Michael Winter

Trends in Logic 25

Goguen Categories

A Categorical Approach to L-fuzzy Relations

Springer

GOGUEN CATEGORIES

TRENDS IN LOGIC
Studia Logica Library

VOLUME 25

SCOPE OF THE SERIES

Trends in Logic is a bookseries covering essentially the same area as the journal *Studia Logica* – that is, contemporary formal logic and its applications and relations to other disciplines. These include artificial intelligence, informatics, cognitive science, philosophy of science, and the philosophy of language. However, this list is not exhaustive, moreover, the range of applications, comparisons and sources of inspiration is open and evolves over time.

Volume Editor
Ewa Orłowska

The titles published in this series are listed at the end of this volume.

GOGUEN CATEGORIES

A Categorical Approach to L-fuzzy Relations

by

MICHAEL WINTER
Brock University,
St. Catharines,
ON, Canada

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-1-4020-6163-9 (HB)
ISBN 978-1-4020-6164-6 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

To my family

Contents

INTRODUCTION

In a wide variety of problems one has to treat uncertain or incomplete information. Some kind of exact science is needed to describe and understand existing methods, and to develop new attempts. Especially in applications of computer science, this is a fundamental problem. To handle such information, Zadeh [44], and simultaneously Klaua [22, 23], introduced the concept of fuzzy sets and relations. In contrast to usual sets, fuzzy sets are characterized by a membership relation taking its values from the unit interval $[0, 1]$ of the real numbers. After its introduction in 1965 the theory of fuzzy sets and relations was ranked to be some exotic field of research. The success during the past years even with consumer products involving fuzzy methods causes a rapidly growing interest of engineers and computer scientists in this field. Nevertheless, Goguen [12] generalized this concept in 1967 to $\mathcal{L}$-fuzzy sets and relations for an arbitrary complete Brouwerian lattice $\mathcal{L}$ instead of the unit interval $[0, 1]$ of the real numbers. He described one of his motivating examples as follows:

> A housewife faces a fairly typical optimization problem in her grocery shopping: she must select among all possible grocery bundles one that meets as well as several criteria of optimality, such as cost, nutritional value, quality, and variety. The *partial ordering* of the bundles is an intrinsic quality of this problem. (Goguen [12] 1967)

It seems to be unnatural – comparing apples to oranges – to describe the criteria of optimality by a linear ordering as the unit interval. Why should the nutritional value of a given product be described by 0.6 (instead of 0.65, or any other value from $[0, 1]$), and why should a product with a high nutritional value be better than a product with high quality since those criteria are usually incomparable?

This observation has led to the theory of *multiobjective* or *multicriteria* optimization problems (cf. [13]). Instead of combining several criteria into a single number, and choosing the highest value, the concept of *Pareto optimailty* is used. In this approach the elements that are not dominated are taken for further considerations. Here an element x is said to dominate an element y if the value x_i for each objective i is greater than or equal to the corresponding value y_i of y. Traditional techniques of optimization and search have been applied in this area. Recently, even genetic algorithms have been used to search for multicriteria optima (e.g., [30, 31]).

One important notion within fuzzy theory is 0-1 crispness. A 0-1 crisp set or relation is described by the property that their characteristic function supplies either the least element 0 or the greatest element 1 of the unit interval $[0, 1]$ or more general the complete Brouwerian lattice $\mathcal{L}$. The class of 0-1 crisp fuzzy sets or relations may be seen as the subclass of regular sets or relations within the fuzzy world. Especially in applications, this notion is fundamental. We want to demonstrate this by considering two examples.

In fuzzy decision theory the basic problem is to select a specific element from a fuzzy set of alternatives. Therefore, several cuts are used [9, 24]. Basically, an α-cut of a fuzzy set M is a set N such that an element x is in N if, and only if, x is in M with a degree $\geq \alpha$. Analogously, an α-cut of a fuzzy relation R is a crisp relation S such that a pair of elements is related in S if, and only if, they are related in R with a degree $\geq \alpha$. Some variants of this notion may also be used. By definition, these cut operations are strongly connected to the notion of crispness. In particular, using the notion of crispness, one may define cut operations, and a cut operation naturally implies a notion of crispness.

Another example might be the development of a fuzzy controller. Usually the output of the controller has to be a 0-1 crisp value since it is used to control some nonfuzzy physical or software system. Therefore, a procedure, called defuzzification, is applied to transform the fuzzy output into some 0-1 crisp value. This list of examples may be continued. The bottom line is that a convenient theory for $\mathcal{L}$-fuzzy relations should be able to express the notion of crispness.

Today, fuzzy theory as well as its application is usually formulated as a variation of set theory or some kind of many-valued logic (e.g., cf. [2, 14, 26]). Although many algebraic laws are developed, these formalizations are not algebraic themselves. But an algebraic description would have several advantages. Applications of fuzzy theory may be described by simple terms in this language. In this way, we get in some sense a denotational semantics of the application, and, hence, a mathematical theory to reason about notions as correctness. One may prove such properties using the calculus of the algebraic theory, which is quite often more or less equational. Furthermore, this denotational semantics may be used to get a prototype of the application.

On the other hand, the calculus of binary relations has been investigated since the middle of the nineteenth century as an algebraic theory for logic and set theory [36, 37]. A first adequate development of such algebras was given by de Morgan and Peirce. Their work has been taken up and systematically extended by Schröder in [34]. More than 40 years later, Tarski started with the exhaustive study of relation algebras [35], and more generally, Boolean algebras with operators [17].

The papers above deal with relational algebras presented in their classical form. Elements of such algebras might be called *quadratic* or *homogeneous*; relations over a fixed universe. Usually a relation acts between two different kinds of objects, e.g., between customers and products. Therefore, a variant of the theory of binary relations has evolved that treats relations as *heterogeneous* or *rectangular*. A convenient framework to describe such kind of typing is given by category theory [3, 10, 27, 28, 32, 33].

There are some attempts to extend the calculus of relations to the fuzzy world. In [21] the concept of fuzzy relation algebras was introduced as an algebraic formalization of fuzzy relations with sup-min composition. These algebras are equipped with a semiscalar multiplication, i.e., an operation mapping an element from $[0,1]$ and a fuzzy relation to a fuzzy relation. In the standard model this is done by componentwise multiplication of the real values. Fuzzy relation algebras and their categorical counterpart [11], so-called Zadeh categories, constitute a convenient algebraic theory for fuzzy relations. Using the semiscalar multiplication it is also possible to characterize 0-1 crisp relations. Unfortunately, there is no way to extend or modify this approach for $\mathcal{L}$-fuzzy relations since for an arbitrary complete Brouwerian lattice such a semiscalar multiplication may not exist.

Another approach is based on Dedekind categories and was introduced in [27]. It was shown that the class of $\mathcal{L}$-fuzzy relations constitutes such a category. Unfortunately, the notion of 0-1 crispness causes some problems. Using the notion of scalar elements, i.e., a set of partial identities corresponding to the lattice $\mathcal{L}$, several notions of crispness in an arbitrary Dedekind category were introduced in [11, 20]. It was shown that the notion of s-crispness as well as the notion of l-crispness coincides with 0-1 crispness under an assumption concerning the underlying lattice. This assumption is fulfilled by all linear orderings, e.g., the unit interval. Unfortunately, it was also shown that both classes of crisp relations are trivial if the underlying lattice is a Boolean lattice. Actually, it can be shown (Theorem 5.1) that the notion of 0-1 crispness cannot be formalized in the language of Dedekind categories, i.e., this theory is too weak to express this property. Therefore, an extended theory is needed: the theory of Goguen categories.

In this book, we want to focus on Goguen categories introduced in [40] and some weaker structures as a convenient algebraic/categorical framework for $\mathcal{L}$-fuzzy relations and their application in computer science. In particular, we are interested in the development process of fuzzy controllers using the method of Mamdani [25]. One major problem is to ensure totality of the controller, i.e., the controller should produce an output value for each input. If the controller is described by a relation R within a Goguen category, this property can be proved by showing $\mathbb{I} \sqsubseteq R;R^{\smile}$, where $\mathbb{I}$ is the identity relation, ; is composition of relations, and $R^{\smile}$ is the converse of R. In most applications the controller is constructed by several components, which are combined using t-norms and t-conorms. The actual choice of the norms and their parameters is often done by experts using their experiences. Especially in complex applications, such a development process might easily lead to "holes" in the domain of the controller, i.e., to a partially defined controller. On the other hand, the relational description R of the controller can be parametric in those norms. From a generic proof of R being total (which is necessarily parametric too) we can generate a set of conditions that have to be satisfied in order to ensure the totality of R. The expert may now select a convenient set of norms and parameters fulfilling these conditions. The controller generated is guaranteed to be total. We will give an example of the development process sketched above in Chapter 6.

This book is organized as follows. In Chapters 1 and 2, we will introduce several mathematical notions as sets and lattices. The basic properties of $\mathcal{L}$-fuzzy relations are investigated in Chapter 3. Afterwards, we will concentrate on the categorical description of relations, i.e., we will introduce several categories of relations in Chapter 4. Furthermore, their basic properties are proved, and their connections to $\mathcal{L}$-fuzzy relations are studied. Chapter 5 is dedicated to Goguen categories and several weaker structures. We will prove some basic properties of those kinds of categories, focus on their representation theory, concentrate on derived connectives from a generalized notion of t-norms and t-conorms, and investigate the validity of equations in the substructure of crisp relations. In the last chapter we will give an applications of Goguen categories in computer science. We want to construct an $\mathcal{L}$-fuzzy controller with respect to a given set of rules. This controller is not based on the unit interval. Furthermore, we will construct the controller without deciding in advance which norms and parameters should be used. From a generic proof of the totality of the controller we derive properties that can be used by an engineer to finally decide about those parameters.

The writing of this book extended over almost 5 years. The early version grew out of the Habilitation thesis of the author in Munich, in 2003. In the following years several parts were revised and extended. In particular, Sections 5.1–5.4 were added in order to provide a more detailed overview of categories of $\mathcal{L}$-fuzzy relations.

The author would like to thank Gunther Schmidt and Yasuo Kawahara for their constant support during the Habilitation. Ivo Düntsch has to be thanked not only as a colleague, but also as a source of suggestions and advice.

The RelMiCS (Relational Methods in Computer Science) community was not only a source of useful comments and criticism, but also of friendship.

The first draft of this book was written in Munich. The author would like to thank the Department of Computer Science of the University of the Federal Armed Forces, Munich, Germany, for its support during this phase. The revision and the writing of the final version took place in St. Catharines. The author would like to thank the Department of Computer Science of Brock University, St. Catharines, Canada, for its support during the later phase.

Last but not least, a special thanks goes to Ewa Orlowska, who suggested to publish the result of the Habilitation in a book.

1
SETS, RELATIONS, AND FUNCTIONS

Sets are fundamental in mathematics. In this chapter we briefly introduce the concepts and notations from set theory we will use throughout the book. We assume that the reader is familiar with the basic concepts of set theory. He may use some kind of naive set theory or a formal theory as ZF or ZFC [18], i.e., the Zermelo-Fraenkel axioms of set theory. As usual, we denote the fact that "x is an element of a set A" by $x \in A$. The set with no elements is called the *empty set*, and is denoted by $\emptyset$. If every element of a set A is also an element of the set B, we say A is a *subset* of B denoted by $A \subseteq B$.

The *set comprehension* "the set of all elements of a set A fulfilling a predicate $\mathfrak{P}$" is denoted by $\{x \in A \mid \mathfrak{P}(x)\}$. If it is clear from the context or if it is insignificant which A is meant, we simply write $\{x \mid \mathfrak{P}(x)\}$.

Union, intersection, and set difference are defined as usual:

$$\begin{array}{lccl} \textit{union} & A \cup B & := & \{x \mid x \in A \text{ or } x \in B\}, \\ \textit{intersection} & A \cap B & := & \{x \mid x \in A \text{ and } x \in B\}, \\ \textit{set difference} & A \setminus B & := & \{x \mid x \in A \text{ and } x \notin B\}. \end{array}$$

The *complement* $\overline{A}$ of a set A in respect to a set $B \supseteq A$ is just the set difference $B \setminus A$. The binary operations union and intersection may be generalized to an arbitrary set of sets as argument. Suppose A_i for $i \in I$ are sets. Then,

$$\begin{array}{lccl} \text{union} & \bigcup\limits_{i \in I} A_i & := & \{x \mid \exists i \in I : x \in A_i\}, \\ \text{intersection} & \bigcap\limits_{i \in I} A_i & := & \{x \mid \forall i \in I : x \in A_i\}. \end{array}$$

The *Cartesian product* of two sets A and B is the set of all pairs (x,y) with $x \in A$ and $y \in B$, and is denoted by $A \times B$. The set of all subsets of A is called the *power set* of A, and is denoted by $\mathcal{P}(A)$.

A *binary relation* R between two sets A and B is an element of $\mathcal{P}(A \times B)$. A is called the *source* and B the *target* of R. If $A = B$, the relation R is also called an *endorelation* or *homogeneous*. To indicate that a binary relation R has source A and target B we usually write $R : A \to B$.

Apart from the set theoretic operations, we consider two further operations on binary relations. Let R be a relation between A and B and S between B and C. Then we define

$$\begin{array}{lrcl} \textit{conversion} & R^{\mathsf{T}} & := & \{(y,x) \mid (x,y) \in R\}, \\ \textit{composition} & R \circ S & := & \{(x,z) \mid \exists y \in B : (x,y) \in R \text{ and } (y,z) \in S\}. \end{array}$$

Due to the definition above, a composition $Q \circ R$ has to be read from the left to the right, i.e., first Q, and then R. We usually write $R(x,y)$ instead of $(x,y) \in R$. Notice that $R^{\mathsf{T}^{\mathsf{T}}} = R$, and that composition is *associative*, i.e., for all relation $Q : A \to B, R : B \to C$ and $S : C \to D$ we have $(Q \circ R) \circ S = Q \circ (R \circ S)$. The *identity relation* $\mathbb{I}_A$ on a set A is defined as the set $\{(x,x) \mid x \in A\}$. Then for all relations $R : A \to B$ we have $R = \mathbb{I}_A \circ R = R \circ \mathbb{I}_B$.

The *range* or *image* $\mathrm{ran}(R)$ of a relation $R : A \to B$ is defined as the set $\{y \in B \mid \exists x \in A : R(x,y)\}$. Dually, the *domain* $\mathrm{dom}(R)$ of R is defined as the set $\{x \in A \mid \exists y \in B : R(x,y)\}$. Obviously, we have $\mathrm{dom}(R) = \mathrm{ran}(R^{\mathsf{T}})$ and $\mathrm{ran}(R) = \mathrm{dom}(R^{\mathsf{T}})$.

A function f from A to B is a binary relation $f : A \to B$ which is

univalent	$f(x,y_1)$ and $f(x,y_2)$ implies $y_1 = y_2$ for all $x \in A$ and $y_1, y_2 \in B$,
total	for all $x \in A$ there exists some $y \in B$ so that $f(x,y)$.

Both properties may be expressed using the relational constructions. The first property is equivalent to $f^{\mathsf{T}} \circ f \subseteq \mathbb{I}_B$, and the second to $\mathbb{I}_A \subseteq f \circ f^{\mathsf{T}}$. The image of a function $f : A \to B$ will also be denoted by $f(A)$. As indicated above, arbitrary binary relations are denoted by uppercase and functions by lowercase letters. If f is a function, we usually write $f(x)$ to indicate the (necessarily unique) y so that $f(x,y)$. The set of all functions from A to B will be denoted by $A \to B$.

A relation $R : A \to B$ is called

(1) *injective* iff[1] $R(x_1,y)$ and $R(x_2,y)$ implies $x_1 = x_2$ for all $x_1, x_2 \in A$ and $y \in B$,

(2) *surjective* iff for all $y \in B$ there exists some $x \in A$ so that $R(x,y)$,

(3) *bijective* iff it is injective and surjective.

[1] We use the phrase "iff" as an abbreviation for "if and only if."

Obviously, a relation is injective iff R^{T} is univalent, surjective iff R^{T} is total, and bijective iff R^{T} is a function. A bijective function is also called a bijection. For historical reasons, the converse of a bijection f is denoted by f^{-1}. Notice that we have

$$f^{-1}(f(x)) = x \quad \text{and} \quad f(f^{-1}(y)) = y$$

for all bijections $f : A \to B$ and $x \in A, y \in B$.

The cartesian product construction is associative up to a bijection, i.e., the function $\alpha_{A,B,C} : (A \times B) \times C \to A \times (B \times C)$ defined by $\alpha_{A,B,C}((a, b), c) = (a, (b, c))$ is bijective for arbitrary sets A, B, and C. We define n-ary products by iterating binary products. Due to the associativity this is well-defined.

Given an n-ary function $f : A_1 \times \cdots \times A_n \to B$ we will use the extended set comprehension scheme $\{f(x_1, \ldots, x_n) \mid x_1 \in A_1, \ldots, x_n \in A_n\}$ as an abbreviation for

$$\{y \mid \exists x_1 \in A_1 \cdots \exists x_n \in A_n : y = f(x_1, \ldots, x_n)\}.$$

The concept of a Cartesian product of sets may be further generalized using functions. Let A_i for $i \in I$ be sets. The *I-indexed product* $\prod\limits_{i \in I} A_i$ of the sets A_i is defined as the set of all functions f from I to $\bigcup\limits_{i \in I} A_i$ so that $f(i) \in A_i$ for all $i \in I$. For a finite set $I = \{1, \ldots, n\}$ we get the usual n-ary product of $A_1, \ldots, A_n$. Notice, if $A_i = A_j =: A$ for all $i, j \in I$, i.e., all components of the product are equal, $\prod\limits_{i \in I} A_i$ is just $I \to A$.

We introduce some notations for commonly known sets:

$\mathbb{B}$ set of Boolean values $\{\mathbf{t}, \mathbf{f}\}$ or $\{0, 1\}$ ($\mathbf{t} = 1 \mathrel{\hat{=}} \text{true}$, $\mathbf{f} = 0 \mathrel{\hat{=}} \text{false}$),
$\mathbb{N}$ set of the natural numbers $\{0, 1, 2, \ldots\}$,
$\mathbb{N}^\infty$ set of the natural numbers with an additional greatest element ∞,
$\mathbb{R}$ set of real numbers,
$[0, 1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ unit interval of real numbers.

The concept of a homomorphism between structured sets, i.e., sets with some operations and/or relations defined on them, is usually somewhat informal. One may obtain a formal definition using the theory of universal algebras. In this book a homomorphism is a function reflecting the structure of the corresponding sets. For example, a homomorphism between the semigroups $(G_1, +_1, 0_1)$ and $(G_2, +_2, 0_2)$ is a function $f : G_1 \to G_2$ respecting the group operation $+$ and the neutral element 0, i.e., $f(x +_1 y) = f(x) +_2 f(y)$ for all $x, y \in G_1$ and $f(0_1) = 0_2$. As usual, a bijective homomorphism f so that f^{-1} is also a homomorphism is called an isomorphism. In this situation the source and the target of f are called isomorphic.