Many-Expert Decision Trees

Guillermo Badia[1], Carles Noguera[2], Alberto Paparella[3] and Guido Sciavicco[3]

[1]University of Queensland, Australia
[2]University of Siena, Italy
[3]University of Ferrara, Italy

Abstract
Taking inspiration from the literature fuzzy decision trees, and leveraging many-valued logics, we propose a novel, and more general variety of decision trees.

Keywords
Decision Trees, Many-Valued Logics, Many-Expert, Symbolic Learning

1. Introduction

Decision trees (*DT*s) [1] have been permeating machine learning literature for nearly four decades, thanks to their interpretability, cost-efficiency, and performance when applied to classification and regression from tabular data; their recent extension to the modal case [2, 3] opened up the possibility of applying decision trees to non-tabular data as well. Since the problem of learning an optimal decision tree from a given dataset is NP-hard [4], the common solution is to use sub-optimal, statistical approximation algorithms for this purpose, including ID3 [1], C4.5 [5], and CART [6].

Generally, a sub-optimal decision tree learning algorithm involves two steps: *splitting*, that is, the process of dividing a node into two or more sub-nodes based on certain statistical measures such as *Gini impurity*, *entropy*, or *variance reduction*, aiming to make the child nodes as homogeneous as possible concerning the target variable, and *pruning*, that is, the process of removing sub-nodes of a decision tree to reduce its complexity and prevent overfitting; this can be obtained using criteria to stop the tree growth early (*pre-pruning*), or by removing branches from a fully grown tree (*post-pruning*).

It is well-known that a decision tree has a logical counterpart consisting of a set of *(propositional) logical rules*.

One common approach for improving the performances and enhancing the interpretability of DTs, and in particular of their corresponding set of rules, is that of resorting to *non-crisp* logic. Classical, *crisp* propositional logic is characterized by being based on the Boolean two-valued algebra; non-crisp logic relaxes this assumption by allowing the existence of more than two truth values. The set of truth values forms an algebra, and when the domain of such an algebra is the set of all real numbers from 0 to 1 with the usual ordering (that is, it is *standard*), the corresponding logic is called *fuzzy*. In the accepted terminology, 'fuzzy' and 'non-crisp' are synonyms in the context of logic; noteworthy examples of varieties of fuzzy algebras include *Gödel* algebras (G) [7], on which *Gödel* logic is founded, *MV*-algebras [8] (MV) on which *Łukasiewicz* logic is based [9], and *product* algebras (Π) [10], which are the backbone of *product* logic. Non-crisp logics, however, may be based on algebras whose domain is not necessarily linear and can be both finite or infinite, such as the case of *Heyting* algebras (H) on which *intuitionistic* logic is based [11].

OVERLAY 2024, 6th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, November 28–29, 2024, Bolzano, Italy
*Corresponding author.
g.badia@uq.edu.au (G. Badia); carles.noguera@unisi.it (C. Noguera); alberto.paparella@unife.it (A. Paparella); guido.sciavicco@unife.it (G. Sciavicco)
https://sites.google.com/site/guillermobadialogic (G. Badia); https://docenti.unisi.it/en/noguera-clofent (C. Noguera); https://alberto-paparella.github.io (A. Paparella); https://sites.google.com/unife.it/guido (G. Sciavicco)
0000-0003-2795-3728 (G. Badia); 0000-0003-4910-599X (C. Noguera); 0009-0007-1653-3660 (A. Paparella); 0000-0002-9221-879X (G. Sciavicco)

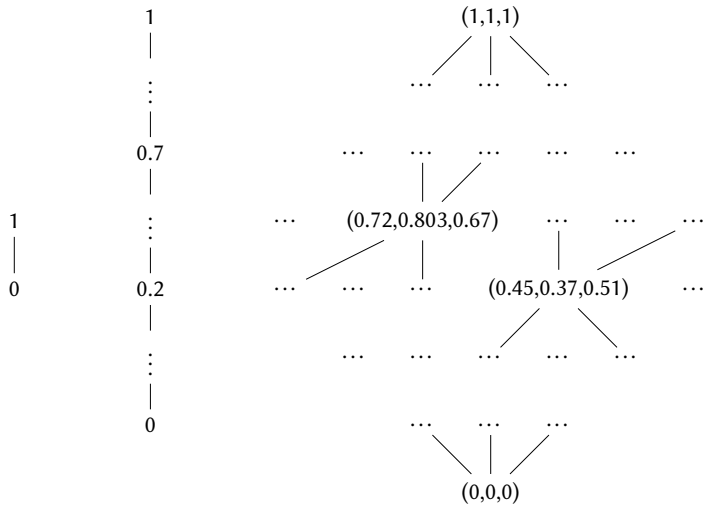

Figure 1: From left to right: the lattice structures of $\mathfrak{B}$, the one for some standard FL_{ew} $\mathfrak{A}$, and the one for $\mathfrak{A}_3$.

A *fuzzy decision tree* (*FDT*s) is essentially a decision tree that corresponds to a set of fuzzy propositional logic rules. Existing FDT learning methods range from algorithms to synthesize a tree from a user-fuzzyfied dataset, such as FuzzyID3 [12, 13], FuzzyC4.5 [14], to techniques for the fuzzyification of already learned decision tree rules, such as FuzzyCART [15]; however, the literature concerning fuzzy decision tree learning is too wide to be reviewed here, and we refer the interested reader to the recent survey [16].

Two traits that are common to essentially all existing proposals for FDT models and their learning algorithms are: (i) they are based on some standard fuzzy logic, and (ii) they are generally not included in open-source, available frameworks for learning and reasoning; the latter, in particular, makes it difficult to evaluate their effectiveness in real situations.

Towards a unifying approach to generalize learning algorithms to the non-crisp case, we consider here a more general variety of algebras, known as FL_{ew} algebras [17] (FL_{ew}). FL_{ew}-algebras are more general than G-, MV-, Π-, and H-algebras, and they allow the underlying domain of truth values to be not necessarily linearly ordered. As suggested by Fitting [18], a non-linear domain may be a suitable formalization of *many experts* situations, that is, situations in which different experts provide an opinion on the events. Logics based on FL_{ew}-algebras are called *many-valued* logics.

In this paper, we theorize the *many-expert decision tree* model (*MEDT*), that corresponds to a set of many-valued propositional logic rules. We consider a specific class of FL_{ew}-algebras, general enough to capture all typical fuzzy algebras commonly used in the FDT literature and beyond. MEDTs are parametric, so that the number of *experts* (which in some way can be thought of as the non-linearity degree of the algebra of truth values) can be varied and the operators for their opinions to be combined (that is, the algebra operators) can be customized within the degrees of freedom of FL_{ew}-algebras.

2. Many-Expert Decision Trees

A *complete FL_{ew}-algebra* is a tuple of the type

$$\mathfrak{A} = \langle \mathbb{A}, \cap, \cup, \cdot, +, 0, 1 \rangle,$$

where $\langle \mathbb{A}, \cap, \cup, 0, 1 \rangle$ is a bounded complete lattice with upper bound 1 and lower bound 0, and $\langle \mathbb{A}, \preceq \rangle$ corresponds to its lattice-ordered set. The two operations $\cdot$ and $+$ are such that $\langle \mathbb{A}, \cdot, 1 \rangle$ and $\langle \mathbb{A}, +, 0 \rangle$ form commutative monoids, with both operations being monotone with respect to $\preceq$. Specifically, if $\gamma \preceq \alpha$ and $\delta \preceq \beta$, then $\gamma \cdot \delta \preceq \alpha \cdot \beta$ and $\gamma + \delta \preceq \alpha + \beta$. The implication operation $\hookrightarrow$ in a FL_{ew}-algebra is defined as $\alpha \hookrightarrow \beta = \max\{\gamma \mid \alpha \cdot \gamma \preceq \beta\}$. In this context, we refer to $\cap$ as *meet*, $\cup$ as *join*, $\hookrightarrow$ as

Temperature (°C)	Humidity (%)	Wind (km/h)	Play Tennis
21.1	65	8.0	Yes
22.2	68	12.9	Yes
20.0	80	19.3	No
23.9	90	11.3	No
26.7	85	16.1	No
25.6	75	22.5	No
29.4	60	32.2	Yes
32.2	70	16.1	Yes

Table 1
Example of a labeled dataset; a typical classification problem on such an example could be the problem of deciding whether to play tennis based on the available data.

implication, $\cdot$ as *t-norm*, and $+$ as *t-co-norm*. A FL_{ew}-algebra is termed *linearly ordered* (or *chain*) if its lattice order is total, *standard* if its lattice reduct is the real unit interval $[0, 1]$, and *finite* if its lattice comprises only a finite number of elements.

Given a FL_{ew}-algebra $\mathfrak{A} = \langle \mathbb{A}, \cap, \cup, \cdot, +, 0, 1 \rangle$ and a set of propositional letters $\mathcal{P}$, the formulas of the propositional $\mathfrak{A}$-*logic* ($\mathfrak{A}$-*formulas*) are obtained by the grammar:

$$\varphi ::= \alpha \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi,$$

where $\alpha \in \mathbb{A}$ and $p \in \mathcal{P}$. A $\mathfrak{A}$-*model* I[1] is a map from each propositional letter in $p \in \mathcal{P}$ to some truth value $I(p) \in \mathfrak{A}$, and, given a $\mathfrak{A}$-formula φ, its value $I(\varphi)$ is computed recursively as follows:

$$\begin{aligned} I(\alpha) &= \alpha \\ I(\varphi \wedge \psi) &= I(\varphi) \cdot I(\psi) \\ I(\varphi \vee \psi) &= I(\varphi) + I(\psi) \\ I(\varphi \to \psi) &= I(\varphi) \hookrightarrow I(\psi). \end{aligned}$$

As it can be observed, we use $\alpha, \beta, \ldots$ for both algebra values and symbols to represent them.

The Boolean two-element algebra $\mathfrak{B}$ is a simple example of FL_{ew}-algebra; the propositional $\mathfrak{B}$-logic is the classical propositional logic, and its semantics reduces to the obvious one. Another setting of interest for us, given a natural number d, is the FL_{ew}-algebra $\mathfrak{A}_d = \langle \mathbb{A}_d \subseteq \mathbb{R}^d, \cap, \cup, \cdot, +, 0, 1 \rangle$, where $[r_1, \ldots, r_d] \preceq [s_1, \ldots, s_d]$ if and only if, for every i, $r_i \leq s_i$, $0 = [0, \ldots, 0]$, and $1 = [1, \ldots, 1]$. The operators $\cdot$ and $+$ are left unspecified; they, as well as the value of d and the cardinality of $\mathbb{A}_d$ will be treated as parameters. We call the propositional $\mathfrak{A}_d$-logic *many-expert* propositional logic. Fixed a model I, an algebra $\mathfrak{A}$, and a formula φ, we write $I_{\mathfrak{A}}(\varphi)$ to denote the value of φ assuming $\mathfrak{A}$ as an algebra; so, for example, for a model I and formula φ we have that $I_{\mathfrak{B}}(\varphi) = 1$ is an alternative notation for $I \models \varphi$, where $\models$ is the classic symbol for propositional satisfaction. Examples of lattice structures are given in Fig. 1.

Decision trees are extracted from datasets.

Definition 1. *A* dataset *is a set of* m instances $\mathcal{I} = \{I_1, \ldots, I_m\}$*, each one of which is described by the values of* n attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$.

Without loss of generality, we assume that the value of each attribute in an instance is a real number. Several problems are usually associated with datasets; in the case of *supervised* learning, each instance is also associated with a *label* (or *class*) $L \in \mathcal{L}$ and a dataset is termed *labeled*. Given a labeled dataset $\mathcal{I}$, supervised *classification* consists of synthesizing an algorithm (a *classifier*) that is able to classify the instances of an unlabelled dataset $\mathcal{J}$ whose instances are defined on the same set of attributes. An example of a dataset can be found in Tab. 1.

[1] In the following, the symbol I is also used to denote an instance; this is intentional, as in the symbolic context instances are seen as logical models.

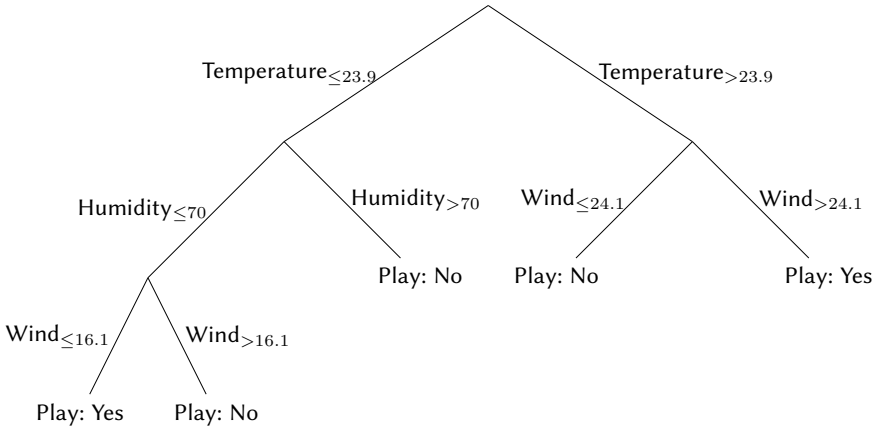

Figure 2: Simple decision tree for classifying whether a person should play tennis based on temperature, humidity, and wind conditions.

In the symbolic context, instances are seen as logical models. To help this interpretation, one takes into consideration that datasets are naturally associated with a logical vocabulary $\mathcal{P}$ of propositional letters, from which formulas are built. A simple choice for such a vocabulary is

$$\mathcal{P} \quad = \quad \{A_{\bowtie a} \mid a \in \mathbb{R}, \bowtie \in \{<, \leq, \geq, >\}\}.$$

Definition 2. *Let $\mathcal{L}$ be a set of classes, $\mathcal{P}$ a finite set of propositional letters, and $\mathfrak{A}$ an FL_{ew}-algebra. A* $\mathfrak{A}$-decision tree on $\mathcal{L}$ and $\mathcal{P}$ *is an object of the type*

$$\tau = \langle V, E, l, e \rangle,$$

where $\langle V, E\rangle$ is a full binary directed tree, l is a leaf-labelling function *that assigns a class from $\mathcal{L}$ to each leaf node in V, and e is an* edge-labelling function *that assigns a* decision *from $\mathcal{P}$ to each edge in E. To each branch $\pi = e_1 e_2 \dots e_k$ ($e_i \in E$, for every $1 \leq i \leq k$) in a decision tree τ is associated a* branch-formula $\varphi_\pi = e(e_1) \wedge \dots \wedge e(e_k)$. *Given an instance I, I is classified as $L \in \mathcal{L}$ by τ if and only if there exists $\pi \in \tau$ such that its leaf is labelled by L and that $I_{\mathfrak{A}}(\varphi_\pi) \succeq I_{\mathfrak{A}}(\varphi_{\pi'})$ for every $\pi' \in \tau$, $\pi' \neq \pi$.*

As it can be seen, a decision tree is a syntactical object. In the following, we simply use the term *decision tree* (*DT*) to denote a $\mathbb{B}$-decision tree, that is, a classical propositional decision tree. Also, a *fuzzy decision tree* (*FDT*) is an $\mathbb{A}$-decision tree for some standard FL_{ew}-algebra $\mathfrak{A}$. Finally, a *many-expert decision tree* (*MEDT*) is a $\mathfrak{A}_d$-decision tree. An example of decision tree can be found in Fig. 2.

In the case of DTs, the decisions that label two outgoing edges from the same node are always semantically opposite; in terms of the propositional vocabulary as we have defined it, this means that two edges outgoing from the same node are labeled, respectively, with$A_{\bowtie a}$ and $A_{\bowtie' a}$, where $\bowtie'$ is $<$ (resp., $\leq, \geq, >$) if $\bowtie$ is $\geq$ (resp., $>, <, \leq$).

Decision trees classify a certain instance I by executing a model checking algorithm. In the case of classical DTs, checking a branch-formula can be performed by progressively checking, step-by-step, each of its individual propositions/decisions, which makes classification with DTs particularly efficient; this is no longer true upon generalising DTs to FDTs and then MEDTs, but efficiency of classification can be, at least partially, preserved.

MEDTs are obtained from DTs as the result of two generalisation steps. First, we introduce a mechanisms to soften individual decisions, obtaining, as a matter of fact, a FDT in the process; to this end, let us fix a standard FL_{ew}-algebra $\mathfrak{A}$. Consider a dataset $\mathcal{I}$. In the classic setting, to the purpose of learning a classical DT τ, $\mathcal{I}$ is randomly separated into two subsets $\mathcal{I}_{train}$, used in the learning phase, and $\mathcal{I}_{test}$, used to test τ. By adding a further division, that is, by separating, instead, $\mathcal{I}$ into $\mathcal{I}_{train}, \mathcal{I}_{ft}$, and $\mathcal{I}_{test}$, and using $\mathcal{I}_{ft}$ as a *fine tuning* portion of the dataset, we take a first step into

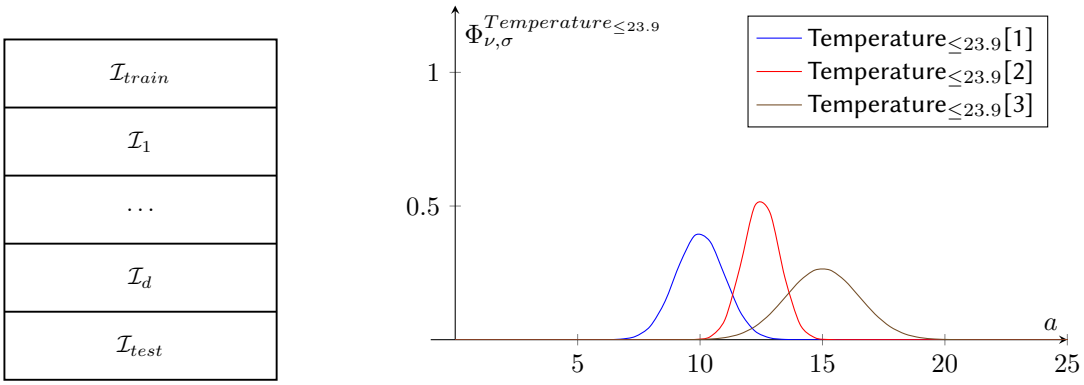

Figure 3: On the left-hand side, an example of dataset partition for MEDT learning with d experts. On the right-hand side: an example of different membership functions representing 3 different experts' opinions on the value of a decision/proposition.

avoiding typical overfitting phenomena. Keeping a portion of the data for fine tuning is a well-known strategy, sometimes applied for post-pruning purposes; as already proposed in the literature, we can use it for assigning a non-binary truth value to each proposition on the original tree. This can be obtained, after having learned a classical DT $\tau = \langle V, E, l, e\rangle$ from $\mathcal{I}_{train}$ using any algorithm, as follows: (i) for every $v \in V$, we associate the set $\mathcal{I}_{ft}^v = \{I \in \mathcal{I}_{ft} \mid I_{\mathfrak{B}}(A_{\bowtie a}) = 1\} \subseteq \mathcal{I}_{ft}$, that is, the portion of $\mathcal{I}_{ft}$ that falls into v, to v itself; (ii) for every $v \in V$, we associate the normal distribution $\Phi_{\nu,\sigma}^{A,\bowtie,a}$ computed on the set I_{ft}^v to the decision $A_{\bowtie a}$ that labels the edge between the parent of v and v itself; and (iii) for every instance I and decision $A_{\bowtie a}$ we define $I_{\mathfrak{A}}(A_{\bowtie a}) = \Phi_{\mathcal{I}_{ft}}^{A}(a')$, where a' is the value of A in I[2]. Observe that now the value of $I_{\mathfrak{A}}(\varphi_\pi)$ can be computed for every branch $\pi \in \tau$.

Stepping from a FDT, as defined above, to a MEDT requires switching from (a standard) $\mathfrak{A}$ to (a concretization of) $\mathfrak{A}_d$. In other words, to every instance I and decision $A_{\bowtie a}$ we need to associate a value $I_{\mathfrak{A}_d}(A_{\bowtie a})$, which is a vector of d real values. To this end, we replace $\mathcal{I}_{ft}$ with a family $I_1, \ldots, I_d$ of fine tuning portions of datasets. Each one of them plays the role of an expert. Therefore, proceeding as above, for every instance I and decision $A_{\bowtie a}$ we compute $I_{\mathfrak{A}_d}(A_{\bowtie a}) = [\Phi_{\mathcal{I}_1}^{A,\bowtie,a}(a'), \ldots, \Phi_{\mathcal{I}_1}^{A,\bowtie,a}(a')]$, where a' is the value of A in I.

An example of DT->MEDT generalisation of the DT in Fig. 2 is shown, partially, in Fig. 3. Consider, in particular, the decision $Temperature_{\leq 23.9}$. In Fig. 3, right, we assume $d = 3$, so that three different distributions for the portion of the fine tuning dataset whose instances show a value of $Temperature$ less than or equal to 23.9 are computed.

As a final observation it is worth noticing that classification with MEDTs is not as efficient as it is with DTs: the exponentially many different many-valued formulas that occurr on the branches of a MEDT should be checked in full to classify each instance individually. The tree structure of a MEDT, however, allows for the implementation of sub-optimal classification strategies, such as, for example, progressively checking all branches up to a fixed height, and then focusing on the sub-tree rooted at the chosen node only.

3. Conclusions

A many-expert decision tree is a non-crisp decision tree based on many-valued logic. This model generalizes both crisp and fuzzy decision trees, and can be obtained as the result of a fine tuning step upon learning a standard decision tree. We intend to carefully design, implement, and test the MEDT model, and include it in an already existing, comprehensive, end-to-end open-source framework for symbolic learning and reasoning.

[2]Obviously, FDTs and MEDTs can be defined with any other membership function.

References

[1] J. Quinlan, Induction of decision trees, Machine learning 1 (1986) 81–106.

[2] A. Brunello, G. Sciavicco, I. Stan, Interval temporal logic decision tree learning, in: Proc. of the 16th European Conference on Logics in Artificial Intelligence (JELIA), volume 11468 of *LNCS*, Springer, 2019, pp. 778–793.

[3] D. Della Monica, G. Pagliarini, G. Sciavicco, I. Stan, Decision trees with a modal flavor, in: Proc. of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA), number 13796 in LNCS, Springer, 2023, pp. 47 – 56.

[4] R. Rivest, Learning Decision Lists, Machine Learning 2 (1987) 229–246.

[5] J. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.

[6] L. Breiman, Classification and regression trees, Routledge, 2017.

[7] M. Baaz, N. Preining, R. Zach, First-order Gödel logics, Annals of Pure and Applied Logic 147 (2007) 23–47.

[8] C. C. Chang, Algebraic analysis of many valued logics, Transactions of the American Mathematical society 88 (1958) 467–490.

[9] A. Rose, Formalisations of further $\aleph_0$-valued Łukasiewicz propositional calculi, Journal of Symbolic Logic 43 (1978) 207–210. doi:10.2307/2272818.

[10] P. Hájek, The Metamathematics of Fuzzy Logic, Kluwer, 1998.

[11] L. Esakia, G. Bezhanishvili, W. H. Holliday, A. Evseev, Heyting Algebras: Duality Theory, Springer, 2019.

[12] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita, Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems, in: Proc. of 3rd IEEE International Fuzzy Systems Conference, IEEE, 1994, pp. 2113–2118.

[13] C. Z. Janikow, Fuzzy decision trees: issues and methods, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28 (1998) 1–14.

[14] M. E. Cintra, M. C. Monard, H. A. Camargo, FuzzyDT—a fuzzy decision tree algorithm based on C4.5, in: Proc. of the Brazilian Congress on Fuzzy Systems, 2012, pp. 199–211.

[15] J.-S. R.Jang, Structure determination in fuzzy modeling: a fuzzy CART approach, in: Proceedings of 1994 IEEE 3rd international fuzzy systems conference, IEEE, 1994, pp. 480–485.

[16] Z. A. Sosnowski, L. Gadomer, Fuzzy trees and forests - review, WIREs Data Mining Knowl. Discov. 9 (2019). URL: https://doi.org/10.1002/widm.1316. doi:10.1002/WIDM.1316.

[17] P. Cintula, P. Hájek, C. Noguera (Eds.), Handbook of Mathematical Fuzzy Logic, volume 37-38 of *Studies in Logic. Mathematical Logic and Foundation*, College publications, 2011.

[18] M. Fitting, Many-valued modal logics, Fundamenta Informaticae 15 (1999). doi:10.3233/FI-1991-153-404.