
- The positions that we occupy in the space are governed by the law of physics.
- ...

Propositional logic

Propositional logic studies the way new sentences are derived from a set of given sentences (usually called axioms).

Example

If there is no fuel the car does not start.

There is no fuel in this car.

This car will not start.

If you own a boat you can travel in the see.

If you can travel in the see you can reach Elba island.

If you own a boat you can reach Elba island.

Propositional logic

Definition

The *objects* in propositional logic are sentences, built from an alphabet.
The language of propositional logic is given by:

- A set V of propositional variables (the alphabet): $\{X_1, \ldots, X_n, \ldots\}$
- Connectives: $\vee$, $\wedge$, $\neg$, $\rightarrow$ (conjunction, disjunction, negation and implication).
- Parenthesis (and).

Sentences of propositional calculus

Definition

Sentences (or formulas) of propositional logic are defined in the following way.

i) Every variable is a formula.
ii) If P and Q are formulas then $(P \vee Q)$, $(P \wedge Q)$, $(\neg P)$, $(P \rightarrow Q)$ are formulas.
iii) All formulas are constructed only using i) and ii).

Parenthesis are used in order to avoid confusion. They can be omitted whenever there is no risk of misunderstandings.

The axioms of propositional logic

The axioms of propositional logic are :

1. $(A \rightarrow (B \rightarrow A))$
2. $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
3. $((\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A))$

plus modus ponens: if $A \rightarrow B$ is true and A is true, then B is true.

A deduction is a sequence of instances of the above axioms and use of the rule modus ponens. The other connectives are defined as

- $A \vee B =_{def} \neg A \rightarrow B$
- $A \wedge B =_{def} \neg(\neg A \vee \neg B) = \neg(\neg\neg A \rightarrow \neg B)$

A deduction in propositional logic

Example

An instance of 1 gives

$$\neg X_1 \to (X_2 \to \neg X_1)$$

and an instance of 2 gives

$$\neg X_1 \to (X_2 \to \neg X_1) \to ((\neg X_1 \to X_2) \to (\neg X_1 \to \neg X_1)),$$

the use of modus ponens leads

$$(\neg X_1 \to X_2) \to (\neg X_1 \to \neg X_1)$$

which, by definition, can be written as

$$(\neg X_1 \to X_2) \to (X_1 \vee \neg X_1).$$

The semantics of a calculus

Just as happens in mathematics, where one makes calculations with numbers and those numbers *represent*, e.g. physical quantities, or amount of money, or points in a space, one can associate to a logic one (or several) interpretation, called the semantics of the logic.

Evaluations

Definition

An evaluation of propositional variables is a function

$$v : V \to \{0, 1\}$$

mapping every variable in either the value 0 (False) or 1 (True).

In order to extend evaluations to formulas we need to interpret connectives as operations over $\{0, 1\}$.
In this way we establish a homomorphism between the algebra of formulas (with the operation given by connectives) and the Boolean algebra on $\{0, 1\}$:

$$v : Form \to \{0, 1\}$$

The semantics of connectives

The evaluation v can be extended to a function $\mathbf{v}$ total on $Form$ by using induction:

- Variables: $\mathbf{v}(X_1) = v(X_1), \dots, \mathbf{v}(X_n) = v(X_n)$.
- $\mathbf{v}(P \wedge Q) = 1$ if both $\mathbf{v}(P) = 1$ and $\mathbf{v}(Q) = 1$.
  $\mathbf{v}(P \wedge Q) = 0$ otherwise.
- $\mathbf{v}(P \vee Q) = 1$ if either $\mathbf{v}(P) = 1$ or $\mathbf{v}(Q) = 1$.
  $\mathbf{v}(P \vee Q) = 0$ otherwise.
- $\mathbf{v}(P \rightarrow Q) = 0$ if $\mathbf{v}(P) = 1$ and $\mathbf{v}(Q) = 0$.
  $\mathbf{v}(P \rightarrow Q) = 1$ otherwise.
- $\mathbf{v}(\neg P) = 1$ if $\mathbf{v}(P) = 0$, and vice-versa.

A formula is a tautology if it only takes values 1. Tautologies are always true, for every valuation of variables.

Truth tables

The above rules can be summarized by the following tables:

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Conjunction

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Disjunction

A	B	$A \to B$
1	1	1
1	0	0
0	1	1
0	0	1

Implication

A	$\neg A$
1	0
0	1

Negation

Truth tables

Using the tables for basic connectives we can write tables for any formula:

Example

Let us consider the formula $X \to (Y \vee \neg X)$:

X	Y	$\neg X$	$Y \vee \neg X$	$X \to (Y \vee \neg X)$
1	1	0	1	1
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

Adding a third truth values

It is easy now to figure out how to extend the previous logical apparatus with a third truth value, say 1/2.
We keep the same syntactical structure of formulas: we just change the semantics.
Evaluations are now functions from the set of variables into $\{0, 1/2, 1\}$.
Accordingly to the definitions of truth tables for connectives we have different three-valued logics.

Kleene's logic

Kleene strong three valued logic is defined as

A and B	0	1/2	1
0	0	0	0
1/2	0	1/2	1/2
1	0	1/2	1

Conjunction

A or B	0	1/2	1
0	0	1/2	1
1/2	1/2	1/2	1
1	1	1	1

Disjunction

A implies B	0	1/2	1
0	1	1	1
1/2	1/2	1/2	1
1	0	1/2	1

Implication

A	not A
1	0
1/2	1/2
0	1

Negation

Łukasiewicz three valued logic

Łukasiewicz three valued logic is given by the following stipulation:

$A \odot B$	0	1/2	1
0	0	0	0
1/2	0	0	1/2
1	0	1/2	1

Conjunction

$A \oplus B$	0	1/2	1
0	0	1/2	1
1/2	1/2	1	1
1	1	1	1

Disjunction

$A \rightarrow B$	0	1/2	1
0	1	1	1
1/2	1/2	1	1
1	0	1/2	1

Implication

A	$\neg A$
1	0
1/2	1/2
0	1

Negation

We can also consider more than three values, and also infinitely many values, for example interpreting formulas in the real interval $[0, 1]$.

t-norms in logic

Here come back the t-norm functions defined earlier. Indeed one can think t-norms as possible semantics for the connective “conjunction”.

To rescue an implication from the t-norm, one can ask for desirable properties which relate the two connectives; a very important one is

$$(A \wedge B) \to C \cong A \to (B \to C).$$

Residuum

Proposition

Let $*$ be a continuous t-norm. Then, for every $x, y, z \in [0, 1]$, there is a unique operation satisfying the property:

$$(x * z) \leq y \quad \text{if and ony if} \quad z \leq (x \Rightarrow y)$$

and it is defined by

$$x \Rightarrow y = \max\{z \mid x * z \leq y\}$$

The operation $\Rightarrow$ is called the residuum of the t-norm $*$.

Example

The following are residua of the three main continuous t-norms:

	T-norm	**Residuum**
Ł	$x *_L y = \max(x+y-1, 0)$	$x \to_L y = \min(1, 1-x+y)$
P	$x *_P y = x \cdot y$	$x \to_P y = \begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise} \end{cases}$
G	$x *_G y = \min(x, y)$	$x \to_G y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}$

Negation

Once we have implication we can also define negation.
Indeed in classical logic a formula that implies a false formula is false itself. Hence

$$\neg A = A \to 0.$$

In case of Łukasiewicz t-norm, we have

$$\neg x = x \to 0 = \min(1, 1 - x + 0) = 1 - x$$

For Gödel and Product logic

$$\neg x = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

The complete picture

Completing the table

	T-norm $x * y$	**Residuum** $x \Rightarrow y$	**Negation** $\neg x$
Ł	$\max(x + y - 1, 0)$	$\min(1, 1 - x + y)$	$1 - x$
P	$x \cdot y$	$\begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{ow} \end{cases}$
G	$\min(x, y)$	$\begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}$	$\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{ow} \end{cases}$

So each of these logics is specified only by the t-norm.

Implication

We have seen that logical systems can be approached in two different ways:

- Specifying the syntax, that means fixing axioms and deduction rules
- Specifying the semantics, that means fixing the interpretation of formulas.

In the first approach the connective of implication plays a very important role, since it is the main ingredient of the basic deduction rule of Modus ponens:

If A and $A \to B$ are theorems, then B is a theorem.

Implication

On the other hand, the implication can be defined as an operation between sets by

$$A \to B = \neg A \cup B.$$

This means that if A and B are subsets of X, then $A \to B = X$ if and only if $A \subseteq B$ that is equivalent to say that $\chi_A(x) \leq \chi_B(x)$.

Going to the fuzzy level, implication takes care of order between membership values.

Later, we shall come back to implication in fuzzy logic.

Summing up

- We have seen that it is possible to formalise inside mathematics what a logical system is.
- Logical systems can be presented syntactically by specifying axioms and rules or semantically by giving devising the truth tables of the connectives.
- Just as happens in classical logic, where the concept of intersection corresponds to the connective hand, we have seen that t-norms can be used as generalised truth tables for conjunction.
- Clearly one can build any logical system whatsoever, but in order to obtain good deductive properties it is important to relate in some way the connectives

next aim: we wish now to push these methods to infinite values and show that syntax and semantic can be reunified back.

Part III

Many-valued logics

Contents of Part III

Many-valued logic

Definition

The language $\mathcal{L}$ for a propositional many-valued logic is given by a countable set $V = \{X_1, X_2, \ldots\}$ of propositional variables, a set C of connectives and a function $\nu : C \to \mathbb{N}$. A connective $\Diamond \in C$ is n-ary if $\nu(\Diamond) = n$.

Definition

The set Form of propositional formulas of a language $\mathcal{L} = \langle V, C, \nu\rangle$, is inductively defined as follows:

- Each $X \in V$ is a formula.
- If $\Diamond \in C$, $\nu(\Diamond) = k$ and $A_1, \ldots, A_k$ are formulas, then $\Diamond(A_1 \cdots A_k)$ is a formula.

Semantical interpretation

Definition

A many-valued propositional logic is a triple $\mathcal{P} = (S, D, F)$, where

- S is a non-empty set of truth-values,
- $D \subset S$ is the set of designated truth values,
- F is a (finite) non-empty set of functions such that for any $\Diamond \in C$ there exists $f_\Diamond \in F$ with $f_\Diamond : S^{\nu(\Diamond)} \to S$.

The functions in F are intended to give the interpretation of the connectives of the logic.

Finite and infinite valued logics

Definition

A triple (S, D, F) is an infinite-valued logic if it is a many valued logic and S is an infinite set. The triple (S, D, F) is a finite-valued logic if S is a finite set.

If $S_N = \{s_1, \ldots, s_N\}$ is a set such that $D \subseteq S_N \subseteq S$ and it is closed with respect to the functions in F, then the logic (S, D, F) naturally induces an N-valued logic (S_N, D, F') where each function in F' is the restriction to S_N of a function in F.

Propositional logic

The definition of a many-valued logic is in fact a generalisation of the classical case, so it should be not surprising that we can recover Propositional logic just by considering two truth values.

Propositional logic B can be written down as

$$\mathrm{B} = (\{0,1\}, \{1\}, \{f_\wedge, f_\neg\}),$$

where

$$f_\wedge(x,y) = \min(x,y) \quad \text{and} \quad f_\neg(x) = 1 - x.$$

Łukasiewicz (infinite-valued) logic

Just as noticed above we can choose a t-norm as interpretation of the conjunction. If we choose the Łukasiewicz t-norm and [0,1] as set of truth values, we get Łukasiewicz (infinite-valued) logic

$$\text{Ł}_\infty = ([0,1], \{1\}, \{f_\odot, f_\to\}).$$

where

$$f_\odot(x,y) = \max(0, x+y-1) \quad \text{and} \quad f_\to(x,y) = \min(1, 1-x+y).$$

Łukasiewicz finite valued logic

For each integer $n > 0$, let S_n be the set $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$.
Łukasiewicz (n+1)-valued logic is defined as

$$Ł_n = (S_n, \{1\}, \{f_\odot, f_\rightarrow\}),$$

where again

$$f_\odot(x, y) = \max(0, x+y-1) \quad \text{and} \quad f_\rightarrow(x, y) = \min(1, 1-x+y).$$

Gödel logic

If we choose the t-norm minimum as interpretation for the conjunction, then we get what is known as Gödel (infinite-valued) logic

$$\mathrm{G}_\infty = ([0,1], \{1\}, \{f_\wedge, f_{\neg_G}, f_{\to_G}\}),$$

where

$$f_{\to_G}(x,y) = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{if } x > y \end{cases} \quad \text{and} \quad f_{\neg_G}(x) = f_{\to_G}(x,0).$$

Product logic

Product logic is

$$\Pi_\infty = ([0,1], \{1\}, \{f_\cdot, f_{\neg_G}, f_{\to_\Pi}\}),$$

where

$$f_\cdot(x,y) = xy \quad \text{and} \quad f_{\to_\Pi}(x,y) = \begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise.} \end{cases}$$

Kleene's logic

Kleene strong three valued logic is defined as

$$\mathbf{K} = (\{0, 1/2, 1\}, \{1\}, \{f_\neg, f_\vee, f_{\to_k}\})$$

where

$$f_\neg(x) = 1 - x, \qquad f_\vee(x, y) = \max(x, y)$$

and

$f_{\to_k}$	0	1/2	1
0	1	1	1
1/2	1/2	1/2	1
1	0	1/2	1

Truth functionality

An important property of the logics described above is that the truth value of the compound formula $\diamondsuit(A, B)$ is determined only by the truth values of A and B. This is called truth-functionality and makes the study of the system much easier, for the interpretations are built in an inductive way.

Assignments

Definition

An assignment for $\mathcal{L}$ is a function $v : V \to S$. Any assignment can be uniquely extended to the whole set of formulas as follows:

$$v[\Diamond(A_1 \cdots A_k)] = f_{\Diamond}(v[A_1], \ldots, v[A_k]).$$

A formula A is satisfied in $\mathcal{L}$ by an assignment v if $v[A] \in D$. A formula A is valid in $\mathcal{L}$ (or a tautology), in symbols

$$\models_{\mathcal{L}} A,$$

if A is satisfied by all assignments, i.e. for every v, $v[A] \in D$.

Truth tables

So given a formula $A(X_1, \ldots, X_n)$, whose variables are among $X_1, \ldots, X_n$, the truth table of A is the function

$$f_A : [0,1]^n \to [0,1] \text{ such that } v(X_1), \ldots, v(X_n) \xmapsto{f_A} v[A].$$

In the following, if $\diamondsuit$ is a binary connective we will write $A_1 \diamondsuit A_2$ to denote $\diamondsuit(A_1, A_2)$ (or $f_\diamondsuit(A_1, A_2)$). Similarly if $\diamondsuit$ is unary.

Deducing tautologies

Truth tables provide a simple method to check whether a formula is a tautology or not. Unfortunately such a method is highly inefficient: one has to check that the truth table of a formula outputs a designated value for any possible assignment of its variables.

This is why one searches for an axiomatisation of a certain logical system. Axioms provide the first step towards an efficient automated deduction system.

Deductive systems

Definition

A deductive system Γ is given by:

- a set of formulas Φ,
- a set of rules Λ.

A deduction in Γ is a sequence of formulas which either belong to Φ or are obtained as application of rules in Λ to preceding formulas of the sequence.
A formula is derivable if it is the last formula of some deduction.
The derivable formulas are called theorems.

Syntax vs. Semantics

A deductive system may be used to find particular formulas in a language, in particular it may be used to find the tautologies of a logic.

There are two important issues here:

1. Only formulas which are tautologies must be derivable in the deductive system.
2. All the tautologies should be derivable from the deductive system.

The issue number 1 is often called soundness and the issue number 2 is called completeness.

Completeness

Definition

Let $\mathcal{P}$ be a many-valued logic and Γ a deductive system. We say that Γ is complete with regard to $\mathcal{P}$ if the set of theorems of Γ coincides with the set of tautologies of $\mathcal{P}$.

Definition

A deductive system Γ is standard complete when there exists a many-valued logic $\mathcal{P} = \langle [0,1], \{1\}, F \rangle$ such that Γ is complete with respect to $\mathcal{P}$. In this case the set of functions in F are called the standard interpretations of the connectives.

An example

The deductive system associated to Łukasiewicz logic has four axioms and one rule:

- $A \to (B \to A)$;
- $(A \to B) \to ((B \to C) \to (A \to C))$;
- $((A \to B) \to B) \to ((B \to A) \to A)$;
- $(\neg A \to \neg B) \to (B \to A)$.

The only rule is modus ponens.

Theorem (Chang 1957)

The deductive system for Łukasiewicz logic is standard complete.

Duality between syntax and semantics

Syntax	Semantics
Propositional variables	Truth values
Connectives	Functions
Theorems	Tautologies
Deductions	Truth tables

Axioms systems for many-valued logics

All the logics seen above have been characterised by means of a finite set of axioms and rules. What can be more surprising is that even the logic of all continuous t-norms can be presented in a particularly neat way. Such a system is called BL (for Basic Logic) and was introduced by P. Hájek.

Basic Logic

Definition

The system BL is given by the following set of axioms:

1. $(A \to B) \to ((B \to C) \to (A \to C))$,
2. $(A * B) \to A$,
3. $(A * B) \to (B * A)$,
4. $(A * (A \to B)) \to (B * (B \to A))$,
5. $(A \to (B \to C)) \to ((A * B) \to C)$,
6. $((A \to B) \to C) \to (((B \to A) \to C) \to C)$,
7. $\bot \to A$.

The only rule is modus ponens

$$\frac{A \quad A \to B}{B}$$

Standard completeness for BL

The completeness of BL is more subtle than the other logic seen above, indeed BL is the logic of all continuous t-norms and their residua.

Theorem

A formula is a theorem of the system BL if, and only if, it is a tautology for any logic $\langle [0,1], 1, \{*, \Rightarrow\} \rangle$ *where* $*$ *is a continuous t-norm and* $\Rightarrow$ *is its residum.*

Other forms of completeness

So the standard completeness guarantees that a certain deductive system can provide exactly all the tautologies of a given many-valued logic based on some (continuous) t-norm.

This is similar to the result in classical logic which states that the formulas derivable in its deductive system are exactly the formulas whose truth tables show only one's in their last column (i.e. tautologies of classical logic).

We will see later that other kinds of completeness are available and they provide deeper comprehension and powerful tools to study logics and deductive systems.

Basic logic

Basic logic can be defined as the logic of all continuous t-norms. On the other hand we have seen that every continuous t-norm is an locally isomorphic to either Łukasiewicz, Gödel or product t-norm.

In fact BL has other important features: the deductive systems of Łukasiewicz, Gödel or product logic can be obtained from the BL system by adding just one simple axiom.

More precisely we have that:

BL	+	$\neg\neg A = A$ (involution)	=	Łukasiewicz logic
BL	+	$A * A = A$ (idempotency)	=	Gödel logic
BL	+	$\neg A \vee ((A \rightarrow (A * B)) \rightarrow B)$ (weak cancellation)	=	Product logic.

SBL

Negation as defined for Gödel and Product logic can be axiomatized by the axiom

$$A \vee \neg A. \tag{S}$$

The extension of BL obtained by adding (S) as an axiom is called SBL (Strict Basic Logic).

This axioms doesn't hold, in general, with the involutive negation $\neg x = 1 - x$.

MTL

The continuity of a t-norm is required in order to guarantee the existence of the corresponding residuum.
In fact, what is equivalent to the existence of a residual operation is just left-continuity. So, just as BL is the logic of all

continuous t-norms, the monoidal t-norm based logic (MTL) is the logic of all left-continuous t-norms.
MTL is obtained from BL by just dropping a single axiom.

Other logics

Partially matching the hierarchy of schematic extensions of BL, there is a similar pattern for MTL and stronger logics given by schematic extensions.

In particular in the literature we find:
Involutive MTL: IMTL $=$ MTL $+ \neg\neg A \rightarrow A$,
Strict MTL: SMTL $=$ MTL $+ (A \vee \neg A)$,
Product MTL: ΠMTL $=$ MTL $\neg A \vee ((A \rightarrow (A \cdot B)) \rightarrow B)$
...but MTL $+ A \rightarrow (A * A)$ is just Gödel logic.

Additional logics in the MTL hierarchy are

the Weak nilpotent minimum:
WNM $= \mathrm{MTL} + \neg(A * B) \vee ((A \wedge B) \to (A * B))$
and the Nilpotent minimum:
NM $= \mathrm{WNM} + \neg\neg A \to A$.
Standard completeness has been proved for all logics introduced.

ŁΠ logic

Weakening the axioms system yelds logics that are more general. But in practical cases one needs sufficiently expressive logic, in order to describe the state of facts.

Another logic which has been studied extensively in the literature is ŁΠ logic.
ŁΠ is obtained obtained by combining connectives from both Łukasiewicz and Product logics.
The expressive power of ŁΠ is quite remarkable:

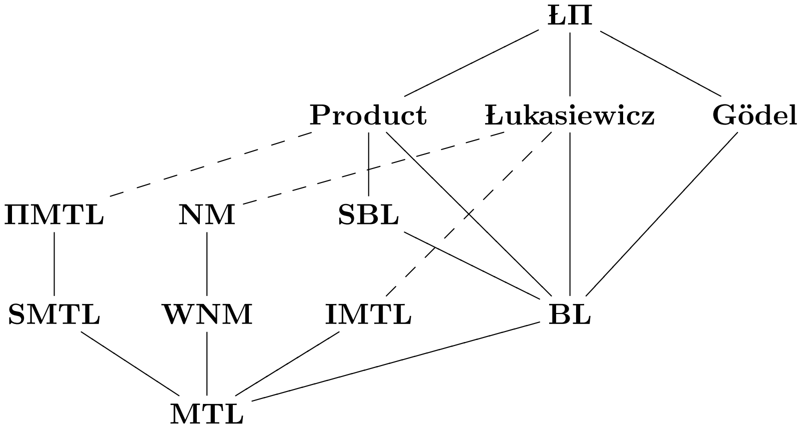
Theorem

Łukasiewicz, Product and Gödel logics are interpretable in ŁΠ

Theorem

Any many-valued logic based on a continuos t-norm with a finite number of idempotents is interpretable in ŁΠ *logic.*

The forest of many-valued logics



Tautology problem

The tautology problem of a given logic $\mathcal{L}_*$ consists of deciding whether a given formula A is a tautology or not, in symbols

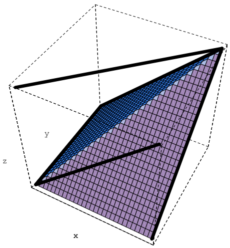
$$A \in Taut(\mathcal{L}_*)?$$

The problem is dual to find a countermodel for a formula. Finding a countermodel for a formula φ in $\mathcal{L}_*$ amounts to find an assignment of propositional variables giving A a value less than 1.

In the case of Łukasiewicz logic one can exploit some geometrical property of the points where formulas can take value 1. This allows to prove that the complexity of the tautology problem is co-NP-complete by showing that for any formula which is not a tautology a suitably small countermodel always exists.

Tautologies in Gödel

By looking at the general form of functions associated with Gödel formulas: we can establish that in order to check if they are equal to one it is enough to check in points with coordinates in $\{0, 1/3, 2/3, 1\}$.



Countermodels of many-valued logics

A Łukasiewicz formula A is not a tautology then A fails to be a tautology of a finite-valued Łukasiewicz logic with a number of truth-values that is polynomially bounded by the length of φ. Then

Theorem

The satisfiability problem for Łukasiewicz logic is NP-complete.

Similar results hold for Gödel and Product logic.

Theorem

The tautology problem for BL is coNP-complete.

Truth tables for Łukasiewicz logic

Which kind of truth tables are associated with formulas of Łukasiewicz infinite-valued logic?
We have to consider all the possible combinations of conjunction, implication and negation.
It is easy to check that the interpretation of any formula gives rise to a function that is continuous (all connectives are continuous and combining one with the other we still have a continuous function.)
Further they are composed by linear pieces, with integer coefficients.

Functional representation

Definition

A McNaughton function $f : [0,1]^n \to [0,1]$ is a continuous, piecewise linear function such that each piece has integer coefficients. In other words, there exist finitely many polynomials $p_1, \ldots, p_{m_f}$ each p_i being of the form

$$p_i(x_1, \ldots, x_n) = a_{i1}x_1 + \ldots + a_{in}x_n + b_i \text{ with } a_{i1} \ldots, a_{in}, b_i \text{ integers,}$$

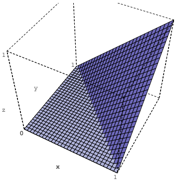
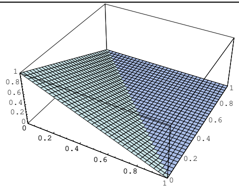
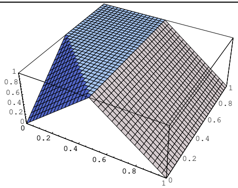
such that, for any $\mathbf{x} \in [0,1]^n$, there exists $j \in \{1, \ldots, m_f\}$ for which $f(\mathbf{x}) = p_j(\mathbf{x})$.

McNaughton theorem

A function $f : [0,1]^n \to [0,1]$ is a truth table of a Łukasiewicz formula if and only if it is a McNaughton function.

McNaughton Theorem

If f is a continuous piecewise linear function, such that every linear piece has integer coefficients, then there exists a Łukasiewicz formula A such that f is the truth table of A.

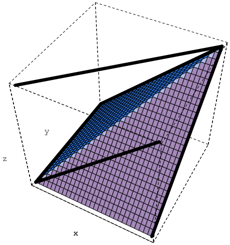
		
Łukasiewicz conjunction	Implication	$2(\neg x \oplus \neg(y \oplus 2x))$

Gödel logic

In Gödel logic we can express the characteristic function of 0 ($\neg x$), but not the characteristic function of 1.

$$x \wedge y = \min(x, y) \qquad x \rightarrow_{\wedge} y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}$$

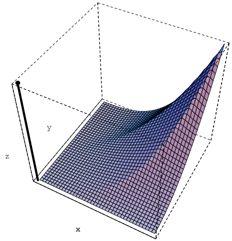
$$\neg x = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{ow} \end{cases}$$



Product Logic

Truth tables of Product formulas have a discontinuity in 0.

$$x \cdot y \quad x \rightarrow. y = \begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise} \end{cases} \qquad \neg x = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{ow} \end{cases}$$



It is a piecewise monomial function, with possible discontinuities in 0.

Summing up

- We have presented a mathematical formalism which allows to describe reasoning with more than two truth values.
- We have seen that such formalisms can be proved to catch exactly the semantics that we have in mind.
- This framework is quite general and provide a neat characterisations of many tools used in fuzzy logic.
- At the same time the simple axiomatisations given allow to build automated deductive system for those logics.
- Finally, we have seen that, when dealing with many valued logics, the truth functions acquire a geometrical interest.

next aim: we only scratched the surface of the abundance of results discovered in many-valued logic, we will see now other interpretations and some applications.

Part IV

Advances in many-valued logic

Contents of part IV

The Ulam game

In his book “Adventures of a Mathematician”, Ulam describes the following game between two players A and B:

> *Player B thinks of a number between one one million (which is jus t less than 2^{20}). Player A is allowed to ask up to twenty questions, to each of which Player B is supposed to answer only yes or no. Obviously the number can be guessed by asking first: Is the number in the first half million? then again reduce the reservoir of numbers in the next question by one-half, and so on. Finally the number is obtained in less than $\log_2(1'000'000)$.*
> *Now suppose Player B were allowed to lie once or twice, then how many questions would one need to get the right answer?*

Optimal strategies

Here we model questions Q as subsets of X.

The problem is to find strategies for A that minimize the number of questions in the worst cases, i.e. whatever is the initial choice of the secret number and whatever is the behavior of B.

In case all questions are asked independently of the answers, optimal searching strategies in this game are the same as optimal k-error-correcting coding strategies.

Limits of classical logic

Now, in the particular case when $k = 0$ (corresponding to the familiar game of Twenty Questions) the state of knowledge of Player A is represented by the classical conjunction of all the pieces of information obtained from the answers of Player B.

In case $k > 0$ classical logic no longer yields a natural formalization of the answers.

Indeed

(a) The conjunction of two equal answers to the same repeated question need not be equivalent to a single answer. Thus, the classical idempotence principle fails. mypause

(b) The conjunction of two opposite answers to the same repeated question need not lead to contradiction.

Łukasiewicz logic in action

When inconsistent information are added to the knowledge, $(k+2)$-valued Łukasiewicz logic comes in play.
Player A can record the current knowledge of the secret number by taking the Łukasiewicz conjunction of the pieces of information contained in the answers of B.

Answers as fuzzy sets

More precisely, let $L = S_{k+1}$. For every question $Q \subseteq X$, the positive L-answer to Q is the L-set $Q^{yes} : X \to L$ given by

$$Q^{yes}(y) = \begin{cases} 1, & \text{if } y \in Q; \\ \frac{k}{k+1}, & \text{if } y \notin Q. \end{cases}$$

Elements $y \in X$ such that $Q^{yes}(y) = 1$ are said to satisfy L-answer Q^{yes}; the remaining elements falsify the answer. The dependence of Q^{yes} and Q^{no} on the actual value of k is tacitly understood.

Describing the state of knowledge

We define the L-subset $\mu_n : X \to L$ of possible numbers resulting after a sequence of questions $Q_1, \dots, Q_n$ with their respective answers $b_1, \dots, b_n$ ($b_i \in \{yes, no\}$), is the Łukasiewicz conjunction

$$\mu_n = Q_1^{b_1} \odot \cdots \odot Q_n^{b_n}.$$

At the stage 0, in which now question has been asked yet, the L-subset is the function constantly equal to 1 over X.

Initially all numbers are possible and have “truth value” 1 (we have no information), at the final step only one number is possible (we have maximum information).

Describing the state of knowledge

Proposition

Let $x \in X$ and let μ_n be the L-subset of possible numbers resulting after the questions $Q_1, ..., Q_n$ and the answers $b_1, \ldots, b_n$ $(b_i \in \{yes, no\})$. Then:

$$\mu_n(x) = \begin{cases} 1 - \frac{i}{k+1}, & \text{if } x \text{ falsifies precisely} \\ & i \leq k+1 \text{ of the } Q_1^{b_1}, ..., Q_n^{b_n} \\ & \\ 0 & \text{otherwise.} \end{cases}$$

Player A will know the correct answer when the L-subset μ_n of possible numbers becomes an L-singleton. More precisely, when there is $a \in X$ such that $\mu_n(x) = 0$ for all $x \in X$ such that $x \neq a$.
Then a is the secret number.

Pavelka-style Fuzzy logic

Fuzzy logic in the style of Pavelka enables to build deductions which prove partially true sentences from partially true assumptions.

The necessary device to build such a calculus is to endow the syntax with a new truth constant $\bar{r}$ for each rational number r. Moreover the following set of bookkeeping axioms must be added to the system:

$$(\overline{r \rightarrow s}) \leftrightarrow (\bar{r} \rightarrow \bar{s})$$

The logic introduced by Pavelka was proved to be a conservative extension of Łukasiewicz first order logic. It can be obtained form Łukasiewicz first order logic by just adding the axioms and constants above.

Pavelka-style Completeness

For Rational Łukasiewicz logic a strong kind of completeness holds:
Let us say that a valuation v is a model of a set of sentences Γ if $v(\gamma) = 1$ for all $\gamma \in \Gamma$

- The Truth degree of A w.r.t. a theory Γ is given by

$$||A||_\Gamma = \inf\{v(A) \mid v \text{ is a model of } \Gamma\}$$

- The Provability degree of A w.r.t. a theory Γ is given by

$$|A|_\Gamma = \sup\{r \mid \text{ from } \Gamma \text{ is derivable } r \to A\}$$

- The Completeness theorem, in Pavelka style, states:

$$|A|_\Gamma = ||A||_\Gamma$$

Fuzzy Controls

Let us consider a simplification of a control system.
Suppose that the input and output sets of a control function are intervals $[a, b]$, $[c, d]$ of $\mathbb{R}$.
We can normalize these intervals and we can assume that the control function is a map $f : [0, 1]^n \to [0, 1]$.

Let $n = 1$. Knowing the behavior of the control function in points $(x_i, f(x_i)) \in \mathbb{Q}^2$ we want to describe a set of rules as a two variables Łukasiewicz formula A. The truth table f_A of A will be a "two-dimensional" approximation of f.
f_A represents for every pair (x, y) how much the value y is appropriate to the input value x in order to achieve the control of the system.
Then the process of defuzzification associates f_A with the final control function from $[0, 1]$ into $[0, 1]$.

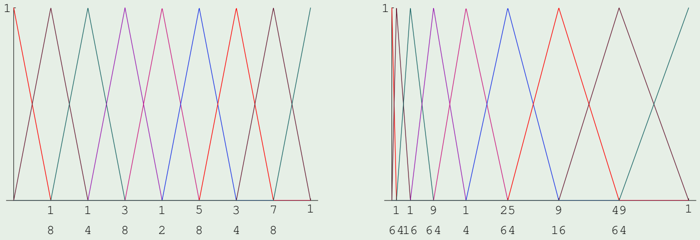
Fuzzy presentations

Let $f : [0, 1] \to [0, 1]$ be a function and let $T \subseteq [0, 1]^2$ be a finite set of couples $(x_i, f(x_i)) \in \mathbb{Q}^2$ of rational numbers, with $i = 1, \ldots, n$.
Then we can construct by means of Łukasiewicz formulas, the two fuzzy presentations $\{A_i\}_{i=1}^n$ and $\{B_i\}_{i=1}^n$ of sets $\{x_1, \ldots, x_n\}$ and $\{f(x_1), \ldots, f(x_n)\}$.

Fuzzy presentations

Example

Fuzzy presentations for sets $Ł_8 = \{0, 1/8, \ldots, 7/8, 1\}$ and $Ł_8^2 = \{0, 1/64, 4/64, \ldots, 49/64, 1\}$ $(f(x) = x^2)$:



Fuzzy presentations

Hence we use two sets of Łukasiewicz formulas to fuzzify the information given by the couples $(x_i, f(x_i))$.
The fuzzy presentations are the simplest way to make this fuzzification.
We can also take care of external information about the function, say the opinion of an expert or the results obtained in a previous computation: for example, if we do not want to give much importance to the information carried by $(x_i, f(x_i))$ we can consider formulas δA_i and τB_i (for suitable operator δ and τ) instead of formulas A_i and B_i.

Fuzzy rules

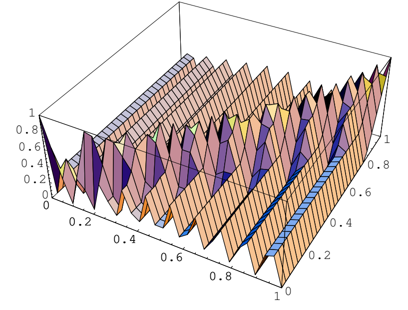
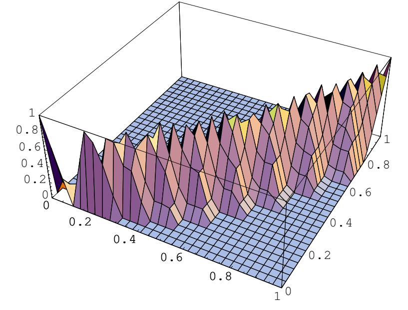
Fuzzy rules have of the form:

$$\begin{cases} \textit{IF } X \textit{ is } A_1 \textit{ THEN } Y \textit{ is } B_1 \\ \dots \\ \textit{IF } X \textit{ is } A_n \textit{ THEN } Y \textit{ is } B_n. \end{cases} \tag{1}$$

Formulas interpreting these rules can have different forms:

$$\bigwedge_{i=1}^{n}(A_i \to B_i) \qquad \text{or} \qquad \bigodot_{i=1}^{n}(A_i \to B_i).$$

In the following picture, the graphics of the truth tables of such formulas are depicted for the case $f(x) = x^2$ and $n = 8$, with respect to fuzzy presentations as before.

	
$\bigwedge_{i=1}^{n}(\varphi_i \to \psi_i)$	$\bigodot_{i=1}^{n}(\varphi_i \to \psi_i)$

On the other hand, we can transform (1) into

$$\begin{cases} \text{Either } X \text{ is } A_1 \text{ and } Y \text{ is } B_1 \\ \text{or } \dots \\ \text{or } X \text{ is } A_n \text{ and } Y \text{ is } B_n, \end{cases} \quad (2)$$

and then we can interpret this system with four different formulas:

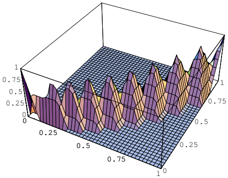
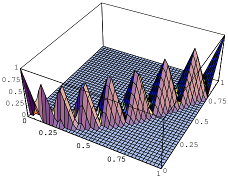
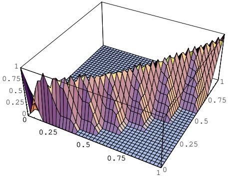
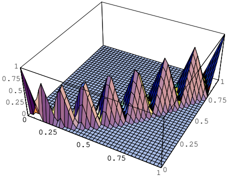
$$\bigvee_{i=1}^{n}(A_i \wedge B_i) \qquad \bigvee_{i=1}^{n}(A_i \odot B_i)$$

$$\bigoplus_{i=1}^{n}(A_i \wedge B_i) \qquad \bigoplus_{i=1}^{n}(A_i \odot B_i).$$

Since for every $x, y \in [0,1]$, we have $x \odot y \leq x \wedge y$ and $x \vee y \leq x \oplus y$, then we can partially compare the above formulas.

$$\bigoplus_{i=1}^{n}(A_i \wedge B_i) \geq \bigvee_{i=1}^{n}(A_i \wedge B_i) \geq \bigvee_{i=1}^{n}(A_i \odot B_i)$$

$$\bigvee_{i=1}^{n}(A_i \odot B_i) \leq \bigoplus_{i=1}^{n}(A_i \odot B_i) \leq \bigoplus_{i=1}^{n}(A_i \wedge B_i).$$

	
$\bigvee_{i=1}^{n}(A_i \wedge B_i)$	$\bigvee_{i=1}^{n}(A_i \odot B_i)$
	
$\bigoplus_{i=1}^{n}(A_i \wedge B_i)$	$\bigoplus_{i=1}^{n}(A_i \odot B_i)$

Defuzzyfication

In all fuzzy control systems that we have considered, an approximation of the function f can be obtained by the defuzzification of f_A.

Accordingly to the kind of defuzzification used, one can make a further choice on the connectives involved. The capacity of the control system to approximate f depends strongly on regularity of the function f, such as continuity and Lipschitz-like hypothesis.

Relations and order

A binary relation R on a set S is a subset of $S \times S$. If $(x, y) \in R$ we write xRy.

A binary relation R is called order relation if:

- For every $x \in S$, xRx (reflexivity)
- If xRy and yRx then $x = y$ (antisymmetry)
- If xRy and yRz then xRz. (transitivity)

Order relations are denoted by $\leq$. An order relation over a set S is total (or linear) if for every $x, y \in S$ either $x \leq y$ or $y \leq x$.

Binary relations that are reflexive, transitive and symmetric (i.e., xRy if and only if yRx) are called equivalence relations.

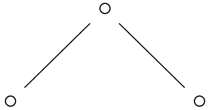
A partially order set (poset) is a set equipped with an order relation.

Examples

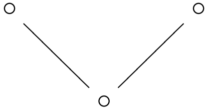
Orders can be described by diagrams. For example orders between three elements are the following ((a) is a chain):

a)

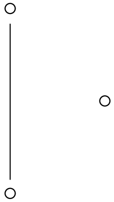
b)



c)



d)

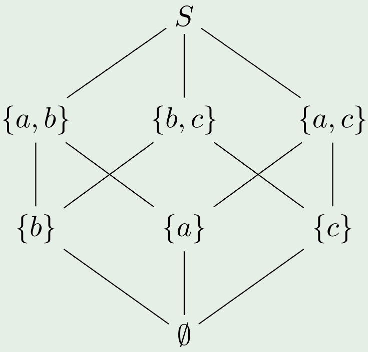


e)

More examples

Example

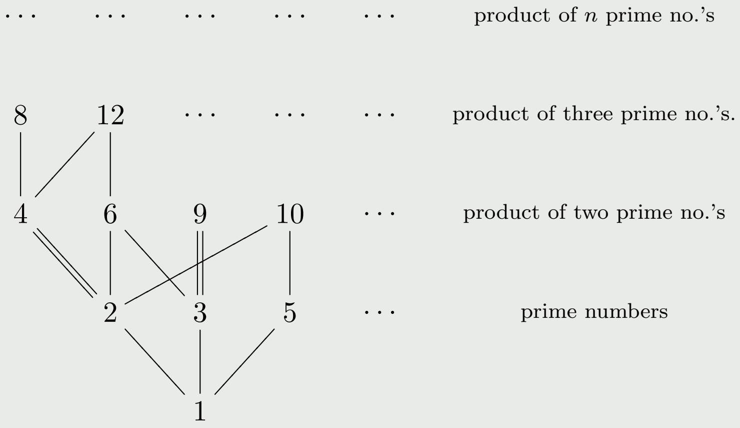
1. The sets $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ with usual order are examples of totally ordered posets (chains).
2. If S is a set, say $S = \{a, b, c\}$, then the set $\mathscr{P}(S)$ of all subsets of S with the order given by inclusion is a poset that is not a chain.



Some examples

In $\mathbb{Z}$ we can consider the order relation / defined by x/y if, and only if, x divides y.

A graphical representation



Lattices

Given a set X let $\sup X$ be the least element that is greater than any other element of X. Similarly let $\inf X$ be the greatest element that is smaller than any other element of X.

Definition

A lattice is a poset R such that for every pair $x, y \in R$ there exist $\inf\{x, y\} \in R$ and $\sup\{x, y\} \in R$. In lattice theory sup and inf of finite sets are often denoted by $x \vee y$ and $x \wedge y$, repsectively. A lattice R is complete if for every subset X of R there exists $\sup X$ and $\inf X$.

It is easy to check that every finite lattice is complete.

Examples of lattices

Example

- Every chain is a lattice, where $\vee$ and $\wedge$ coincide with maximum and minimum.
- $\mathbb{R}$ with usual order is a lattice that is not complete.
- The poset $\mathscr{P}(S)$ is a lattice.

More on lattices

A lattice L containing $0 = \inf L$ e $1 = \sup L$ is called bounded.
If L is bounded, the complement of $a \in L$ is an element $b \in L$ such that

$$a \vee b = 1 \quad \text{and} \quad \wedge b = 0.$$

A lattice L is distributive if the following holds for every $a, b, c \in L$:

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

In a distributive lattice, if the complement of a exists it is unique and it is denoted by a'.
A Boolean lattice is a distributive lattice with 0 and 1 such that every element has a complement.

Boolean algebras

Usually Boolean lattices are seen with the algebraic structure: a Boolean algebra is a structure $(B, \wedge, \vee, 0, 1, ')$ such that

(i) $(B, \wedge, \vee)$ is a distributive lattice.

(ii) $a \vee 0 = a$ and $a \wedge 1 = a$ for every $a \in B$.

(iii) $a \vee a' = 1$ and $a \wedge a' = 0$ for every $a \in B$.

Example

The set $\{0, 1\}$ equipped with operations $\wedge, \vee$ and $\neg$ is a Boolean algebra.
The lattice $\mathscr{P}(S)$ is a Boolean algebra.

Theorem

An equation φ holds in all Boolean algebras if, and only if, it holds in the Boolean algebra $\{0, 1\}$.

Propositional logic and Boolean algebras

The basic relationship between propositional classical logic and Boolean algebras is that the set $\{0, 1\}$ in which we evaluate the formulas is a Boolean algebra.
Further, the evaluation map sends every formula in a value 0 or 1 in such a way that the connectives are mapped in the boolean operations. So

> *evaluations are morphisms of the boolean algebra of formula where operations are the connectives, and the boolean algebra* $\{0, 1\}$.

Such an algebraic representation extends also fuzzy logic.

Residuated lattices

One very straightforward solution could be to consider residuated lattices:
A bounded lattice is said to be residuated if it is equipped with a couple of operations $(*, \rightarrow)$ (called the adjoint couple) such that $*$ is associative, commutative, $x * 1 = x$ and

$$x \rightarrow y * z \text{ iff } x \leq y \rightarrow z$$

Note that Boolean algebras are residuated lattice when we take $(\wedge, \rightarrow)$ as adjoint couple, where $x \rightarrow y = \neg x \vee y$.

BL and MTL algebras

But residuated lattices are structures still too general.
The algebraic counterpart of Basic Logic and MTL are BL-algebras and MTL-algebras respectively.

$$\text{MTL- algebras} = \text{Residuated lattice} + \quad ((x \to y) \vee (y \to x) = 1)$$

pre-linearity

$$\text{BL-algebras} = \text{MTL} + \quad (x \wedge y = x * (x \to y))$$

divisibility

BL and MTL algebras

Example

- Any Boolean algebra is both a BL algebra and a MTL algebra.
- The real interval [0,1] endowed with a continuous t-norm and its residum is a BL algebra.
- The real interval [0,1] endowed with a left continuous t-norm and its residum is a MTL algebra.

Interpretations

So the notion of evaluation given before is just a particular instance of what we call interpretation

Definition

Let $\mathcal{P} = (S, D, F)$ be a many-valued logic and $\mathbf{K}$ a class of residuated lattices with additional operation such that for any connective in $\diamond \in F$ there is an operation $\diamond^*$ in the algebras of $\mathbf{K}$ with the same ariety. Then a function i form the formulas of $\mathcal{P}$ into an algebra $K \in \mathbf{K}$ is called an interpretation if:

- For any propositional variable X, $i(X) \in K$;
- if $A \diamond B$ is a formula then $i(A \diamond B) = i(A) \diamond^* i(B)$

Algebraic Completeness

So the formulas of a logic can be interpreted as element of an algebra. Hence we can associate to any logic $\mathcal{P}$ a class of algebras $\mathbf{K}^{\mathcal{P}}$.

When dealing with the above presented logics one has a number of properties:

- Examples of algebras in $\mathbf{K}^{\mathcal{P}}$ are the unit interval $[0, 1]$ with truth functions of connectives of $\mathcal{P}$ as operations and the algebra of classes of provably equivalent formulas.
- If $A \in Form(\mathcal{P})$ is provable then $i(A) = 1$ is valid in all algebras of $\mathbf{K}^{\mathcal{P}}$.
- Each algebra in $\mathbf{K}^{\mathcal{P}}$ is a subalgebra of the direct product of some linearly ordered algebra.
- If $i(\varphi) = 1$ is valid in the algebra $[0, 1]$ then it is valid in all linearly ordered algebras, in particular in the algebra of classes of formulas, which means that A is a provable formula.

An example: Łukasiewicz logic

Originally, Łukasiewicz infinite-valued logic was axiomatized (using implication and negation as the basic connectives) by the following schemata:

- $A \to (B \to A)$
- $(A \to B) \to ((B \to C) \to (A \to C))$
- $((A \to B) \to B) \to ((B \to A) \to A)$
- $(\neg A \to \neg B) \to (B \to A)$
- $((A \to B) \to (B \to A)) \to (B \to A)$.

Chang and Meredith proved independently that the last axiom is derivable from the others.
In order to prove the completeness of this schemata of axioms with respect to semantics of the interval $[0, 1]$, Chang introduced MV-algebras.

MV-algebras

An MV-algebra is a structure $A = (A, \oplus, \neg, 0, 1)$ satisfying the following equations:

- $x \oplus (y \oplus z) = (x \oplus y) \oplus z$
- $x \oplus y = y \oplus x$
- $x \oplus 0 = x; \qquad x \oplus 1 = 1$
- $\neg 0 = 1 \ ; \qquad \neg 1 = 0$
- $\neg\neg x = x$
- $\neg(\neg x \oplus y) \oplus y = \neg(\neg y \oplus x) \oplus x.$

Each MV-algebra contains as a subalgebra the two-element boolean algebra $\{0, 1\}$.

MV-algebras

As proved by Chang, boolean algebras coincide with MV-algebras satisfying the additional equation $x \oplus x = x$ (idempotency). The set $B(A)$ of all idempotent elements of an MV-algebra A is the largest boolean algebra contained in A and is called the boolean skeleton of A.
The algebra $(A, \oplus, 0)$ and $(A, \odot, 1)$ are isomorphic via the map

$$\neg : x \mapsto \neg x.$$

Further any MV-algebra A is equipped with the order relation

$$x \leq y \quad \text{if and only if} \quad \neg x \oplus y = 1 \qquad (x \to y = 1).$$

MV-algebras turn out to coincide with those BL-algebras satisfying the equation $\neg\neg x = x$.

Examples

Example

(i) The set $[0, 1]$ equipped with operations

$$x \oplus y = \min\{1, x + y\}, \quad x \odot y = \max\{0, x + y - 1\}, \quad \neg x = 1 - x$$

is an MV-algebra.

(ii) For each $k = 1, 2, \ldots,$ the set

$$Ł_{k+1} = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\},$$

equipped with operations as before, is an MV-algebra.

The algebra of fuzzy sets

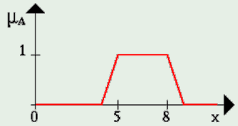
Example

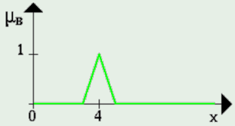
(iii) If X is any set and A is an MV-algebra, the set of functions $f : X \to A$ obtained by pointwise application of operations in A is an MV-algebra:

$$\begin{aligned} (\neg f)(x) &= \neg f(x) \\ (f \odot g)(x) &= f(x) \odot g(x) \end{aligned}$$

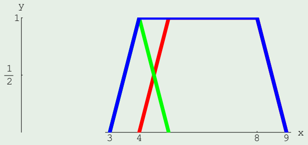
In particular, considering a set X and the MV-algebra $[0, 1]$ we obtain the MV-algebra of fuzzy sets where operations are no more minimum and maximum but Łukasiewicz operations:

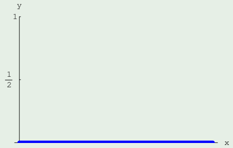
Example





$$\mu_{F\cup S}(x) = \mu_F(x) \oplus \mu_S(x) \quad \mu_{F\cap S}(x) = \mu_F(x) \odot \mu_S(x)$$





Another example

The set of all McNaughton functions (i.e., functions from $[0,1]^n$ into $[0,1]$ that are continuous and piece-wise linear, and such that each linear piece has integer coefficients), with operations obtained as pointwise application of operations as above, is an MV-algebra.
This is the free MV-algebra over n free generators.
McNaughton functions can be considered as very special case of fuzzy subsets of $[0,1]$.

Di Nola representation Theorem

Can we have for MV-algebras a result similar to Stone representation theorem for Boolean algebras?
The answer for the general case is rather complicated, but a simple results has been established for a class of MV-algebras:

Theorem

Any semi-simple MV-algebra is isomorphic to an algebra of fuzzy sets.

Theorem

Any MV-algebra is isomorphic to an algebra of fuzzy sets where the characteristic functions may have non-standard values.

Algebraic completeness for MV-algebras

Chang's Completeness Theorem states:

Theorem

The following are equivalent:

- *The formula A holds in the Łukasiewicz calculus,*
- *The equation* $i(A) = 1$ *holds in every MV-algebra*
- *The equation* $i(A) = 1$ *holds in the MV-algebra* $[0, 1]$ *equipped with operations* $x \oplus y = \min\{1, x + y\}$, $x \odot y = \max\{0, x + y - 1\}$ *and* $\neg x = 1 - x$.

This theorem was proved by Chang using quantifier elimination for totally ordered divisible abelian groups. There are several alternative proofs in literature: the syntactic proof by Rose and Rosser, the algebraic proof by Cignoli and Mundici and the geometric proof by Panti.

Relation with groups

Definition

A lattice-ordered group (ℓ-group) $G = (G, 0, -, +, \wedge, \vee)$ is an abelian group $(G, 0, -, +)$ equipped with a lattice structure $(G, \wedge, \vee)$ such that:

$$\text{for every } a, b, c \in G, \quad c + (a \wedge b) = (c + a) \wedge (c + b).$$

An element $u \in G$ is a strong unit of G if for every $x \in G$ there exists $n \in \mathbb{N}$ such that $nu \geq x$.

Theorem

There exists an equivalence functor $\mathbf{\Gamma}$ *from the category of ℓ-groups with strong unit to the category of MV-algebras:*

If G is an ℓ-group and u is a strong unit for G, the MV-algebra $\mathbf{\Gamma}(G, u)$ has the form $\{x \in G \mid 0 \leq x \leq u\}$ and operations are defined by $x \oplus y = u \wedge x + y$ and $\neg x = u - x$. If A is an MV-algebra we shall denote by G_A the ℓ-group corresponding to A via $\mathbf{\Gamma}$.

Gödel Logic and Gödel algebras

Gödel infinite-valued propositional logic G_∞ is the triple $([0,1], \{1\}, \{\wedge, \rightarrow_G\})$, where

$$\begin{aligned} x \wedge y &= \min(x, y) \\ x \rightarrow_G y &= \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases} \end{aligned}$$

Finite-valued Gödel propositional logics G_n were introduced to prove that intuitionistic propositional logic cannot be viewed as a system of finite-valued logic.
Dummett proved completeness of such system.

Gödel propositional logic can be defined as the fragment of intuitionistic logic satisfying the axiom $(\alpha \to \beta) \vee (\beta \to \alpha)$. Theorems of Gödel logic are exactly those formulas which are valid in every linearly ordered Heyting algebra, where Heyting algebras are the structure naturally associated with intuitionistic logic.
In Hájek framework, Gödel logic is obtained adding to axioms of Basic Logic the axiom

$$(G1) \qquad \varphi \to (\varphi * \varphi)$$

stating the idempotency of $*$.
Gödel algebras are BL-algebras satisfying the identity $x * x = x$.

Product Logic and PL algebras

Product Logic Π is the triple $([0,1], \{1\}, \{\cdot, \rightarrow_\Pi\})$ where:

$$x \cdot y \quad \text{is the usual product of reals}$$

$$x \rightarrow y \quad = \quad \begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise.} \end{cases}$$

Axioms for Product Logic can be obtained by adding

$$\neg\neg\varphi \rightarrow ((\varphi \rightarrow \varphi \cdot \psi) \rightarrow \psi \cdot \neg\neg\psi) \tag{P1}$$

to Axioms of Basic Logic.
Product logic algebras, or PL-algebras for short, were introduced to prove the completeness theorem for Product logic.

Summing up

What you should have learned from this course:

Fuzzy logic is not imprecise reasoning, it is formal reasoning about imprecise statements.

In the sense above, fuzzy logic is a generalisation of classical logic which admits phenomena of inconsitency, vagueness, uncertainty, etc.

There is not just one fuzzy logic, one can choose a suitable description for a particular application. Yet, there is a general framework which subsumes all the possible choices.

There are well established logical foundations for fuzzy logic which can be exploited from a number of perspectives:

- automated reasoning,
- geometrical studies,
- algebraic characterisations.