A Unified Model for Set-based Computations

Y.Y. Yao and N. Noroozi
Department of Mathematical Sciences, Lakehead University
Thunder Bay, Ontario, Canada P7B 5E1
E-mail: yyao@flash.lakeheadu.ca

Abstract *In many situations with incomplete or vague information, it may not be possible to obtain the exact values of certain parameters. A plausible solution to this problem is to use a set-valued representation scheme, in which a parameter is assigned to a set of all possible values based on the available information. This paper examines a unified model for set-based computations. Within the proposed framework, a number of existing methods are analyzed, including interval-number algebra, interval-set algebra, interval-valued fuzzy reasoning, and set-based information systems. It is argued that the proposed approach may be useful in establishing a unified framework for uncertainty representation and approximate reasoning.*

1 Introduction

In the design of intelligent information systems, it is important to develop tools for representing uncertainty and reasoning under uncertainty. The extensive studies on these topics have resulted in many useful extensions of classical mathematical tools. A class of such tools is the interval-based computations, which use a pair of lower and upper bounds to specify the range within which lies the true value. Examples of interval-based approaches are interval-number algebra [11], interval-set algebra [7, 27], interval algebra [14], rough-set model [16], incidence calculus [2], fuzzy-set model [29], belief and plausibility functions [22, 23], necessity and possibility functions [4], interval-valued probabilistic reasoning [1, 17, 21], and interval-valued fuzzy reasoning [9]. Recently, Wong, Wang and Yao introduced the concept of interval structures [26]. To some extent, their work unifies some of these various notions.

The main motivation of interval-based computations is that in many practical situations it may be impossible to specify the exact values of certain parameters under consideration. For example, it may be difficult to measure the precise value of the temperature. One can therefore give only a lower and an upper bound to indicate its range. It may also be the case that an expert is unable to give the exact probability values of certain events, but is able to provide probability intervals. From this point of view, interval-based computations are more flexible, as they use less restrictive knowledge representation schemes. However, a few important issues must not be overlooked. It is necessary to establish a unified framework as a basis for the study of interval-based computation methods. Such a framework may provide a common ground for various approaches using different notations. This kind of work has been done in other context of uncertainty management [19]. An underlying assumption of interval-based computations is that intervals are defined on ordered sets. For instance, interval numbers are defined on the set of real numbers using the "greater than or equal to" relation $\geq$. Interval sets are defined on the power set of a set using the set inclusion relation $\supseteq$. In many cases, an order relation may not exist on the values of certain parameters. For example, it is not always possible to define an order relation on the set of languages. If the available information suggests that a given person may speak English or French, it is impossible to represent such information in terms of intervals. In this case, a set-valued representation scheme may be used instead. By observing the fact that an interval indeed specifies a set of values, it is natural to generalize interval-based computations into set-based computations. An added advantage of set-based computations is that a parameter may takes more than one value. In fact, set-valued attributes have been investigated in database systems [6].

The main objective of this paper is to investigate some basic issues of set-based computations. Operations on set values are defined based on the corresponding point-based (i.e., single-value-based) operations on their members. The properties of set-based computations are examined in connection to the corresponding properties of the point-based computations.

Particularly, we analyze properties of binary operations, such as the commutativity and the associativity, and properties of binary relations, such as the reflexivity and transitivity. Within the proposed framework, we present a critical analysis of a number of existing set-based computation methods. This provides further evidence supporting the proposed model. To a large extent, the present study may be regarded as a more explicit re-examination of methods that have been implicitly used in many studies, using a unified notion. The results of such an investigation will be useful in establishing a framework for more systematic study of set-based computations.

2 Set-based Computations

Various types of computational methods can be modeled by different mathematical systems. For example, in numerical computations, one uses the numerical system $(\Re, +, -, \times, /, \geq, =)$, where $\Re$ is the set of real numbers, $+, -, \times$ and $/$ are binary operations on $\Re$, and $\geq$ and $=$ are binary relations on $\Re$. In logical inference, one uses the system $(V, \wedge, \vee, \neg)$, where V is a set of truth values, and $\wedge$, $\vee$ and $\neg$ are operations defined on V. Abstracting from these examples, we use the concept of a relational system [20]. A *relational system* is an ordered $(p+q+1)$-tuple,

$$RS = (U, \circ_1, \ldots, \circ_p, R_1, \ldots, R_q), \tag{1}$$

where U is a nonempty set, $\circ_1, \ldots, \circ_p$ are operations on U, and $R_1, \ldots, R_q$ are (not necessarily binary) relations on U. Although the definition of relational systems is very general, we usually deal with a small number of operations and relations. In this section, we only consider unary and binary operations, and binary relations.

In a relational system, all the operations and relations are defined on elements of the set U. When it is impossible to represent a physical quantity using a single element of U in practice, a subset of U may be used. To accommodate this set-based representation, one has to extend the operations and relations on the elements of U into operations and relations on the subsets of U. We refer to the former as point-based computations, and the latter as set-based computations. The extended relational system must be defined in such a way that it preserves important characteristics of the original system. In particular, when only singleton sets of U are used, set-based computations must reduce to point-based computations.

A binary operation $\circ$ on U can be extended into a binary operation $\circ'$ on 2^U by applying it to the members of the subsets of U. For any two elements $a, b \in U$, $a \circ b$ is an element of U. Given two subsets $A, B \in 2^U$, one can derive another subset $C \in 2^U$ by collecting all elements $a \circ b$, where $a \in A$ and $b \in B$. That is, we may adopt the following definition of the extended operation. For clarity, we will not consider the case when one of the operands is an empty set.

Definition 1. Suppose $\circ$ is a binary operation on U. An extended binary operation $\circ'$ on $2^U - \{\emptyset\}$ is defined by:

$$A \ \circ' \ B = \{a \circ b \mid a \in A, b \in B\}. \tag{2}$$

If only singleton subsets of U are used, operation $\circ'$ reduces to $\circ$. By definition, many properties of operations in the original system can be carried over by the extended operations. The theorem given below shows that the commutativity and associativity are preserved.

Theorem 1. Suppose $\circ$ is a binary operation on U, and $\circ'$ on $2^U - \{\emptyset\}$ is the extended binary operation defined by equation (2). Then,

(a). if $\circ$ is commutative, $\circ'$ is commutative,

(b). if $\circ$ is associative, $\circ'$ is associative.

In general, it is not necessary to use the entire set 2^U to construct a system for set-based computations. In many situations, one may find that it is more meaningful to use a subset of 2^U. For example, if U is an ordered set, one may consider the set of all closed intervals, which is only a subset of 2^U. Section 3 examines such systems and shows that other properties of $\circ$ may be carried over by $\circ'$.

The extension of a binary relation R can be done in a similar manner. However, the process is more complicated because the relation may only hold for some elements of two subsets of U. Given two subsets $A, B \in 2^U - \{\emptyset\}$, relation R may not hold for any pair (a, b), where $a \in A$ and $b \in B$. In the case when the relation R does hold between elements of A and B, it may hold for only one pair, two pairs, ..., or all pairs. We may, for example, define a family of graded relations based on the number of pairs for which the relation holds. In the present study, we consider the two extreme points of this spectrum.

Definition 2. Suppose R is a binary relation on U. A pair of extended binary relations $\langle R_*, R^* \rangle$ on $2^U - \{\emptyset\}$ is defined by:

$$\begin{aligned} A \ R_* \ B &\iff (\forall a \in A, \forall b \in B) \ aRb, \\ A \ R^* \ B &\iff (\exists a \in A, \exists b \in B) \ aRb. \end{aligned} \tag{3}$$

These two relations can be interpreted as representing necessity and possibility. If $A\ R_*\ B$, an element of A is necessarily related to an element of B. For example, we write $[1,2] \geq_* [0,1]$ to represent the fact that any number in the interval $[1,2]$ is greater or equal to a number in the interval $[0,1]$. If $A\ R^*\ B$, an element of A is possibly related to an element of B. We write $[1,2] \geq^* [1,3]$ to represent the fact that a number in the interval $[1,2]$ may be greater or equal to a number in the interval $[1,3]$. It can be seen that the following relationship holds: for $A, B \in 2^U - \{\emptyset\}$,

$$A\ R_*\ B \Longrightarrow A\ R^* B. \tag{4}$$

That is, if a pair of elements from two sets are necessarily related, they are possibly related. If only singleton subsets of U are used, both R_* and R^* reduce to relation R.

The following theorem shows that properties of R may be carried over by $\langle R_*, R^* \rangle$.

Theorem 2. Suppose R is a binary relation on U, and $\langle R_*, R^* \rangle$ on $2^U - \{\emptyset\}$ is a pair of extended binary relations defined by equation (3). Then,

(a). if R is reflexive, R^* is reflexive,

(b). if R is symmetric, R_* and R^* are symmetric,

(c). if R is transitive, R_* is transitive.

From Definitions 1 and 2, given a relational system

$$RS = (U, \circ_1, \ldots, \circ_p, R_1, \ldots, R_q),$$

equation we can extend it into a set-based system:

$$RS' = (\Gamma(2^U), \circ'_1, \ldots, \circ'_p, \langle R_{1*}, {R_1}^* \rangle, \ldots, \langle R_{q*}, {R_q}^* \rangle).$$

The set $\Gamma(2^U) \subseteq 2^U$ should be chosen such that the extended operations $\circ'$ are closed. System RS' is a natural generalization of pointed-based system RS. It should be noted that we have not put any constraints on the elements of U. Elements of U may in fact be sets themselves. Thus, the proposed framework provides a simple, yet powerful enough, model of set-based computations.

Example 1. This simple example illustrates the basic concepts introduced in this section. Consider the following relational system,

$$RS = (U, \circ, R),$$

where $U = \{a, b, c\}$, and the binary operation $\circ$ and the binary relation R are defined by the following tables:

$\circ$	a	b	c
a	a	a	a
b	a	b	a
c	a	a	b

R	a	b	c
a	1	0	1
b	0	1	0
c	1	0	1

An entry in the relation table with value 1 indicates that the the relation holds between the elements in the corresponding row and column. Operation $\circ$ is commutative and associative, and relation R is both reflexive, symmetric and transitive, i.e., R is an equivalence relation. From equation (2), the extended operation $\circ'$ is given by:

$\circ'$	a	b	c	ab	ac	bc	abc
a	a	a	a	a	a	a	a
b	a	b	a	ab	a	ab	ab
c	a	a	b	a	ab	ab	ab
ab	a	ab	a	ab	a	ab	ab
ac	a	a	ab	a	ab	ab	ab
bc	a	ab	ab	ab	ab	ab	ab
abc	a	ab	ab	ab	ab	ab	ab

In the above table, we have represented a subset by its members. For instance, ab stands for the subset $\{a, b\}$. It can be checked that the extended operation $\circ'$ is also commutative and associative. According to equations (3), the extended relations R_* and R^* are given by the following tables:

R_*	a	b	c	ab	ac	bc	abc
a	1	0	1	0	1	0	0
b	0	1	0	0	0	0	0
c	1	0	1	0	1	0	0
ab	0	0	0	0	0	0	0
ac	1	0	1	0	1	0	0
bc	0	0	0	0	0	0	0
abc	0	0	0	0	0	0	0

R^*	a	b	c	ab	ac	bc	abc
a	1	0	1	1	1	1	1
b	0	1	0	1	0	1	1
c	1	0	1	1	1	1	1
ab	1	1	1	1	1	1	1
ac	1	0	1	1	1	1	1
bc	1	1	1	1	1	1	1
abc	1	1	1	1	1	1	1

One can see that R_* is symmetric and transitive but not reflexive, whereas R^* is reflexive and symmetric but not transitive. In this example, if the extended operation and relations are restricted to the singleton subsets of U, the original operation and relation are obtained. □

Yiyu Yao and Noroozi
A unified model for set-based computations,
Proceedings of the Third International Workshop on Rough Sets and Soft Computing,
T.Y. Lin (Ed.), San Jose State University, pp. 236-243, 1994.

3 Examples of Set-based Computations

To illustrate the usefulness the proposed framework, this section analyzes briefly a few existing set-based computation approaches. It will be shown that all these models conform to the same basic structure.

3.1 Interval-number algebra

An *interval number* $[a_1, a_2]$ with $a_1 \le a_2$ is an interval of real numbers, where a_1 is referred to as the lower bound and a_2 the upper bound. It defines the following subset of real numbers:

$$[a_1, a_2] = \{x \mid a_1 \le x \le a_2\}. \tag{5}$$

The set of all interval numbers is denoted by $I(\Re)$. Degenerate intervals of the form $[a, a]$ are equivalent to real numbers.

According to Definition 1, arithmetic with interval numbers can be carried out through the arithmetic operations on their members. Let I and J be two interval numbers, and let $*$ denote an arithmetic operation $+$, $-$, $\times$ or $/$ on pairs of real numbers. Then an arithmetic operation $*$ may be extended to pairs of interval numbers I, J by

$$I * J = \{x * y \mid x \in I, y \in J\}. \tag{6}$$

The result $I * J$ is again a closed bounded interval unless $0 \in J$ and the operation $*$ is division (in which case, $I * J$ is undefined). In fact, they are defined by the following formulas:

$$\begin{aligned}
[a_1, a_2] + [b_1, b_2] &= [a_1 + b_1, a_2 + b_2], \\
[a_1, a_2] - [b_1, b_2] &= [a_1 - b_2, a_2 - b_1], \\
[a_1, a_2] \times [b_1, b_2] &= [\min(a_1b_1, a_1b_2, a_2b_1, a_2b_2), \\
&\qquad \max(a_1b_1, a_1b_2, a_2b_1, a_2b_2)], \\
[a_1, a_2] \;/\; [b_1, b_2] &= [a_1, a_2] \times [1/b_2, 1/b_1] \\
&\qquad \text{for } \; 0 \notin [b_1, b_2].
\end{aligned} \tag{7}$$

The arithmetic system defined above is called *interval arithmetic* or an *interval-number algebra*. Following the discussion of Section 2, we can see that many properties of normal arithmetic operations can be carried over to interval-number operations. For example, interval-number addition is commutative and associative.

By Definition 2, we can define two binary relations $\ge_*$ and $\ge^*$ on interval numbers using the binary relation $\ge$:

$$\begin{aligned}
I \ge_* J &\Longleftrightarrow \text{for all } x \in I \text{ and } y \in J, \quad x \ge y; \\
I \ge^* J &\Longleftrightarrow \text{there exist a } x \in I \\
&\qquad \text{and a } y \in J \text{ such that } x \ge y.
\end{aligned} \tag{8}$$

The relation $\ge_*$ is not reflexive but transitive, whereas $\ge^*$ is reflexive but not transitive. One may similarly define two relations $=_*$ and $=^*$ based on the equality relation $=$ on the real numbers.

3.2 Interval-set algebra

Let U be a finite nonempty set called the universe and 2^U its power set. Given two sets $A_1, A_2 \in 2^U$ with $A_1 \subseteq A_2$, the following subset of 2^U,

$$\begin{aligned}
\mathcal{A} &= [A_1, A_2] \\
&= \{X \in 2^U \mid A_1 \subseteq X \subseteq A_2\},
\end{aligned} \tag{9}$$

is called a closed *interval set*. The set A_1 is called the lower bound of the interval set and A_2 the upper bound. That is, an interval set is a set of subsets bounded by two particular elements of the Boolean algebra formed by the power set.

Let $\cap, \cup$ and $-$ denote set intersection, union and difference, respectively. According to Definition 1, we have the corresponding binary operations on interval sets. For two interval sets $\mathcal{A} = [A_1, A_2]$ and $\mathcal{B} = [B_1, B_2]$, the interval-set intersection, union and difference are defined as:

$$\begin{aligned}
\mathcal{A} \sqcap \mathcal{B} &= \{X \cap Y \mid X \in \mathcal{A}, Y \in \mathcal{B}\}, \\
\mathcal{A} \sqcup \mathcal{B} &= \{X \cup Y \mid X \in \mathcal{A}, Y \in \mathcal{B}\}, \\
\mathcal{A} \setminus \mathcal{B} &= \{X - Y \mid X \in \mathcal{A}, Y \in \mathcal{B}\}.
\end{aligned} \tag{10}$$

Let $I(2^U)$ denote the set of all closed interval sets. Then the above defined operations are closed on $I(2^U)$, namely, $\mathcal{A} \sqcap \mathcal{B}$, $\mathcal{A} \sqcup \mathcal{B}$ and $\mathcal{A} \setminus \mathcal{B}$ are interval sets. They can be explicitly computed by:

$$\begin{aligned}
\mathcal{A} \sqcap \mathcal{B} &= [A_1 \cap B_1, A_2 \cap B_2], \\
\mathcal{A} \sqcup \mathcal{B} &= [A_1 \cup B_1, A_2 \cup B_2], \\
\mathcal{A} \setminus \mathcal{B} &= [A_1 - B_2, A_2 - B_1].
\end{aligned} \tag{11}$$

The interval-set complement $\neg[A_1, A_2]$ of $[A_1, A_2]$ is defined by $[U, U] \setminus [A_1, A_2]$. This is equivalent to $[U - A_2, U - A_1] = [A_2^c, A_1^c]$, where $A^c = U - A$ denotes set complement of A. Clearly, $\neg[\emptyset, \emptyset] = [U, U]$ and $\neg[U, U] = [\emptyset, \emptyset]$. Degenerate interval sets of the form $[A, A]$ are equivalent to ordinary sets. For degenerate interval sets, the proposed operations $\sqcap$, $\sqcup$, $\setminus$, and $\neg$ reduce to the usual set-theoretic operations.

As discussed in Section 2, properties of interval-set operations can be derived from the properties of set operations. For instance, the idempotent, commutativity, associativity, distributivity, absorption, and De Morgan's laws hold for interval-set intersection and union. The double negation law holds for interval-set complement. However, for an interval set $\mathcal{A}$, $\mathcal{A} \sqcap \neg\mathcal{A}$ is not necessarily equal to $[\emptyset, \emptyset]$, $\mathcal{A} \sqcup \neg\mathcal{A}$ is not necessarily equal to $[U, U]$, and $\mathcal{A} \setminus \mathcal{A}$ is not necessarily equal to $[\emptyset, \emptyset]$. Therefore, $I(2^U)$ is a completely distributive lattice but not a Boolean algebra, whereas 2^U is a Boolean algebra [12].

From Definition 2, we can derive two binary relations $\supseteq_*$ and $\supseteq^*$ on interval sets:

$$\begin{aligned} \mathcal{A} \supseteq_* \mathcal{B} &\Longleftrightarrow \text{for all } X \in \mathcal{A} \text{ and } Y \in \mathcal{B}, \quad X \supseteq Y; \\ \mathcal{A} \supseteq^* \mathcal{B} &\Longleftrightarrow \text{there exist a } X \in \mathcal{A} \\ &\qquad \text{and a } Y \in \mathcal{B} \text{ such that } X \supseteq Y. \end{aligned} \tag{12}$$

Relation $\supseteq_*$ is not reflexive but transitive, whereas $\supseteq^*$ is reflexive but not transitive.

3.3 Interval-valued logic

Let L be a lattice with operations $\otimes$ and $\oplus$. If L is a Boolean lattice, the symbol $\sim$ denotes the complement operation. Given two elements $a_1, a_2 \in L$ with $a_1 \preceq a_2$, an interval $[a_1, a_2]$ is defined by the set:

$$[a_1, a_2] = \{x \in L \mid a_1 \preceq x \preceq a_2\}. \tag{13}$$

That is, $[a_1, a_2]$ consists of these elements of L that are bounded by a_1 and a_2. It is a sublattice of L. An element $a \in L$ may be represented as a degenerate interval of the form $[a, a]$.

Let $I(L)$ denote the set of all intervals formed from L. Using Definition 1, we may extend the operations $\otimes$, $\oplus$ and $\sim$ over to elements of $I(L)$ as follows:

$$\begin{aligned} [a_1, a_2] \otimes [b_1, b_2] &= \{x \otimes y \mid a_1 \preceq x \preceq a_2, \\ &\qquad b_1 \preceq y \preceq b_2\}, \\ [a_1, a_2] \oplus [b_1, b_2] &= \{x \oplus y \mid a_1 \preceq x \preceq a_2, \\ &\qquad b_1 \preceq y \preceq b_2\}, \\ \sim [a_1, a_2] &= \{\sim x \mid a_1 \preceq x \preceq a_2\}. \end{aligned} \tag{14}$$

For simplicity, the same set of symbols has been used for operations on both L and $I(L)$. The extended operations on $I(L)$ are closed and can be computed by:

$$\begin{aligned} [a_1, a_2] \otimes [b_1, b_2] &= [a_1 \otimes b_1, a_2 \otimes b_2], \\ [a_1, a_2] \oplus [b_1, b_2] &= [a_1 \oplus b_1, a_2 \oplus b_2], \\ \sim [a_1, a_2] &= [\sim a_2, \sim a_1]. \end{aligned} \tag{15}$$

The set $I(L)$ with the above operations forms a lattice. To differentiate it from the original lattice, we call $I(L)$ an interval lattice. Many properties of $\otimes$, $\oplus$ and $\sim$ on L are carried over by their corresponding operations on the interval lattice. For example, if L is a complete distributive lattice, then $I(L)$ is a complete distributive lattice. However, if L is a Boolean lattice, $I(L)$ is not a Boolean lattice but a complete distributive lattice.

Consider a multi-valued logic system in which the truth values are taken from a lattice [5]. We assume that well formed formulas are defined to be exactly the same as those in two-valued propositional logic. Let $v(\phi) \in L$ denote the truth value of a proposition or formula ϕ. The evaluation of logical conjunction, disjunction, and negation are defined as the greatest lower bound ($\otimes$), the least upper bound ($\oplus$), and the complement ($\sim$), respectively. They may be evaluated using the following rules:

$$\begin{aligned} v(\phi \wedge \psi) &= v(\phi) \otimes v(\psi), \\ v(\phi \vee \psi) &= v(\phi) \oplus v(\psi), \\ v(\neg\phi) &= \sim v(\phi), \\ v(\phi \rightarrow \psi) &= \sim v(\phi) \oplus v(\psi), \\ v(\phi \leftrightarrow \psi) &= (\sim v(\phi) \oplus v(\psi)) \otimes \\ &\qquad (v(\phi) \oplus \sim v(\psi)). \end{aligned} \tag{16}$$

The adoption of a lattice for the definition of a many-valued logic implies that the logical connectives $\wedge$, $\vee$, and $\neg$ must have the same properties as that of $\otimes$, $\oplus$, and $\sim$. For example, if a lattice L is a distributive lattice, the logical connectives must be distributive. Conversely, if logical connectives are distributive, one must choose a distributive lattice to represent the truth values.

In a interval-based logic system, we assume that an interval $[v_*(\phi), v^*(\phi)]$ may be assigned to the proposition to indicate the range within which lies the truth value. Any element between $v_*(\phi)$ and $v^*(\phi)$ may be the actual truth value. Within the framework of set-based computations introduced in Section 2, the following rules can be used for logical conjunction, disjunction, negation, implication and equivalence:

$$\begin{aligned} &[v_*(\phi \wedge \psi), v^*(\phi \wedge \psi)] \\ &\quad = [v_*(\phi) \otimes v_*(\psi), v^*(\phi) \otimes v^*(\psi)], \\ &[v_*(\phi \vee \psi), v^*(\phi \vee \psi)] \\ &\quad = [v_*(\phi) \oplus v_*(\psi), v^*(\phi) \oplus v^*(\psi)], \\ &[v_*(\neg\phi), v^*(\neg\phi)] \\ &\quad = [\sim v^*(\phi), \sim v_*(\phi)], \\ &[v_*(\phi \rightarrow \psi), v^*(\phi \rightarrow \psi)] \\ &\quad = [\sim v^*(\phi) \oplus v_*(\phi), \sim v_*(\phi) \oplus v^*(\phi)] \\ &[v_*(\phi \leftrightarrow \psi), v^*(\phi \leftrightarrow \psi)] \\ &\quad = [(\sim v^*(\phi) \oplus v_*(\psi)) \otimes (v_*(\phi) \oplus \sim v^*(\psi)), \\ &\qquad (\sim v_*(\phi) \oplus v^*(\psi)) \otimes (v^*(\phi) \oplus \sim v_*(\psi))]. \end{aligned} \tag{17}$$

These rules are extensions of rules defined by equation (16) that use a single element of L as the truth value of a proposition. The assignment of interval truth values suggests that the interval lattice $I(L)$, or a sublattice of $I(L)$, should be used. To reflect this property, we refer to such a logic as an interval-valued logic.

The interval-valued system examined above is related to a number of systems. If the lattice $([0,1], \max, \min)$ is used, we obtain the interval-valued fuzzy logic system investigated by Kenevan and Neapolitan [9]. All their inference rules have a counterpart in our framework. If three truth values T (true), F (false) and I (unknown or undermined) are used [18], one can draw the corresponding between such a three-valued logic and interval-valued logic [28]. This can be simply done by interpreting the truth value I as either an interval $[F, T]$ or as its equivalent set representation $\{F, T\}$, where it is assumed that $T \succ F$. If we take the lattice to be the Boolean algebra $(2^U, \cap, \cup, {}^c)$, interval-set algebra is immediately derived.

3.4 Set-based information systems

Following Lipski [10], Orlowska and Pawlak [13], Pawlak [15], and Vakarelov [24], we define a set-based information system to be a quadruple,

$$S = (O, A, \{V_a \mid a \in A\}, \{f_a \mid a \in A\}), \qquad (18)$$

where

O is a nonempty set of objects,

A is a nonempty set of attributes,

V_a is a nonempty set of values of $a \in A$,

$f_a : O \times A \longrightarrow 2^{V_a}$ is an information function.

If all information functions map an object only to singleton sets of attribute values, we obtain a degenerate set-based information system commonly used in the rough-set model [16]. The notion of information systems provides a convenient tool for the representation of objects in terms of their attribute values. By definition, a database system is an information system. More examples can be found in [15].

Set-based computations introduced in Section 2 can be easily applied in the set-based information systems. We use the following information system to demonstrate the main idea.

	AGE	HEIGHT	LANGUAGE
o_1	{35}	{tall}	{English, French}
o_2	[30, 35]	{medium}	{French}
o_3	{20}	[medium, tall]	{English}
o_4	{60, 61}	{short}	{English, French}
o_5	{54}	[short, tall]	{English}

The set-based representation in this example may arise in several ways. The available information may be insufficient to determine the exact value of an attribute. For instance, based on the given information, one may only infer that the age of o_2 is between 30 and 35. In the worst case, if one is totally ignorant of the value of an attribute a, the entire set V_a may be used to represent such an unknown value [8]. Any possible attribute value may in fact be the actual value of the attribute. The assignment of [short, tall] to the attribute HEIGHT of o_5 reflects such a situation. It is possible that an attribute takes a subset of V_a as its value. For example, o_1 speaks both English and French. An expert may feel that the prefixed grades in the system is not fine enough, and would rather use an interval formed by two adjacent values, say [medium, tall], as an additional value. From above discussion, it is obvious that the flexibility of set-based representation leads to a richer and more complicated semantics of set-based information systems. Applications of set-based computations depend on a well defined semantics of an information system.

Suppose all attributes take a single value and a set-based information system is used to represent uncertainty in specifying the actual value. One can immediately apply the extended relations introduced in Section 2 to carry out the retrieval process, one of the basic operations in information system. For an unordered set of attribute values, a valid operation is the comparison of equality. In the case, we have two possible retrieved sets, i.e., the sets Ret_* and Ret^* derived from $=_*$ and $=^*$, respectively. Consider a query

$$q_1 : \quad \text{LANGUAGE} \ = \ \text{English}.$$

It produces the following two sets:

$$Ret_*(q_1) = \{o_3, o_5\},$$
$$Ret^*(q_2) = \{o_1, o_3, o_4, o_5\}.$$

Elements of Ret_* definitely satisfy the query, whereas elements of $Ret^* - Ret_*$ may satisfy the query. The pair (Ret_*, Ret^*) defines an interval set $[Ret_*, Ret^*]$, indicating the range of the set of objects that actually satisfies the query. They may also be interpreted as lower and upper approximations in the context rough-set model. For an ordered set of attribute values, in addition to the comparison of equality, we may also make comparisons using the ordered relation. For a given ordered relation $\succeq$ defined on V_a, it induces two extended relation $\succeq_*$ and $\succeq^*$. Accordingly, two sets Ret_* and Ret^* will be produced. For example, for the query,

$$q_2: \quad \text{AGE} \ \geq \ 34,$$

we have:

$$Ret_*(q_2) = \{o_1, o_4, o_5\},$$
$$Ret^*(q_2) = \{o_1, o_2, o_4, o_5\}.$$

They are a pair of lower and upper approximations.

By combining queries q_1 and q_2, we obtain two composite queries:

$$\begin{aligned} q_3: \quad & q_1 \text{ and } q_2 \\ & (\text{LANGUAGE} = \text{English}) \text{ and } (\text{AGE} \geq 34), \\ q_4: \quad & q_1 \text{ or } q_2 \\ & (\text{LANGUAGE} = \text{English}) \text{ or } (\text{AGE} \geq 34). \end{aligned}$$

Using these queries, we derive the following two pairs of retrieved sets:

$$\begin{aligned} & Ret_*(q_1 \text{ and } q_2) = \{o_5\}, \\ & Ret^*(q_1 \text{ and } q_2) = \{o_1, o_4, o_5\}; \\ & Ret_*(q_1 \text{ or } q_2) = \{o_1, o_3, o_4, o_5\}, \\ & Ret^*(q_1 \text{ or } q_2) = \{o_1, o_2, o_3, o_4, o_5\}. \end{aligned}$$

Obviously, the following properties hold:

$$\begin{aligned} Ret_*(q_1 \text{ and } q_2) &= Ret_*(q_1) \cap Ret_*(q_1), \\ Ret^*(q_1 \text{ and } q_2) &= Ret^*(q_1) \cap Ret^*(q_1), \\ Ret_*(q_1 \text{ or } q_2) &= Ret_*(q_1) \cup Ret_*(q_1), \\ Ret^*(q_1 \text{ or } q_2) &= Ret^*(q_1) \cup Ret^*(q_1). \end{aligned} \quad (19)$$

They correspond to the operations of the interval-set algebra. These rules may be considered as a generalization of the rules used in database systems. They may be used in a retrieval process of a set-based information systems. However, it should be noted that these rule may not generated the tightest bounds.

The proposed two operations in a set-based information system are essentially the same as the modal operators proposed by Lipski in the study of incomplete databases [10]. If a different semantic interpretation of a set-based information system is used, one may introduce other types extended operations [10]. For example, Vakarelov [24] used set equality to define an indiscernibility relation, and $=^*$ to define similarity relations. Our analysis may be extended to the case where a query itself uses a nonsingleton set.

4 Conclusion

In this paper, we have presented and analyzed a framework for set-based computations. This framework is particularly useful in situations where it is difficult to obtain a precise value of certain parameter, or where set-valued attributes play an important role. To be consistent with point-valued computations, operations on set values are defined by carrying over the corresponding point-based operations. Based on this definition, properties of point-based operations and relations are extended to their set-based counterparts. For example, the commutativity and associativity of a point-based binary operations ensure that the set-based operation is also commutative and associative. For a point-based relation, we define a pair of set-based relations, which represent two extreme cases. Within the proposed framework, we have explicitly shown that a number of existing set-based computation methods, such as interval-number algebra, interval-set algebra, and interval-valued fuzzy reasoning, and set-valued information systems, are special cases.

A major part of the paper focuses on the re-examination of existing approaches in the context of the proposed framework. Such an integration of various models is a preliminary and an important step towards the establishment of a unified framework for a systematic study of set-based computations. The definition of the extended system suggested in this paper represents only one of many possible ways. It is entirely possible that a different definition may be more appropriate in different applications. For example, one can mix the quantifiers in Definition 2 to obtain additional extended relations. In fact, with respect to the interval-set operations, similar and alternative proposals have been suggested by Calabrese [3], and Goodman, Nguyen and Walker [7] in their study of conditional events. Wellman and Simmons [25] discussed more set operations in designing mechanisms for reasoning about sets. It is worthwhile to incoporpate the results of these studies into the proposed framework.

References

[1] Bonissone, P. Plausible reasoning. *Encyclopedia of Artificial Intelligence*, 854-863, 1987.

[2] Bundy, A. Incidence calculus: a mechanism for probabilistic reasoning. *Journal of Automated Reasoning*, **1**, 263-283, 1985.

[3] Calabrese, P. An algebraic synthesis of the foundations of logic and probability. *Information Sciences*, **42**, 187-237, 1987.

[4] Dubois, D. and Prade, H. *Possibility Theory: a Approach to the Computerized Processing of Uncertainty*, Plenum Press, New York, 1988.

[5] Edmonds, E. A. Lattice fuzzy logics. *International Journal of Man-Machine Studies*, **13**, 455-465, 1980.

[6] Gilula, M.M. *The Set Model for Database and Information Systems*, Addison-Wesley, New York, 1994.

[7] Goodman, I.R., Nguyen, H.T., and Walker, E.A. *Conditional Inference and Logic for Intelligent Systems: A Theory of Measure-free Conditioning*, North-Holland, 1991.

[8] Grzymala-Busse, J.W. On the unknown attribute values in learning from examples. *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems.* Lecture Notes in Artificial Intelligence, vol. 542, 368-377, 1991.

[9] Kenevan, J.R. and Neapolitan, R.E. A model theoretic approach to propositional fuzzy logic using Beth tableaux. In L. Zadeh and J. Kacprzyk (Eds.), *Fuzzy Logic for the Management of Uncertainty*, Wiley, New York, 141-157, 1992.

[10] Lipski, W. Jr. On databases with incomplete information. *Journal of the ACM*, **28**, 41-70, 1981.

[11] Moore, R.E. *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[12] Negoiţă, C.V. and Ralescu, D.A. *Applications of Fuzzy Sets to Systems Analysis*, Basel, Birkhäuser Verlag, 1975.

[13] Orlowska, E. and Pawlak, Z. Logical foundations of knowledge representation. *ICS PAS Reports*, **537**, 1-108, 1984.

[14] Parsons, S. Interval algebra and order of magnitude reasoning. Artificial Intelligence in Engineering, 945-961, 1992.

[15] Pawlak, Z. Information systems – theoretical foundations. *Information Systems*, **6**, 205-218, 1981.

[16] Pawlak, Z. Rough sets. *International Journal of Computer and Information Sciences*, **11**, 341-356, 1982.

[17] Quinlan, J.R. Inferno: a cautious approach to uncertain inference. *The Computer Journal*, **26**, 255-269, 1983.

[18] Rescher, N. *Many-valued Logic*, McGraw-Hill, New York, 1969.

[19] Resconi, G., Klir, G.J., Clair, U.S., and Harmanec, D. On the integration of uncertainty theories. *International Journal of Uncertainty, Fuzziness and Knowledge-based systems.* **1**, 1-18, 1993.

[20] Roberts, F.S. *Measurement Theory*, Addison-Wesley, Reading, Massachusetts, 1979.

[21] Ruspini, E.H. Approximate reasoning: past, present, future. *Information Sciences*, **57-58**, 297-313, 1991.

[22] Shafer, G. Perspectives on the theory and practice of belief functions. *International Journal of Approximate Reasoning*, **4**, 323-362, 1990.

[23] Smets, P. Belief functions (with discussions). In P. Smets, A. Mamdani, D. Dubois, H. Prade (Eds.), *Non-standard Logics for Automated Reasoning*, Academic Press, New York, 253-285, 1988.

[24] D. Vakarelov, A modal logic for similarity relations in Pawlak knowledge representation systems. *Fundamenta Informaticae*, **XV**, 61-79, 1991.

[25] Wellman, M. and Simmons, R.G. Mechanisms for reasoning about sets. *Proceedings of the 7th AAAI National Conference on Artificial intelligence*, 398-402.

[26] Wong, S.K.M., Wang, L.S., and Yao, Y.Y. Interval structures: a framework for representing uncertain information. *Proceedings of Eighth Conference on Uncertainty in Artificial Intelligence*, 336-343, 1992.

[27] Yao, Y.Y. Interval-set algebra for qualitative knowledge representation. *Proceedings of the Fifth International Conference on Computing and Information*, 370-374, 1993.

[28] Y.Y. Yao, and X. Li, Uncertain reasoning with interval-set algebra. *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, First Edition, 191-202, 1993.

[29] Zadeh, L.A. Fuzzy sets. *Information and Control*, **8**, 338-353, 1965.