Chapter 4
Bipolarity in Database Querying: Various Aspects and Interpretations

Sławomir Zadrożny and Janusz Kacprzyk

Abstract A crucial problem in database querying is how to devise a query to best reflect the very intentions and preferences of the user. A new line of research in this area aims at taking into account the polarity of preferences what should considerably enhance the functionality and usefulness of flexible database querying systems. Bipolar queries constitute an important concept in this area. They are meant here, in general, as queries involving negative and positive information. In a special, promising interpretation they can be viewed in terms of necessary and possible conditions. The purpose of this paper is to critically analyze, recast in a unified perspective and clarify with respect to conceptual, algorithmic and implementation related aspects of various ways to deal with bipolarity. This should open new perspectives for research and commercial applications of bipolar and related queries which should provide more comprehensive, enhanced and more human consistent querying capabilities.

1 Introduction

A crucial problem in database querying is how to formulate, and then represent and process a query to best reflect the very intentions and preferences of the user. Traditionally, databases are meant to store highly structured information and to support

---

To Patrick, Professor Patrick Bosc, a friend and peer, who has been for a long time stimulating and amplifying our interest in flexible database querying.

S. Zadrożny (✉) · J. Kacprzyk
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
e-mail: Slawomir.Zadrozny@ibspan.waw.pl

J. Kacprzyk
e-mail: Janusz.Kacprzyk@ibspan.waw.pl

S. Zadrożny
Kazimierz Pulaski University of Technologies and Humanities in Radom, Radom, Poland

O. Pivert and S. Zadrożny (eds.), *Flexible Approaches in Data, Information and Knowledge Management*, Studies in Computational Intelligence 497,
DOI: 10.1007/978-3-319-00954-4_4, © Springer International Publishing Switzerland 2014

equally highly structured and precise query languages. However, nowadays databases find their application in many various settings and are expected to provide information to growing population of end users without a relevant IT/ICT training. Moreover, a growing complexity of application domains requires some more sophisticated forms of queries to make it possible to reflect the real users' intentions, preferences and interests, the semantics of which is not obvious and straightforward. Advanced graphical user interfaces (GUIs) alleviate to some extent the problem of man-machine communication, but, alone, do not help much with respect to the semantic representation of the users' information needs. Thus, some more conceptually advanced and sophisticated information access methods are needed to make a full use of the potential brought by the vast amount of information gathered in modern databases. First of all, a query is usually first conceived in natural language—the primary, maybe the only fully natural way of articulation, communication and "information processing" for the human being. Then, it has to be translated to a form required by a given database management system (DBMS). This translation is often lossy and thus those users are in need of another access path. This need is addressed by the traditional research on query answering systems which ultimately aims at providing the users with a fully natural language based query interface. An important issue is the very modeling of linguistic terms which may be used in queries and here many interesting approaches have been developed by the fuzzy logic community.

An important novel line of research concerning advanced querying of databases addresses the issue of the bipolar nature of conditions describing data sought by the user. Namely, the user looking for data usually can specify some disqualifying (negative information) and some desired (positive information) features of data. Classical query formalisms do not allow to express such requirements. The problem becomes particularly complex when these features are specified in an imprecise way. There is quite a rich literature dealing with this problem in the framework of fuzzy logic in general, or for some specific applications, notably decision making. A few other chapters in this volume also belong to this direction.

Various existing approaches to the representation and processing of such bipolar queries are based on different assumptions, paradigms and formal tools. As always in such a case, there is an acute need for some deeper analysis of views and perspectives within which various authors deal with the problem so that crucial differences and similarities be discovered. Such a comprehensive study of many various possible interpretations of bipolarity related to database queries seems to be missing. This is the purpose of this chapter in which we first provide a quick review of known approaches and interpretations and then propose our own contribution to the understanding of this phenomenon with special emphasis on the discussion of various scales of bipolarity which play a particular role for our purpose.

2 Background

2.1 Basic Concepts

The starting point for our considerations is the seminal Zadeh's concept of a *fuzzy set* [44] which may be conveniently identified with its *membership function*. Namely, a fuzzy set A in a universe U will be in what follows usually identified with its membership function:

$$\mu_A : U \rightarrow [0, 1]$$

such that $\mu_A(x)$ denotes degree to which element $x \in U$ is a member of the fuzzy set A.

We will skip most of the basic concepts related to fuzzy sets as they are clearly superfluous in this volume. We will just remind briefly a few concepts which will be important for the further discussion. We refer an interested reader to a vast literature, notably to the recent paper by Dubois and Prade [26] who provide a perspective on the notions of fuzziness, uncertainty and bipolarity which are all very important in the context of data modeling and database querying.

The *support* and the *core* of a fuzzy set A in universe U, denoted $Support(A)$ and $Core(A)$ respectively, are "classical" (crisp) sets defined as follows:

$$x \in Support(A) \Leftrightarrow \mu_A(x) > 0 \tag{1}$$

$$x \in Core(A) \Leftrightarrow \mu_A(x) = 1 \tag{2}$$

The concept of the *twofold fuzzy set* is an extension of the concept of the regular fuzzy set [21]. A twofold fuzzy set A over a given universe of discourse U is defined by two membership functions $\pi_A : U \rightarrow [0, 1]$ and $\eta_A : U \rightarrow [0, 1]$ such that:

$$\eta_A(x) > 0 \Rightarrow \pi_A(x) = 1 \qquad \forall x \in U \tag{3}$$

Intuition behind the condition (3) is such that $\pi_A(x)$ may denote the degree to which it is possible that x belongs to A, while $\eta_A(x)$ may denote the degree to which it is necessary that x belongs to A, and then (3) is a natural consequence of the essence of possibility theory. This condition may also be expressed in the following way. Assuming that both membership functions specify regular fuzzy sets, it is required that the support of the fuzzy set defined by η_A must be contained in the core of the fuzzy set defined by π_A.

Another extension of the concept of the regular fuzzy set is *Atanassov's intuitionistic fuzzy set* (AIFS, for short) [1]; for a debate about the appropriateness of the term "intuitionistic", cf. Dubois et al. [20]. An AIFS A over a given universe of discourse U is defined by two functions: a membership function $\mu_A : U \rightarrow [0, 1]$ and a non-membership function $\nu_A : U \rightarrow [0, 1]$ such that:

$$\mu_A(x) + \nu_A(x) \leq 1 \qquad \forall x \in U \tag{4}$$

Thus in this approach the membership and the non-membership of an element to an AIFS may be determined to some extent independently. The consistency condition (4) finds an interesting interpretation in the context of the bipolar queries against the database, discussed in this paper, cf. [23].

2.2 Classical and Flexible Queries: A Brief Overview

We adopt the basic terminology of the *relational data model*. In particular, we will mainly refer to a single *relation* (or, more precisely, relational variable), comprising a set of tuples $T = \{t_i\}$, characterized by a set of attributes $At = \{X, Y, \ldots\}$.

In this chapter we focus on the conditions in a query which specify which data is sought. Looking from the perspective of the SQL language, we are concerned with the `WHERE` clause of a simple `SELECT-FROM-WHERE` query. Some aspects discussed here may be further extended to the case of more complex SQL queries, e.g., involving the join operator though this goes beyond the scope of the current paper. Our study concerns the following scenario. A classical ("crisp") relational database is considered, against which queries of the `SELECT-FROM-WHERE` type are addressed. However, these queries may contain some non-standard conditions in their `WHERE` clause. First, some imprecision (fuzziness) may be present, exemplified by a query "Find all middle-aged employees", which is meant to express in a better and more direct way the user requirements than, e.g., a query "Find all employees whose age falls in the interval between 35 and 45". We follow here the line of research of Bosc and Pivert [3, 4, 6–8, 10, 12–14, 16, 17] and Kacprzyk et al. [30–35, 46, 47, 51] in which it is assumed that such imprecise conditions are construed using some linguistic terms as, e.g., "middle-aged" in the previous example, which are modeled using fuzzy logic. Second, we assume that the condition may be composed of some positive and negative components, what is meant to reflect bipolarity of user preferences. A study of the latter feature is the main topic of this chapter.

The simplest form of bipolarity may be ascribed to any classical "crisp" query against a database. Let us consider a simple query involving just one atomic condition concerning one attribute. Let this condition be:

$$\texttt{price} \leq \texttt{USD 500K}, \tag{5}$$

as in Example 1 (cf. p. 7). Then, the values in the domain of the attribute `price` which are above 500 K are rejected (i.e., treated as "negative"), while the values lower or equal than that are accepted ("positive"). This is however a very specific type of bipolarity which is of a lesser interest, at least due to the two following reasons:

1. the rejection/acceptance is binary
2. there is no notion of "neutrality", and, what is closely related, an element of the domain under consideration which is not "negative" is necessarily "positive" and vice versa.

The classical fuzzy approach to querying obviously alleviates the first limitation but not the second: still, if an element x of the domain is indicated as "positive" to a degree $\mu(x)$, then it is automatically treated as "negative" to the degree $1-\mu(x)$. This type of bipolarity may be referred to, after Dubois and Prade [24, 26], a *symmetric univariate bipolarity*. There is still another feature of the bipolarity which is not properly addressed by the classical, either crisp or fuzzy, approaches to the querying. Namely, there should be available for the user some specific aggregation operators, which take into account the bipolarity of preferences while computing the overall matching degree. In order to be in accord with psychological observations, such operators should make it possible to treat negative and positive preferences in a different way [29].

3 Unipolar and Bipolar Fuzzy Conditions

3.1 Classical Fuzzy Approach to the Modeling of Query Conditions and Bipolarity

A fuzzy logic based perspective has been adopted to model conditions of a query against a database since the early days; cf., e.g., Tahani [40]. The flexibility of modeling provided by the concept of a fuzzy set finds an immediate application for the purposes of query conditions specification. Here we will only very briefly summarize the advantages of the classical fuzzy logic based approach so as to clearly show later the difference that taking into account of bipolarity makes.

Let us consider a database of real-estate properties offer for sale by a real estate agency. Let the particular houses be characterized by some attributes exemplified by: `price`, `location`, and `size` (in square meters). Let us further assume that a customer of the agency is looking for a house of a *low* price (usually a customer will require a few conditions that should be met by the house he or she is looking for, but here for the sake of presentation clarity we will focus on one attribute). Using, e.g., a flexible querying interface provided by Kacprzyk and Zadrożny's FQUERY for Access [32, 33], he or she can form a query using the *linguistic term* "low" directly to express the constraint on the price. The set of acceptable prices will be modeled by a fuzzy set A in a universe U, characterized by its membership function:

$$\mu_A : U \to [0, 1] \tag{6}$$

In our example, the term "low" will be modeled by such a fuzzy set A (in the universe U which is identified then with the domain of given attribute; here: `price`) that $\mu_A(x)$ denotes to which degree a given price x is low, and this degree will be treated as the matching degree of a house with the given price against the query under consideration. Thus the user is released from artificially distinguishing the prices which (fully) are low from those which are not. This way of modeling is clearly more human consistent, i.e., more in line with human perception of such linguistic terms as "low", "moderate", "high", etc. A direct consequence is then the possibility to order the tuples of a database (real estate properties, in the case of our example) according to their matching degrees of the query condition.

We will denote a classical fuzzy query concerning attribute X and using a linguistic term modeled by a fuzzy set A as:

$$X \text{ is } A \tag{7}$$

Referring to our previous example, X in (7) denotes the attribute `price`, while fuzzy set A represents the linguistic term "low".

From the point of view of this chapter the interpretation of the membership degrees related to (6) is the most interesting. It is worthwhile to note that in the classical fuzzy approach a unipolar scale is tacitly associated with (6). Namely, $\mu_A(x)$ denotes the degree to which a given attribute value is compatible with the meaning of a given linguistic term and, in consequence, the degree to which this value satisfies query condition. There is no explicit distinction between "negative" ("rejected", "bad") and "positive" ("accepted", "good") values. It may be argued that such a distinction is usually made by the user in the framework of the classical fuzzy approach but it is of a slightly different nature than the one considered here. Namely, usually the query languages and interfaces proposed (c.f., e.g., [14, 33]) presume the use of a threshold in a query which indicates that only tuples matching the query to a degree higher than this threshold are shown to the user. Thus, in a sense, the mechanism related to such a threshold makes the distinction between "negative" and "positive" tuples which may be further interpreted as the distinction concerning the attribute values—if the query condition is atomic and refers to just one attribute (there are also approaches in which such a threshold may be associated with each atomic condition separately [39]). However, this is a binary distinction and is made externally with respect to the query condition, and thus of a lesser interest to us here.

Following the arguments mentioned in the introduction we assume that very often the user preferences are inherently bipolar. This bipolarity may manifest itself *at the level of each attribute domain* or *at the level of the comprehensive evaluation* [29] of the whole tuple. In the former case, the user may see particular elements of the domain as "negative", "positive" or "neutral", to a degree. This classification should, of course, influence the matching degree of a tuple having a particular element of the domain as the value of the attribute under consideration. In the latter case the user is expected to express some conditions, involving possibly many attributes, which when satisfied by a tuple (to a degree) make it "negative" or "positive" (to a degree). Thus, effectively, in the latter case some combinations of multiple attributes values are seen as “negative”/“positive”. The former case may be seen as a special case of the latter, when both the “negative” and “positive” conditions concern the same attribute and thus demarcate the “negative” and “positive” values of a given attribute. However, the distinguishing of the former case is worthwhile as it is somehow less intuitive but still practically useful and there do exist some formal means which provide for its elegant formal representation which will be discussed later on. Moreover, the distinguishing of this case makes it easier to study various approaches known in the literature. Let us illustrate that on two examples.

Example 1 Let us consider a customer of our real-estate property agency. He or she may have the following view on the domain of the `price` attribute:

(a) the price above USD 500 K is definitely negative,
(b) the price below USD 300 K is definitely positive,
(c) the remaining prices are neither negative nor positive, i.e., are neutral.

In Example 1 the bipolarity is defined in the crisp way on the level of the attribute domain. Now let us consider more complex preferences of the user.

Example 2 Let us consider another customer of our real-estate property agency. He or she finds:

(a) the properties more expensive than USD 500 K and, at the same time, of the size less than 100 sq. m. as definitely negative,
(b) the properties located in Waterfront as definitely positive,
(c) the remaining properties as neither negative nor positive, i.e., as neutral.

In Example 2 the bipolarity concerns a combination of attribute values, or, equivalently the whole tuples (here: the real-estate properties) possessing these combinations of values. In this example (and also in Example 1, which is however possibly less obvious) the point (b) requires some discussion concerning the compatibility of condition given there with the condition given in the point (a)—such a discussion will be provided later on while presenting alternative ways to formally represent the bipolarity in query conditions.

Both Examples 1 and 2 are crisp, however may be easily “fuzzified” in practical scenarios by using such (subjectively defined) linguistic terms as “very expensive”, “rather cheap”, “small” instead of the numbers expressing the price and size of the property. What is worth noting is that the user preferences expressed both in the crisp and fuzzy versions of these examples cannot be properly expressed using the classical crisp or fuzzy approaches to database querying. On the other hand, one may argue that some special cases of such preferences are representable using the classical approaches. Namely, while considering a modification of the first example, one can claim that the condition “price <USD 500 K” properly represents such bipolar preferences if the values of the price higher than USD 500 K are “negative” and the values below or equal to the same threshold are “positive”. More generally, in the specific case where the subset of “positive” values in the domain of given attribute is the complement of the subset of “negative” values, it may be seen as corresponding

to the bipolarity defined by using the symmetric bipolar univariate scale (called a symmetric univariate bipolarity by Dubois and Prade [25]).

Example 1 clearly explains the terminology often used in the literature (cf., e.g., [23, 48]) when referring to the negative and positive parts of the scale: the former is used as the scale for the required conditions and the latter as the scale for the desired conditions.

Thus, the classical fuzzy approach makes it possible for the user to clearly specify in the query a distinction between the "negative" data he or she rejects (to a degree) and the "positive" data he or she accepts (to a degree). It is worth to emphasize this concept of the negative/positive traits of data the user has in mind, which is, of course, relative to a given query, and which is accompanied with some affect. This distinction between the negative and positive traits of data should then be properly taken into account during the computation of the overall matching degree using appropriate aggregation operators—we will discuss this in more detail later on. Without this distinction one can argue that already in the traditional, "crisp" approaches a query defines the set of rejected and accepted data.

3.2 Bipolarity: Which Scale to Use

3.2.1 Bipolarity in the Query Condition via a Univariate Bipolar Scale

In this case it is assumed that for the data under consideration, being either an attribute domain element or the whole tuple, the user may evaluate its "negative" and "positive" sides and he or she is in a position to combine these evaluations and expresses an overall evaluation on one univariate bipolar scale. It may be instructive to consider two cases, depending on the level at which this bipolarity is expressed.

Univariate bipolarity at the level of an attribute domain element

Here we assume that the user has a bipolar evaluation of each element of a domain dom_X of a given attribute X. For convenience, we assume that such an evaluation is of the form (cf., e.g., [29] for a discussion):

$$\xi_X : dom_X \rightarrow [-1, 1] \tag{8}$$

and for $x \in dom_X$ the value $\xi_X(x) > 0$ denotes x's degree of "positiveness", $\xi_X(x) < 0$ denotes its degree of "negativeness" and $\xi_X(x) = 0$ means that x is neutral from the point of view of the user, concerning given query.

In such a case, the user preferences may be properly modeled using a twofold fuzzy set (3) in dom_X. Thus in (7) fuzzy set A will be now replaced by a twofold fuzzy set. This twofold fuzzy set will be interpreted as follows:

- the membership function π_A is used to represent the negative evaluations of the elements of dom_X; its values equal the evaluation for these elements plus 1:

$$\pi_A(x) = \min(1 + \xi_X(x), 1) \tag{9}$$

  More precisely, the values of $\pi_A(x)$ form a reversed negative scale: value 1 denotes no negative evaluation, value 0 the strongest negative evaluation, while intermediate values represent some degrees of negative evaluation—the closer to 0 they are the stronger negative evaluation it is.
- the membership function η_A is used to represent the positive evaluations of the elements of dom_X:

$$\eta_A(x) = \max(\xi_X(x), 0) \tag{10}$$

  i.e., the value 0 of $\eta_A(x)$ denotes no positive evaluation, the value 1 the strongest positive evaluation, while intermediate values represent some degrees of positive evaluation—the closer to 1 they are the stronger positive evaluation it is.

The idea is illustrated in Fig. 1. Thanks to the very property of the twofold fuzzy set (3) there is a one-to-one mapping between the degrees of evaluation ξ_X given by (8) and the membership functions of the corresponding twofold fuzzy set. The mapping from $\xi_X(x)$ to a pair $(\pi_A(x), \eta_A(x))$ is given by Eqs. (9)–(10). The reverse transformation is given by the following formula:

$$(\pi_A(x), \eta_A(x)) \rightarrow \xi_X(x) \tag{11}$$

$$\xi_X(x) = \begin{cases} \eta_A(x) & \text{for } \pi_A(x) = 1 \\ \pi_A(x) - 1 & \text{otherwise} \end{cases} \tag{12}$$

In this scenario the user is assumed to express his or her bipolar preferences with respect to an attribute X using a univariate bipolar scale. In order to do so one can choose two linguistic terms from the dictionary (or define them; cf., e.g., details of the user interface of the FQUERY for Access system [32, 33]) which are represented by fuzzy sets forming together a twofold fuzzy sets, i.e., whose membership functions satisfy condition (3). An illustration is shown in Example 3.

Example 3 Let us consider a customer who does not like small houses and would be most satisfied with a house of the size around 350 sq. m. Then he or she may express his or her preferences defining or choosing from the dictionary two linguistic terms "small" and "around 350 sq. m." and form a twofold fuzzy set A with the following membership functions $(\pi_A(x), \eta_A(x))$:

$$\pi_A(x) = \mu_{\text{“not small”}}(x)$$
$$\eta_A(x) = \mu_{\text{“around 350 sq. m.”}}(x)$$

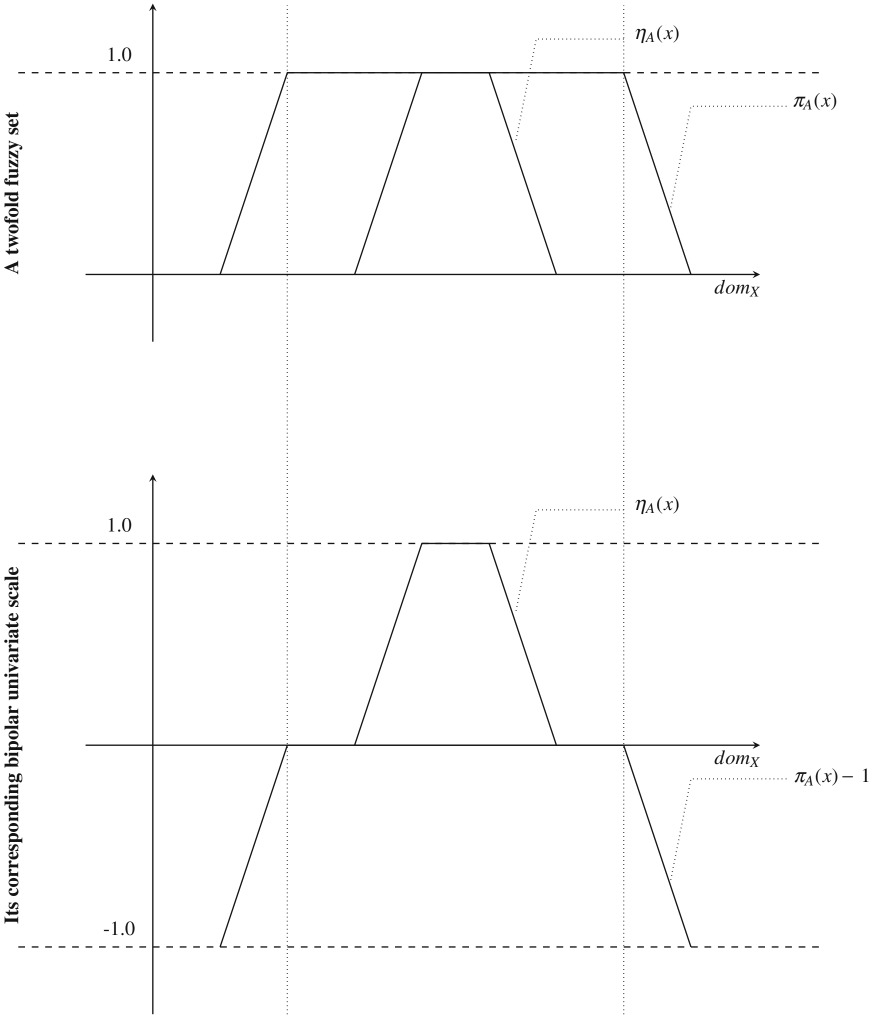

Fig. 1 Illustration of the bipolar univariate scale representation using the twofold fuzzy set

assuming that the support of the fuzzy set representing "around 350 sq. m." is a subset of the core of the complement of the fuzzy set representing the linguistic term "small", i.e., the fuzzy set representing the linguistic term "not small".

Note, that in this case it may be fairly easily checked that the membership functions $\pi_A(x)$ and $\eta_A(x)$ really form a twofold fuzzy set, i.e., satisfy the condition (3). Thus, it is reasonable to assume that the user, properly supported by the user interface, picks up an appropriate pair of fuzzy sets.

Univariate bipolarity at the level of a tuple (at the comprehensive evaluation level)

This is a more general case than the previous one as now it is assumed that the user has a comprehensive evaluation of the whole tuple expressed using a univariate bipolar scale. Thus, as previously, we assume an evaluation ξ_T ranging over the interval $[-1, 1]$ but this time its domain is the set of tuples T:

$$\xi_T : T \rightarrow [-1, 1] \tag{13}$$

Again, a formal representation of this evaluation is obtained using a twofold fuzzy set denoted by a pair of membership functions $(\pi_T(t), \eta_T(t))$. Now we will assume that the user defines two conditions denoted $C(t)$ and $P(t)$, respectively, that will in turn define these membership functions, i.e.,

$$\pi_T(t) = C(t) \tag{14}$$

$$\eta_T(t) = P(t) \tag{15}$$

Here, and in what follows, we will denote by C and P both fuzzy predicates identified by the respective conditions and the fuzzy sets of tuples satisfying, to a degree, these predicates. Moreover, by $C(t)$ and $P(t)$ we will denote the membership function values of the particular tuples $t \in T$ to these fuzzy sets. Let us illustrate that with a "fuzzified" version of Example 2, which is given below as Example 4.

In this case the link between (13), and (14) and (15) is analogous as in the case of univariate bipolarity at the level of an attribute domain elements as discussed earlier.

Example 4 Let us consider a customer of our real-estate property agency. He or she finds:

(a) very expensive and, at the same time, small properties as definitely negative,
(b) properties located in eastern districts of the city as definitely positive.

In Example 4 the fuzzy predicates C and P are defined as "not (*very expensive and small*)", and "located in *eastern* districts", respectively (we assume that "eastern districts" is a gradual notion, well represented by a fuzzy predicate). Note that in the case of a comprehensive evaluation it is rather unreasonable to expect that the respective membership functions satisfy condition (3), or—to put that more precisely—that the user may be somehow aware if they do or do not. For example, there may exist a property located in the "totally eastern" (i.e., to the degree 1) district but very expensive and small. Thus, in fact, (15) has to be modified so as to force the satisfaction of (3). The simplest way to do that is to use the following variant of (15):

$$\eta_T(t) = \begin{cases} P(t) & \text{for } C(t) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

3.2.2 Bipolarity in the Query Condition via a Bivariate Bipolar Scale

In this approach the user is assumed to define separately positive and negative traits of the data sought. This may be done again, as in the previous case, at the level of an attribute domain element or at the level of a tuple. However, here still another distinction should be made regarding the semantics of these separate positive and negative evaluations. Namely, we will distinguish two cases in which:

1. the negative evaluation is treated as related to the violation (to a degree) of a constraint and the positive evaluation is treated as related to the satisfaction of a desire, i.e., of a somehow supplementary condition; thus the positive evaluation plays here a subsidiary role—the elements violating the constraint are thus treated as rejected (to a degree); in what follows we will refer to this case as the "required/desired semantics",
2. both evaluations are treated "equally".

Thus, in the first case a specific semantics of "positive" and "negative" evaluations is assumed. This implies a need for some consistency conditions which express the fact that something may be desired at most to a degree to which it satisfies the constraints, i.e., to a degree to which it is not rejected. We will discuss that issue in Sect. 4 in a more detailed way. It should be noted that this semantics is adopted in most of the works related to bipolar queries; cf., e.g. [23, 48]. It is definitely very intuitive and of a high practical value. However, second case, in which the treatment of bipolarity is more general in the sense of just reflecting the existence of a positive and negative condition without any specific interpretation of their relations and interplay, deserves more attention and research.

Bivariate bipolarity at the level of an attribute domain element—the general case

Here it is assumed that the user has a bipolar evaluation of each element of a given attribute X domain dom_X and can separately evaluate its positive and negative traits (his "liking" and "disliking" of an element). For convenience, we assume that such an evaluation is expressed via two functions and such that

$$\xi_X^+ : dom_X \rightarrow [0, 1] \tag{17}$$

$$\xi_X^- : dom_X \rightarrow [0, 1] \tag{18}$$

where ξ_X^+ and ξ_X^- denote how "good" and "bad", respectively, the element x is. Let us illustrate this case with the following example.

Example 5 Let us consider a customer, who especially cares for the location of a house which is given in the database by the name of the district of a city. For each district he can list some "pros" and "cons" (possibly, of varying importance/strength). For example, district D is well communicated with the rest of the city but is known for its relatively high crime level. The user is able to separately aggregate the lists of the arguments and to come up with a separate positive and negative evaluation of each location.

It should be noted that such a bipolar bivariate evaluation makes sense basically only in the case when there is a set of criteria that may be related to the elements of the domain in question but are not directly represented in a database. If the locations of the houses were represented in the database of Example 5 at a more detailed level (including the communication convenience, crime level etc.), then the preferences of the user would probably be better expressed using the bipolar univariate or even just unipolar scale, with respect to the domains of attributes comprising this more detailed representation.

Bivariate bipolarity at the level of an attribute domain element—the required/desired semantics

Here it is assumed that the user has a bipolar evaluation of each element of a given attribute X domain dom_X in the sense that he or she can distinguish a (fuzzy) set R of rejected elements and the (fuzzy) set P of really desired (preferred) elements. For convenience, such an evaluation may be expressed by two (membership) functions:

$$\xi_X^R : dom_X \rightarrow [0, 1] \tag{19}$$

$$\xi_X^P : dom_X \rightarrow [0, 1] \tag{20}$$

and for $x \in dom_X$ the value $\xi_X^R(x)$ denotes degree to which x is rejected, while $\xi_X^P(x)$ denotes degree to which it is desired. It should be noticed that if the sets R and P are complements of each other then only one of them have to be specified, i.e., it refers to the case of the classical fuzzy logic based querying: ξ_X^P may be identified with (6). Thus, this case is interesting only if $C = \bar{R} \neq P$ and such a bipolar query may be represented equivalently by the pairs of (fuzzy) sets (R, P) or (C, P).

Due to the postulated required/desired semantics it is rational to impose a consistency condition $P \subseteq C$, which states that an element has first to be non-rejected before it can be desired (preferred). Thus, the user preferences may be here properly modeled using an AIFS (4) in dom_X. Thus in (7) fuzzy set A will be now replaced by an AIFS which will be interpreted as follows:

- the membership function μ_A is used to define the degree to which a particular element is desired:

$$\mu_A(x) = \xi_X^P(x)$$

- the non-membership function ν_A is used to define the degree to which a particular element is rejected:

$$\nu_A(x) = \xi_X^R(x)$$

The consistency condition $P \subseteq C$, which may be expressed equivalently as $P \subseteq \bar{R}$ or $\xi_X^P(x) \leq 1 - \xi_X^R(x)$ coincides with the condition (4) characteristic for the AIFS's.

Thus, in this scenario the user is assumed to express his or her bipolar preferences with respect to an attribute X using a bivariate bipolar scale. In order to do that one can choose two linguistic terms from the dictionary, representing sets R and P, respectively, the membership functions of which have to satisfy condition (4). An illustration is shown in Example 6 which is a modified version of Example 3.

Example 6 Let us consider a customer who does not like small houses and would be most satisfied with a house of the size around 350 sq. m. Then, he or she may expresses his or her preferences by defining or choosing from the dictionary two linguistic terms "small" and "around 350 sq. m." and by forming the following AIFS A $(\mu_A(x), \nu_A(x))$:

$$\nu_A(x) = \mu_{\text{"small"}}(x)$$
$$\mu_A(x) = \mu_{\text{"around 350 sq. m."}}(x)$$

assuming that for all values representing the size the sum of its membership degrees to these two fuzzy sets is not larger than 1.0.

Note, that in this case the preferences of the user may be expected to be consistent and, if the two above mentioned fuzzy sets adequately represent his or her subjective understanding of the linguistic terms "small" and "around 350 sq. m", then the query formed in such a way will represent the user's preferences in a fair way.

4 Semantics of the Bipolar Bivariate Conditions: An Aggregation Perspective

In the previous sections we were mainly concerned with the identification of various forms of bipolarity in queries and their representations. Here we discuss an interpretation of bipolar queries in terms of an ordering of tuples they imply. In particular, we focus on a specific semantics of bipolar conditions, referred to as "required/desired" semantics in the previous section. Here we will discuss this semantics in a more detailed way, in a specific perspective.

4.1 A General View

The most general interpretation of the bivariate bipolarity in queries is the one mentioned in Sect. 3.2.2—with the positive and negative conditions treated as equally important and independent. Thus, we have two conditions and each tuple is evaluated against them yielding a pair of satisfaction (matching) degrees. The natural question is then how to order data in an answer to such a query.

Basically, while doing that we should take into account the very nature of both matching degrees, i.e., the fact that they correspond to the positive and negative conditions. The situation here may be compared to that of decision making under risk. Namely, in the latter context a decision maker who is risk-averse may not accept actions leading with some non-zero probability to a loss. On the other hand, a risk-prone decision maker may ignore risk of an even serious loss as long as there are prospects for a high gain. Similar considerations apply in the case of bipolar queries. Some users may be more concerned about negative aspects and will reject a piece of data with a non-zero matching degree of the negative condition. Some other users may be more oriented towards the satisfaction of the positive conditions and may be ready to accept the fact that a given piece of data satisfies to some extent the negative conditions.

The conclusion from the above considerations is such that the bipolar query meant in such a general sense should be evaluated in a database in a way strongly dependent on the specific attitude of the user. In the extreme cases, the above-mentioned analogs of "risk-averse" and "risk-prone" attitudes would be represented by lexicographic orders. In the former case the lexicographic ordering would be first non-decreasing with respect to the negative condition matching degree and then non-increasing with respect to the positive condition matching degree. The less extreme attitudes of the users may be represented by various aggregation operators producing a scalar overall matching degree of a bipolar query.

An approach to a comprehensive treatment of such generally meant bipolar queries has been proposed by De Tré and Matthé [38], and further developed in [19, 37]. In this approach a pair of matching degrees of the positive and negative conditions is referred to as a *bipolar satisfaction degree* (BSD). The respective matching degrees are denoted as s and d, and called the *satisfaction degree* and the *dissatisfaction degree*, respectively. The ranking of data retrieved against a bipolar query in this approach may be obtained in various ways. One of the options is based on the difference $s - d$ of the two matching degrees. In this case a "risk-neutral" attitude of the user is modeled: he or she does not favor neither the positive nor the negative evaluation.

The BSDs are assumed to be assigned at the attribute level and then are aggregated so that an overall BSD for the whole query is obtained. In [19, 37] it is proposed how such an aggregation should be carried out in case of the standard logical connectives. See also a paper by Matthé et al. in this volume which reports on further developments in this research direction.

4.2 The Required/Desired Semantics Once Again

The semantics in question supports the following interpretation of the *positive* and *negative* conditions in the bipolar query: the data items sought have to satisfy the complement of the latter conditions unconditionally while the former conditions is of somehow secondary importance. For example, a house the user is looking for may have to be cheap and then among cheap houses those which are closer to a railway station are preferred. The negative condition is here "not being cheap" while the positive condition is "being close to the railway station". Usually, the complement of the negative condition will be specified in such a query (denoted C), which may therefore be interpreted as a *required* condition. On the other hand, the positive condition is expressed directly and may be referred to as a *desired* condition (denoted C). It is worth noting that this interpretation is close to the mode of *aggregation* of a hierarchy of conditions proposed in 1987 in a seminal work of Lacroix and Lavency [36], of course without any reference to the notion of bipolarity at that time. This type of aggregation may be seen as based on the "and possibly" operator: to satisfy the required conditions *and if possible also* the desired conditions. We develop this idea further in the following sections.

Whatever the interpretation of the positive and negative conditions is adopted, the main practical problem is how to order the tuples based on their satisfaction degrees of these conditions. In case of the "required/desired" semantics we denote a pair of these conditions as (C, P). The problem mentioned may be solved in many ways.

The simplest approach is to use the matching degree with respect to the desired condition just to order the data items which satisfy the required condition. This idea leads to the use of the lexicographic order which is promoted by many authors, notably Dubois and Prade; cf., e.g., [23]. This interpretation is in fact predominant in the literature dealing with bipolar queries. The early works of Bosc and Pivert [9, 11] which aim at introducing a fuzzified version of the operator for aggregating the conditions in the spirit of Lacroix and Lavency also belong to this category. In those papers, as well as in the sophisticated possibility theory based interpretations by Dubois and Prade [24, 25] focus is on a proper treatment of *multiple* required and preferred conditions, basically assuming the lexicographic order as the way of combining the required (negative) and desired (positive) conditions, cf. also Bosc et al. [18]. However, if a fine (detailed) scale for the satisfaction of the required condition is adopted then a smallest possible dominance of one tuple over another with respect to the satisfaction of the required condition makes it "better" even if the other tuple is much better with respect to the desired condition. The solution proposed is to use a coarser scale of required condition satisfaction degrees but still it is a rather artificial solution.

Another approach consists in employing an aggregation operator which combines the degrees of matching (satisfaction) of conditions C and P and yields an overall matching degree which is then used to order tuples in the usual way. In particular the operator introduced by Lacroix and Lavency [36] may be used. Then, the whole query may be interpreted as expressing the following condition:

$$C \text{ and possibly } P \tag{21}$$

In the literature such aggregation operators have been studied by many authors under different names, and sometimes in slightly different contexts. However, in the framework of database querying Lacroix and Lavency proposed first such an approach. Zadrożny [45] proposed a direct "fuzzification" of the approach by Lacroix and Lavency, Zadrożny and Kacprzyk [49, 52] studied some properties of that solution.

4.3 The "and possibly" Operator Based Aggregation

The essence of the "and possibly" operator consists in taking into account the whole dataset while combining the matching degrees related to the required and desired conditions. Namely, if there is a piece of data which satisfies both conditions, then and only then it is actually *possible* to satisfy both of them and each piece of data has to meet both of them. Thus, the (C, P) query reduces to the usual conjunction $C \wedge P$. On the other hand, if there is no such a piece of data, then it is *not possible* to satisfy both conditions and the desired one can be disregarded. Thus, the (C, P) query reduces to C. These are however two extreme cases and actually it may be the case that the two conditions may be simultaneously satisfied to some degree. Then, the matching degree of the (C, P) query against a piece of data lies somewhere between its matching degrees of $C \wedge P$ and C. This may be formally written for the crisp case as [36]:

$$C(t) \underline{\text{ and possibly }} P(t) \equiv C(t) \wedge \exists s (C(s) \wedge P(s)) \Rightarrow P(t) \tag{22}$$

and for the fuzzy case as [45, 48]:

$$C(t) \underline{\text{ and possibly }} P(t) \equiv \min\left(C(t), \max(1 - \max_{s \in T} \min(C(s), P(s)), P(t))\right) \tag{23}$$

where T denotes the whole dataset being queried.

The formula (23) is derived from (22) using the classic fuzzy interpretation of the logical connectives via the maximum and minimum operators. In Zadrożny and Kacprzyk [49, 50, 52] we analyze the properties of the counterparts of (23) obtained by using a broader class of operators modeling the logical connectives.

The "and possibly" aggregation operator that is implicit in the Lacroix and Lavency's proposal [36] has been later proposed independently by Dubois and Prade [22] in the context of default reasoning and by Yager [42, 43] in the context of multicriteria decision making for the case of so-called *possibilistically qualified criteria*. Yager [43] intuitively characterizes a possibilistically quantified criterion as such which should be satisfied unless it interferes with the satisfaction of other criteria. This is in fact the essence of the aggregation operator "and possibly" as we understand it here. The concept of this operator was also used by Bordogna and Pasi [2] in the context of textual information retrieval.

Recently, the modeling of the aggregations operators in the spirit of the "and possibly" operator is gaining a broad interest. Usually, they lack the dependence on the whole data set what is a distinguishing characteristic feature of the operator based on the Lacroix and Lavency approach. However, they may have some importance for the implementation of bipolar queries and some of them are proposed to this aim. Dujmović [27] already in 1979 defined an aggregation operator combining two arguments in such a way that one of them controls the influence of the other ones on the result of their combination. Bosc and Pivert [15] also consider similar operators. Tudorie [41] introduced the "among" operator which is similar to the "and possibly" operator and is used to form queries such as "find data satisfying a condition P among those satisfying a condition C". The evaluation of a query with the "among" operator is expressed in terms of the rescaling of fuzzy predicates used to specify condition P.

5 Concluding Remarks

The idea of taking into account bipolarity of user preferences expressed in the form of database queries is gaining a growing popularity. However, there are still some basic questions open. This paper is an attempt to describe the very essence of bipolarity in the considered context, in a slightly more general way by concentrating on the presentation of various possible views and perspectives, and then attempting to find a unifying view. We also briefly review relevant literature to support our line of reasoning and views, and to show a line of logical developments which have occurred in the research efforts related to bipolar queries. In particular, we distinguish various possible approaches depending on the following aspects:

1. the type of a bipolar scale used to express preferences,
2. the existence (and type of) or lack of consistency constraints imposed on the positive and negative preferences, and
3. the level of data at which these evaluations are given.

We hope that this provides a better perspective on the research on bipolar queries. In particular, it shows that the approaches currently predominant in the literature cover only a part of the spectrum of possible interpretations.

The concepts and relations developed have been illustrated by numerous partial examples. However, due to space limitation, it has been impossible to present an in-depth analysis of one of many applications of the method proposed, notably in the area of querying real estate databases. Basically, due to the very essence of this domain and a relevance of interaction with the human customer, the presentation to be meaningful would have required a detailed coverage of many aspects exemplified by dictionaries of terms, analyses of preferences, multicriteria choice processes, etc.

Acknowledgments This contribution is partially supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from The European Union within the Innovative Economy Operational Programme (2007–2013) and European Regional Development Fund.

References

1. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**, 87–96 (1986)
2. Bordogna, G., Pasi, G.: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. Int. J. Intel. Syst. **10**(2), 233–248 (1995)
3. Bosc, P., Galibourg, M.: Flexible selection among objects: A framework based on fuzzy sets. In: Proceedings of the SIGIR Conference, Grenoble, France (1988)
4. Bosc, P., Galibourg, M., Hamon, G.: Fuzzy querying with SQL: extensions and implementation aspects. Fuzzy Sets Syst. **28**, 333–349 (1988)
5. Bosc, P., Kacprzyk, J. (eds.): Fuzziness in Database Management Systems. Physica, Heidelberg (1995)
6. Bosc, P., Lietard, L.: Quantified statements and some interpretations for the OWA operators. In: Yager, R., Kacprzyk, J. (eds.) The Ordered Weighted Averaging Operators: Theory and Applications, pp. 241–257. Kluwer, Boston (1997)
7. Bosc, P., Lietard, L., Pivert, O.: Quantified statements and database fuzzy querying. In: Bosc, Kacprzyk [5], pp. 275–308
8. Bosc, P., Lietard, L., Prade, H.: An ordinal approach to the processing of fuzzy queries with flexible quantifiers. In: Hunter, A., Parsons, S. (eds.) Applications of Uncertainty Formalisms, LNAI, vol. 1455, pp. 58–75. Springer, Berlin (1998)
9. Bosc, P., Pivert, O.: Discriminated answers and databases: fuzzy sets as a unifying expression means. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 745–752, San Diego, USA (1992)
10. Bosc, P., Pivert, O.: Fuzzy querying in conventional databases. In: Zadeh, L., Kacprzyk, J.(eds.) Fuzzy Logic for the Management of Uncertainty, pp. 645–671. Wiley, New York (1992)
11. Bosc, P., Pivert, O.: An approach for a hierarchical aggregation of fuzzy predicates. In: Proceedings of the Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93), pp. 1231–1236, San Francisco, USA (1993)
12. Bosc, P., Pivert, O.: Flexible queries, discriminated answers and fuzzy sets. In: Proceedings of the Fifth IFSA World Congress, pp. 525–528. Seoul, Korea (1993).
13. Bosc, P., Pivert, O.: On the evaluation of simple fuzzy relational queries: principles and measures. In: Lowen, R., Roubens, M. (eds.) Fuzzy Logic: State of the Art, pp. 355–364. Kluwer Academic Publishers, Boston (1993)
14. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. IEEE Trans. Fuzzy Syst. **3**(1), 1–17 (1995)
15. Bosc, P., Pivert, O.: On three fuzzy connectives for flexible data retrieval and their axiomatization. In: Chu, W.C., Wong, W.E., Palakal, M.J., Hung, C.C. (eds.) SAC, pp. 1114–1118. ACM (2011)
16. Bosc, P., Pivert, O., Farquhar, K.: Integrating fuzzy queries into an existing database management system: an example. Int. J. Intel. Syst. **9**, 475–492 (1994)
17. Bosc, P., Pivert, O., Lietard, L.: Aggregate operators in database flexible querying. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2001), pp. 1231–1234, Melbourne, Australia (2001)

18. Bosc, P., Pivert, O., Mokhtari, A., Lietard, L.: Extending relational algebra to handle bipolarity. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) SAC, pp. 1718–1722. ACM (2010)
19. De Tré, G., Zadrożny, S., Matthe, T., Kacprzyk, J., Bronselaer, A.: Dealing with positive and negative query criteria in fuzzy database querying. Lect. Notes Comput. Sci. **5822**, 593–604 (2009)
20. Dubois, D., Gottwald, S., Hájek, P., Kacprzyk, J., Prade, H.: Terminological difficulties in fuzzy set theory—the case of "intuitionistic fuzzy sets". Fuzzy Sets Syst. **156**(3), 485–491 (2005)
21. Dubois, D., Prade, H.: Twofold fuzzy sets and rough sets—some issues in knowledge representation. Fuzzy Sets Syst. **23**, 3–18 (1987)
22. Dubois, D., Prade, H.: Default reasoning and possibility theory. Artif. Intel. **35**(2), 243–257 (1988)
23. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002, LNAI, vol. 2522, pp. 174–182. Springer, Berlin (2002)
24. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: Galindo [28], pp. 97–114
25. Dubois, D., Prade, H.: An overview of the asymmetric bipolar representation of positive and negative information in possibility theory. Fuzzy Sets Syst. **160**(10), 1355–1366 (2009)
26. Dubois, D., Prade, H.: Gradualness, uncertainty and bipolarity: Making sense of fuzzy sets. Fuzzy Sets Syst. **192**, 3–24 (2012)
27. Dujmović, J.: Partial absorption function. J. teh Univ. Belgrade EE Dept. **659**, 156–163 (1979)
28. Galindo, J. (ed.): Handbook of Research on Fuzzy Information Processing in Databases. Information Science Reference, New York (2008)
29. Grabisch, M., Greco, S., Pirlot, M.: Bipolar and bivariate models in multicriteria decision analysis: descriptive and constructive approaches. Int. J. Intel. Syst. **23**, 930–969 (2008)
30. Kacprzyk, J.: Fuzzy logic in DBMSs and querying. In: Proceedings of Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, pp. 106–109. IEEE Computer Society Press, Los Alamitos (1995)
31. Kacprzyk, J., Zadrożny, S.: Fuzzy querying for Microsoft Access. In: Proceedings of the Third IEEE Conference on Fuzzy Systems (FUZZ-IEEE'94), vol. 1, pp. 167–171, Orlando, USA (1994)
32. Kacprzyk, J., Zadrożny, S.: FQUERY for access: fuzzy querying for a windows-based DBMS. In: Bosc and Kacprzyk [5], pp. 415–433
33. Kacprzyk, J., Zadrożny, S.: Computing with words in intelligent database querying: standalone and internet-based applications. Inf. Sci. **134**(1–4), 71–109 (2001)
34. Kacprzyk, J., Zadrożny, S.: SQLf and FQUERY for access. In: Proceedings of the Conference IFSA/NAFIPS 2001, pp. 2464–2469, Vancouver, Canada (2001)
35. Kacprzyk, J., Zadrożny, S., Ziółkowski, A.: FQUERY III+: a "human consistent" database querying system based on fuzzy logic with linguistic quantifiers. Inf. Syst. **14**(6), 443–453 (1989)
36. Lacroix, M., Lavency, P.: Preferences: putting more knowledge into queries. In: Proceedings of the 13 International Conference on Very Large Databases, pp. 217–225, Brighton, UK (1987)
37. Matthé, T., De Tré, G., Zadrożny, S., Kacprzyk, J., Bronselaer, A.: Bipolar database querying using bipolar satisfaction degrees. Int. J. Intel. Syst. **26**(10), 890–910 (2011)
38. Matthé, T., Tré, G.D.: Bipolar query satisfaction using satisfaction and dissatisfaction degrees: bipolar satisfaction degrees. In: Shin, S.Y., Ossowski, S. (eds.) SAC, pp. 1699–1703. ACM (2009)
39. Medina, J.M., Pons, O., Miranda, M.A.V.: Gefred: a generalized model of fuzzy relational databases. Inf. Sci. **76**(1–2), 87–109 (1994)
40. Tahani, V.: A conceptual framework for fuzzy query processing: a step toward very intelligent database systems. Inf. Proces. Manage. **13**(5), 289–303 (1977)
41. Tudorie, C.: Qualifying objects in classical relational database querying. In: Galindo [28], pp. 218–245

42. Yager, R.: Higher structures in multi-criteria decision making. Int. J. Man Mach. Stud. **36**, 553–570 (1992)
43. Yager, R.: Fuzzy logic in the formulation of decision functions from linguistic specifications. Kybernetes **25**(4), 119–130 (1996)
44. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
45. Zadrożny, S.: Bipolar queries revisited. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) Modelling Decisions for Artificial Intelligence (MDAI 2005), LNAI, vol. 3558, pp. 387–398. Springer, Berlin (2005)
46. Zadrożny, S., Kacprzyk, J.: Fuzzy querying using the 'query-by-example' option in a windows-based DBMS. In: Proceedings of Third European Congress on Intelligent Techniques and Soft Computing EUFIT'95, vol. 2, pp. 733–736, Aachen, Germany (1995)
47. Zadrożny, S., Kacprzyk, J.: Multi-valued fields and values in fuzzy querying via FQUERY for access. In: Proceedings of the Fifth International Conference on Fuzzy Systems (FUZZ-IEEE'96), vol. 2, pp. 1351–1357, New Orleans, USA (1996)
48. Zadrożny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: Proceeding of the 17th International Conference on Database and Expert Systems Applications (DEXA'06), pp. 415–419. IEEE Computer Society, Krakow, Poland (2006)
49. Zadrożny, S., Kacprzyk, J.: Bipolar queries using various interpretations of logical connectives. In: Melin, P., Castillo, O., Aguilar, L., Kacprzyk, J., Pedrycz, W. (eds.) Foundations of Fuzzy Logic and Soft Computing, Lecture Notes in Computer Science, pp. 181–190. Springer (2007)
50. Zadrożny, S., Kacprzyk, J.: Bipolar queries: an approach and its various interpretations. In: Carvalho, J.P., Dubois, D., Kaymak, U., da Costa Sousa, J.M. (eds.) IFSA/EUSFLAT Conference, pp. 1288–1293 (2009)
51. Zadrożny, S., Kacprzyk, J.: Issues in the practical use of the OWA operators in fuzzy querying. J. Intel. Inf. Syst. **33**(3), 307–325 (2009)
52. Zadrozny, S., Kacprzyk, J.: Bipolar queries: an aggregation operator focused perspective. Fuzzy Sets Syst. **196**, 69–81 (2012)