Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 262 (2015) 1–20

FUZZY sets and systems

www.elsevier.com/locate/fss

# Fuzzy logic programming reduced to reasoning with attribute implications

Tomas Kuhr *, Vilem Vychodil

*Data Analysis and Modeling Laboratory (DAMOL), Dept. Computer Science, Palacky University, Olomouc 17. listopadu 12, CZ-77146 Olomouc, Czech Republic*

Received 26 September 2013; received in revised form 13 April 2014; accepted 18 April 2014

Available online 28 April 2014

**Abstract**

We present a link between two types of logic systems for reasoning with graded if–then rules: the system of fuzzy logic programming (FLP) in sense of Vojtáš and the system of fuzzy attribute logic (FAL) in sense of Belohlavek and Vychodil. We show that each finite theory consisting of formulas of FAL can be represented by a definite program so that the semantic entailment in FAL can be characterized by correct answers for the program. Conversely, we show that for each definite program there is a collection of formulas of FAL so that the correct answers can be represented by the entailment in FAL. Using the link, we can transport results from FAL to FLP and *vice versa* which gives us, e.g., a syntactic characterization of correct answers based on Pavelka-style Armstrong-like axiomatization of FAL. We further show that entailment in FLP is reducible to reasoning with Boolean attribute implications and elaborate on related issues including properties of least models.


*Keywords:* Logic programming; Attribute implications; Functional dependencies; Ordinal scales; Residuated lattices; Least model property

## 1. Introduction

This paper contributes to the field of reasoning with graded if–then rules and presents a link between two logic systems that have been proposed and studied independently in the past. Namely, we focus on fuzzy logic programming in sense of [38] and fuzzy attribute logic presented in [8]. Both systems play an important role in computer science and artificial intelligence as they can be used for approximate knowledge representation and inference, description of dependencies found in data, representing approximate constraints in relational similarity-based databases, etc. Although the systems are technically different and were developed to serve different purposes, they share common features: (i) they are based on residuated structures of truth degrees, (ii) they use truth-functional interpretation of logical connectives, (iii) both the systems can be used to describe if–then dependencies in problem domains when one requires a formal treatment of inexact matches, (iv) models of theories form particular closure systems and semantic

* Corresponding author. +420 585 634 721.
*E-mail address:* tomas.kuhr@upol.cz (T. Kuhr).

http://dx.doi.org/10.1016/j.fss.2014.04.013

entailment (from theories) can be expressed by means of least models. It is therefore appealing to look closer at their mutual relationship. Furthermore, a possible link between the two systems can bring forth new results. For instance, in fuzzy attribute logic there is a known reduction to Boolean-case reasoning. That means, each theory in fuzzy attribute logic can be represented by its crisp counterpart which can be directly obtained from the theory by a simple algorithm. We are not aware of any analogous (and straightforward) Boolean-case reductions in fuzzy logic programming. By establishing the link between the two systems, one can adopt the Boolean-case reduction procedure from the fuzzy attribute logic.

The discovery of connections between FAL and FLP may also be interesting from the point of view of data analysis. For instance, fuzzy attribute implications can be seen as an alternative description of concept lattices [20] induced by graded object-attribute data [3,6,35]. Therefore, by finding fuzzy logic programs corresponding to sets of fuzzy attribute implications, one may introduce an alternative description of fuzzy concept lattices by fuzzy logic programs which we think may be an interesting issue for future work.

The aim of this paper is to show that the fundamental notions of correct answers and semantic entailment that appear in the systems are mutually reducible and allow to transport results from one theory to the other and *vice versa*. In addition to the reductions, we study particular implications of the results. In the rest of this section, we outline the form of the rules under our consideration. Section 2 presents preliminaries and recalls technical details from FLP and FAL. Further sections are devoted to the reductions. As a part of the new results, we also extend the existing Pavelka-style [34] Armstrong-like [1] axiomatization of FAL over infinite attribute sets and over arbitrary complete residuated lattices taken as the structure of truth degrees. This paper is an extension of our initial conference paper [28] on this topic which outlined the technique we develop in this paper.

*Fuzzy logic programming* [16,31,38] is a generalization of the ordinary logic programming [29] in which logic programs consist of facts and complex rules containing a head (an atomic predicate formula) and a body (a formula composed from atomic predicate formulas using connectives and aggregations interpreted by monotone truth functions) connected by a residuated implication. In addition, each rule (and fact) in a program is assumed to be valid to a degree (i.e., programs are theories in sense of Pavelka's abstract fuzzy logic [24,34]). As a consequence, fuzzy logic programs are capable of expressing graded dependencies between facts. As an example, we can consider the following rule:

$$\mathit{suitable}(\boldsymbol{X}) \overset{0.8}{\Leftarrow} \boldsymbol{wa}\big(\mathit{near}(\boldsymbol{X}, \mathsf{stadium}) \mathbin{/\!\!\!\wedge} \mathit{near}(\boldsymbol{X}, \mathsf{center}), \mathit{quality}(\boldsymbol{X}), \mathit{cost}(\boldsymbol{X})\big), \tag{1}$$

which expresses how much a hotel (variable $\boldsymbol{X}$) is suitable for a sport fan. This rule describes the degree of hotel suitability (atomic formula $\mathit{suitable}(\boldsymbol{X})$) as weighted average (aggregator $\boldsymbol{wa}$) of degrees of being conveniently located, having high quality ($\mathit{quality}(\boldsymbol{X})$), and having low prices ($\mathit{cost}(\boldsymbol{X})$). The convenience of hotel location is specified here as a conjunction ($/\!\!\!\wedge$) of being near to a stadium ($\mathit{near}(\boldsymbol{X}, \mathsf{stadium})$) and being near to a city center ($\mathit{near}(\boldsymbol{X}, \mathsf{center})$). The rule is valid to degree 0.8, which can be understood so that we put "almost full emphasis (in the veristic sense) on the rule".

The basic result of FLP is the completeness which puts in correspondence the declarative and procedural semantics of logic programs [38, Theorem 1 and Theorem 3] represented by correct answers and computed answers.

*Fuzzy attribute logic* [8] was developed primarily for the purpose of describing if–then dependencies that hold in object-attribute relational data where objects are allowed to have attributes to degrees. The formulas of FAL, so-called fuzzy attribute implications (FAIs), can be seen as implications $A \Rightarrow B$ between two graded sets of attributes (features), saying that if an object has all the attributes from $A$ (the antecedent) then it has all the attributes from $B$ (the consequent). The fact that $A$ and $B$ are graded sets (fuzzy sets) allows us to express graded dependencies between attributes. As an example

$$\big\{{}^{0.7}/\mathit{lowAge}, {}^{0.9}/\mathit{lowMileage}\big\} \Rightarrow \big\{{}^{0.6}/\mathit{highFuelEconomy}, {}^{0.9}/\mathit{highPrice}\big\} \tag{2}$$

is an attribute implication saying that cars with low age (at least to degree 0.7) and low mileage (at least to 0.9) have also high fuel economy (at least to 0.6) and high price (at least to 0.9). Formulas of this form can be prescribed by an expert or inferred from object-attribute relational data [23].

FAIs have an alternative interpretation as similarity-based functional dependencies [7] in relational databases [13,30]. For instance