Towards a representation system for possibilistic relations

Patrick Bosc, Olivier Pivert

IRISA/ENSSAT
Technopole Anticipa BP 447
22305 Lannion Cedex France

bosc@enssat.fr, pivert@enssat.fr

Abstract

This paper is situated in the area of databases containing ill-known attribute values represented by possibility distributions. In order to manipulate such databases in a safe and efficient way, a constrained framework has been previously suggested, where a restricted number of querying operations are permitted (selections and fk-joins calling on attributes taking imprecise values). The key for efficiency resides in the fact that these operators do not require to make computations explicitly over all the more or less possible worlds. However, this framework has several limitations and an extended model based on the notion of nested relations enabling to represent possibility distributions over sets of attributes is proposed.

1 Introduction

Nowadays, applications are more and more sophisticated and they require the support of intelligent softwares (systems), among which database management systems. One may think that in the near future, in order to reflect faithfully the reality, databases will have to manage imperfect data, i.e., data tainted with imprecision and/or uncertainty. Such a perspective imposes that DBMSs are able to store and manage (if possible efficiently) such data, in particular in terms of querying. Some works have been carried out in the area of null values [9], but it turns out that it is out of reach to design a complete extension of the relational algebra in this context, i.e., to design a model inside which the result of any operation is still a relation of the model and has a thorough interpretation in terms of worlds. Similar results have been achieved for disjunctive data (or-sets,

probability or possibility distributions) which can be seen as refinements of null values. This situation is not specific to the relational model and the same conclusions hold for object-oriented models or extended relational ones. For instance, if we think of a join in case of a joining attribute taking imprecise values, the problem is that a tuple of the first relation can be joined with several tuples of the second one, which results in several different tuples which are not independent as shown hereafter. Let us consider a tuple <{a1 **or** a2 **or** a3}, b> to join with the three tuples <{a1 **or** a2}, c1>, <a3, c2>, and <{a1 **or** a2}, c3>. The representation of the result as three tuples: <{a1 **or** a2 **or** a3}, b, {a1 **or** a2}, c1>, <{a1 **or** a2 **or** a3}, b, a3, c2>, <{a1 **or** a2 **or** a3}, b, {a1 **or** a2}, c3> is not correct for (at least) two reasons: i) the second tuple cannot co-occur with the other two and ii) the first and third tuples are not independent since the choice <a1, b, a1, c1> for the first one and <a2, b, a2, c3> for the other are inconsistent.

This paper considers a possibilistic extension of the relational model of data, in which imprecise values are allowed and represented as weighted disjunctions of candidates. The objective of this paper is to show that it is possible to devise a representation system valid for a restricted set of operations including selections, projections, joins on attributes taking precise values (called "regular" joins in the remainder) and fk-joins which represent a special type of join, that is likely to appear frequently in practice. It is important to emphasize that the interpretation of all these operations in terms of worlds remains the correct reference. However, for efficiency purposes, the explicitation of worlds is undesirable and computations are made on the initial possibilistic database D (i.e., queries are processed in a "compact" way on D itself, with the guarantee that the result is the same as that of Q over the set of regular databases issued from D, according to the notion of a strong representation system). Such an approach can be seen as an attempt to go beyond the proposal made by Prade and Testemale in the mid 80s which did not tackle the issue of representation systems and was limited to selections and projections (see [12] for details). The contribution of this paper is then twofold : i) the model underlying the representation system (for the above operations) and ii) the possibility for using a certain type of join operation in presence of imprecise values. To our knowledge, there does not exists any previous research works of the same nature. One can nevertheless mention some works that are somewhat related to certain points tackled here, such as: i) [2, 8, 9] where the representation systems valid in the presence of null values are studied from a theoretical point of view, ii) [10, 11] where the or-set model is introduced, iii) [1] where the interest and the properties of non-normalized relations (i.e., including nested relations) are studied, and iv) several proposals of extended algebras in the probabilistic framework [6, 7] where the authors do not aim at defining strong representation systems.

The rest of the paper is structured as follows. In section 2, the two main types of elements that have a significant influence on the model are presented. They concern the choice of the possibilistic framework to represent imprecise values and the interpretation of a possibilistic database in terms of usual databases containing only precise information on the one hand and the fact that it is sometimes necessary to take into account dependencies between candidate values coming from different attributes on the other hand. A presentation of the principal aspects of the data model underlying our representation system is made in section 3. The key idea is to allow possibility distributions over tuples of values, which indeed correspond to a structure involving nested relations. In section 4, four operations working with this model are presented with a focus on two of them (selection and fk-join). Section 5 is dedicated to some conclusions and lines for future works.

2 Basic elements of the model

In this section, two principal aspects impacting the model are pointed out. The first one concerns the representation of imprecise values as possibility distributions and the interpretation of such databases in terms of regular ones. The second point is related to the fact that some operations may require to express that candidate values pertaining to different attributes are not independent.

2.1 Imprecise values, possibility distributions and worlds

Possibility theory [14] provides an ordinal model for uncertainty where imprecision is represented by means of a preference relation coded by a total order over the possible situations. This framework is closely related to fuzzy set theory [13] since the idea is to constrain the values taken by a variable thanks to a (normalized) fuzzy set called a possibility distribution. More formally, a possibility distribution is an application π from a given domain X to the unit interval ([0, 1]) and $\pi(a)$ states the degree to which a is possible for the considered variable. The normalization condition imposes that at least one of the values of the domain (a_0) is completely possible, i.e., $\pi(a_0) = 1$. This setting is particularly suited to take into account uncertainty represented by linguistic terms such as high, large, medium and so on. When the domain is discrete, a possibility distribution is written $\{\pi_1/a_1 + \ldots + \pi_n/a_n\}$ where a_i is a candidate value and π_i its possibility degree. A measure of possibility Π is associated to any event such that: i) $\Pi(\varnothing) = 0$, ii) $\Pi(X) = 1$ and iii) $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$. Due to the non additivity of the model, the value of $\Pi(\bar{A})$ cannot be deduced from that of $\Pi(A)$ and a second measure (of certainty or necessity) N has been introduced, $N(A) = 1 - \Pi(\bar{A})$.

This approach proposes a unified framework for representing precise values, as well as imprecise ones (regular or fuzzy sets), including the so-called "null values" (see [5] for more details). Similarly to the probabilistic model, it can be seen as a refinement of disjunctive values, also called or-sets. More formally, an or-set can be modeled as a possibility distribution where any candidate is completely possible. One of the advantages of the possibilistic model is its ability to deal simultaneously with imprecise values and fuzzy queries since the semantics of preferences is the same.

From a semantic point of view, a database D containing ill-known values, represented as possibility distributions, can be interpreted as a set of usual databases D_1, ... , D_p (denoted as rep(D)), each of which being more or less possible. Any D_i is obtained by choosing a candidate in each possibility distribution appearing in D and one of these (regular) databases D_k is supposed to correspond to the actual state of the world. The degree assigned to a world (D_i) is the minimum (conjunction operator) of the degrees tied to each of the chosen candidates. For instance, the following relation emp over attributes #id, years and job:

emp

#id	years	job
1	{1/5 + 0.8/4 + 0.3/3}	cashier
2	{1/8 + 0.5/9}	{1/salesman + 0.6/cashier}
3	{1/4 + 0.8/5}	salesman
4	2	salesman

can be associated with 24 (3 * 2 * 2 * 2) more or less possible worlds. The choices: i) 5 for years in the first tuple, ii) 8 for years and cashier for job in the second one and iii) 5 for years in the third one lead to a world which is 0.6 possible.

Since we aim at the design of a strong representation system, it must be possible to "derive" all the (more or less) possible worlds including those in which some tuples are absent (which correspond to the removal of at least one candidate). To achieve this goal, any tuple t of a possibilistic relation must be provided with a necessity degree N telling the extent to which it is certain that t belongs to the considered relation. N equals 1 when no candidate has been discarded by some operation (which is the case for initial relations). When N differs from 1, its value serves to compute the possibility degree (1 − N) attached to the choice of a world where the tuple is absent. Let us illustrate this with the example of the previous relation onto which the selection job = "salesman" is applied. Clearly, the first tuple does not qualify, whereas the last two do. In order to have a faithfull representation (in the sense of a strong representation system), for the second tuple, it is necessary to keep track of the fact that the condition

may be satisfied although it is not certain. In this case, we will use the following representation of the resulting relation:

salesmen

#id	years	job	N
2	{1/8 + 0.5/9}	salesman	0.4
3	{1/4 + 0.8/5}	salesman	1
4	2	salesman	1

which tells us that it is indeed possible for the first tuple to be absent in certain worlds (moreover the highest possibility degree for this equals 1 – 0.4 = 0.6). For instance, the world:

#id	years	job
3	5	salesman
4	2	salesman

is possible at the degree: min(0.6, 0.8, 1) = 0.6 and it corresponds to the choice of the value “cashier” in the initial relation emp.

1.2 Non independence of candidate values

The regular possibilistic model is based on the fact that attributes may take values which are possibility distributions. Implicitly, such a model states that candidate values appearing in distinct attributes of a given tuple are independent. For instance, in the tuple <Jones, {1/3 + 1/4 + 0.7/5 + 0.3/6}, {1/salesman + 1/manager + 0.3/cashier}>, there is no constraint tying the choices of a number of years and a job. If this seems reasonable in this example, this is not always the case, especially for relations stemming from intermediate relational expressions. Once again, let us illustrate the need through an example with a relation emp defined over the attributes name, years, job, l-city and b-city, these two attributes describing respectively the living and birth city of the considered employee. Moreover, let us consider the case of a selection operation over this relation with the criterion l-city ≠ b-city. The representation of all possible worlds makes it necessary to make explicit the pairs of candidates which comply with the condition. For instance, with the tuple <Jones, 4, salesman, {1/Paris + 0.4/Nice}, {0.7/Marseille + 1/Nice}>, the (three) acceptable pairs of cities (along with their possibility degree) are: 0.7/(Paris, Marseille), 1/(Paris, Nice) and 0.4/(Nice, Marseille). It is obvious that such a situation cannot be expressed by independent sets of candidate

values over l-city and b-city. It is then necessary to extend the model so as to meet this need.

3 An overview of the extended possibilistic model

As it has been pointed out before, there is a need for expressing that, inside a given tuple, some candidate values coming from different attributes are tied to each other. The idea is then to improve the model so that some attribute values are possibility distributions over several domains rather than over atomic domains only (here, we restrict ourselves to attributes whose domain is discrete). In other words, some candidates are (weighted) tuples and the model is based on the concept of nested relations. At that point, it is of prime importance to emphasize two particular aspects, both strongly connected with the fact that these relations are used to support possibility distributions: i) tuples of nested relations are weighted since an element of a possibility distribution is assigned a level of preference and ii) the extension of a nested relation has a disjunctive meaning as a possibility distribution describes a set of alternative candidates.

One originality of the model proposed here, is to restrict the number of nesting levels to a single one. In the remainder, the notation $R(A_1, \dots, A_m, X_1(A_p, \dots, A_q), \dots, X_n(A_k, \dots, A_l))$ stands for a schema in which A_1 to A_m are elementary attributes whose values are either precise or possibility distributions and $X_i(A_h, \dots, A_j)$ represents a "structured" attribute whose values are possibility distributions over tuples built on the attributes A_h to A_j. Indeed, an elementary attribute can be seen as a special case of a structured one where the arity of the nested relation is 1. Obviously, such relations have an interpretation in terms of worlds as it is the case for ordinary possibilistic relations. Indeed, when one moves to a given world, the schema becomes unnested, since a structured candidate value can be split into atomic values.

Let us now talk about some relationships between ordinary possibilistic relations and extended ones. It is clear that, from an ordinary possibilistic relation, it is possible to build an equivalent extended relation involving any number of nested relations. For instance, the schema R(A, B, C, D) can be represented as R1(A, X(B, C), D), but also as R2(X(A, C, D), B), or even R3(X(A, B, C, D)). Of course, the converse is not feasible (in general) for the reasons already mentioned in section 2. The interesting point lies in the equivalence of all these representations in terms of worlds and this property will be exploited in the next section where it serves as a basis for the definition of selection and join operations. Last, from a storage point of view, it is obvious that R(A, B, C, D) is the most convenient since the equivalent nested relations involve Cartesian products which require more space.

From a concrete point of view, let us emphasize the fact that initial relations (i. e., those which are permanent in the database) remain usual possibilistic relations and that

the extended model is used (when necessary) during calculations to represent in a correct fashion the result of intermediate operations including the final result.

According to this model, it would be possible to have the relation schema R(ss#, date, X(car-t, brand)) describing the type (car-t) and brand of cars purchased at a certain date by persons (ss#) in order to represent some intermediate result. A possible extension of a relation r(R) is:

r	ss#	date	X			N
			car-t	brand	Π	
	1	d1	camry	toyota	1	1
	2	d3	camry taurus avensis	toyota ford toyota	1 0.7 0.4	1

Names and brands of cars are represented as pairs of (more or less possible) candidates unlike it is done in example 5. In this case, there is only one candidate for the nested relation over the schema X(car-t, brand) in the first tuple and 3 in the second tuple. For space reason, the alternate equivalent representation is used in the remainder:

r	ss#	date	X	N
			car-t brand	
	1	d1	{1/<camry, toyota>}	1
	2	d3	{1/<camry, toyota>+0.7/<taurus, ford>+0.4/<avensis, toyota>}	1

If the second candidate value is chosen in the second tuple, we get the ordinary relation: <1, d1, camry, toyota>, <2, d3, taurus, ford> which is part (since the database may involve other relations) of a world which is at most 0.7 possible.

4 The operators of the model

In this section, we focus on the operations which work in the context of the representation system, i.e., those which comply with the principle: rep(Q(D)) = Q(rep(D)). In other words, the result of a query Q processed on the "compact" database D is the same as that of Q over the set of regular databases issued from D. Four operations are considered and more attention is paid to two of them which deserve some details, even if, for space reason, the proofs of validity of these operators (in the sense of the above

equality) are not given here. The first one is the join of (possibilistic) relations over attributes taking always precise values. This operation raises no specific problem and works as usual (i.e., a tuple is built the precise values match in both input tuples). The second operator is the projection, which, appears usually (at least) as a terminal operation in order to remove useless information (attributes). Here, two aspects must be dealt with : i) the fact that such attributes may be part of a nested relation and ii) the impact of removal of attributes on the value of N in each tuple. For the sake of brevity, details on that are not given here. The last two operators (selection and fk-join) are presented in the next subsections.

4.1 Selection

With respect to the initial proposition made by Prade and Testemale [12], the selection has to be revisited in order to i) keep in the resulting relation only the candidate values complying with the selection criterion and ii) take into account criteria which establish a dependency between attributes. The first point is fairly easy to manage since it is just necessary to modify the considered imprecise value and retain the candidates which satisfy the condition, then modify the necessity degree (N) attached to the tuple, depending on the degrees of the candidates that have been discarded. As to the second point, two situations where a selection operation leads to a resulting relation involving a nested one may occur. The first situation concerns atomic selection criteria made of two attributes, i.e., of type: "A_i θ A_k" where θ is a comparison operator. Here, the resulting relation schema involves the nested relation $X(A_i, A_k)$ in order to represent the dependency between candidate values coming from A_i on the one hand and A_k on the other hand. In this case, assuming that the input relation r has the nonnested schema $R(A_1, \ldots, A_n)$, we have:

$$\begin{aligned}&\text{sel}(r, A_i\ \theta\ A_k) = \\ &\quad \{N'/<a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n, \{\min(\lambda_{i,p}, \lambda_{k,l})/<(a_{i,p}, a_{k,l}>> \mid \\ &\quad \exists\, t = N/<a_1, \ldots, a_n> \textbf{ and } (t \in r) \textbf{ and } (a_{i,p}\ \theta\ a_{k,l})\} \textbf{ and } a_j = \{\lambda_{j,1}/a_{j,1} + \ldots + \lambda_{j,n_j}/a_{j,n_j}\} \\ &\quad \textbf{and } N' = \min(N, 1 - \sup_{p,l \mid \text{not } (a_{i,p}\ \theta\ a_{k,l})} \min(\lambda_{i,p}, \lambda_{k,l}))\}.\end{aligned}$$

The other case is when the condition is made of a disjunction of elementary conditions of the form: $c_i(A_i)$ **or** … **or** $c_k(A_k)$ where c_j is an atomic condition of type "A_j θ value" with θ a comparison operator. Then, the result of the selection is a relation whose schema involves the nested relation $X(A_i, \ldots, A_k)$.

Let us illustrate this with the relation schema PUR(ss#, date, car-t, l-city, p-city) describing the type (car-t) of cars purchased at a certain date in a given city (p-city) by persons (ss#) living in a city (l-city). The query searching for those people who have

made a purchase outside their living city or who have bought an Avensis or a Taurus, is a selection (car-t ∈ {avensis, taurus} **or** p-city ≠ l-city). With p(PUR):

p

ss#	date	car-t	l-city	p-city
1	d1	taurus	{1/antibes + 0.3/nice}	nice
1	d2	jeep	paris	{1/paris + 0.5/nice}
2	{1/d3 +1/d4}	{0.5/avensis + 1/camry}	{0.4/madrid + 1/granada}	madrid

we get the result:

ss#	date	X			N
		car-t	l-city	p-city	
1	d1	{1/<taurus, antibes, nice> + 0.3/<taurus, nice, nice>}			1
1	d2	{0.5/<jeep, paris, nice>}			0
2	{1/d3 + 1/d4}	{1/<camry, granada, madrid> + 0.4/<avensis, madrid, madrid> + 0.5/<avensis, granada, madrid>}			0.6

Here again, the value of N in the last tuple (0.6) expresses that the most possible candidate which did not satisfy the condition has the degree 0.4 (corresponding to the choice <camry, madrid, madrid> in p). In fact, the result of the selection can be seen as a restriction of the candidates issued from the Cartesian product of the initial candidates over the concerned attributes to those satisfying the condition. Then, the desired property with respect to worlds is achieved.

1.2 Foreign key join

This operator is used inside queries calling on a (regular) relation s over the schema S(X, Y) describing a precise function between the sets of attributes X and Y on the one hand and a (possibilistic) relation r whose schema is R(X, Z) where X (and Z) may take imprecise values. As discussed in [3], such a case is connected with the notion of a functional dependency. Indeed, relations r and s come from the decomposition of an initial relation (gathering X, Y and Z) aiming at the limitation of redundancy (tied to the presence of a function connecting X and Y). This specific join [4], called foreign key join (fk-join for short), is an equi-join which aims at computing the image of a

joining attribute value (through the function). Each tuple of the result can be seen as a kind of completion of a tuple of r (via its X-component) thanks to the knowledge of the function connecting X and Y-values materialized in s. It is worth mentioning that the values appearing in the distribution for X in a resulting tuple correspond to those candidates that have an image in s (according to the function). By construction, this operation expresses a connection between X and Y values (antecedents and images) and it is necessary to maintain it in the result in order to comply with the requirement of a strong representation system. According to the model presented in section 3, the nested relation mechanism is used to achieve this goal. If we start with the possibilistic relation r and the regular relation (function) s whose respective schemas are $R(A_1, \ldots, A_m, B_1, \ldots, B_q)$ and $S(A_1, \ldots, A_m, C)$, the fk-join is defined as:

fk-join(r, s, $\{A_1, \ldots, A_m\}$) =

$\{N'/<\{ \ldots + \min(\lambda_{1, k_1}, \ldots, \lambda_{m, k_m})/<a_{1, k_1}, \ldots, a_{m, k_m}> + \ldots\}, c, b_1, \ldots, b_q> \mid$
$\exists\, t = N/<a_1, \ldots, a_m, b_1, \ldots, b_q>$ **and** $\exists\, u = <a_{1, k_1}, \ldots, a_{m, k_m}, c>$ **and** $(t \in r)$ **and**
$(u \in s)$ **and** $a_i = \{\lambda_{i, 1}/a_{i, 1} + \ldots + \lambda_{i, p_i}/a_{i, p_i}\}$ **and** $\forall j \in [1, m], a_{j, k_j} \in scv(a_i)$ **and**
$N' = \min(N, 1 - \sup_{k_i \mid <a_{1, k_1}, \ldots, a_{m, k_m}> \notin proj(s, \{A_1, \ldots, A_m\})} \min(\lambda_{1, k_1}, \ldots, \lambda_{m, k_m}))\}$

where $scv(a_i)$ is the set of candidate values appearing in a_i. In other words, one looks for the image in s of each candidate value over the attributes $A_1, \ldots, A_m$ of a tuple t of relation r.

If we use the schemas R(A, B, C) and S(A, B, D) and the extensions:

r

A	B	C
{1/a1 + 1/a2 + 0.7/a3}	{1/b1 + 0.3/b2}	c1
{1/a1 + 0.3/a4}	b2	c2

s

A	B	D
a1	b1	d1
a1	b2	d2
a2	b1	d3
a3	b2	d2

the result of fk-join(r, s, {A, B}) is:

X			C	N
A	B	D		
{1/<a1, b1, d1> + 0.3/<a1, b2, d2> + 1/<a2, b1, d3> + 0.3/<a3, b2, d2>}			c1	0.3
{1/<a1, b2, d2>}			c2	0.7

Here again, the soundness of the procedure stems from the fact that any initial (non nested) possibilistic relation r has an equivalent representation involving a nested relation. Then, the elements retained in the result are only those for which there is a match. Similarly to the selection, if an attribute pertaining to those involved in the joining condition is already part of a nested relation in r, the entire schema of this nested relation is incorporated in that of the resulting relation.

5 Conclusion

In this paper, an extension of the relational possibilistic model of data formerly devised by Prade and Testemale, is proposed. Its major characteristic lies in the introduction of single level (weighted) nested relations inside schemas, which leads to attribute values being possibility distributions over Cartesian products of atomic domains. Such a model offers a number of significant advantages with respect to the initial one. In particular, it constitutes a strong representation system for the operations of selection (with a wide range of criteria), fk-join (including the multiple attribute case), projection and regular join.

This work should be pursued along several directions including the study of i) possibility distributions over continuous domains and ii) complexity of queries addressed to databases in the context of the new model. It is clear that the use of nested relations entails manipulations which are more costly than those performed on regular possibilistic relations and it would be useful to identify queries for which the nesting mechanism can be ignored without threatening the correctness of the final answer. Moreover, it would be of interest to implement a system managing imprecise information and offering the manipulation capabilities evoked in this paper.

References

[1] S. Abiteboul, P.C. Fischer, and H.J. Schek. *Nested relations and complex objects in databases*. Lecture Notes in Computer Science, vol. 361. Springer-Verlag, Berlin, 1989.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*. Addison-Wesley, 1995.

[3] P. Bosc, L. Liétard, and O. Pivert. An approach to functional dependencies in fuzzy databases. In *Proc. 9th In.l Conf. on Fuzzy Systems (FUZZ-IEEE'2000)*, pages 917–922, 2000.

[4] P. Bosc, L. Liétard, and O. Pivert. A function-based join for the manipulation of possibilistic relations. In *Proc.of the ACM Symposium on Applied Computing (SAC'2001),* pages 472–476, 2001.

[5] P. Bosc and H. Prade. An introduction to the fuzzy set and possibility theory-based treatment of flexible queries and uncertain and imprecise databases. In A. Motro and P. Smets, editors, *Uncertainty Management in Information Systems*, pages 285–324. Kluwer Academic Publishers, 1997.

[6] D. Dey, and S. Sarkar. A probabilistic relational model and algebra. *ACM Transactions on Database Systems*, 21:339–369, 1996.

[7] T. Eiter, J.J. Lu, T. Lukasiewicz, and V.S. Subrahmanian. Probabilistic object bases. *ACM Transactions on Database Systems*, 26:264–312, 2001.

[8] G. Grahne. *The problem of incomplete information in relational databases*. Lecture Notes in Computer Science, vol. 554. Springer-Verlag, Berlin, 1991.

[9] T. Imielinski and W. Lipski W. Incomplete information in relational databases. *Journal of the ACM*, 31:761–791, 1984.

[10] W. Lipski. On semantic issues connected with incomplete information databases. *ACM Transactions on Database Systems*, 4:262–96, 1979.

[11] W. Lipski. On databases with incomplete information. *Journal of the ACM*, 28:41–70, 1981.

[12] H. Prade and C. Testemale. Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries. *Information Sciences*, 34:115–143, 1984.

[13] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[14] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.