



HAL
open science

Provenance Calculus and Possibilistic Logic: A Parallel and a Discussion

Salem Benferhat, Didier Dubois, Henri Prade

► **To cite this version:**

Salem Benferhat, Didier Dubois, Henri Prade. Provenance Calculus and Possibilistic Logic: A Parallel and a Discussion. 17th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2023), CRIL, Université d'Artois - CNRS, Sep 2023, Arras, France. pp.427-441, 10.1007/978-3-031-45608-4_32 . hal-04414547

HAL Id: hal-04414547

<https://hal.science/hal-04414547>

Submitted on 24 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Provenance calculus and possibilistic logic: a parallel and a discussion

Salem Benferhat¹, Didier Dubois², and Henri Prade²

¹ CRIL, University of Artois, CNRS UMR 8188
Rue Jean Souvraz, Lens, France
benferhat@cril.fr

² IRIT-CNRS, Université Paul Sabatier, 118, route de Narbonne, 31062 Toulouse Cedex 9,
France
{didier.dubois, henri.prade}@irit.fr

Abstract. Provenance calculus has been introduced, about fifteen years ago, for complementing relational algebra calculations in databases, with semiring operations in order to handle data lineage, incomplete or probabilistic information. Possibilistic logic has started to be developed twenty years before, initially for dealing with epistemic uncertainty, using the max-min semiring. Since then, several variants and generalizations of possibilistic logic have been proposed, involving various semirings. All these forms of possibilistic logic are surveyed and paralleled with provenance calculus, through logical counterparts of relational algebra operations. The paper ends with a discussion of the parallel between the two research trends.

1 Introduction

Semirings are a mathematical structure that involves two associative operations with distinct identity elements, the former, viewed as an “addition”, being commutative, and the latter, viewed as a “product”, being distributive with respect to the former; moreover the identity element of the former (denoted ‘0’) is an annihilating element for the latter. When the product-like operation is commutative, one speaks of commutative semiring. Such structures, clearly weaker than the computational structure with real numbers, are currently encountered in many fields related to the treatment of information, such as “tropical” semirings in automata theory [47], dioids in discrete event processes [21], or various semirings in flexible constraint satisfaction problems in artificial intelligence [54, 12], but also in fuzzy relational equations [23].

It has been proposed, about fifteen years ago, to associate tuples in relational databases with different types of annotations [40, 42]. Annotations may be integers, subsets, polynomials, but also Boolean expressions, probabilities, or levels in a finite scale. In each case, semiring operations can be applied to these annotations, giving birth to various forms of so-called provenance calculi. Thus, we can in particular describe what tuples have been involved in the computation of an output tuple belonging to the result of a query (“why-provenance”), or how the output tuple is derived (“how-provenance”), where pieces of data are copied from (“where-provenance”) [19]. The idea of provenance is often related to the lineage [22] of output tuples. Yet, the term “provenance” applies to any calculus propagating tuple annotations by means of semiring operations.

Quite similar to the annotation of tuples is the association of logical formulas with weights or labels in weighted logics [25]. A prototypical example of this idea is the possibilistic logic, introduced in the mid 1980's [32]. Basic possibilistic logic formulas are pairs made of a formula and its certainty level, understood as a lower bound of a necessity measure. Since the necessity of the conjunction of two formulas is the minimum of the necessities of each formula, and we look for the derivation of formulas with their greatest certainty level, possibilistic uncertainty is propagated by means of a max-min semiring. But there exists a number of variants of possibilistic logic involving different semirings. Interestingly enough, queries to a relational database where tuples are associated with certainty levels can be handled in the possibilistic logic setting [50].

This paper aims at emphasizing the parallel between data provenance calculi with possibilistic logic and related logics, and at identifying how provenance can be handled in possibilistic logic for explanation purposes for instance. The paper is organized as follows. Section 2 provides a brief presentation of the main provenance calculi. Section 3 surveys the various semirings at work in possibilistic and related logics. Section 4 provides a final discussion of the parallel between these two research trends.

2 Provenance calculus

The idea of provenance was already present in many database works at the beginnings of years 2000's, e.g. [22], [15], [14]. Provenance is then often linked to the annotation of data. However, the proposal of endowing annotations with operations having a semiring structure appeared in the work of Green and Tannen (and their co-workers) from 2006 [42, 40]. Detailed studies or introductions to provenance calculus in databases can be found respectively in [19] and in [20, 52]. Provenance calculus aims at propagating tuple annotations when computing query answers using relational algebra operations. We use an example, due to Green and Tannen, for explaining the main ideas [42].

Let us consider two relations R and S respectively defined on attributes (A, B, C) and (D, B, E) . Let abc and dbe be two tuples of R and S respectively, with respective annotations p and r . The product-like operation (denoted \cdot) of the semiring associates $p \cdot r$ to the tuple $abcde$ of the relation $R \bowtie S$ obtained as the join of R and S on B and defined on attributes (A, B, C, D, E) . Given two relations R and S that both contain the tuple abc with respective annotations p and r , an addition-like relation denoted by $+$ associates $p+r$ to the tuple abc in the union $R \cup S$ of R and S . This operation is also used in case of projection operation (denoted by π) for keeping track of the different tuples whose projection yields the same tuple. Let us take an example.

Example 1. Consider the query $Q = \sigma_{C=e} \pi_{AC} (\pi_{AC} R \bowtie \pi_{BC} R \cup \pi_{AB} R \bowtie \pi_{AC} R)$ addressed to a relation R defined on attributes (A, B, C) (π stands for projection, σ for selection, \bowtie for join). Suppose R contains the 3 tuples abc , dbe and fge , respectively annotated by p , r and s . Then it can be checked that the output relation is given in the Table 1 thereafter.

In this table, for final selection operation, we multiply with two special annotations, 1 and 0, depending if $C = e$ or not ; p^2 is short for $p \cdot p$. One may replace $p + p$ by $2p$, and $p^2 + p^2$ by $2p^2$. In fact, the agreement of the calculus on formal annotations

A	C	
a	c	$(p^2 + p^2) \cdot 0$
a	e	$p \cdot r \cdot 1$
d	c	$r \cdot p \cdot 0$
d	e	$(r^2 + r \cdot s + r^2) \cdot 1$
f	e	$(s^2 + s \cdot r + s^2) \cdot 1$

Table 1: Example of provenance calculus with polynomials

with the properties of union, join, projections and selections in relational algebra lead to require that this calculus follows the properties of a commutative semiring [40].

As can be seen in the example, formal polynomials are used for encoding the way the tuples are obtained. For instance, the output tuple $d e$ in Table 1 is annotated with the polynomial $2r^2 + r \cdot s$. It acknowledges the fact that there are three different ways to derive $d e$ from relation R , two of them use r only (but twice), while the third way uses r and s only once.

Now if we replace the formal annotations, p , r and s by integers (e.g., 2, 5 and 1 respectively) understood as the number of copies that there exists in R for each tuple, we can count the number of ways of obtaining the output tuple $d e$ for instance, by applying the polynomial $2r^2 + r \cdot s$ (namely, $2 \times 5^2 + 5 \times 1 = 55$). This is the bag semantics, where annotations are multiplicities. Then the semiring is just $(\mathbb{N}, +, \times, 0, 1)$.

Another reading of the polynomials is to regard annotations, p , r and s in the above example, as Boolean variables, where $+$ and \cdot are taken as the Boolean disjunction \vee and conjunction \wedge respectively. Then Table 2 is the rewrite of Table 1 using a Boolean semiring (where \perp and \top stand for “false” and “true” respectively):

A	C	
a	c	$((p \wedge p) \vee (p \wedge p)) \cdot \perp = \perp$
a	e	$(p \wedge r) \wedge \top = p \wedge r$
d	c	$(r \wedge p) \wedge \perp = \perp$
d	e	$((r \wedge r) \vee (r \wedge s) \vee (r \wedge r)) \wedge \top = r$
f	e	$((s \wedge s) \vee (s \wedge r) \vee (s \wedge s)) \wedge \top = s$

Table 2: Example of provenance calculus with a Boolean semiring

If for example, we allocate the value “false” to r , meaning that the tuple $d b e$ is not in R , then the output relation reduces to $f e$, with the annotation s . Thus, one can determine under which truth condition on the tuples in R , an output tuple belongs to the answer. This corresponds to the Boolean semiring $(\mathbb{B}, \vee, \wedge, \perp, \top)$. However, note that there is no negation in the logical expressions.

This is very similar to conditional tables (c-tables) introduced as early as 1984 for handling incomplete information [44] where tuples can be associated with truth conditions. In a c-table (see [1] for details), tuples can be annotated with logical expressions

involving \neg , \vee , or \wedge , and atomic conditions that can be combined, such as $(x = a)$ or $(x = y)$ (where a is an attribute domain value); moreover logical constraints on variables can be introduced. C-tables have been recognized as a powerful setting for representing incomplete information in databases. This is now exemplified [1, 39].

Example 2. Suppose we know that Sally is taking math ($z = 0$) or computer science (CS) ($z \neq 0$) (but not both) and another course (x); Alice takes biology if Sally takes math (i.e., $z = 0$), and math ($t = 0$) or physics ($t \neq 0$) but not both if Sally takes physics ($= x$). This is represented thereafter by the *c*-table on Table 3.

<i>Student</i>	<i>Course</i>	<i>Condition</i>
	$(x \neq \text{math}) \wedge (x \neq \text{CS})$	
<i>Sally</i>	<i>math</i>	$(z = 0)$
<i>Sally</i>	<i>CS</i>	$(z \neq 0)$
<i>Sally</i>	<i>x</i>	
<i>Alice</i>	<i>biology</i>	$(z = 0)$
<i>Alice</i>	<i>math</i>	$(x = \text{physics}) \wedge (t = 0)$
<i>Alice</i>	<i>physics</i>	$(x = \text{physics}) \wedge (t \neq 0)$

Table 3: Example of a *c*-table

Instead of annotating tuples with Boolean variables, one may use sets as well. Such a set may represent an event to which a probability degree is attached that represents the probability that the tuple is present in the database [41]. Then the probabilities of tuples in the output of a query can be computed from the resulting annotated table, assuming the independence of the events annotating the tuples in the relational database [37].

Some applications of provenance calculus may involve semirings other than the ones involving integers, logical expressions, or sets we have already mentioned. A good example is provided by access control levels [36]: Assume now that an XML database is annotated with security information, where not only the tuples as a whole, but possibly attribute values may be labelled with access levels belonging to the following totally ordered scale $\mathbb{A} = P < C < S < T < 0$, where P means “public”, C “confidential”, S “secret”, and T “top-secret” (0 stands for something as “completely unaccessible”). Then a database tuple may already involve “product” of access levels in case of attributes that are annotated. Generally speaking, the output tuples are associated with polynomials that have to be interpreted using the semiring $(\mathbb{A}, \min, \max, 0, P)$, e.g., an output tuple annotated with $C \cdot S \cdot T + C^2 \cdot S = S$ corresponds to a “secret” piece of information, applying the semiring operations. Another case where not only tuples but also attribute values within tuples can be annotated with provenance information is the handling of queries involving aggregate operations (such as min, max, sum, or average) on attribute values [2].

Thus the merit of provenance semiring is to provide a means to answer queries such as “Is this piece of data derivable from trusted tuples?” or “What score should this answer receive, given initial scores of the base tuples?” [45]. As emphasized in [20] “Most work on provenance in databases focused on finding minimal subsets of a

dataset that witness the existence of a tuple in the result, as well as which parts of the dataset are the tuple copied from.”

Among other applications of provenance, let us mention causality: for example, in [18], they investigate structural causal models as a semantics for Open Provenance Model graphs (which have a provenance interpretation in the sense of the Semantic Web). Provenance semirings have been also considered for description logics, such as attributes-based DL-lite [13, 16].

3 Possibilistic logic

We first present the basic possibilistic logic (and some variants) where the weights associated to the formulas belong to a completely ordered scale. Then we review various extensions where weights belong to partially ordered structures, especially lattices. After which, we briefly survey applications (fusion, revision, inconsistency handling) where “products” other than min, or polynomials are useful. Lastly, we recall how (basic) possibilistic logic can help to handling uncertainty in databases.

3.1 Basic possibilistic logic and variants

Possibilistic logic (PL) is a special form of weighted logic [25]. It starts with the idea of associating a classical logic formula with a certainty level. A basic PL formula [32] is a pair (p, α) made of a logical formula p associated with a certainty level $\alpha \in (0, 1]$ (or in any bounded totally ordered scale \mathbb{S}), viewed as a lower bound of a *necessity measure* N , i.e., (p, α) is semantically understood as $N(p) \geq \alpha$.³ Formulas of the form $(p, 0)$, contain no information ($N(p) \geq 0$ always holds), and are not considered. Thanks to the minitivity property for conjunction that characterizes necessity measures, i.e., $N(p \wedge q) = \min(N(p), N(q))$, a PL base, i.e., a set of PL formulas, can be always put in an equivalent clausal form. The necessity measure N is associated by duality with a possibility measure $\Pi(p) = 1 - N(\neg p) = \max_{\omega \models p} \pi(\omega)$, where π is a possibility distribution on interpretations.

In PL, the following inference rule is valid: $(\neg p \vee q, \alpha), (p \vee r, \beta) \vdash (q \vee r, \min(\alpha, \beta))$. It can be shown that this weakest link resolution rule yields the greatest lower bound that can be attached to $q \vee r$. This resolution rule is used repeatedly in a refutation-based proof procedure that is sound and complete w. r. t. the semantics of propositional possibilistic logic [28].

The semantics of a PL base $\mathcal{B} = \{(p_k, \alpha_k) \mid k = 1, \dots, n\}$ is in terms of a possibility distribution $\pi_{\mathcal{B}}(\omega) = \min_{k=1, \dots, n} \max([p_k](\omega), 1 - \alpha_k)$ where $[p_k]$ denotes the set of models of p_k , and $\max([p_k](\omega), 1 - \alpha_k)$ is the possibility distribution (fuzzy set) interpreting (p_k, α_k) . The possibility distribution $\pi_{\mathcal{B}}$ is the fuzzy intersection of the n such fuzzy sets. It expresses that an interpretation ω is all the more possible as it does not violate a formula having a higher certainty. The necessity measure associated with the distribution $\pi_{\mathcal{B}}$ is defined by $N(p) = \min_{\omega \not\models p} 1 - \pi_{\mathcal{B}}(\omega)$. Moreover a level of inconsistency $inc(\mathcal{B})$ is associated to \mathcal{B} ; it is the largest weight with which the contradiction

³ Possibilistic logic can also be used for modeling preferences, then (p, α) is understood as a goal p with priority level α .

can be inferred from \mathcal{B} . The set of all formulas with certainty levels strictly larger than $\text{inc}(\mathcal{B})$ is consistent.

The inference exploits a refutation method in order to reach the empty clause with the greatest possible certainty level. The computations use the semiring $(\mathbb{S}, \max, \min, 0, 1)$. This semiring is similar to the already encountered semiring $(\mathbb{A}, \min, \max, 0, P)$, except that the order in the scale is reversed. The complexity remains similar to the one of classical logic (it is multiplied by the logarithm of the number of distinct certainty levels present in the PL base we start with). Introductions, details, applications to various artificial intelligence problems can be found in [32, 28, 33, 34].

We may wonder if a similar calculus would be possible with probabilities. Indeed there exists a probabilistic counterpart of the resolution rule, namely, $P(\neg p \vee q) \geq \alpha, P(p \vee r) \geq \beta \vdash P(q \vee r) \geq \max(0, \alpha + \beta - 1)$ where the Frechet bound obtained is the greatest lower bound that can be proved valid. Unfortunately, the repeated use of this rule does not lead to an inference process that is complete [28].

Possibilistic logic can be viewed as a special case of a labelled deductive system [38] where logical formulas are associated with various kinds of formulas or weights belonging to some lattice structure. Basically, a PL formula is a pair made of a classical logic formula and a label that qualifies in which conditions or to what extent the classical logic formula is regarded as certainly true.

One may think of associating “labels” other than certainty levels. It may be lower bounds of other measures in possibility theory, such as in particular “strong possibility” measures Δ , which are characterized by the decomposability property $\Delta(p \vee q) = \min(\Delta(p), \Delta(q))$. They obey the “resolution-like” rule $\Delta(\neg p \wedge q) \geq \alpha, \Delta(p \wedge r) \geq \beta \vdash \Delta(q \wedge r) \geq \min(\alpha, \beta)$. $\Delta(p)$ can be interpreted as a degree of evidential support for p , since we have $\Delta(p) = \min_{\omega \models p} \pi_{\mathcal{D}}(\omega)$, where $\pi_{\mathcal{D}}(\omega) = \max_{j=1, \dots, m} \min([q_j](\omega), \beta_j)$ is the possibility distribution associated with a base of Δ constraints $\{\Delta(q_j) \geq \beta_j \mid j = 1, \dots, m\}$ expressing that each q_j is guaranteed possible at least at level β_j . Another interpretation of $\Delta(p)$ is in terms of desire [29]. Again computations in this logic uses the semiring $(\mathbb{S}', \max, \min, 0, 1)$ (where \mathbb{S}' is the scale for possibility levels).

Let us mention a construction similar to possibilistic logic, but made in an additive setting where each formula is associated with a cost (in $\mathbb{N} \cup \{+\infty\}$). This logic associates, to each formula of the logic base, the price to pay if this formula is violated. The weight (cost) attached to an interpretation is the sum of the costs of the formulas in the base violated by the interpretation; this is the starting point of *penalty logic* [35, 48]. It contrasts with possibilistic logic, where weights are combined by an idempotent operation. The so-called “cost of consistency” of a formula is then defined as the minimum of the weights of its models (which is a ranking function in the sense of Spohn [53], or the counterpart of a possibility measure defined on $\mathbb{N} \cup \{+\infty\}$ where now 0 expresses full possibility, and $+\infty$ complete impossibility since it is a cost that cannot be paid). The best model has a cost equal to 0 if the set of formulas is consistent. This logic relies on the semiring $(\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$; by a logarithmic transformation, one can move to the semiring $([0, 1], \max, \text{product}, 0, 1)$ instrumental in product-based possibilistic logic.

3.2 Weights in partially ordered structures

Logical formulas may be also associated with labels taking values in partially ordered structures, such as lattices. This can be motivated by different needs, as briefly reviewed in the following.

A rather recent example, *interval-based possibilistic logic*, has been proposed in [10] where classical logic formulas are associated with intervals, supposed to gather certainty levels provided by different sources. More precisely, an interval-based possibilistic knowledge base is a set of weighted formulas of the form $\mathcal{GK} = \{(\phi_i, \mathcal{G}_i) : i = 1, \dots, n\}$ where $\mathcal{G}_i = [\alpha_i, \beta_i]$ is a closed sub-interval of $]0, 1]$. The pair (ϕ_i, \mathcal{G}_i) , called an interval-based weighted formula, means that the weight associated with ϕ_i belongs to \mathcal{G}_i . This interpretation of (ϕ_i, \mathcal{G}_i) is different from the one used in [26], where (ϕ_i, \mathcal{G}_i) is understood as $\forall \alpha_i \in \mathcal{G}_i, (\phi_i, \alpha_i)$ is true. Unlike standard possibilistic logic, an interval-based possibilistic logic only induces a partial pre-order over the set of interval-based weighted formulas. Let (ϕ_1, \mathcal{G}_1) and (ϕ_2, \mathcal{G}_2) be two interval-based formulas of the interval-based possibilistic knowledge base, with $\mathcal{G}_1 = [\alpha_1, \beta_1]$ and $\mathcal{G}_2 = [\alpha_2, \beta_2]$. For reasoning from interval knowledge bases, the semiring $(\mathcal{G}^{[0,1]}, \mathcal{M}, m, [0,0], [1,1])$ is used, where $\mathcal{G}^{[0,1]}$ represents the set of all closed subintervals of $]0, 1]$, \mathcal{M} is defined by

$$\mathcal{M}([\alpha_1, \beta_1], [\alpha_2, \beta_2]) = [\max(\alpha_1, \alpha_2), \max(\beta_1, \beta_2)].$$

Similarly, the minimum of two intervals is defined by

$$m([\alpha_1, \beta_1], [\alpha_2, \beta_2]) = [\min(\alpha_1, \alpha_2), \min(\beta_1, \beta_2)].$$

The corresponding partial ordering associated with the above lattice is $\mathcal{G}_1 \geq \mathcal{G}_2$ if and only if $\alpha_1 \geq \alpha_2$ and $\beta_1 \geq \beta_2$. An interesting point is that with the use of this semiring, standard possibilistic logic reasoning has been extended to deal with interval-based weighted formulas without inducing extra computational cost.

Timed possibilistic logic [26] has been the first proposed extension of this kind. Logical formulas are then associated with sets of time instants where the formula is known as being certainly true. More generally certainty may be graded as in basic possibilistic logic, and then formulas are associated with *fuzzy* sets of time instants where the grade attached to a time instant is the certainty level with which the formula is true at that time. In such a reified temporal logic it is important to make sure that the knowledge base remains consistent over time. At the semantic level, it leads to an extension of necessity (and possibility) measures now valued in a distributive lattice structure where necessity functions are (fuzzy) set-valued. We are thus working with the commutative semiring $(\mathbb{S}^{\mathbf{T}}, \cup_{\max}, \cap_{\min}, \emptyset, \mathbf{T})$ where \mathbf{T} is the set of time instants, $\mathbb{S}^{\mathbf{T}}$ the set of \mathbb{S} -graded fuzzy sets over \mathbf{T} , \cup_{\max} and \cap_{\min} denote the max- and min-based fuzzy set union and intersection respectively.

Taking inspiration of possibilistic logic, Lafage, Lang and Sabbadin [46] have proposed a *logic of supporters*, where each formula p is associated with a “supporter”, that is a subset of subsets of “assumptions” that encodes a disjunction of conjunctions whose truth “supports” the truth of p . This logic is another lattice-based generalisation of possibilistic logic, where support measures (which play the role of necessity measures) are valued on the power set of the power set of the set of assumptions. Thus,

this corresponds to the use of the semiring $(2^{2^{\mathbb{H}}}, \oplus, \otimes, \{\}, \{\{\}\})$ where \mathbb{H} is the set of assumptions, \oplus, \otimes are the lattice operations associated with the partial order between supporters ($S_2 \sqsubseteq S_1$ iff $\forall E_2 \in S_2, \exists E_1 \in S_1$ s.t. $E_1 \subseteq E_2$, which intuitively means that S_1 is easier to satisfy than S_2 .)

Still another, simple, example of a lattice-based extension of possibilistic logic is multi-source possibilistic logic [27], where each formula is associated with a set of distinct explicit sources that support its truth. Again, a certainty / confidence level (belonging to \mathbb{S} may be attached to each source, and then formulas are associated with fuzzy sets of sources. This corresponds to working with the semiring $(\mathbb{S}^{\mathbf{S}}, \cup_{\max}, \cap_{\min}, \emptyset, \mathbf{S})$ where \mathbf{S} is the set of all sources.

In possibilistic logic, as well as in its extensions and variants, we deal with pairs where formulas and weights are handled simultaneously but on their own. Still, literals can be moved to the “weight lot”. Indeed, a formula such as $(\neg p \vee q, \alpha)$ can be rewritten under the semantically equivalent form $(q, \min([p], \alpha))$, where $[p] = 1$ if p is true and $[p] = 0$ if p is false ($[p]$ can be viewed as the characteristic function of the set of models of p). This latter formula now reads “ q is α -certain, provided that p is true”, and can be used in hypothetical reasoning in case no formula (p, γ) is deducible from the available information [8, 31]. In the special case where all the certainty weights are equal to 1 and where we would start with a base made of a set of pairs associating a formula with the set of models of an hypothetical formula, we would deal with a semiring of the form $(2^{\mathcal{G}}, \cup, \cap, \emptyset, \mathcal{G})$ where \mathcal{G} is the set of interpretations induced by the language of the hypothetical formulas.

It is also possible to move the weight inside the formula. Namely, a possibilistic formula (p, α) is rewritten as a classical two-sorted clause $p \vee ab_\alpha$, where ab_α means *the situation is α -abnormal*, and thus the clause expresses that *p is true or the situation is abnormal*, while more generally $(p, \min(\alpha, \beta))$ is rewritten as the clause $p \vee ab_\alpha \vee ab_\beta$. This leads to a possibilistic-like many-sorted propositional logic, first presented in [11], which was proposed for handling *partial orderings* between weights. Then a known constraint between unknown weights such as $\alpha \geq \beta$ is translated into a clause $\neg ab_\alpha \vee ab_\beta$. Another slightly different approach [17] handles the unknown weights in a purely symbolic manner, i.e., computes the level from a derived formula as a symbolic expression. For instance, $\mathcal{B} = \{(p, \alpha), (\neg p \vee q, \beta), (q, \gamma)\} \vdash (q, \max(\min(\alpha, \beta), \gamma))$. There still exists a partial order between formulas based on the partial order between symbolic levels; this leads to a logic (for which completeness has been proven) that comes close to the logic of supporters.

Multiple agent possibilistic logic was outlined in [4], but its underlying semantics and completeness results have been laid bare only in [5]. A multiple agent propositional formula is a pair (p, A) , where p is a classical propositional formula of a language \mathcal{L} and A is a non-empty subset of agents, where $A \subseteq All$ (All denotes the finite set of all considered agents; note that All may not be known in extension). The intuitive meaning of formula (p, A) is that *at least all* the agents in A believe that p is true. More general formulas of the form $(p, \alpha/A)$ are also considered; they mean that at least all the agents in A are certain that p is true *at least* at level α . In spite of the obvious parallel with possibilistic logic (where propositions are associated with levels expressing the strength with which the propositions are believed to be true), (p, A)

should not be just used as another way of expressing the strength of the support in favor of p (the larger A , the stronger the support), but rather as a piece of information linking a proposition with a group of agents. The resolution rule is now if $A \cap B \neq \emptyset$, then $(\neg p \vee q, \alpha/A), (p \vee r, \beta/B) \vdash (q \vee r, \min(\alpha, \beta)/(A \cap B))$. This multiple agent logic should not be confused with multiple source logic. In the former, each agent may be viewed as a source, but what is manipulated is thus a subset of sources taken as a whole; what matters in multiple agent logic is the collective consistency of *subsets* of agents (while the collection of the beliefs held by the whole set of agents may be inconsistent). We are dealing here with the semiring $(\mathbb{S} \times \mathbb{B}(\mathbb{L}_A), (\max, \cup), (\min, \cap), (0, \emptyset), (1, All))$ where $\mathbb{B}(\mathbb{L}_A)$ is the Boolean algebra induced by the subsets appearing in the formulas of the base.

3.3 Other products and polynomials

The semantics of a possibilistic logic base is a possibility distribution over the set of interpretations. In information fusion, the combination of possibility distributions can be equivalently performed in terms of PL bases: The syntactic counterpart of the point-wise combination of two possibility distributions π_1 and π_2 into a distribution $\pi_1 \otimes \pi_2$ by any monotonic combination operator \otimes such that $1 \otimes 1 = 1$, can be computed in the following way: Namely, if the PL base \mathcal{B}_1 is associated with π_1 and the base \mathcal{B}_2 with π_2 , a PL base $\mathcal{B}_{1 \otimes 2}$ semantically equivalent to $\pi_1 \otimes \pi_2$ is given by [9]:

$$\{(p_i, 1 - (1 - \alpha_i) \otimes 1) \text{ s.t. } (p_i, \alpha_i) \in \mathcal{B}_1\} \cup \{(q_j, 1 - 1 \otimes (1 - \beta_j)) \text{ s.t. } (q_j, \beta_j) \in \mathcal{B}_2\} \\ \cup \{(p_i \vee q_j, 1 - (1 - \alpha_i) \otimes (1 - \beta_j)) \text{ s.t. } (p_i, \alpha_i) \in \mathcal{B}_1, (q_j, \beta_j) \in \mathcal{B}_2\}.$$

where $1 - (\cdot)$ is the order reversing map of the scale \mathbb{S} . For $\otimes = \min$, we get $\mathcal{B}_{1 \otimes 2} = \mathcal{B}_1 \cup \mathcal{B}_2$ with $\pi_{\mathcal{B}_1 \cup \mathcal{B}_2} = \min(\pi_1, \pi_2)$ as expected (conjunctive combination). For $\otimes = \max$ (disjunctive combination), we get $\mathcal{B}_{1 \otimes 2} = \{(p_i \vee q_j, \min(\alpha_i, \beta_j)) \text{ s.t. } (p_i, \alpha_i) \in \mathcal{B}_1, \text{ and } (q_j, \beta_j) \in \mathcal{B}_2\}$. With non idempotent \oplus operators, some reinforcement effects may be obtained. We thus deal with the semiring $(\mathbb{S}, \max, \otimes, 0, 1)$ (if \otimes is associative).

In [7] a plausibility relation representation, based on polynomials, has been proposed. This work was done in the context of belief revision where the primary goal is to study reversible revision mechanisms. In this representation, the interpretations, as well as the formulas of propositional logic, are associated with plausibility values defined as polynomials. The polynomials used are with only one variable and where coefficients can only take two possible values 0 and 1. Let $\mathcal{B} = \{0, 1\}$ be the set of values 0 and 1 and let $\mathcal{B}[x]$ be the set of polynomials (with a single variable) whose coefficients belong to \mathcal{B} . The terms of the polynomials $p \in \mathcal{B}[x]$ can have positive or negative degrees and are of the form: $p = \sum_{k=1}^n p_k x^{-k} + \sum_{i=0}^m p_i x^i$ (here we abuse notations since we accept polynomials with negative exponents). Different operators have been used on the polynomials $\mathcal{B}[x]$ for the revision process. First shift operations, materialized by multiplication by x (right shift) and multiplication by x^{-1} (left shift) are used to take into account new information. Then, the maximum operator (defined with respect to the degree of the polynomials) is used to define the polynomials associated with the interpretations from the polynomial-based knowledge bases. Finally, the lexicographical order (on the set of degrees of the polynomials terms) is used for the comparison of the polynomials of $\mathcal{B}[x]$.

3.4 Databases and possibilistic logic

It has been shown [50] that the complexity of handling uncertainty in databases is considerably reduced if the uncertainty takes the form of certainty levels associated to attribute values or to tuples for relational algebra queries. The case of aggregate queries is addressed in [49]. This kind of information indeed corresponds to possibilistic logic formulas. For instance, if we go back to the Example 2 of Section 2: the information $take(Sally, math) \vee take(Sally, CS)$, corresponds to the possibility distribution $\pi_{take(Sally, \cdot)}(math) = 1 = \pi_{take(Sally, \cdot)}(CS)$, or if we prefer $\pi(z = 0) = \pi(z \neq 0) = 1$. Indeed “Sally is taking math or computer science” is expressed by $(take(Sally, math) \vee take(Sally, CS), 1)$ and the additional constraint “but not both” by $(\neg take(Sally, math) \vee \neg take(Sally, CS), 1)$.

Let us now examine the rest of Example 2. We can take for the domain of attribute *Course* the set $D_{Course} = \{math, CS, biology, physics, others\}$ that involves all the topics mentioned in the example and leave room for others. Then the information “Sally takes another course” (apart from “math” or “CS”) writes in possibilistic logic $(take(Sally, physics) \vee take(Sally, biology) \vee take(Sally, others), 1)$ while “Alice takes biology if Sally takes math, and math or physics (but not both) if Sally takes physics” writes $(take(Alice, biology), [take(Sally, math)]),$
 $(take(Alice, math) \vee take(Alice, physics), [take(Sally, physics)]),$
 $(\neg take(Alice, math) \vee \neg take(Alice, physics), 1),$

where symbolic weights are between $[\]$. We could equivalently write $(take(Alice, biology) \vee \neg take(Sally, math), 1)$ in place of $(take(Alice, biology), [take(Sally, math)])$, and this applies as well to the possibilistic formula after. Thus, the conditional table represented in Table 3 translates easily in a possibilistic logic base, and obviously can be extended to certainty levels less than 1, if needed.

Possibilistic Description Logics [43, 30] are extensions of standard Description Logic frameworks based on possibility theory that allow query answering from uncertain ontologies. The so-called lightweight ontologies are interesting fragments of DLs since they provide a good trade-off between expressive power and computational complexity. They are particularly appropriate for applications where query answering is the most important reasoning task. An example of a lightweight ontology language is DL-lite which has been extended to the framework of possibility theory [6]. In the same spirit as the standard possibilistic logic, in the logics of description possibilities, degrees of priority or importance are assigned to the axioms of TBox (terminological knowledge base) and to the assertions of ABox (assertional knowledge base). The same algebraic structures of semirings, defined in the framework of possibilistic logic, have been used in possibilistic description logics. An important point to note is that the extensions of the description logics to the framework of possibility theory have been made without additional computational cost. This is particularly true in the presence of inconsistent ontologies where the computation of the assertion repair is done in polynomial time for for both totally and partially ordered possibilistic DL-lite [6, 3].

4 Provenance and possibilistic logic - A final discussion

Provenance calculus and possibilistic logic have been motivated by different concerns: keeping track of the origin of the tuples obtained in a query on the one hand and the

handling of epistemic uncertainty on the other hand. Still, the evaluation of a query in face of a database, which, using Datalog, may be turned into a particular kind of inference problem, is not so different from the deduction from a knowledge base in the setting of some logical representations. Indeed in both cases, evaluating a query or trying to prove a formula can be associated to a graph describing the different paths leading to the output. Moreover we have seen with the variety of the different semantics associated with weights that similar concerns may be encountered in the two fields of research, as for instance, in the case of access control levels, leading to the use of isomorphic semirings.

In the provenance calculus, the product (for join) and sum (for union and projection) operators are used. In possibilistic (propositional or DL-lite) logic, the two semirings, based on (\min, \max) and $(\text{product}, \max)$, are both used. However, from a computational point of view, the use of semiring (\min, \max) offers better results and in particular it preserves the tractability of DL-lite's query-answering (which is not the case with a computation based on the product and maximum operators).

In both settings, the need for explanations seems to be a common, implicit concern. Explanations may be of different kinds in possibilistic-like logics: proof leading to the highest certainty level, best arguments supporting a conclusion, or sources involved in it. One may also need a symbolic expression keeping track of all the paths leading to the conclusion in order to determine what could influence its certainty.

To a large extent, database and AI are fields that have been developed separately. However, remarkably enough, it seems there has been absolutely no mutual exchanges between the ideas underlying provenance and epistemic uncertainty in spite of their proximity. Perhaps that's a pity. Besides, let us also note that some concerns such as consistency are proper to possibilistic-like logics where it can specialize in different forms. Besides, in case of semirings based on \min and \max operations, it may be useful to refine these operations lexicographically for breaking ties, as done, e.g., in [24].

In the presence of incoherent ontologies, the propagation mechanisms of the numerical or symbolic degrees of certainty attached to the assertions, based on the algebraic structures of semirings, make it possible to determine whether an assertion is accepted or not and whether a response to a query is considered as valid or not. In standard possibilistic logic, it is easy to provide a degree of plausibility of an answer to a query and to evaluate the degree of inconsistency of an ontology. The task becomes difficult if we have to give the best support for an answer. In the case of partially ordered bases, the challenge is above all put on finding effective methods to replace the inconsistent ontology by one of its repairs. Once a preferred repair is computed, standard mechanisms are then used on the repair and thus ignore the initial annotations necessary to justify the validity of a given conclusion. It would then be interesting to define a notion of annotated repair which would keep enough information from the initial incoherent ontology to be able to retrieve the origins of the derived conclusions.

Acknowledgements: This research has received support from the European Union's Horizon research and innovation programme under the MSCA-SE (Marie Skłodowska-Curie Actions Staff Exchange) grant agreement 101086252; Call: HORIZON-MSCA-2021-SE-01; Project title: STARWARS (STormwAteR and WastewAteR networkS het-

erogeneous data AI-driven management).

This research has also received support from the French national project ANR (Agence Nationale de la Recherche) EXPIDA (EXplainable and parsimonious Preference models to get the most out of Inconsistent DATabases), grant number ANR-22-CE23-0017.

References

1. Serge Abiteboul, Richard Hull, Victor Vianu. *Foundations of Databases*. Addison-Wesley, Reading (1995)
2. Yael Amerdamer, Daniel Deutch, Val Tannen. Provenance for aggregate queries. *Proc. 30th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS'11)*, June 12-16, Athens, ACM, (M. Lenzerini, Th. Schwentick, eds.), ACM, 153-164, 2011.
3. Sihem Belabbes, Salem Benferhat. Computing a possibility theory repair for partially pre-ordered inconsistent ontologies. *IEEE Trans. Fuzzy Syst.* 30 (8), 3237-3246, 2022.
4. Asma Belhadi, Didier Dubois, Faiza Khellaf-Haned, Henri Prade. Multiple agent possibilistic logic. *J. Appl. Non Class. Logics*, 23 (4), 299-320, 2013.
5. Asma Belhadi, Didier Dubois, Faiza Khellaf-Haned, Henri Prade. Reasoning with multiple-agent possibilistic logic. *Proc. 10th Int. Conf. on Scalable Uncertainty Management (SUM'16)*, (S. Schockaert, P. Senellart, eds.) Nice, Sept. 21-23, LNCS 9858, Springer, 67-80, 2016.
6. Salem Benferhat and Zied Bouraoui. Min-based possibilistic DL-Lite. *J. of Logic and Computation*, 27 (1), 261-297, 2017.
7. Salem Benferhat, Didier Dubois, Sylvain Lagrue, Odile Papini. Making revision reversible: an approach based on polynomials. *Fundam. Informaticae* 53 (3-4), 251-280, 2002.
8. Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade. Hypothetical reasoning in possibilistic logic: basic notions and implementation issues. In: *Between Mind and Computer, Fuzzy Science and Engineering*, (P.Z. Wang, K.F. Loe, eds.), World Scient. Publ., 1-29, 1994.
9. Salem Benferhat, Didier Dubois, Henri Prade. From semantic to syntactic approaches to information combination in possibilistic logic. In: *Aggregation and fusion of imperfect information*. (B. Bouchon-Meunier, ed.), Physica Verlag, Heidelberg, 141-161, 1998.
10. Salem Benferhat, Julien Hué, Sylvain Lagrue, Julien Rossit. Interval-based possibilistic logic. *Proc. 22nd Inter. Joint Conf. on Artificial Intelligence (IJCAI'11)*, (T. Walsh, ed.), Barcelona, July 16-22, 750-755, 2011.
11. Salem Benferhat, Henri Prade. Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic. *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, (L. Pack Kaelbling, A. Saffiotti, eds.), Edinburgh, July 30 - Aug.5, Professional Book Center, 1281-1286, 2005.
12. Stefano Bistarelli, Hélène Fargier, Ugo Montanari, Francesca Rossi, Thomas Schiex, Gérard Verfaillie. Semiring-based CSPs and valued CSPs: Basic properties and comparison. In: *Over-Constrained Systems 1995*: M. Jampel, E. C. Freuder, M. J. Maher, eds.), LNCS 1106, Springer, 111-150, 1996
13. Camille Bourgaux, Ana Ozaki. Querying attributed DL-Lite ontologies using provenance semirings. *Proc. 33rd Conf. on Artificial Intelligence (AAAI'19)*, Honolulu, 2719-2726, 2019.
14. Peter Buneman, Adriane Chapman, James Cheney, Stijn Vansummeren. A provenance model for manually curated data. *Proc. Int. Provenance and Annotation Workshop (IPAW'06)*, (L. Moreau, I. Foster, eds.), LNCS 4145, Springer, 162-170, 2006.
15. Peter Buneman, Sanjeev Khanna, Wang-Chiew Tan. Why and where: A characterization of data provenance. *Proc. 8th Int. Conf. on Database Theory (ICDT'01)*, (J. Van den Bussche, V. Vianu, eds.), London, Jan. 4-6, LNCS 1973, Springer, 316-330, 2001.

16. Diego Calvanese, Davide Lanti, Ana Ozaki, Rafael Peñaloza, Guohui Xiao. Enriching ontology-based data access with provenance. Proc. 28th Int. Joint Conf. on Artificial Intelligence (IJCAI'19), (S. Kraus, ed.), Macao, Aug. 10-16, 1616-1623, 2019.
17. Claudette Cayrol, Didier Dubois, Fayçal Touazi. Symbolic possibilistic logic: completeness and inference methods. J. Log. Comput. 28 (1), 219-244, 2018.
18. James Cheney. Causality and the semantics of provenance. Proc. 6th Workshop on Developments in Computational Models: Causality, Computation, and Physics (DCM'10), (S. B. Cooper, P. Panangaden, E. Kashefi, eds.), Edinburgh, 9-10th July, EPTCS 26, 63-74, 2010.
19. James Cheney, Laura Chiticariu and Wang-Chiew Tan. Provenance in databases: Why, how, and where. Foundations and Trends in Databases, 1 (4), 379-474, 2009.
20. James Cheney, Wang-Chiew Tan. Provenance in databases. Encyclopedia of Database Systems (2nd ed.), (Ling Liu, M. Tamer Özsu, eds.), Springer, 2018.
21. Guy Cohen, Didier Dubois, Jean-Pierre Quadrat, Michel Viot. A linear system-theoretic view of discrete event processes and its use for performance evaluation in manufacturing. IEEE Transactions on Automatic Control 30 (3), 210-220, 1985.
22. Yingwei Cui, Jennifer Widom, Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Syst. 25 (2), 179-227, 2000.
23. Bernard De Baets. Analytical methods for fuzzy relational equations. In: Fundamentals of Fuzzy Sets (D. Dubois, H. Prade, eds.), The Handbooks of Fuzzy Sets Series, Kluwer Acad. Publ., 291-340, 2000.
24. Didier Dubois, Philippe Fortemps. Selecting preferred solutions in the minimax approach to dynamic programming problems under flexible constraints. Eur. J. Oper. Res. 160 (3), 582-598, 2005.
25. Didier Dubois, Lluís Godo, Henri Prade. Weighted logics for artificial intelligence - An introductory discussion. Int. J. Approx. Reason. 55 (9), 1819-1829, 2014.
26. Didier Dubois, Jérôme Lang, Henri Prade. Timed possibilistic logic. Fundam. Informaticae 15 (3-4), 211-234, 1991.
27. Didier Dubois, Jérôme Lang, Henri Prade. Dealing with Multi-source information in possibilistic logic. Proc. 10th Europ. Conf. on Artificial Intelligence (ECAI'92), (B. Neumann, ed.), Vienna, Aug. 3-7, John Wiley and Sons, 38-42, 1992.
28. Didier Dubois, Jérôme Lang, Henri Prade. Possibilistic logic. In: D. M. Gabbay, C. J. Hogger, J. A. Robinson, D. Nute (Eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3* (pp. 439-513) Oxford University Press, 1994.
29. Didier Dubois, Emiliano Lorini, Henri Prade. The strength of desires. A logical approach. Minds and Machines, 27 (1), 199-231, 2017.
30. Didier Dubois, Jérôme Mengin, Henri Prade. Possibilistic uncertainty and fuzzy features in description logic. a preliminary discussion. In: Fuzzy Logic and the Semantic Web. Volume 1 of Capturing Intelligence, pp. 101-113, 2006.
31. Didier Dubois, Henri Prade. Combining hypothetical reasoning and plausible inference in possibilistic logic. Int. J. of Multiple-Valued Logic, 1, 219-239, 1996.
32. Didier Dubois, Henri Prade. Possibilistic logic: a retrospective and prospective view. Fuzzy Sets Syst., 144 (1), 3-23, 2004.
33. Didier Dubois, Henri Prade. Possibilistic logic - An overview. In : Computational Logic. Handbook of the History of Logic 9, (J. H. Siekmann, ed.), Elsevier 283-342, 2014.
34. Didier Dubois, Henri Prade. Possibilistic logic: From certainty-qualified statements to two-tiered logics - A prospective survey. Proc. 16th Europ. Conf. on Logics in Artificial Intelligence (JELIA'19), (F. Calimeri, N. Leone, M. Manna, eds.), Rende, May 7-11, LNCS 11468, Springer, 3-20, 2019.
35. Florence Dupin de Saint-Cyr, Jérôme Lang, Thomas Schiex: Penalty logic and its link with Dempster-Shafer theory. Proc. 10th Ann. Conf. Uncertainty in Artif. Intellig.(UAI'94), (R. López de Mántaras, D. Poole, eds.), Seattle, July 29-31, Morgan Kaufmann, 204-211, 1994.

36. J. Nathan Foster, Todd J. Green, Val Tannen. Annotated XML: queries and provenance. Proc. 27th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS'08), (M. Lenzerini, D. Lembo, eds.), June 9-11, Vancouver, 271-280, 2008.
37. Norbert Fuhr, Thomas Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. ACM Trans. Inf. Syst. 15 (1), 32-66, 1997.
38. Dov Gabbay. Labelled Deductive Systems. Volume 1. Oxford Univ. Press, Oxford, 1996.
39. Sergio Greco, Cristian Molinaro, Francesca Spezzano. Incomplete Data and Data Dependencies in Relational Databases. Springer, 2012.
40. Todd Green, Grigoris Karvounarakis, Val Tannen Provenance semirings. Proc. 26th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS'07), (Leonid Libkin, ed.), June 11-13, Beijing, China. ACM, 31-40, 2007
41. Todd J. Green, Val Tannen. Models for incomplete and probabilistic information. IEEE Data Eng. Bull. 29 (1), 17-24, 2006.
42. Todd J. Green, Val Tannen: The semiring framework for database provenance. Proc. 36th ACM SIGMOD-SIGACT-SIGAI Symp. on Principles of Database Systems (PODS'17), (E. Sallinger, J. Van den Bussche, F. Geerts, eds.) Chicago, May 14-19, ACM, 93-99, 2017.
43. B. Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics . Int. J. of Approximate Reasoning, 12 (2), 85-109, 1995.
44. Tomasz Imielinski, Witold Lipski Jr. Incomplete information in relational databases. J. of ACM 31 (4), 761-791, 1984.
45. Grigoris Karvounarakis, Zachary G. Ives, Val Tannen. Querying data provenance. Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'10), (A. K. Elmagarmid, D. Agrawal, ed.), Indianapolis, June 6-10, 951-962, 2010.
46. Céline Lafage, Jérôme Lang, Régis Sabbadin. A logic of supporters. In: Information, Uncertainty and Fusion.(Bouchon-Meunier, B., Yager, R., Zadeh, L., eds.), 381-392. Kluwer, 1999.
47. Jean-Éric Pin. Tropical semirings. In Idempotency, (J. Gunawardena, ed.) Publications of the Newton Institute, Vol. 11, 50-69, Cambridge University Press, 1998.
48. Gadi Pinkas, Ronald Prescott Loui. Reasoning from inconsistency : A taxonomy of principles for resolving conflict. Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92), (J. Allen, R. Fikes, and E. Sandewall, eds.), Cambridge, MA, 709- 719, Morgan-Kaufmann, 1992.
49. Olivier Pivert, Henri Prade. Dealing with aggregate queries in an uncertain database model based on possibilistic certainty. Proc. 15th Int. Conf. on Inform. Process. and Management of Uncert. in Knowledge-based Syst. (IPMU'14), Part III, (A. Laurent, O. Strauss, B. Bouchon-Meunier, R. Yager, eds.), Montpellier, July 15-19, CCIS 444, Springer, 150-159, 2014.
50. Olivier Pivert, Henri Prade. A certainty-based model for uncertain databases. IEEE Trans. Fuzzy Syst., 23 (4), 1181-1196, 2015.
51. Olivier Pivert, Henri Prade. Possibilistic conditional tables. Proc. 9th Int. Symp. on Foundations of Information and Knowledge Systems (FoIKS'16), (M. Gyssens, G. R. Simari, eds.), Linz, Austria, March 7-11, 2016. LNCS 9616, Springer, 42-61, 2016.
52. Pierre Senellart: Provenance in databases: Principles and applications. In: Reasoning Web. Explainable Artificial Intelligence - 15th Int. Summer School 2019, Bolzano, Sept.20-24, Tutorial Lectures, (M. Krötzsch, D. Stepanova, eds.), LNCS 11810, Springer, 104-109, 2019.
53. Wolfgang Spohn. The Laws of Belief: Ranking Theory and its Philosophical Applications, Oxford University Press, UK, 2012.
54. L.A. Zadeh. Calculus of fuzzy restrictions. In: Fuzzy Sets and their Applications to Cognitive and Decision Processes (Zadeh L. A., Fu K. S., Tanaka K. and Shimura M., eds.), Academic Press, New York, 1-39, 1975.