Constraints and Wishes in Quantified Queries Merged by Asymmetric Conjunction

Miroslav Hudec

1 Introduction

A query condition usually consists of several atomic predicates P_i ($i = 1, \ldots, n$) merged by the *and* connective, i.e. $\bigwedge_{i=1}^{n} P_i$. The answer to such a query is empty regardless either all P_i are not satisfied or only one P_i is not satisfied and moreover attributes' values of several tuples are very close to meet this predicate.

Flexible conditions such as A_1 *is small and* A_2 *is high and* A_3 *is about 500* (where A_i is i-th attribute in a database) mitigate this problem but empty answer might also appear. Discussions and approaches related to the empty answer problems can be found in e.g. [2–4, 19].

On the other hand, users may be interested in tuples, which meet most of atomic predicates due to the human way of approximate reasoning. The option is usage of linguistic quantifiers as in Zadeh [23] in query conditions. The aim is to find all the tuples such that most of predicates from a given set are satisfied. In a general way query is of structure *select tuples where most of (about half, few …) of* $\{P_1, P_2, \ldots, P_n\}$ *is satisfied* [14]. For instance, user may be interested to know in which municipalities air pollution is severe considering several pollutants. The pollution is considered as serious when most of pollutants exceed their respective limits. In order to solve this kind of queries Kacprzyk et al. [15] suggested the FQUERY III+ tool.

Furthermore, not all atomic predicates have the same importance for users. When people express their requirements for data they can have in mind constraints (have to be satisfied) and wishes (is nice if they are satisfied). Keeping aforementioned in

M. Hudec (✉)
Faculty of Economic Informatics, University of Economics in Bratislava, Dolnozemská cesta 1, 852 35 Bratislava, Slovakia
e-mail: miroslav.hudec@euba.sk

A. Meier et al. (eds.), *The Application of Fuzzy Logic for Managerial Decision Making Processes*, Fuzzy Management Methods,
DOI 10.1007/978-3-319-54048-1_3

mind, quantified queries can be expressed in a from *most of* P_i^C *and if possible most of* P_j^W is satisfied, where $i = 1, \ldots, n, j = 1, \ldots, m$, C stands for constraints and W for wishes.

These tasks can be solved by bipolar query approaches. More about bipolar queries can be found in e.g. [6, 21, 24]. The way for solving constraints and wishes in quantified queries by bipolar approach has been suggested in [13]. This paper is focused on the other option: the noncommutative operators, which are examined in e.g. [1, 22].

In order to present option for solving constraints and wishes in quantified queries by asymmetric conjunction, the paper is structured in the following way. Sections 2 and 3 gives some preliminaries of linguistically quantified queries and asymmetric conjunctions consequently. Section 4 is devoted to quantified queries by asymmetric conjunction. Section 5 is dedicated to illustrative example and discussion. Finally, Sect. 6 concludes this paper and gives a touch for future research topics.

2 Preliminaries of Quantified Queries

In this section we recall basic notions of quantified sentences in sense of Zadeh [23] related to linguistic quantifiers and in sense of Yager [20] focused on the summarization. A quantified sentence has the form *Q r are P*, where Q is a linguistic quantifier such as *most of*, *about half* and *few*, r is a set of entities (e.g. relation in a relational database) and P is an atomic or compound predicate connected by suitable connectives [7, 11]. The truth value is calculated in the following way [12]:

$$Tr(Qx(P(x))) = \mu_Q(\frac{1}{n}\sum_{i=1}^{n}\mu_P(x_i)) \tag{1}$$

where n is the number of tuples or the scalar cardinality of database, $\frac{1}{n}\sum_{i=1}^{n}\mu_P(x_i)$ is the proportion of records in a database that satisfy predicate P and μ_Q is the membership function of chosen relative quantifier.

The truth value Tr is calculated by the linguistic quantifier. In this paper we are focused on the quantifier *most of*. It can be constructed independently by equations offered in e.g. [12] or as one granule from the family of uniformly distributes quantifiers constructed on the [0, 1] interval shown in Fig. 1 [8], where $y = \frac{1}{n}\sum_{i=1}^{n}\mu_P(x_i)$. When, instead of numbers, parameters are used, the quantifier *most of* yields [9] (Fig. 2):

$$\mu_Q(y) = \begin{cases} 1, & \text{for } y \geq n \\ \frac{y-m}{n-m}, & \text{for } m < y < n \\ 0, & \text{for } y \leq m \end{cases} \tag{2}$$

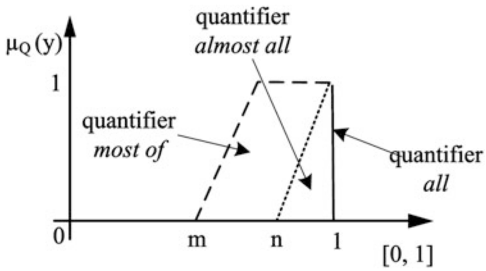

Fig. 1 Family of relative quantifiers constructed on the [0, 1] domain

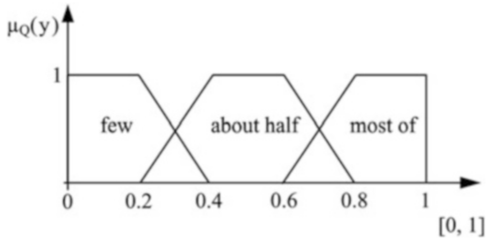

Fig. 2 Parametrized linguistic quantifier *most of*

where $0 \leq m \leq n \leq 1$. When $m = n = 1$ quantifier becomes the crisp quantifier *all*. When $0.8 \lesssim m \leq n = 1$ the quantifier becomes the fuzzy quantifier *almost all*.

The solution of such a query is the validity or truth value (from the unit interval) of a quantified sentence, not set of retrieved tuples. The solution is set of tuples when quantified query condition is a nested subquery, i.e. *select regions where most of municipalities have small population density and altitude around 1000 m*. In the next sections we are focused on adjusting quantified queries to retrieve tuples which meet majority of atomic predicates.

3 Preliminaries of Preferences in Queries

Connectives are able to specify that some predicates are more important than others. One class of connectives deals with merging constraints (negative preferences, i.e. predicates which have to be satisfied) and wishes (positive preferences, that is, it is nice if these predicates are satisfied). This connective is expressed as

$$P^C \textit{ and if possible } P^W \tag{3}$$

where P^C stands for set of predicates appearing in the constraint part and P^W stands for set of predicates appearing in the wish part. Hence, this formula expresses a weak and asymmetric conjunction.

This category of aggregation can be solved by bipolar approaches. A way how can bipolar queries handle preferences and wishes of quantified queries is suggested in [13]. Aggregation of constraints and wishes of bipolar queries (Bipolar Satisfaction Degree) is examined in [18].

Our focus is on developing way for handling this kind of queries by non-commutative aggregation. Bosc and Pivert [1] created the following six axioms in order to formally write operator (3):

- is less drastic than the *and* operator (P^C *and* P^W);
- is more drastic when only constraints (P^C) appear;
- is increasing in constraint argument;
- is increasing in wish argument;
- has asymmetric behaviour, i.e. $\alpha(\mu_{P^C}, \mu_{P^W}) \neq \alpha(\mu_{P^W}, \mu_{P^C})$ (where α is noncommutative operator and μ stands for matching degrees of constraints and wishes);
- P^C *and if possible* P^W is equivalent to P^C *and if possible* (P^C *and* P^W);

Hence, function of the structure:

$$\alpha(\mu_{P^C}, \mu_{P^W}) = \min(\mu_{P^C}, h(\mu_{P^C}, \mu_{P^W})) \tag{4}$$

is sought. Further, the *min* operator could be replaced with t-norm function, but it does not always hold as is shown later.

Function which meets aforementioned axioms and structure (4) is [1]:

$$\alpha(\mu_{P^C}, \mu_{P^W}) = \min(\mu_{P^C}, k \cdot \mu_{P^C} + (1-k) \cdot \mu_{P^W}) \tag{5}$$

where $k \in [0, 1]$. When $k = 0$ the operator becomes ordinal *and* operator merged by the minimum t-norm. On the other end of interval ($k = 1$) the wish part does not influence the result.

The second definition is based on the weighted conjunction [21]:

$$\alpha(\mu_{P^C}(r), \mu_{P^W}(r)) = \min(\mu_{P^C}(r), \max(f(P^C, P^W), \mu_{P^W}(r))) \tag{6}$$

where $f(P^C, P^W) = 1 - \max_{s \in T} \min(\mu_{P^C}(s), \mu_{P^W}(s))$. This approach refers to the concept of the conditional possibility of satisfying a predicate when another one is satisfied. The formula (6) meets requirements expressed in (4).

The formula (6) corresponds to the "global" interpretation of the term *and if possible* whereas the formula (5) corresponds to the "local" interpretation. Satisfying the constraint gives a benefit to the tuple, but there is no need to compare with other tuples from the data set [10]. Hence, tuples are analysed independently.

4 *And If Possible* Connective in Quantified Queries by Asymmetric Conjunction

Essences and main properties of quantified and noncommutative queries are discussed in Sects. 2 and 3 respectively.

Generally speaking, handling constraints P^C and wishes P^W can be realized by variety of forms. One form is quantified query. An example of such a query is *select tuples where most of* $\{P_1, P_2$ and $P_3\}$ *and if possible about half of* $\{P_4, P_5, P_6\}$ are met.

The truth value of the constraint is expressed as

$$Tr^C(Q_C(P^C)) = \mu_{Q_C}(\frac{\sum_{i=1}^{n_c} \mu_{c_i}(r)}{n_c}) \tag{7}$$

where n_c is the number of atomic predicates in the constraint part of a quantified query, P^C has the same meaning as in (3), $\mu_{c_i}(r)$ is a satisfaction degree of tuple r to the i-th predicate and Q_C is the membership function of chosen quantifier.

Analogously, the truth value of the wish is expressed as

$$Tr^W(Q_W(P^W)) = \mu_{Q_W}(\frac{\sum_{j=1}^{n_w} \mu_{w_j}(r)}{n_w}) \tag{8}$$

where n_w is the number of atomic predicates in the wish part of a quantified query, P^W has the same meaning as in (3), $\mu_{w_j}(r)$ is a satisfaction degree of tuple r to the j-th predicate and Q_C is the membership function of chosen quantifier.

Generally, quantifiers in (7) and (8) can be different. The *and if possible* operator is a relaxation of the *and* operator. When $Q_C \succ Q_W$, then we got further relaxation as was shown in example at the beginning of this section.

Let us firstly assume $Q_W \equiv Q_C$. From (5), (7) and (8) yields:

$$\begin{aligned}&\alpha(Tr^C(Q_C(P^C)), Tr^W(Q_W(P^W))) = \min(Tr^C(Q_C(P^C)),\\ &k \cdot Tr^C(Q_C(P^C)) + (1-k) \cdot Tr^W(Q_W(P^W)))\end{aligned} \tag{9}$$

or simplified

$$\alpha(\alpha_1, \alpha_2) = \min(\alpha_1, k \cdot \alpha_1 + (1-k) \cdot \alpha_2) \tag{10}$$

where $k \in [0, 1]$, $\alpha_1 = Tr^C(Q_C(P^C))$ and $\alpha_2 = Tr^W(Q_W(P^W))$.

When $k = 0$ *and if possible* becomes ordinal *and* operator merged by the minimum t-norm. For the non-quantified asymmetric condition (5) it works. Moreover, instead of minimum operator any other t-norm can be used, depending on the users preferences [10].

However, $k = 0$ cannot be applied in quantified queries (9) when $Q_C \equiv Q_W$ (i.e. both are expressed as *most of*). In this case sets P^C and P^W are equally important. If all P_i^C $(i = 1, \ldots, n)$ and few of P_j^W $(j = 1, \ldots, m)$ are satisfied, the truth value of answer is significantly under 0.5 regardless $n >> m$. Thus, all predicates should be covered by one quantifier.

When $k = 1$, then α_2 is not considered, i.e. the wish part is excluded. It implies that only minimum t-norm is suitable. The solution should be $\alpha = (\alpha_1, \alpha_1) = \alpha_1$. Only minimum t-norm has algebraic property of idempotency for all $\alpha_1 \in [0, 1]$ [16]. It means that other t-norms should be excluded. To summarize, for $Q_C \equiv Q_W$ holds $k \in (0, 1]$ and for $k = 1$ only minimum t-norm is suitable.

Regarding the case $Q_C \neq Q_W$, $k = 0$ is acceptable value. The condition *most of* P^C *and about half* P^W is covered by two different quantifiers and therefore it is not possible to mount all atomic predicates into one quantifier.

This approach works as a local interpretation, whereas approach [13] works as a global interpretation of constraints and wishes. The drawback of our approach is that influences of other tuples are not considered. The advantage of our approach is on reduced computational burden, which may have significant impact on large data sets and especially big data sets, which nowadays are hot topic for research. The next section is focused on illustrative example and further discussion.

5 Illustrative Example and Discussion

This section illustrates approach suggested in Sect. 4 and provides further discussion.

5.1 *Illustrative Example*

The task is to find suitable municipality for building cottage for holiday purposes. Relevant predicates are: altitude above sea level around 1000 m (P_1), small population density (P_2), medium area of municipality size (P_3), low pollution (P_4), low unemployment (P_5), short distance to the district capital (P_6) and positive opinion about municipality (P_7). It is highly presumable that none of municipalities meets all predicates in a query of the structure $\bigwedge_{i=1}^{7} P_i$, even though predicates have flexible boundaries.

In order to solve this problem user may say that municipality should be considered if it meets most of predicates. Furthermore, not all predicates are equally important. Let us say that P_1, P_2, P_3 and P_4 are constraints and P_5, P_6 and P_7 are wishes.

Matching degrees of municipalities to respective predicates are shown in Table 1. Results are obtained in the following way:

$\alpha_1 = \mu_{Q_C}(\frac{\sum_{i=1}^{4} \mu_{C_i}(r)}{4})$; $\alpha_2 = \mu_{Q_W}(\frac{\sum_{j=1}^{3} \mu_{C_j}(r)}{3})$; $\alpha = min(\alpha_1, \frac{\alpha_1 + \alpha_2}{2})$ (applying (10) for $k = 0.5$).

Table 1 Matching degrees to atomic predicates, constraints, wishes and to overall query condition

Municipality	P_1	P_2	P_3	P_4	P_5	P_6	P_7	α_1	α_2	α
M 1	0.8	0.9	0.6	0	1	0.7	0.8	0.21	0.95	0.21
M 2	0	0	0	0.3	0.2	0	0	0	0	0
M 3	1	0	1	1	1	1	1	0.71	1	0.71
M 4	0.2	0	0.4	0	1	1	0.9	0	1	0
M 5	0.9	0.9	0.8	1	0	0	0.1	1	0	0.5
M 6	0.9	0.9	0	1	0.6	0.8	0.6	0.57	0.48	0.52

The quantifier *most of* is expressed by (2) with parameters $m = 0.5$ and $n = 0.85$.

5.2 Discussion

It is obvious from Table 1 that the ordinal *and* operator results in a empty answer, i.e. no municipality is selected. The value of 0 is the annihilator for conjunction calculated by any t-norm.

When quantified constraint is not met, then the solution is 0 (*M 4*), because the influence of the wish part is in this case irrelevant. Furthermore, low degree of constraint fully influences solution (*M 1*). But when quantified wish is not met, then the solution is lower than only constraint is considered (*M 5*). Therefore, the best option is municipality *M 3* followed by *M 6*, because relatively high value of constraints allows wishes to influence solution.

If all atomic predicates are constraints, then result is calculated by formula $Tr(Q_C(P^C)) = \mu_{Q_C}(\frac{\sum_{i=1}^{7}\mu_{C_i}(r)}{7})$ instead of (10) for $k = 0$.

In addition, lets *M a* and *M b* meet constraints and wishes with degrees shown in Table 2. These two municipalities are in the separate table due to two reasons: (1) to keep empty answer problem of a non-quantified query $\bigwedge_{i=1}^{7} P_i$ (Table 1) obvious; (2) to compare results with one of bipolar approaches based on the lexicographic ordering [5].

In the lexicographic bipolar approach degrees for P^C and P^W are evaluated independently [17] because no aggregation between the constraint and the wish is performed (it is assumed full independence between them). Tuple r_1 is referred against r_2 if [5]:

$$\begin{aligned} & r_1 \succ r_2 \Leftrightarrow (P^C(r_1) > P^C(r_2)) \text{ or} \\ & (P^C(r_1) = P^C(r_2) \text{ and } P^W(r_1) > P^W(r_2)) \end{aligned} \tag{11}$$

It is clear from Table 2 that *M b* is slightly worse than *M a* in constraint, but significantly better in wish. Our suggested approach considers this fact and prefers

Table 2 Matching degree for constraints and wishes by (10) for $k = 0.5$

Municipality	α_1	α_2	α
M a	0.80	0.70	0.75
M b	0.78	0.95	0.78

$M\ b$ over $M\ a$. Regarding (11), the answer is opposite. That ordering may be useful in cases when constraint is hard condition. Broad discussions related to satisfaction degrees of bipolar queries can be found in [17] and [18].

This short discussion illustrates benefits of our approach. Firstly, we have shown that asymmetric conjunction is suitable for quantified queries consisted of constraints and wishes. Secondly, this approach has lower computation burden because each tuple is independently and only once considered in comparison with bipolar approaches. Furthermore, our approach is not complex such as bipolar approaches. For the users of limited knowledge in fuzzy logic and related topics, searching for a suitable bipolar approach (regarding scales for measuring constraints and wishes and aggregating into bipolar satisfaction degrees) can be hard task. On the other hand, drawbacks of our approach lies in non-existence of bipolarity in tasks, where it is required.

Last but not the least, the suggested approach contributes to the field dealing with the empty answer problems.

6 Conclusion

When we consider queries with fuzzy linguistic quantifiers, non-equally relevant atomic predicates in the form of constraints and wishes, and needs for lower computational burden for large databases, then merging constraints and wishes of quantified queries by asymmetric conjunction is a solution.

The drawback of non-handling bipolarity may be compensated by higher computational efficiency. The approaches of bipolar quantified queries and non-commutative ones are not competitive, but rather complementing to meet variations of users expectations. Furthermore, as was demonstrated in illustrative example, without the quantification a query with higher number of atomic predicates might easily return an empty result.

Further research topics can be focused on deeper comparison between our approach and bipolar ones, adjusting quantified queries handled with noncommutative conjunction for querying big data sources, merging with other approaches for solving empty answer problems and developing software tools.

References

1. Bosc, P., Pivert, O.: On four noncommutative fuzzy connectives and their axiomatization. Fuzzy Sets Syst. **202**, 42–60 (2012)
2. Bosc, P., Hadjali, A., Pivert, O.: Weakening of fuzzy relational queries: and absolute proximity relation-based approach. Mathware Soft Comput. **14**, 35–55 (2007)
3. Bosc, P., Hadjali, A. Pivert, O.: Empty versus overabundant answers to flexible relational queries. Fuzzy Sets Syst. **159**, 1450–1467 (2008)
4. Bosc, P., Brando, C., Hadjali, A., Jaudoin, H., Pivert, O.: Semantic proximity between queries and the empty answer problem. In: Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, Lisbon, pp. 259–264 (2009)
5. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS (LNAI), vol. 2522, pp. 174–182. Springer, Berlin, Heidelberg (2002)
6. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 97–114. Information Science Reference, Hershey (2008)
7. George, R., Srikanth, R.: Data summarization using genetic algorithms and fuzzy logic. In: Herrera, F., Verdegay, J.L. (eds.) Genetic Algorithms and Soft Computing, pp. 599–611. Physica-Verlag, Heidelberg (1996)
8. Hudec, M.: Issues in construction of linguistic summaries. In: Mesiar, R., Bacigál, T. (eds.) Proceedings of Uncertainty Modelling 2013, pp. 35–44. STU, Bratislava (2013)
9. Hudec, M.: Linguistic summaries applied on statistics - case of municipal statistics. Austrian J. Stat. **43**, 63–75 (2014)
10. Hudec, M., Vučetić, M.: Some issues of fuzzy querying in relational databases. Kybernetika **51**(6), 994–1022 (2015)
11. Hudec, M., Vučetić, M., Vujošević, M.: Synergy of linguistic summaries and fuzzy functional dependencies for mining knowledge in the data. In: 18th IEEE International Conference on System Theory, Control and Computing (ICSTCC 2014), Sinaia, pp. 335–340 (2014)
12. Kacprzyk, J., Zadrożny, S.: Protoforms of linguistic database summaries as a human consistent tool for using natural language in data mining. Int. J. Software Sci. Comput. Intell. **1**(1), 100–111 (2009)
13. Kacprzyk, J., Zadrożny, S.: Compound bipolar queries: combining bipolar queries and queries with fuzzy linguistic quantifiers. In: 8th Conference of the European Society of Fuzzy Logic and Technology (EUSFLAT 2013), Milan, pp. 848–855 (2013)
14. Kacprzyk, J., Ziółkowski, A.: Database queries with fuzzy linguistic quantifiers. IEEE Trans. Syst. Man Cyber. SMC **16**(3), 474–479 (1986)
15. 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)
16. Klement, E.P., Mesiar, R., Pap. E.: Triangular norms: basic notions and properties. In: Klement, E.P., Mesiar, R. (eds.) Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms, pp. 17–60. Elsevier, Amsterdam (2005)
17. Matthé, T., De Tré, G.: Ranking of bipolar satisfaction degrees. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part II, CCIS 298, pp. 461–470. Springer, Berlin, Heidelberg (2012)
18. Matthé, T., Nielandt, J., Zadrożny, S., De Tré, G.: Constraint–Wish and Satisfied–Dissatisfied: an overview of two approaches for dealing with bipolar querying. In: Pivert, O., Zadrożny, S. (eds.) Flexible Approaches in Data, Information and Knowledge Management, Studies in Computational Intelligence, vol. 497, pp. 21–44. Springer International Publishing, Cham (2014)

19. Smits, G., Pivert, O., Hadjali, A.: Fuzzy cardinalities as a basis to cooperative answering. In: Pivert, O., Zadrożny, S. (eds.) Flexible Approaches in Data, Information and Knowledge Management, Studies in Computational Intelligence, vol. 497, pp. 261–289. Springer International Publishing, Cham (2014)
20. Yager, R.R.: A new approach to the summarization of data. Inf. Sci. **28**, 69–86 (1982)
21. Yager, R.R.: Higher structures in multi-criteria decision making. Int. J. Man-Mach. Stud. **36**, 553–570 (1992)
22. Yager, R.R.: Fuzzy sets and approximate reasoning in decision and control. In: 1992 IEEE International Conference on Fuzzy Systems, San Diego, pp. 415–428 (1992)
23. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Comput. Math. Appl. **9**, 149–184 (1983)
24. Zadrożny, S., Kacprzyk, J.: Bipolar queries: a way to enhance the flexibility of database queries. In: Ras, Z.W., Dardzinska, A. (eds.) Advances in Data Management. Studies in Computational Intelligence, vol. 223, pp. 49–66. Springer, Berlin, Heidelberg (2009)