International Journal of Approximate Reasoning 70 (2016) 51–67

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Automated prover for attribute dependencies in data with grades

Radim Belohlavek [a], Pablo Cordero [b], Manuel Enciso [b], Ángel Mora [b], Vilem Vychodil [a]

[a] Dept. Computer Science, Palacky University, Olomouc, Czech Republic
[b] Universidad de Málaga, Spain

ARTICLE INFO

Article history:
Received 19 May 2015
Received in revised form 20 October 2015
Accepted 8 December 2015
Available online 15 December 2015

Keywords:
Attribute implication
Functional dependency
Fuzzy logic
Ordinal data
Ordinary and graded completeness
Similarity

ABSTRACT

We present a new axiomatization of logic for dependencies in data with grades, which includes ordinal data and data over domains with similarity relations, and an efficient reasoning method that is based on the axiomatization. The logic has its ordinary-style completeness characterizing the ordinary, bivalent entailment as well as the graded-style completeness characterizing the general, possibly intermediate degrees of entailment. A core of the method is a new inference rule, called the rule of simplification, from which we derive convenient equivalences that allow us to simplify sets of dependencies while retaining semantic closure. The method makes it possible to compute a closure of a given collection of attributes with respect to a collection of dependencies, decide whether a given dependency is entailed by a given collection of dependencies, and more generally, compute the degree to which the dependency is entailed by a collection of dependencies. We also present an experimental evaluation of the presented method.

1. Introduction

We present a complete axiomatization of a logic for dependencies in data with grades and an efficient reasoning method based on this axiomatization. The dependencies are expressed by formulas of the form

$$A \Rightarrow B, \tag{1}$$

such as

$$\{0.2/y_1, y_2\} \Rightarrow \{0.8/y_3\}. \tag{2}$$

Formulas of the form (2) have two different interpretations whose entailment relations coincide. First, an interpretation given by object-attribute data with grades in which (2) means: every object that has attribute y_1 to degree at least 0.2 and attribute y_2 to degree 1 (i.e. fully possesses y_2) has attribute y_3 to degree at least 0.8. Second, an interpretation given by ranked tables over domains with similarities—a particular extension of Codd's model of relational data—in which (2) means: every two tuples that are similar on attribute y_1 to degree at least 0.2 and are equal on attribute y_2 are similar on attribute y_3 to degree at least 0.8. We assume that the degrees form a partially ordered set equipped with particular

E-mail addresses: radim.belohlavek@acm.org (R. Belohlavek), pcordero@uma.es (P. Cordero), enciso@uma.es (M. Enciso), amora@ctima.uma.es (Á. Mora), vychodil@acm.org (V. Vychodil).

http://dx.doi.org/10.1016/j.ijar.2015.12.007
0888-613X/

aggregation operations. If 0 and 1 are the only degrees, the first interpretation coincides with the well-known attribute dependencies in binary data saying that presence of certain attributes implies presence of other attributes, and the second one with functional dependencies in the ordinary Codd's model.

The logic based on the presented axiomatization obeys two types of completeness. The ordinary-style completeness says that $A \Rightarrow B$ semantically follows from a set T of dependencies if and only if $A \Rightarrow B$ is provable from T. The graded-style completeness characterizes the possibly intermediate degrees of entailment in that it says that the degree to which $A \Rightarrow B$ semantically follows from T equals the appropriately defined degree to which $A \Rightarrow B$ is provable from T, leaving classic entailment and non-entailment particular cases. The logic enables a new method of automated reasoning whose efficiency derives from a new rule, called a simplification equivalence, which makes it possible to replace theories by equivalent but simpler ones. The algorithm we present computes a closure of a given collection of attributes with respect to a given collection of formulas. A simple modification of the algorithm results in a procedure that decides entailment, i.e. decides whether $A \Rightarrow B$ follows from T, and more generally, computes the degree to which $A \Rightarrow B$ follows from T. The experimental evaluation we present demonstrates that the proposed method is computationally efficient and outperforms the previously proposed method based on the classic Closure algorithm [20].

2. Preliminary notions and results

In this section, we present preliminaries from complete residuated lattices and fuzzy attribute logic. Details can be found in [3,13–16].

2.1. Complete residuated lattices and related structures

We use complete residuated lattices as the structures of degrees. A complete residuated lattice is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with 0 and 1 being the least and greatest element, respectively; $\langle L, \otimes, 1\rangle$ is a commutative monoid (i.e. $\otimes$ is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$); $\otimes$ and $\rightarrow$ satisfy the following adjointness property:

$$a \otimes b \leq c \quad \text{iff} \quad a \leq b \rightarrow c \tag{3}$$

for any $a, b, c \in L$. As usual in the context of fuzzy logics in narrow sense, we interpret elements in L as degrees (of truth) with the following comparative meaning: if $a = \|\varphi\|_e$ and $b = \|\psi\|_e$ are degrees from L assigned to formulas φ and ψ by evaluation e and if $a \leq b$, then φ is less true than ψ under e. Operations $\otimes$ and $\rightarrow$ (residuum) represent truth functions of logical connectives "fuzzy conjunction" and "fuzzy implication". Note that from (3) it follows that $a \leq b$ iff $a \rightarrow b = 1$.

Remark 1. Complete residuated lattices include infinite as well as finite structures. For instance, a large family of structures defined on the real unit interval with its natural ordering with $\otimes$ being left-continuous t-norms and $\rightarrow$ being the corresponding residua [3,16,19]. An important family of finite and linearly ordered complete residuated lattices results by considering finite substructures on t-norm based complete residuated lattices on the real unit interval. In particular, for $L = \{0, 1\}$, $\mathbf{L}$ may be identified with the two-element Boolean algebra of classical logic. Namely, $\wedge$ and $\vee$ then become the truth functions of classical conjunction and disjunction, $\otimes$ coincides with $\wedge$, and $\rightarrow$ becomes the truth function of classical implication.

We equip complete residuated lattices with an additional unary connective: an idempotent truth-stressing hedge (shortly, a hedge) on a complete residuated lattice $\mathbf{L}$ is a map ${}^*\colon L \rightarrow L$ satisfying the following conditions: (i) $1^* = 1$, (ii) $a^* \leq a$, (iii) $(a \rightarrow b)^* \leq a^* \rightarrow b^*$, (iv) $a^{**} = a^*$ for each $a, b \in L$. Truth-stressing hedges were investigated from the point of view of fuzzy logic in narrow sense by Hájek [17], see also a recent general approach in [11,12]. In fuzzy logics, truth-stressing hedges serve as truth functions for unary connectives like "very true". For instance, Hájek [17] introduces a unary connective "*vt*" and formulas of the form $vt\varphi$ which read "φ is very true". Such formulas and evaluated (under e) so that $\|vt\varphi\|_e = (\|\varphi\|_e)^*$ with * being a truth-stressing hedge. Two important boundary cases of hedges are identity (i.e. $x^* = x$ for all $x \in L$) and so-called globalization [23] (i.e. $1^* = 1$ and $x^* = 0$ for all $1 \neq x \in L$).

Considering $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ as the structure of degrees, we introduce the basic notions of fuzzy relational systems. An $\mathbf{L}$-set (a fuzzy set with degrees in $\mathbf{L}$) A in universe Y is any map $A\colon Y \rightarrow L$, $A(y)$ being interpreted as "the degree to which y belongs to A". The collection of all $\mathbf{L}$-sets in universe Y is denoted by L^Y. $A \in L^Y$ is called crisp if $A(y) \in \{0, 1\}$ for all $y \in Y$. In that case, we may identify a crisp $\mathbf{L}$-set in Y with an ordinary subset of Y. By a slight abuse of notation, if A is crisp, we may write $y \in A$ and $y \notin A$ to denote that $A(y) = 1$ and $A(y) = 0$, respectively. In each universe Y, we consider two borderline (crisp) $\mathbf{L}$-sets: 0_Y such that $0_Y(y) = 0$ for all $y \in Y$ (an empty $\mathbf{L}$-set in Y); 1_Y such that $1_Y(y) = 1$ for all $y \in Y$. If Y is clear from the context, we write $\emptyset$ and Y instead of 0_Y and 1_Y, respectively.

Operations with $\mathbf{L}$-sets we use in this paper are induced componentwise by the operations of $\mathbf{L}$. For instance, the intersection of $\mathbf{L}$-sets $A, B \in L^Y$ is an $\mathbf{L}$-set $A \cap B \in L^Y$ such that $(A \cap B)(y) = A(y) \wedge B(y)$ for all $y \in Y$. The structure of all $\mathbf{L}$-sets in Y together with the induced operations is in fact a direct power $\mathbf{L}^Y = \langle L^Y, \cap, \cup, \otimes, \rightarrow, \emptyset, Y\rangle$ of $\mathbf{L}$ which is also a complete residuated lattice. Moreover, for $A, B \in L^Y$, we define the degree of inclusion of A in B as follows:

$$S(A, B) = \bigwedge\nolimits_{y \in Y} \big(A(y) \rightarrow B(y)\big). \tag{4}$$

Table 1
Ranked data table.

$\mathcal{D}(t)$	Name	Hair	Skin	Age	Eyes	Factor
1.0	John	Black	dark	34	Brown	10
0.8	Albert	Brown	light	32	Blue	50
0.6	Mary	Auburn	ligint	29	Blue	50
0.4	Dave	Red	light	26	Blue	50
0.1	Noa	White	dark	44	Green	30

In addition, we write $A \subseteq B$ iff $S(A, B) = 1$, i.e. if A is fully included in B. Using adjointness, (4) yields that $A \subseteq B$ iff, for each $y \in Y$, $A(y) \leq B(y)$.

2.2. Formulas and their interpretation

As we have outlined in the introduction, we deal with formulas which can be formalized as implications between **L**-sets of attributes. In order to describe the language, we fix a nonempty set Y whose elements are called attributes. Given **L** and Y, we consider as formulas the expressions of the form (1) where $A, B \in L^Y$. The set of all such formulas (for particular **L** and Y) is denoted by $\mathcal{L}$.

Interpretation in ranked tables over domains with similarities

The formulas in $\mathcal{L}$ may be interpreted in *ranked tables over domains with similarities*, developed e.g. in [6,8,9], which are formal counterparts to tables in an extension of the classical Codd's model in which one takes into account similarity relations on domains as well as degrees of match of tuples w.r.t. similarity-based queries. In particular, let for each attribute $y \in Y$, the domain D_y of y (i.e., the set of permissible values for y) be equipped with a similarity relation $\approx_y$, i.e. a map $\approx_y: D_y \times D_y \to L$ satisfying $u \approx_y u = 1$ (reflexivity) and $u \approx_y v = v \approx_y u$ (symmetry) for every $u, v \in D_y$. A ranked table on Y over $\{\langle D_y, \approx_y\rangle \mid y \in Y\}$ is a map $\mathcal{D}: \prod_{y\in Y} D_y \to L$ assigning a degree $\mathcal{D}(t) \in L$, called the rank of t, to every tuple $t \in \prod_{y\in Y} D_y$ such that only a finite number of tuples is assigned a non-zero rank. Table 1, which illustrates the concept of a ranked table, may be seen as an answer to a similarity query "show all persons with age approximately 34".

For a ranked data table $\mathcal{D}$, tuples t_1, t_2 and a fuzzy set $C \in L^Y$, the degree to which t_1 and t_2 have similar values on attributes from C is defined by

$$(t_1(C) \approx_{\mathcal{D}} t_2(C)) = (\mathcal{D}(t_1) \otimes \mathcal{D}(t_2)) \to \bigwedge\nolimits_{y\in Y} \big[C(y) \to (t_1[y] \approx_y t_2[y])\big],$$

where $t_1[y]$ and $t_2[y]$ denote the y-values of t_1 and t_2, respectively. The degree $\|A \Rightarrow B\|_{\mathcal{D}}$ to which formula $A \Rightarrow B$ is true in $\mathcal{D}$ is defined by

$$\|A \Rightarrow B\|_{\mathcal{D}} = \bigwedge\nolimits_{t_1,t_2} \big[(t_1(A) \approx_{\mathcal{D}} t_2(A))^* \to (t_1(B) \approx_{\mathcal{D}} t_2(B))\big]. \quad (5)$$

Using the basic principles of fuzzy logic [16], $\|A \Rightarrow B\|_{\mathcal{D}}$ is the truth degree of the proposition "for every two tuples, if they have (very) similar values on attributes from A, they have similar values on attributes from B", generalizing thus functional dependencies to domains where equalities (used for exact matches) are replaced by similarities (used for approximate matches). In the particular case $L = \{0, 1\}$, the above concepts coincide with the usual concepts regarding validity of functional dependencies. The generalization of the functional dependency notion may be tackle in different ways and there are a very wide set of works focused on this issue. In this paper we follow the most general approach in the framework of fuzzy logic as was showed in [10].

Interpretation in data with graded attributes

Alternatively, formulas $A \Rightarrow B$ over Y may be interpreted in *data with graded attributes* (ordinal data), i.e. data represented by a triplet $\langle X, Y, I\rangle$ where X is a set of objects and $I: X \times Y \to L$ assigns to every object $x \in X$ and attribute $y \in Y$ the degree $I(x, y)$ to which y applies to x. The following definition can be found e.g. in the overview paper [5]. The degree $\|A \Rightarrow B\|_{\langle X,Y,I\rangle}$ to which $A \Rightarrow B$ is true in $\langle X, Y, I\rangle$ is defined by

$$\|A \Rightarrow B\|_{\langle X,Y,I\rangle} = \bigwedge\nolimits_{x\in X} \big(S(A, I_x)^* \to S(B, I_x)\big), \quad (6)$$

where $I_x \in L^Y$ is the graded set of attributes possessed by the object x, i.e. $I_x(y) = I(x, y)$ for each $y \in Y$, and $S(\cdot,\cdot)$ denotes the subsethood degrees (4). Again, using the basic principles of fuzzy logic [16], $\|A \Rightarrow B\|_{\langle X,Y,I\rangle}$ is the truth degree of the proposition "every object that has all the attributes to degrees given by A has also all attributes to degrees given by B". If $L = \{0, 1\}$, we get the well-known dependencies asserting that the presence of attributes in A implies the presence of attributes in B.

Remark 2. Let us comment on the role of hedges. Observe that in (5) and (6), the hedge * is used to modify the degree to which the antecedent of the formula is satisfied. For instance, in case of (6), due to the interpretation of hedges as (truth functions) for logical connectives like “very true”, $S(A, I_x)^* \to S(B, I_x)$ represents a degree to which “if it is very true that x has attributes from A, then it has all attributes from B”. By setting the hedge * to its two borderline cases (globalization and identity), we obtain two important interpretations. Namely, if * is identity, the expression in (6) becomes $S(A, I_x) \to S(B, I_x)$. In particular, $\|A \Rightarrow B\|_{\langle X,Y,I\rangle} = 1$ then means $S(A, I_x) \le S(B, I_x)$ for all $x \in X$. In words, the degree to which x has (all the attributes from) B is at least as high as the degree to which x has (all the attributes from) A. On the other hand, if * is globalization, then $\|A \Rightarrow B\|_{\langle X,Y,I\rangle} = 1$ means: for each x, if $A(y) \le I_x(y)$ for all $y \in Y$, then $B(y) \le I_x(y)$ for all $y \in Y$. In words, if x has all attributes at least to the degrees prescribed by A, then it has all attributes at least to the degrees prescribed by B. Thus, by choosing to different hedges, we obtain two different (and both important) interpretations of the formulas in question. A similar argument applies to (5). The need for covering several interpretations of if-then rules based on various choices of hedges originated in the study of nonredundant bases of implications, see [5,8] for details. We keep the approach here because of its generality.

2.3. Fuzzy attribute logic

An important property is that the concepts of semantic entailment corresponding to these two above-described interpretations coincide. The semantic entailment is defined as follows. For a set T of formulas (a theory), a ranked data table $\mathcal{D}$ is called a model of T if $\|C \Rightarrow D\|_{\mathcal{D}} = 1$ for all $C \Rightarrow D \in T$. Analogously, $\langle X, Y, I\rangle$ is a model of T if $\|C \Rightarrow D\|_{\langle X,Y,I\rangle} = 1$ for all $C \Rightarrow D \in T$. Then, for a theory T and a formula $A \Rightarrow B$, we may define a degree to which $A \Rightarrow B$ semantically follows from T by

$$\|A \Rightarrow B\|_T = \bigwedge\nolimits_{\mathbf{M}\in \mathrm{Mod}(T)} \|A \Rightarrow B\|_{\mathbf{M}}, \tag{7}$$

where either (i) $\mathrm{Mod}(T)$ denotes all ranked data tables over domains with similarities which are models of T and $\|A \Rightarrow B\|_{\mathbf{M}}$ is (5); or (ii) $\mathrm{Mod}(T)$ denotes all object-attribute tables $\langle X, Y, I\rangle$ which are models of T and $\|A \Rightarrow B\|_{\mathbf{M}}$ is (6). It has been shown that both (i) and (ii) yield the same notion of semantic entailment [5]. As a result, a single proof system (axiomatization) can be used for both interpretations.

The following Armstrong-like [1,18] proof system was proposed in [4] under the name fuzzy attribute logic (FAL):

[Ax] infer $AB \Rightarrow A$ (*Axiom*)
[Cut] from $A \Rightarrow B$ and $BC \Rightarrow D$ infer $AC \Rightarrow D$ (*Cut*)
[Mul] from $A \Rightarrow B$ infer $c^*{\otimes}A \Rightarrow c^*{\otimes}B$ (*Multiplication*)

where $A, B, C, D \in L^Y$ and $c \in L$. In [Ax] and [Cut], we use the convention of writing AB instead of $A \cup B$, and in [Mul], we use $a{\otimes}B$ to denote the so-called a-multiple of $B \in L^Y$ which is an **L**-set such that

$$(a{\otimes}B)(y) = a \otimes B(y) \tag{8}$$

for all $y \in Y$ (i.e., the degrees to which $y \in Y$ belongs to B is multiplied by a constant degree $a \in L$). If A, B, C, D in [Ax] and [Cut] are replaced by ordinary sets instead of **L**-sets, we obtain a system of two deduction rules which are equivalent to the well-known system of Armstrong rules for reasoning with functional dependencies in relational database systems [1,20].

As usual, if $\mathcal{R}$ is an axiomatic system (like that containing the rules [Ax], [Cut], and [Mul]), a formula $A \Rightarrow B$ is said to be *provable* from a theory T using $\mathcal{R}$, denoted by

$$T \vdash_{\mathcal{R}} A \Rightarrow B \tag{9}$$

if there is a sequence $\varphi_1, \dots, \varphi_n$ called a proof such that φ_n is $A \Rightarrow B$, and for each φ_i we either have $\varphi_i \in T$ or φ_i is inferred (in one step) from some of the preceding formulas using some inference rule in $\mathcal{R}$. A deduction rule “from $\varphi_1, \dots, \varphi_n$ infer ψ” is called *derivable* (using $\mathcal{R}$) if $\{\varphi_1, \dots, \varphi_n\} \vdash_{\mathcal{R}} \psi$. Theories T_1 and T_2 are called *equivalent* (under $\mathcal{R}$), denoted $T_1 \equiv_{\mathcal{R}} T_2$, if for all φ, we have $T_1 \vdash_{\mathcal{R}} \varphi$ iff $T_2 \vdash_{\mathcal{R}} \varphi$.

The results in [4] imply that $\mathcal{R}$ consisting of [Ax], [Cut], and [Mul] is complete in the following sense:

Theorem 1 *(Ordinary-style completeness of FAL). Let $\mathcal{R}$ be the axiomatic system given by* [Ax], [Cut], *and* [Mul]. *If* **L** *and Y are finite, then $T \vdash_{\mathcal{R}} A \Rightarrow B$ iff $\|A \Rightarrow B\|_T = 1$.*

3. Fuzzy attribute simplification logic

Our goal is to design an efficient automated prover capable of determining entailment, i.e. of deciding whether $A \Rightarrow B$ follows from T, and computing the degree to which $A \Rightarrow B$ follows from T. We start by proposing an alternative axiomatic system which replaces [Cut] by a new rule, called simplification rule and study properties of the new axiomatization and the related notion of provability. There are several reasons why [Cut] is not suitable for designing of an automated prover. The most important one is that [Cut] can only be used for input formulas which are in a particular form: $A \Rightarrow B$ and $BC \Rightarrow D$. Put in words, the consequent B of the first formula must be included in the antecedent BC of the second one.

Checking this condition creates potential issues for efficiency of the prover. Even if the input formulas are in the correct form, in general there are several ways to choose C in $BC \Rightarrow D$, i.e., [Cut] can infer various formulas for the same input. This leads to an excessive growth of the number of possible derived formulas and makes an efficient deduction practically impossible.

In case of the ordinary functional dependencies, it has been observed that a system equivalent to that of Armstrong can be obtained by taking [Ax] (with **L**-sets replaced by ordinary sets) and the following rule:

[Sim] from $A \Rightarrow B$ and $C \Rightarrow D$ infer $A(C - B) \Rightarrow D$ (*Simplification*)

where A, B, C, D are ordinary sets and $C - B$ is the usual set-theoretic difference. Clearly, [Sim] overcomes the issues of [Cut] we have mentioned because (i) it can be applied to any two formulas, and (ii) it produces exactly one formula as the output.

We propose an axiomatization of fuzzy attribute logic with [Sim] as above with A, B, C, D being **L**-sets of attributes and operation $-$ being a suitable generalization of the ordinary set-theoretic difference. The key issue is how to introduce a difference of **L**-sets that has sufficient properties and is powerful enough to be used in [Sim]. As in case of intersections and unions of **L**-sets, we assume that $-$ is induced componentwise by a binary operation on L. We denote this operation on L by $\setminus$. Clearly, $\setminus$ should be monotone in the first argument and antitone in the second argument and satisfy $1 \setminus 0 = 1$ and $0 \setminus 0 = 0 \setminus 1 = 1 \setminus 1 = 0$ for the borderline degrees 0 and 1 from L. Furthermore, we may postulate that $\setminus$ is in a reasonable relationship to $\vee$. For instance, we may assume that $(a \vee b) \setminus a \leq b$ and $a \leq (a \setminus b) \vee b$ hold true because both of the conditions represent natural properties of "difference" when considered with respect to the "union". We may list other properties that we assume natural for the difference but instead we postulate a simple condition from which many natural properties follow. We assume that $\setminus$ is a binary operation in L satisfying the following adjointness property

$$a \setminus b \leq c \quad \text{iff} \quad a \leq b \vee c \tag{10}$$

for all $a, b, c \in L$. If $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, \setminus, 0, 1\rangle$ a complete residuated lattice equipped with $\setminus$ satisfying (10), then the dual lattice $\langle L, \vee, \wedge, 1, 0\rangle$ equipped with $\setminus$ is a complete Heyting algebra. In addition, **L** can be seen as a particular case of a complete integral commutative double residuated lattice (a DDR algebra in terms of Orłowska and Radzikowska [21]). As a consequence, if $\setminus$ satisfying (10) exists, it is uniquely given and can be expressed as

$$a \setminus b = \bigwedge\{c \in L \mid a \leq b \vee c\}. \tag{11}$$

In particular, if **L** is linearly ordered, we have

$$a \setminus b = \begin{cases} a, & \text{if } a > b, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The following assertion shows that postulating (10) is equivalent to postulating three simple requirements on $\setminus$ and $\vee$ we have mentioned as natural properties of difference.

Theorem 2. *Let* **L** *be a complete residuated lattice and* $\setminus$ *be a binary operation on* L*. Then,* $\setminus$ *and* $\vee$ *satisfy* (10) *for all* $a, b, c \in L$ *iff* $\setminus$ *is monotone in the first argument and the following inequality*

$$(b \vee c) \setminus b \leq c \leq b \vee (c \setminus b), \tag{13}$$

is true for all $b, c \in L$.

Proof. Let (10) hold for all $a, b, c \in L$. We first prove that $\setminus$ is monotone in the first argument. If $a_1 \leq a_2$, using (10) on $a_2 \setminus b \leq a_2 \setminus b$, we get $a_2 \leq b \vee (a_2 \setminus b)$, hence $a_1 \leq b \vee (a_2 \setminus b)$ using the assumption that $a_1 \leq a_2$. Then, applying (10), we obtain $a_1 \setminus b \leq a_2 \setminus b$, showing monotony. Now, using (10), observe that from $b \vee c \leq b \vee c$ it follows that $(b \vee c) \setminus b \leq c$ and from $c \setminus b \leq c \setminus b$ it follows that $c \leq b \vee (c \setminus b)$, i.e. (13) is satisfied.

Conversely, let $\setminus$ be monotone in the first argument and satisfy (13). If $a \setminus b \leq c$, then from $a \leq b \vee (a \setminus b)$ which is an instance of (13), we immediately get $a \leq b \vee c$. In order to prove the converse implication of (10), assume that $a \leq b \vee c$. Now, using $(b \vee c) \setminus b \leq c$ which is an instance of (13) together with the monotony of $\setminus$ in the first argument, we get $a \setminus b \leq c$. Thus, $\setminus$ and $\vee$ satisfy (10). $\square$

From now on, we shall consider a binary operation $-$ on **L**-sets in Y which is induced componentwise by $\setminus$. That is,

$$(A - B)(y) = A(y) \setminus B(y)$$

for all $y \in Y$. We need the following properties of $-$.

Lemma 1. *The following properties hold for all* $A, B, C \in L^Y$*:*

$$A - B \subseteq A, \tag{14}$$

$$A - \emptyset = A, \tag{15}$$

$$A \cup B = A \cup (B - A), \tag{16}$$

$$(A \cup B) - A \subseteq B, \tag{17}$$

$$A \cup ((A \cup B) - C) = A \cup (B - C), \tag{18}$$

$$A - B = \emptyset \textit{ iff } A \subseteq B, \tag{19}$$

Proof. (14) follows from (10) applied to $a \leq b \vee a$; (15) is a direct consequence of (16); (16) follows from $a \leq a \vee (b \setminus a)$ and $b \leq a \vee (b \setminus a)$ which is an instance of (13) together with (14); (17) is an application of (13); (18) follows from the monotony of $-$ in the first argument together with the facts that $b \leq c \vee (b \setminus c)$ and thus $a \vee b \leq c \vee a \vee (b \setminus c)$, i.e. $(a \vee b) \setminus c \leq a \vee (b \setminus c)$; (19) is a consequence of (10) since $a \setminus b \leq 0$ iff $a \leq b \vee 0 = b$. □

The following example illustrates how the fuzzy set difference works:

Example 1. Considering the unit interval as the truthfulness value set, since it is linearly ordered, the set difference is introduced as equation (12) shows. Let $U = \{a, b, c, d\}$ and $X, Y \in [0, 1]^U$ where $X = \{0.2/a, b, 0.5/c\}$ and $Y = \{0.3/b, 0.5/c, 0.7/d\}$. Then,

$X \cup Y = \{0.2/a, b, 0.5/c, 0.7/d\}$ $\quad$ $X \cap Y = \{0.3/b, 0.5/c\}$
$X \setminus Y = \{0.2/a, b\}$ $\quad$ $Y \setminus X = \{0.7/d\}$

Observe that $X \cup Y = (X \setminus Y) \cup (X \cap Y) \cup (Y \setminus X)$, as it holds in the crisp case.

We consider the axiomatic system which consists of [Ax], [Mul], and [Sim] with A, B, C, D being **L**-sets of attributes and $-$ being the difference defined above. In order to distinguish the system from the original axiomatic system of the fuzzy attribute logic (FAL), we call the new system and the corresponding logic a fuzzy attribute simplification logic (FASL). The provability of FASL is denoted by $\vdash$, i.e. $T \vdash A \Rightarrow B$ means that $A \Rightarrow B$ is provable from T using [Ax], [Mul], and [Sim]. We first show two versions of completeness of FASL, the ordinary-style and the so-called graded-style completeness. The ordinary style completeness says that the formulas provable from T are just those which semantically follow from T to degree 1.

Theorem 3 *(Ordinary-style completeness of FASL). Let* **L** *and* Y *be finite, let* T *be a set of formulas. For the axiomatic system given by* [Ax], [Sim], *and* [Mul], *we have* $T \vdash A \Rightarrow B$ *if and only if* $\|A \Rightarrow B\|_T = 1$.

Proof. We proceed by showing that deduction rules [Ax] + [Sim] are equivalent to [Ax] + [Cut]. The rest follows from the completeness of FAL, see Theorem 1. First, the following sequence proves that [Cut] can be derived from [Ax] + [Sim]:

1.	$A \Rightarrow B$	hypothesis
2.	$BC \Rightarrow D$	hypothesis
3.	$A(BC - B) \Rightarrow D$	1., 2., [Sim]
4.	$C \Rightarrow \emptyset$	[Ax]
5.	$C(A(BC - B) - \emptyset) \Rightarrow D$	4., 3., [Sim]
	$CA(BC - B) \Rightarrow D$	restated using (15)
	$AC \Rightarrow D$	restated using (17)

Conversely, consider the following sequence

1.	$A \Rightarrow B$	hypothesis
2.	$C \Rightarrow D$	hypothesis
3.	$BC \Rightarrow C$	[Ax]
4.	$BC \Rightarrow D$	3., 2., [Cut]
	$B(C - B) \Rightarrow D$	estated using (16)
5.	$A(C - B) \Rightarrow D$	1., 4., [Cut]

The sequence is a proof of $A(C - B) \Rightarrow D$ from $A \Rightarrow B$ and $C \Rightarrow D$, showing that [Sim] is derivable using [Ax] and [Cut]. □

Remark 3. The completeness result can be extended to general infinite **L** provided we supply additional infinitary rules. We do not consider such extension here because infinitary rules and, consequently, proofs as infinitely branching trees instead of finite sequences, cannot be handled by automated provers. Moreover, considering infinite **L** is not a practical since the formulas are used to describe dependencies in finite data tables where only finitely many pairwise different degrees can appear.

Interestingly, FASL can be used to characterize syntactically all degrees $\|A \Rightarrow B\|_T$ of semantic entailment, not just the entailment to degree 1 which is subject to Theorem 3. In fuzzy logic in the narrow sense, this kind of result is called a graded-style completeness. Note that in this case, a theory T may be conceived as an **L**-set of formulas, the degree $T(\varphi)$ to which a formula φ belongs to T being interpreted as the degree to which φ is assumed to be valid. A seminal work on graded-style completeness is [22]; further important work includes [14,16].

In our case, we approach the graded-style completeness as follows. First, each **L**-set $T: \mathcal{L} \to L$ is called a theory (note here that L is the set of degrees in **L** whereas $\mathcal{L}$ is the set of all formulas). In this setting, $T(A \Rightarrow B)$ is a degree from L interpreted as a threshold prescribing that $A \Rightarrow B$ must be true in each model of T at least to the degree $T(A \Rightarrow B)$. Note that $T(A \Rightarrow B) = 0$ represents no constraint on models whereas $T(A \Rightarrow B) = 1$ means that $A \Rightarrow B$ must be fully true in all models. Formally, $M \in L^Y$ is a model of a theory T if $\|A \Rightarrow B\|_M \geq T(A \Rightarrow B)$ for each $A \Rightarrow B \in \mathcal{L}$. It is easily seen that the classic concept of a theory we used so far can be understood as a theory T considered as a crisp **L**-set, i.e., $T(A \Rightarrow B) \in \{0, 1\}$ for each $A \Rightarrow B$. Finally, the degree to which $A \Rightarrow B$ semantically follows from T is defined as in (7) with $\mathrm{Mod}(T)$ being the set of all models of the **L**-set T.

Theorem 4 *(Graded-style completeness of FASL). Let* **L** *and* Y *be finite, let* $T: \mathcal{L} \to L$*. For the axiomatic system given by* [Ax], [Sim], *and* [Mul], *we have* $\|A \Rightarrow B\|_T = |A \Rightarrow B|_T$, *where*

$$|A \Rightarrow B|_T = \bigvee\{c \in L \mid crisp(T) \vdash A \Rightarrow c{\otimes}B\}, \tag{20}$$

$$crisp(T) = \{A \Rightarrow T(A \Rightarrow B){\otimes}B \mid S(T(A \Rightarrow B){\otimes}B, A) < 1\}. \tag{21}$$

Proof. Recall that $c{\otimes}B$ and $T(A \Rightarrow B){\otimes}B$ from (20) and (21) are the a-multiples of B for a being c and $T(A \Rightarrow B)$, respectively, see (8). The claim can be shown using the observations that $crisp(T)$ and T have the same models (easy to see using the fact that $c \leq \|A \Rightarrow B\|_M$ iff $\|A \Rightarrow c{\otimes}B\|_M = 1$) and $\|A \Rightarrow B\|_T$ is the supremum of all $c \in L$ such that $\|A \Rightarrow c{\otimes}B\|_T = 1$. Then, apply Theorem 3. □

Remark 4. The degree $|A \Rightarrow B|_T$ is called the provability degree of $A \Rightarrow B$ from T. Since, as is easily seen, $\{A \Rightarrow B_1, A \Rightarrow B_2\} \vdash A \Rightarrow B_1 \cup B_2$, the distributivity of $\otimes$ over $\bigvee$ together with the facts that **L** and Y are finite, we can conclude that (20) is the greatest degree $c \in L$ such that $A \Rightarrow c{\otimes}B$ is provable from $crisp(T)$ using [Ax], [Sim], and [Mul]. Moreover, $crisp(T)$ given by (21) represents an ordinary theory (a set of formulas) which is a counterpart to the **L**-set T. The condition $S(T(A \Rightarrow B){\otimes}B, A) < 1$ ensures that redundant formulas which are instances of [Ax] are not contained in $crisp(T)$.

We now proceed to further properties of FASL related to automated deduction.

Lemma 2. *The following deduction rules are derivable in FASL.*

[Aug] $\{A \Rightarrow B\} \vdash AC \Rightarrow B$ *(Augmentation)*

[Dec] $\{A \Rightarrow BC\} \vdash A \Rightarrow B$ *(Decomposition)*

[Com] $\{A \Rightarrow B, C \Rightarrow D\} \vdash AC \Rightarrow BD$ *(Composition)*

Proof. For any $A, B, C \in L^Y$, consider the following sequence of formulas:

1. $C \Rightarrow \emptyset$ [Ax]
2. $A \Rightarrow B$ hypothesis
3. $C(A - \emptyset) \Rightarrow B$ 1., 2., [Sim]

The last formula is equal to $AC \Rightarrow B$ because $A - \emptyset = A$, see (15). Therefore, $AC \Rightarrow B$ is provable from $\{A \Rightarrow B\}$, i.e., [Aug] is derivable. Analogously,

1. $A \Rightarrow BC$ hypothesis
2. $B \Rightarrow B$ [Ax]
3. $A(B - BC) \Rightarrow B$ 1., 2., [Sim]

The last formula is equal to $A \Rightarrow B$ because $A \cup (B - (B \cup C)) = A$ which is a consequence of (19). This proves that [Dec] is derivable. Finally, observe that

1. $A \Rightarrow B$	hypothesis
2. $C \Rightarrow D$	hypothesis
3. $ABCD \Rightarrow BD$	[Ax]
4. $A(ABCD - B) \Rightarrow BD$	1., 3., [Sim]
5. $A(ABCD - B)CD \Rightarrow BD$	4., [Aug]
$ACD \Rightarrow BD$	restated using (17)
6. $C(ACD - D) \Rightarrow BD$	2., 5., [Sim]
7. $C(ACD - D)A \Rightarrow BD$	6.[Aug]
$AC \Rightarrow BD$	restated using (17)

Hence, the sequence is a proof of $AC \Rightarrow BD$ from $A \Rightarrow B$ and $C \Rightarrow D$ using [Ax], [Sim], and [Aug] which is derivable from [Ax] and [Sim], showing that [Com] is derivable. This concludes the proof. □

Remark 5. Let us note that in the proof of Lemma 2, we have only used [Ax], [Sim], and [Cut]. The rule of multiplication [Mul] has not been used.

We now show that FASL has an analogy of the classic deduction theorem. We use this analogy in the proof of correctness of the automated deduction method. We need the following notions.

Definition 1. For any theory T and $A \in L^Y$, we put

$$Add_A(T) = \{AB \Rightarrow C \mid B \Rightarrow C \in T\}, \quad (22)$$

$$T^+ = \{B \Rightarrow C \mid T \vdash B \Rightarrow C\}. \quad (23)$$

Note T^+ is called the *deductive closure of* T and that $^+\colon 2^{\mathcal{L}} \to 2^{\mathcal{L}}$ is a closure operator. The following assertions describe the mutual relationship between the operators Add_A and $^+$.

Lemma 3. *For any theory T and $A \in L^Y$, we have $Add_A(T^+) \subseteq Add_A(T)^+$.*

Proof. We prove the claim by induction. Take any formula from $Add_A(T^+)$. The formula can be written as $AB \Rightarrow C$ such that $B \Rightarrow C \in T^+$. We distinguish four cases depending on whether $B \Rightarrow C$ belongs to T or it has been derived using [Ax], [Sim], or [Mul], respectively.

Case of $B \Rightarrow C \in T$:
By definition, $AB \Rightarrow C \in Add_A(T)$, whence clearly, $AB \Rightarrow C \in Add_A(T)^+$.

Case of $B \Rightarrow C \in T^+$ derived by [Ax]:
Then, $AB \Rightarrow C$ can also be derived by [Ax], i.e., $AB \Rightarrow C \in Add_A(T)^+$.

Case of $B \Rightarrow C \in T^+$ derived by [Sim]:
In this case, there are formulas $D \Rightarrow E \in T^+$ and $F \Rightarrow C \in T^+$ such that $B = D \cup (F - E)$. By definition of Add_A, we get $AD \Rightarrow E \in Add_A(T^+)$ and $AF \Rightarrow C \in Add_A(T^+)$. By induction hypothesis, $AD \Rightarrow E \in Add_A(T)^+$ and $AF \Rightarrow C \in Add_A(T)^+$. Now, [Sim] yields $AD(AF - E) \Rightarrow C \in Add_A(T)^+$, i.e. $AD(F - E) \Rightarrow C \in Add_A(T)^+$ using (18), showing $AB \Rightarrow C \in Add_A(T)^+$.

Case of $B \Rightarrow C \in T^+$ derived by [Mul]:
In this case, there are $a \in L$ and $D \Rightarrow E \in T^+$ such that $B = a^* \otimes D$ and $C = a^* \otimes E$. Thus, $AD \Rightarrow E \in Add_A(T^+)$ and by induction hypothesis, $AD \Rightarrow E \in Add_A(T)^+$. Applying [Mul], we obtain $a^* \otimes (AD) \Rightarrow a^* \otimes E \in Add_A(T)^+$. By distributivity of $\otimes$ over $\cup$, $(a^* \otimes A)(a^* \otimes D) \Rightarrow a^* \otimes E \in Add_A(T)^+$. Furthermore, [Aug] yields $A(a^* \otimes A)(a^* \otimes D) \Rightarrow a^* \otimes E \in Add_A(T)^+$. Since $A \cup (a^* \otimes A) = A$, it follows that $A(a^* \otimes D) \Rightarrow a^* \otimes E \in Add_A(T)^+$. Thus, $AB \Rightarrow C \in Add_A(T)^+$. □

Lemma 4. *For any theory T and $A \in L^Y$, we get $Add_A(T)^+ \subseteq T^+$. Moreover, for any $B \in L^Y$, $T \vdash A \Rightarrow B$ iff $Add_A(T) \vdash A \Rightarrow B$.*

Proof. The fact that [Aug] is a derived deduction rule yields $Add_A(T) \subseteq T^+$. Since $^+$ is monotone and idempotent, $Add_A(T)^+ \subseteq T^{++} = T^+$ which concludes the first part. If $Add_A(T) \vdash A \Rightarrow B$, then $T \vdash A \Rightarrow B$. Hence, it suffices to show the converse implication. If $T \vdash A \Rightarrow B$, i.e. $A \Rightarrow B \in T^+$, then by definition of Add_A, $A \Rightarrow B \in Add_A(T^+)$. Using Lemma 3, $A \Rightarrow B \in Add_A(T)^+$, i.e. $Add_A(T) \vdash A \Rightarrow B$. □

Recall that the classic deduction theorem of propositional logic says that $T \vdash \varphi \Rightarrow \psi$ if and only if $T \cup \{\varphi\} \vdash \psi$. Using the fact that any propositional formula χ is equivalent to $\overline{1} \Rightarrow \chi$ where $\overline{1}$ denotes a tautology, the classic deduction theorem can be equivalently restated as $T \vdash \varphi \Rightarrow \psi$ if and only if $T \cup \{\overline{1} \Rightarrow \varphi\} \vdash \overline{1} \Rightarrow \psi$ which is close in form to the following assertion.

Theorem 5 *(Deduction Theorem of FASL). Let T be a theory and $A, B \in L^Y$. Then, $T \vdash A \Rightarrow B$ iff $T \cup \{\emptyset \Rightarrow A\} \vdash \emptyset \Rightarrow B$.*

Proof. If $T \vdash A \Rightarrow B$ then clearly $T \cup \{\emptyset \Rightarrow A\} \vdash A \Rightarrow B$. Furthermore, $T \cup \{\emptyset \Rightarrow A\} \vdash \emptyset \Rightarrow A$. Using [Sim] and (19), we get $T \cup \{\emptyset \Rightarrow A\} \vdash \emptyset \Rightarrow B$.

Conversely, let $T \cup \{\emptyset \Rightarrow A\} \vdash \emptyset \Rightarrow B$, i.e., $\emptyset \Rightarrow B \in (T \cup \{\emptyset \Rightarrow A\})^+$. By definition of Add_A and using Lemma 3,

$$A \Rightarrow B \in Add_A((T \cup \{\emptyset \Rightarrow A\})^+) \subseteq Add_A(T \cup \{\emptyset \Rightarrow A\})^+.$$

Observe that $Add_A(T \cup \{\emptyset \Rightarrow A\})^+ = Add_A(T)^+$ since $A \Rightarrow A$ is an instance of [Ax]. Therefore, $A \Rightarrow B \in Add_A(T)^+$, i.e., $Add_A(T) \vdash A \Rightarrow B$. Now, apply Lemma 4 to conclude the proof. □

Remark 6. Note that from the point of view of interpreting $\emptyset \Rightarrow A$ in ordinal data, $\emptyset \Rightarrow A$ can be seen as a formula saying "attributes from A are (unconditionally) present", or more precisely, "every attribute from Y is (unconditionally) present to degree at least $A(y)$".

4. Automated prover

In this section, we develop the foundations of the automated prover based on FASL. In particular, we present several equivalences which can be used to replace theories by equivalent ones which are simpler. Replacing theories by simpler ones can be seen as a rewriting process which terminates after finitely many steps. In this section, provability refers to provability in FASL; moreover, we assume that both **L** and Y are finite.

Theorem 6. *For any $A, B, C, D \in L^Y$, the following equivalences hold true:*

(**DeEq**) $\{A \Rightarrow B\} \equiv \{A \Rightarrow B - A\}$;

(**UnEq**) $\{A \Rightarrow B, A \Rightarrow C\} \equiv \{A \Rightarrow BC\}$;

(**SiEq**) *If $A \subseteq C$ then $\{A \Rightarrow B, C \Rightarrow D\} \equiv \{A \Rightarrow B, A(C - B) \Rightarrow D - B\}$.*

Proof. (**DeEq**): Using [Dec] and (14), we get $\{A \Rightarrow B\} \vdash A \Rightarrow B - A$. In addition to that, the following sequence

1. $A \Rightarrow B - A$	hypothesis
2. $A \Rightarrow A$	[Ax]
3. $A \Rightarrow A(B - A)$	1., 2., [Com]
$A \Rightarrow AB$	restated using (16)
4. $A \Rightarrow B$	3., [Dec]

shows that $A \Rightarrow B$ is provable from $\{A \Rightarrow B - A\}$, proving (**DeEq**).

(**UnEq**) is a direct consequence of [Com] and [Dec], see Lemma 2.

(**SiEq**): Observe that $\{A \Rightarrow B, C \Rightarrow D\} \vdash A(C - B) \Rightarrow D - B$ using [Sim], [Dec], and applying (14). If $A \subseteq C$, the sequence:

1. $A \Rightarrow B$	hypothesis
2. $A(C - B) \Rightarrow D - B$	hypothesis
3. $A(C - B) \Rightarrow B(D - B)$	1., 2., [Com]
$A(C - B) \Rightarrow BD$	restated using (16)
4. $A(C - B) \Rightarrow D$	3., [Dec]
5. $AC(C - B) \Rightarrow D$	4., [Aug]
$C \Rightarrow D$	restated using $A \subseteq C$ and (14)

is a proof of $C \Rightarrow D$ from $\{A \Rightarrow B, A(C - B) \Rightarrow D - B\}$. □

Remark 7. (**DeEq**), called a decomposition equivalence, simplifies right-hand sides of formulas in theories. Analogously, (**UnEq**) called a union equivalence can be used to simplify theories by grouping together formulas with the same antecedent. Therefore, theories can be considered as sets of formulas with pairwise distinct antecedents. Finally, (**SiEq**) called a simplification equivalence allows to substitute $A(C - B) \Rightarrow D - B$ for $C \Rightarrow D$ in a theory provided that $A \Rightarrow B$ is provable from the theory and $A \subseteq C$.

The following theorem describes the core of the prover. The formula $\emptyset \Rightarrow A$ has a special meaning and shall be called a *guide.*

Theorem 7. *For any $A, U, V \in L^Y$ and $A' = A \cup (S(U, A)^* \otimes V)$, we have* (**gSiEq**) $\{\emptyset \Rightarrow A, U \Rightarrow V\} \equiv \{\emptyset \Rightarrow A', U - A' \Rightarrow V - A'\}$;

(**gSiUnEq**) *If* $U - A' = \emptyset$ *then* $\{\emptyset \Rightarrow A, U \Rightarrow V\} \equiv \{\emptyset \Rightarrow A'V\}$;

(**gSiAxEq**) *If* $V - A' = \emptyset$ *then* $\{\emptyset \Rightarrow A, U \Rightarrow V\} \equiv \{\emptyset \Rightarrow A'\}$.

Proof. (**gSiEq**): Since $\emptyset \subseteq U$, using (**SiEq**) it follows that

$$\{\emptyset \Rightarrow A', U \Rightarrow V\} \equiv \{\emptyset \Rightarrow A', U - A' \Rightarrow V - A'\}.$$

Thus, it suffices to show that $\{\emptyset \Rightarrow A, U \Rightarrow V\} \equiv \{\emptyset \Rightarrow A', U \Rightarrow V\}$. Since $A \subseteq A'$, we immediately obtain that $\emptyset \Rightarrow A$ is provable from $\{\emptyset \Rightarrow A', U \Rightarrow V\}$ on account of [Dec]. Furthermore, using the fact that $S(U, A)^* \otimes U \subseteq A$ which is an obvious consequence of the adjointness property, we get

1. $\emptyset \Rightarrow A$	hypothesis
2. $U \Rightarrow V$	hypothesis
3. $S(U, A)^* \otimes U \Rightarrow S(U, A)^* \otimes V$	2., [Mul]
4. $(S(U, A)^* \otimes U) - A \Rightarrow S(U, A)^* \otimes V$	1., 3., [Sim]
$\emptyset \Rightarrow S(U, A)^* \otimes V$	restated by (19) and $S(U, A)^* \otimes U \subseteq A$
5. $\emptyset \Rightarrow A(S(U, A)^* \otimes V)$	1., 4., [Com]

which is a proof of $\emptyset \Rightarrow A'$ from $\{\emptyset \Rightarrow A, U \Rightarrow V\}$, concluding the proof of (**gSiEq**). Furthermore, (**gSiUnEq**) results from (**gSiEq**) and (**UnEq**) utilizing (16). Finally, (**gSiAxEq**) results from (**gSiEq**) and using the fact that $U - A' \Rightarrow \emptyset$ is an instance of [Ax]. □

The following example illustrates how Theorem 7 together with the Deduction Theorem allows to infer new formulas.

Example 2. Given the subset of the unit interval $\{0, 0.1, \dots, 0.9, 1\}$ with the natural ordering, the Łukasiewiz adjoint par and the hedge defined by

$$x^* = \begin{cases} 1 & \text{if } x = 1, \\ 0.5 & \text{if } 0.5 \le x < 1, \\ 0 & \text{otherwise,} \end{cases}$$

since $S(\{0.3/b\}, \emptyset)^* = 0.7^* = 0.5$, by (**gSiEq**), we have that

$$\big\{\{0.3/b\} \Rightarrow \{0.8/a, 0.9/c, 0.5/d\}\big\} \vdash \emptyset \Rightarrow \{0.3/a, 0.4/c\}$$

By the other side, we have that $S(\{0.1/a, 0.7/b\}, \{0.2/b\})^* = 0.5^* = 0.5$. From (**gSiEq**), $\big\{\emptyset \Rightarrow \{0.2/b\}, \{0.1/a, 0.7/b\} \Rightarrow \{0.8/c\}\big\} \vdash \emptyset \Rightarrow \{0.2/b, 0.3/c\}$ and, by Deduction Theorem we finally conclude

$$\{0.1/a, 0.7/b\} \Rightarrow \{0.8/c\} \vdash \{0.2/b\} \Rightarrow \{0.2/b, 0.3/c\}$$

The primary output of the prover is the closure of a given **L**-set A of attributes:

Definition 2. Let $A \in L^Y$ and T be a crisp theory. The *closure of* A (*with respect to* T), denoted by A^+_T, is the greatest **L**-set in Y such that $T \vdash A \Rightarrow A^+_T$. A is called T-*closed* if $A^+_T = A$.

Note that since both **L** and Y are finite, the closure A^+_T exists. Namely, for all B_i such that $T \vdash A \Rightarrow B_i$ $(i \in I)$, we get $T \vdash A \Rightarrow \bigcup_{i \in I} B_i$ by a repeated application of [Com]. Closures in sense of Definition 2 can be used to characterize provability and provability degrees via the full and graded inclusion relations:

Theorem 8. *If* T *is a set of formulas and* $A \in L^Y$, *then*

$$T \vdash A \Rightarrow B \quad \textit{iff} \quad B \subseteq A^+_T. \tag{24}$$

Furthermore, if T *is an* **L**-*set of formulas, then*

$$|A \Rightarrow B|_T = S\Big(B, A^+_{crisp(T)}\Big). \tag{25}$$

Proof. If $B \subseteq A^+_T$ then from $T \vdash A \Rightarrow A^+_T$ we get $T \vdash A \Rightarrow B$ using [Dec]. The converse implication follows from the definition of A^+_T. From (20) and using the first claim

```
   Input: T (a set of formulas), A (an L-set of attributes)
   Output: A^+_T (the closure of A with respect to T)
1  begin
3      T := {U ⇒ V − U | U ⇒ V ∈ T};
5      repeat
6          A_old := A;
7          foreach U ⇒ V ∈ T do
8              A := A ∪ (S(U, A)* ⊗ V);
9              if U ⊆ A or V ⊆ A then
10                 T := T \ {U ⇒ V};
11             else
12                 T := (T \ {U ⇒ V}) ∪ {U − A ⇒ V − A};
13             end
14         end
15     until A = A_old;
17     return A;
18 end
```

Fig. 1. FASL automated prover.

$$|A \Rightarrow B|_T = \bigvee\{c \in L \mid crisp(T) \vdash A \Rightarrow c\otimes B\} = \bigvee \left\{c \in L \mid c\otimes B \subseteq A^+_{crisp(T)}\right\}$$
$$= \bigvee \left\{c \in L \mid c \leq S\big(B, A^+_{crisp(T)}\big)\right\},$$

from which (25) readily follows. □

Fig. 1 shows the main algorithm of the prover. It takes a theory T and an **L**-set A of attributes as its input and produces an **L**-set of attributes as its output. Further in this section, we prove that the output is the closure A^+_T. Hence, Theorem 8 can be used to decide whether $T \vdash A \Rightarrow B$ by checking whether B is included in the result of Algorithm 1. Furthermore, if T is an **L**-set of formulas, Theorem 8 can be used to obtain the degree of provability $|A \Rightarrow B|_T$ by computing the degree of graded inclusion of B in the result $A^+_{crisp}(T)$ of Algorithm 1 for $crisp(T)$ and A as its input arguments. Thus, Algorithm 1 serves three basic purposes: (i) it computes the closure of an **L**-set of attributes, (ii) it checks whether formulas are provable from T, (iii) it computes provability degrees of formulas.

Before we prove the correctness of the algorithm, let us inspect the pseudocode in Fig. 1. In line 3, T is replaced by an equivalent theory with formulas whose consequents are simplified using (**DeEq**). The repeat-until loop between lines 5–17 iterates until A is not changed in the inner for-each loop (lines 7–14). The **L**-set A together with T (which is also changing during the computation) represent a theory $T \cup \{\emptyset \Rightarrow A\}$ which is being replaced by equivalent simpler theories during the computation. Indeed, the inner cycle (lines 7–14) iterates over formulas from T and for each of them it replaces $T \cup \{\emptyset \Rightarrow A\}$ by an equivalent theory by applying (**gSiEq**) to $\emptyset \Rightarrow A$ and $U \Rightarrow V \in T$. Observe that the new value of A computed in line 8 corresponds to A' from Theorem 7. In addition, if $V \subseteq A$ (i.e., if $V - A = \emptyset$) then (**gSiAxEq**) is used which in turn leads to the removal of $U \Rightarrow V$ from T because $U - A \Rightarrow \emptyset$ is an instance of [Ax]. Analogously, if $U \subseteq A$ (i.e., if $U - A = \emptyset$) then (**gSiUnEq**) is used and since A already contains V (due to the fact that $S(U, A) = 1$, see line 8), $U \Rightarrow V$ is removed from T. In the general case, $U \Rightarrow V$ in T is replaced by $U - A \Rightarrow V - A$, see line 12. Thus, the inner for-each cycle can be seen as a cycle which either deletes formulas from T or replaces the formulas by simpler ones while maintaining A.

Consider an **L**-set $A \in L^Y$ of attributes. Observe that in the semantics given by data with graded attributes (Section 2.2), a table $\langle X, Y, I\rangle$ with graded attributes with a single object x for which $I(x, y) = A(y)$ is a model of a set T of formulas if and only if

$$S(U, A)^* \leq S(V, A) \tag{26}$$

for every $U \Rightarrow V \in T$, in which case we call A a model of T as well. It is easy to observe that (26) holds true if and only

$$c^*\otimes U \subseteq A \text{ implies } c^*\otimes V \subseteq A \tag{27}$$

for each $c \in L$. We utilize (27) below. We obtain the following characterization.

Lemma 5. *A is T-closed iff A is a model of T.*

Proof. Let A be T-closed and consider any $U \Rightarrow V \in T$ and $c \in L$ such that $c^*\otimes U \subseteq A$. Now observe that

1. $A \Rightarrow c^*\otimes U$	[Ax]
2. $U \Rightarrow V$	hypothesis
3. $c^*\otimes U \Rightarrow c^*\otimes V$	2., [Mul]
4. $A(c^*\otimes U - c^*\otimes U) \Rightarrow c^*\otimes V$	1., 3., [Sim]
$A \Rightarrow c^*\otimes V$	restated using (19)

shows that $T \vdash A \Rightarrow c^* \otimes V$. Since A is T-closed, A is the greatest **L**-set B such that $T \vdash A \Rightarrow B$, i.e., $c^* \otimes V \subseteq A$. As a consequence, A is a model of T on account of the discussion preceding this lemma.

Let A be a model of T. If A is not T-closed, there exists $B \not\subseteq A$ for which $T \vdash A \Rightarrow B$. Theorem 3 implies $T \models A \Rightarrow B$, i.e. $A \Rightarrow B$ is true in every model of T. In particular, $A \Rightarrow B$ is true in A which means $1 = S(A, A)^* \leq S(B, A)$ from which we get $B \subseteq A$, a contradiction. □

We are now ready to prove the following assertion:

Theorem 9 *(Correctness of prover). For every T and A, Algorithm 1 finishes and returns A^+_T.*

Proof. Denote by T_0 and A_0 the input theory and **L**-set of attributes. Observe that the algorithm finishes after finitely many steps. This follows from the fact that **L** and Y are finite and the update of A in line 8 is nondecreasing. Thus, after finitely many steps, the repeat-until loop between lines 5 and 15 terminates. Denote by T_n and A_n the values of T and A when the algorithm reaches line 17.

We now prove that A_n is T_n-closed. Due to Lemma 5, it suffices to show that A_n is a model of T_n, i.e. to verify (26) for every $U \Rightarrow V \in T_n$. Since T_n and A_n were obtained after the repeat-until loop terminated, for each $U \Rightarrow V \in T_n$, we have $A_n \supseteq A_n \cup (S(U, A_n)^* \otimes V)$, i.e., $S(U, A_n)^* \otimes V \subseteq A_n$ from which (26) directly follows. Due to Lemma 5, A_n is T_n-closed, i.e., $(A_n)^+_{T_n} = A_n$.

Since the modifications to T and A during the computation correspond to applications of (**gSiEq**), (**gSiUnEq**), and (**gSiAxEq**) to $T \cup \{\emptyset \Rightarrow A\}$, we get that $T_n \cup \{\emptyset \Rightarrow A_n\} \equiv T_0 \cup \{\emptyset \Rightarrow A_0\}$. Therefore, for each $B \in L^Y$, $T_n \cup \{\emptyset \Rightarrow A_n\} \vdash \emptyset \Rightarrow B$ iff $T_0 \cup \{\emptyset \Rightarrow A_0\} \vdash \emptyset \Rightarrow B$, i.e. due to Theorem 3, $T_n \vdash A_n \emptyset \Rightarrow B$ iff $T_0 \vdash A_0 \Rightarrow B$. Definition 2 now yields that $(A_n)^+_{T_n} = (A_0)^+_{T_0}$. Since, $A_0 = A$ and $T_0 = T$, Algorithm 1 returns the closure of A w.r.t. T. □

Remark 8. From the point of view of abstract rewriting systems, the simplification procedure utilized in Algorithm 1 represents substituting a theory by another equivalent theory which has a simpler form because the algorithm replaces formulas in form of the left-hand side of (**gSiEq**) by formulas in the form of the right-hand side of (**gSiEq**). Therefore, we may put $T_1 \cup \{\emptyset \Rightarrow A_1\} \rightrightarrows T_2 \cup \{\emptyset \Rightarrow A_2\}$ if $T_2 \cup \{\emptyset \Rightarrow A_2\}$ results from $T_1 \cup \{\emptyset \Rightarrow A_1\}$ by a single application of (**gSiEq**), (**gSiUnEq**), or (**gSiAxEq**). Obviously, $\rightrightarrows$ is terminating. In addition, it can be shown that $\rightrightarrows$ is locally confluent. This follows from the fact that if $T \cup \{\emptyset \Rightarrow A\} \rightrightarrows T_1 \cup \{\emptyset \Rightarrow A_1\}$ and $T \cup \{\emptyset \Rightarrow A\} \rightrightarrows T_2 \cup \{\emptyset \Rightarrow A_2\}$ then T contains formulas $U_1 \Rightarrow V_1$ and $U_2 \Rightarrow V_2$ such that for $T_3 = \{U_1 \Rightarrow V_1, U_2 \Rightarrow V_2\}$ and $A^+_{T_3}$, both $T_1 \cup \{\emptyset \Rightarrow A_1\}$ and $T_2 \cup \{\emptyset \Rightarrow A_2\}$ can be reduced to

$$(T \setminus T_3) \cup \{U_1 - A^+_{T_3} \Rightarrow V_1 - A^+_{T_3}, U_2 - A^+_{T_3} \Rightarrow V_2 - A^+_{T_3}, \emptyset \Rightarrow A^+_{T_3}\}$$

by multiple applications of (**gSiEq**). Analogously in the special cases for (**gSiUnEq**) and (**gSiAxEq**). As a consequence, each $T \cup \{\emptyset \Rightarrow A\}$ has its normal form (a reduced theory) which is in the proof of Theorem 9 denoted by $T_n \cup \{\emptyset \Rightarrow A_n\}$. A practical implication of this observation is that the order in which Algorithm 1 processes formulas of T is not essential for the result. On the other hand, various strategies of processing formulas may be taken into account in the implementation of the prover to improve its efficiency (we do not discuss this issue here).

Remark 9. Algorithm 1 can be modified in several ways if one wants to use it for the particular purpose of testing $T \vdash A \Rightarrow B$ or, more generally, $\|A \Rightarrow B\|_T \geq c$, where $c \in L$. In the first case, the algorithm can be modified so that it accepts T and $A \Rightarrow B$ as the input and terminates the computation with answer "yes" whenever the current value of A contains B. If the repeat-until loop terminates without B being included in A, the algorithm returns "no". In the second case, we can use the same modification of the algorithm which is run with arguments $crisp(T)$ and $A \Rightarrow c \otimes B$.

5. Complexity and performance evaluation

The worst-case time complexity of Algorithm 1 is the same as that of Graded Closure [7] which generalizes the classic Closure [2,20] algorithm for graded attributes. If $|L|$ is considered as a constant (the residuated lattices is not part of the input), Algorithm 1 has the worst-case time complexity in $O(pn^2)$ where p is the number of formulas in T and n is the number of attributes in Y.

Although having the same complexity as Graded Closure, Algorithm 1 significantly outperforms Graded Closure in most cases as we have observed in a series of experiments run on data of various sizes and characteristics. The data for the experiments have been obtained by a random generation of collections of formulas containing from 15,000 up to 100,000 attributes which is currently considered as mid-size data in the domains of data analysis and machine learning. The graph in Fig. 2 indicates that on average Algorithm 1 outperforms Graded Closure [7] whose quadratic growth of the running time is more rapid (dashed curve) than in case of Algorithm 1 (solid curve).

6. Conclusions

New type of data, including data which is imprecise or imperfect in various ways, presents new challenges for data processing methods. In particular, the classic approaches in which yes/no options, such as presence/absence, match/mismatch,

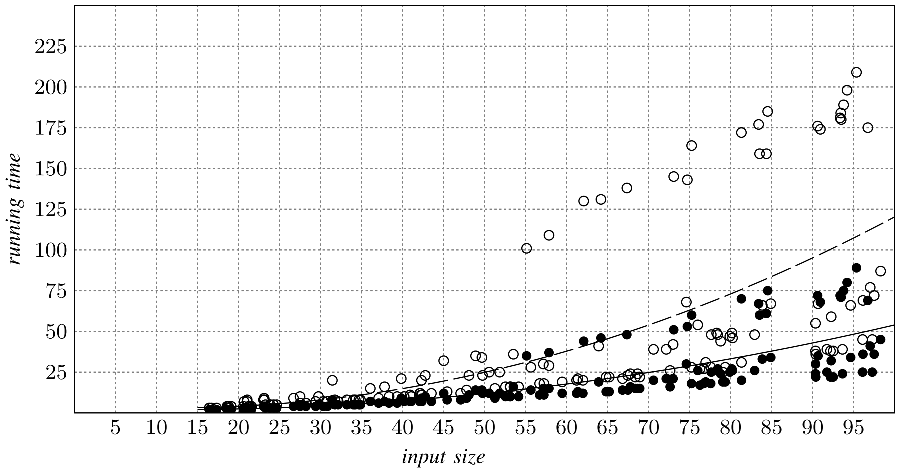

Fig. 2. Comparison of GRADED CLOSURE (denoted "○") and Algorithm 1 (denoted "●") in terms of the running time (y-axis) measured in milliseconds depending on the size of the input (x-axis) measured in thousands of attributes contained in input formulas.

or equality/nonequality get naturally replaced by degrees, such as degree of presence, degree of match, or degree of similarity, may conveniently be extended using the recently developed calculi of fuzzy logics. In this paper, we proposed a new axiomatization of logic for reasoning with attribute dependencies that involve grades. We proposed a new prover for such dependencies that may be used to solve the classic-style problems of computing a closure and deciding entailment as well as a conceptually new problem of computing degrees of entailment. The method utilizes a new rule, called simplification rule, that enables to replace formulas by equivalent but simpler ones. The new rule overcomes the drawbacks of other potentially applicable rules such as the rule of cut. As demonstrated by the experimental evaluation, the methods are computationally feasible practically to the same extent as the classic methods. The presented methods are based on the calculus of residuated lattices, which are used as the basic structures of truth degrees in modern fuzzy logic, and demonstrate how such calculus may be used in automated reasoning.

Acknowledgements

Support by the ECOP (Education for Competitiveness Operational Programme) project No. CZ.1.07/2.3.00/20.0059, which is co-financed by the European Social Fund and the state budget of the Czech Republic, is gratefully acknowledged.

Supported by Grants TIN2011-28084 and TIN2014-59471-P of the Science and Innovation Ministry of Spain.

Appendix A. An execution trace of FASL Automated Prover

In this appendix we illustrate the execution of the Algorithm with two examples, corresponding to the derivability and no-derivability situations.

Example 3. We consider the truthfulness structure described in Example 2. Let T be the following fuzzy theory:

$$
\begin{aligned}
T = \Big\{ \; & \{0.4/a, 0.6/c\} \stackrel{0.6}{\Rightarrow} \{0.8/c, 0.5/d, 0.6/e, 0.7/f\},\\
& \{0.2/d, 0.3/f\} \stackrel{0.9}{\Rightarrow} \{1/d, 0.6/e, 0.9/g\},\\
& \{0.4/d, 0.5/e\} \stackrel{0.8}{\Rightarrow} \{0.6/h, 0.2/d\},\\
& \{0.6/d, 0.4/i\} \stackrel{1}{\Rightarrow} \{0.7/a, 0.7/d\},\\
& \{0.3/c, 0.4/e\} \stackrel{1}{\Rightarrow} \{0.2/h\},\\
& \{0.4/c, 0.6/h\} \stackrel{0.6}{\Rightarrow} \{0.3/b, 0.7/e, 0.8/i\},\\
& \{0.2/g\} \stackrel{0.6}{\Rightarrow} \{0.7/a, 0.4/d\},\\
& \{0.6/c, 0.5/d\} \stackrel{0.8}{\Rightarrow} \{0.4/e\} \Big\}
\end{aligned}
$$

and we want to check if

$$T \vdash \{0.2/c, 0.6/f\} \stackrel{0.8}{\Rightarrow} \{0.5/a, 0.5/d, 0.6/g, 0.6/h\}$$

The trace of the execution of the **FASL** *Automated Prover* is the following:

1. The *guide* is $\{\emptyset \Rightarrow A_1\} := \{\emptyset \Rightarrow \{0.2/c, 0.6/f\}\}$
2. Compute *crisp*(T). That is,

$$
\begin{aligned}
crisp(T) = \{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.4/c, 0.1/d, 0.2/e, 0.3/f\}\\
& \{0.2/d, 0.3/f\} && \Rightarrow \{0.9/d, 0.5/e, 0.8/g\}\\
& \{0.4/d, 0.5/e\} && \Rightarrow \{0.4/h\}\\
& \{0.6/d, 0.4/i\} && \Rightarrow \{0.7/a, 0.7/d\}\\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\}\\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\}\\
& \{0.2/g\} && \Rightarrow \{0.3/a\}\\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\}\ \}
\end{aligned}
$$

and, applying **DeEq** to every formula in *crisp*(T), we obtain

$$
\begin{aligned}
\mathrm{T}_1 = \{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.1/d, 0.2/e, 0.3/f\}\\
& \{0.2/d, 0.3/f\} && \Rightarrow \{0.9/d, 0.5/e, 0.8/g\}\\
& \{0.4/d, 0.5/e\} && \Rightarrow \{0.4/h\}\\
& \{0.6/d, 0.4/i\} && \Rightarrow \{0.7/a, 0.7/d\}\\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\}\\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\}\\
& \{0.2/g\} && \Rightarrow \{0.3/a\}\\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\ \}
\end{aligned}
$$

3. The algorithm applies Equivalences to the *guide* and each $U \Rightarrow V \in \mathrm{T}_i$ as follows:

(**gSiEq**): $A' = \{0.2/c, 0.6/f\}$

$$\{\emptyset \Rightarrow \{0.2/c, 0.6/f\}, \{0.4/a, 0.6/c\} \Rightarrow \{0.1/d, 0.2/e, 0.3/f\}\} \equiv$$
$$\equiv \{\emptyset \Rightarrow \{0.2/c, 0.6/f\}, \{0.4/a, 0.6/c\} \Rightarrow \{0.1/d, 0.2/e\}\}$$

$$
\begin{aligned}
\{\emptyset \Rightarrow A_2\} = \{\ & \emptyset && \Rightarrow \{0.2/c, 0.4/d, 0.6/f, 0.3/g\}\ \}\\
\mathrm{T}_2 = \{\ & \mathbf{\{0.4/a, 0.6/c\}} && \Rightarrow \mathbf{\{0.1/d, 0.2/e\}}\\
& \{0.2/d, 0.3/f\} && \Rightarrow \{0.9/d, 0.5/e, 0.8/g\}\\
& \{0.4/d, 0.5/e\} && \Rightarrow \{0.4/h\}\\
& \{0.6/d, 0.4/i\} && \Rightarrow \{0.7/a, 0.7/d\}\\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\}\\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\}\\
& \{0.2/g\} && \Rightarrow \{0.3/a\}\\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\}
\end{aligned}
$$

(**gSiUnEq**): $A' = \{0.2/c, 0.4/d, 0.6/f, 0.3/g\}$

$$\{\emptyset \Rightarrow \{0.2/c, 0.6/f\}, \{0.3/f\} \Rightarrow \{0.9/d, 0.5/e, 0.8/g\}\} \equiv$$
$$\equiv \{\emptyset \Rightarrow \{0.2/c, 0.4/d, 0.6/f, 0.3/g\}, \emptyset \Rightarrow \{0.9/d, 0.5/e, 0.8/g\}\} \equiv$$
$$\equiv \{\emptyset \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g\}\}$$

$$
\begin{aligned}
\{\emptyset \Rightarrow A_3\} = \{\ & \emptyset && \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g\}\ \}\\
\mathrm{T}_3 = \{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.2/e\}\\
& \{0.4/d, 0.5/e\} && \Rightarrow \{0.4/h\}\\
& \{0.6/d, 0.4/i\} && \Rightarrow \{0.7/a, 0.7/d\}\\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\}\\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\}\\
& \{0.2/g\} && \Rightarrow \{0.3/a\}\\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\}
\end{aligned}
$$

(**gSiAxEq**): $A' = \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}$
$\big\{\emptyset \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g\}, \{0.4/d, 0.5/e\} \Rightarrow \{0.4/h\}\big\} \equiv$
$\equiv \{\emptyset \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\big\}$

$$\begin{aligned}
\{\emptyset \Rightarrow A_4\} = \{\ & \emptyset && \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\ \} \\
\mathrm{T}_4 = \big\{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.2/e\} \\
& \{0.6/d, 0.4/i\} && \Rightarrow \{0.7/a, 0.7/d\} \\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\} \\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\} \\
& \{0.2/g\} && \Rightarrow \{0.3/a\} \\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\big\}
\end{aligned}$$

(**gSiEq**): $A' = \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}$
$\big\{\emptyset \Rightarrow \{0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.5/h\},$
$\{0.6/d, 0.4/i\} \Rightarrow \{0.7/a, 0.7/d\}\big\} \equiv$
$\equiv \{\emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}, \{0.4/i\} \Rightarrow \{0.7/a\}\big\}$

$$\begin{aligned}
\{\emptyset \Rightarrow A_4\} = \{\ & \boldsymbol{\emptyset} && \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\ \} \\
\mathrm{T}_5 = \big\{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.2/e\} \\
& \{\mathbf{0.4/\mathit{i}}\} && \Rightarrow \{\mathbf{0.7/\mathit{a}}\} \\
& \{0.3/c, 0.4/e\} && \Rightarrow \{0.2/h\} \\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\} \\
& \{0.2/g\} && \Rightarrow \{0.3/a\} \\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\big\}
\end{aligned}$$

(**gSiAxEq**): $A' = \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}$
$\big\{\emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\},$
$\{0.3/c, 0.4/e\} \Rightarrow \{0.2/h\}\big\} \equiv$
$\equiv \{\emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.5/h\}\big\}$

$$\begin{aligned}
\{\emptyset \Rightarrow A_4\} = \{\ & \emptyset && \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\ \} \\
\mathrm{T}_6 = \big\{\ & \{0.4/a, 0.6/c\} && \Rightarrow \{0.2/e\} \\
& \{0.4/i\} && \Rightarrow \{0.7/a\} \\
& \{0.4/c, 0.6/h\} && \Rightarrow \{0.3/e, 0.4/i\} \\
& \{0.2/g\} && \Rightarrow \{0.3/a\} \\
& \{0.6/c, 0.5/d\} && \Rightarrow \{0.2/e\}\big\}
\end{aligned}$$

(**gSiEq**): $A' = \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}$

$\big\{\emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\},$
$\{0.4/c, 0.6/h\} \Rightarrow \{0.3/e, 0.4/i\}\big\} \equiv$
$\equiv \{\emptyset \Rightarrow \{0.3/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.5/h\},$
$\{0.4/c, 0.6/h\} \Rightarrow \{0.4/i\}\big\}$

$$\{\emptyset \Rightarrow A_4\} = \{\ \emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\ \}$$

$$\begin{aligned} \mathrm{T}_7 = \big\{\ & \{0.4/a, 0.6/c\} \Rightarrow \{0.2/e\} \\ & \{0.4/i\} \Rightarrow \{0.7/a\} \\ & \{\mathbf{0.4/c, 0.6/h}\} \Rightarrow \{\mathbf{0.4/i}\} \\ & \{0.2/g\} \Rightarrow \{0.3/a\} \\ & \{0.6/c, 0.5/d\} \Rightarrow \{0.2/e\}\big\} \end{aligned}$$

(**gSiAxEq**): $A' = \{0.3/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}$

$$\big\{\emptyset \Rightarrow \{0.2/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}, \{0.2/g\} \Rightarrow \{0.3/a\}\big\} \equiv$$
$$\equiv \{\emptyset \Rightarrow \{0.3/a0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\big\}$$

$$\{\emptyset \Rightarrow A_5\} = \{\ \emptyset \Rightarrow \{0.3/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}\ \}$$

$$\begin{aligned} \mathrm{T}_8 = \big\{\ & \{0.4/a, 0.6/c\} \Rightarrow \{0.2/e\} \\ & \{0.4/i\} \Rightarrow \{0.7/a\} \\ & \{0.4/c, 0.6/h\} \Rightarrow \{0.4/i\} \\ & \{0.6/c, 0.5/d\} \Rightarrow \{0.2/e\}\big\} \end{aligned}$$

4. As the *guide* is $\emptyset \Rightarrow \{0.3/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.5/h\}$ and

$$S(\ \{0.3/a, 0.3/d, 0.4/g, 0.4/h\},$$
$$\{0.3/a, 0.2/c, 0.9/d, 0.5/e, 0.6/f, 0.8/g, 0.4/h\}) = 1$$

then the output is $\mathrm{T} \vdash \{0.2/c, 0.6/f\} \overset{0.8}{\Rightarrow} \{0.5/a, 0.5/d, 06/g, 0.6/h\}$.

In the above example, we have shown a successful derivation. In the following one the derivation fails.

Example 4. We consider the same truthfulness structure described in previous examples. Let T be the following fuzzy theory:

$$\begin{aligned} T = \big\{\ & \{0.2/c, 0.4/f\} \overset{0.8}{\Rightarrow} \{0.9/d\}, \\ & \{0.7/c, 0.3/d\} \overset{0.8}{\Rightarrow} \{0.5/a, 0.5/b, 0.5/c\}, \\ & \{0.1/d, 0.3/f\} \overset{0.9}{\Rightarrow} \{0.3/c, 0.3/e, 0.8/f\}, \\ & \{0.1/a, 0.1/d\} \overset{0.8}{\Rightarrow} \{0.4/e, 0.2/f\}\big\} \end{aligned}$$

and we want to check if

$$T \vdash \{0.2/a, 0.3/f\} \overset{0.8}{\Rightarrow} \{0.5/c, 0.4/d, 0.4/f\}$$

The trace of the execution of the **FASL** *Implication Automated Prover* is the following:

1. The *guide* is $\{\emptyset \Rightarrow A_1\} := \big\{\emptyset \Rightarrow \{0.2/a, 0.3/f\}\big\}$
2. Compute *crisp*(T). That is,

$$\begin{aligned} crisp(T) = \big\{\ & \{0.2/c, 0.4/f\} \Rightarrow \{0.7/d\}, \\ & \{0.7/c, 0.3/d\} \Rightarrow \{0.3/a, 0.3/b, 0.3/c\}, \\ & \{0.1/a, 0.1/d\} \Rightarrow \{0.2/e\}\big\} \end{aligned}$$

and applying **DeEq** to every formula in *crisp*(T) obtains

$$\begin{aligned} \mathrm{T}_1 = \big\{\ & \{0.2/c, 0.4/f\} \Rightarrow \{0.7/d\}, \\ & \{0.7/c, 0.3/d\} \Rightarrow \{0.3/a, 0.3/b\}, \\ & \{0.1/a, 0.1/d\} \Rightarrow \{0.2/e\}\big\} \end{aligned}$$

3. Applying Equivalences to *guide* and each $U \Rightarrow V \in \mathrm{T}_i$:

(**gSiEq**): $A' = \{0.2/a, 0.2/d, 0.3/f\}$

$$\big\{\emptyset \Rightarrow \{0.2/a, 0.3/f\}, \{0.2/c, 0.4/f\} \Rightarrow \{0.7/d\}\big\} \equiv$$
$$\equiv \big\{\emptyset \Rightarrow \{0.2/c, 0.6/f\}, \{0.2/c, 0.4/f\} \Rightarrow \{0.7/d\}\big\}$$

$$
\begin{aligned}
\{\emptyset \Rightarrow A_2\} = \{\ & \emptyset && \Rightarrow \{0.2/a, 0.2/d, 0.3/f\}\ \} \\
\mathrm{T}_2 = \Big\{\ & \{0.2/c, 0.4/f\} && \Rightarrow \{0.7/d\}, \\
& \{0.7/c, 0.3/d\} && \Rightarrow \{0.3/a, 0.3/b\}, \\
& \{0.1/a, 0.1/d\} && \Rightarrow \{0.2/e\}\Big\}
\end{aligned}
$$

(**gSiAxEq**): $A' = \{0.2/a, 0.2/d, 0.2/e, 0.3/f\}$

$$
\begin{aligned}
&\big\{\emptyset \Rightarrow \{0.2/a, 0.2/d, 0.3/f\}, \{0.2/c, 0.4/f\} \Rightarrow \{0.7/d\}\big\} \equiv \\
\equiv &\big\{\emptyset \Rightarrow \{0.2/a, 0.2/d, 0.3/f\}, \emptyset \Rightarrow \emptyset\big\} \\
\equiv &\big\{\emptyset \Rightarrow \{0.2/a, 0.2/d, 0.3/f\}\big\}
\end{aligned}
$$

$$
\begin{aligned}
\{\emptyset \Rightarrow A_3\} = \{\ & \emptyset && \Rightarrow \{0.2/a, 0.2/d, 0.2/e, 0.3/f\}\ \} \\
\mathrm{T}_3 = \Big\{\ & \{0.2/c, 0.4/f\} && \Rightarrow \{0.7/d\}, \\
& \{0.7/c, 0.3/d\} && \Rightarrow \{0.3/a, 0.3/b\},
\end{aligned}
$$

4. A fix point is achieved and it is not possible to apply equivalences and

$$S(\ \{0.5/c, 0.4/d, 0.4/f\}, \{0.2/a, 0.2/d, 0.3/f, 0.2/e\}\ ) < 1$$

the output of the algorithm is

$$\mathrm{T} \not\vdash \{0.2/a, 0.3/f\} \stackrel{0.8}{\Rightarrow} \{0.5/c, 0.4/d, 0.4/f\}$$

References

[1] W. Armstrong, Dependency structures of data base relationships, in: IFIP Congress, 1974, pp. 580–583.
[2] C. Beeri, P.A. Bernstein, Computational problems related to the design of normal form relational schemas, ACM Trans. Database Syst. 4 (March 1979) 30–59.
[3] R. Belohlavek, Fuzzy Relational Systems: Foundations and Principles, Kluwer Academic Publishers, Norwell, MA, USA, 2002.
[4] R. Belohlavek, V. Vychodil, Axiomatizations of fuzzy attribute logic, in: IICAI'05, 2005, pp. 2178–2193.
[5] R. Belohlavek, V. Vychodil, Attribute implications in a fuzzy setting, in: R. Missaoui, J. Schmidt (Eds.), ICFCA, in: Lecture Notes in Computer Science, vol. 3874, Springer, Berlin/Heidelberg, 2006, pp. 45–60.
[6] R. Belohlavek, V. Vychodil, Data tables with similarity relations: functional dependencies, complete rules and non-redundant bases, in: M. Li Lee, K.-L. Tan, V. Wuwongse (Eds.), Database Systems for Advanced Applications, in: Lecture Notes in Computer Science, vol. 3882, Springer, Berlin/Heidelberg, 2006, pp. 644–658.
[7] R. Belohlavek, V. Vychodil, Basic algorithm for attribute implications and functional dependencies in graded setting, Int. J. Found. Comput. Sci. 19 (2) (2008) 297–317.
[8] R. Belohlavek, V. Vychodil, Logical foundations for similarity-based databases, in: L. Chen, C. Liu, Q. Liu, K. Deng (Eds.), Database Systems for Advanced Applications, in: Lecture Notes in Computer Science, vol. 5667, Springer, Berlin/Heidelberg, 2009, pp. 137–151.
[9] R. Belohlavek, V. Vychodil, Query systems in similarity-based databases: logical foundations, expressive power, and completeness, in: ACM SAC, 2010, pp. 1648–1655.
[10] R. Belohlavek, V. Vychodil, Codd's relational model from the point of view of fuzzy logic, J. Log. Comput. 21 (5) (2011) 851–862.
[11] F. Esteva, L. Godo, C. Noguera, Fuzzy logics with truth hedges revisited, in: Proc. EUSFLAT 2011 and LFA 2011. Advances in Intelligent Systems Research, Atlantis Press, 2011, pp. 146–152.
[12] F. Esteva, L. Godo, C. Noguera, A logical approach to fuzzy truth hedges, Inf. Sci. 232 (2013) 366–385.
[13] N. Galatos, P. Jipsen, T. Kowalski, H. Ono, Residuated Lattices: An Algebraic Glimpse at Substructural Logics, 1st edition, vol. 151, Elsevier Science, San Diego, USA, 2007.
[14] G. Gerla, Fuzzy Logic. Mathematical Tools for Approximate Reasoning, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
[15] S. Gottwald, A Treatise on Many-Valued Logics, Studies in Logic and Computation, vol. 9, Research Studies Press, Baldock, Nov. 2000.
[16] P. Hájek, Metamathematics of Fuzzy Logic, Trends in Logic: Studia Logica Library, vol. 4, Kluwer Academic Publishers, Dordrecht, 1998.
[17] P. Hájek, On very true, Fuzzy Sets Syst. 124 (3) (2001) 329–333.
[18] R. Holzer, Knowledge acquisition under incomplete knowledge using methods from formal concept analysis: part I, Fundam. Inform. 63 (1) (2004) 17–39.
[19] E. Klement, R. Mesiar, E. Pap, Triangular Norms. Trends in Logic: Studia Logica Library, Kluwer Academic Publishers, 2000.
[20] D. Maier, The Theory of Relational Databases, Computer Software Engineering Series, Computer Science Press, 1983.
[21] E. Orłowska, A. Radzikowska, Double residuated lattices and their applications, in: H. de Swart (Ed.), Relational Methods in Computer Science, in: Lecture Notes in Computer Science, vol. 2561, Springer, Berlin/Heidelberg, 2002, pp. 171–189.
[22] J. Pavelka, On fuzzy logic I, II, III, Math. Log. Q. 25 (Jan. 1979) 45–52.
[23] G. Takeuti, S. Titani, Globalization of intuitionistic set theory, Ann. Pure Appl. Log. 33 (1987) 195–211.