COMPARISON RELATIONS FOR INTERVALS: A FRAMEWORK

DOMINIQUE LOHEZ

Abstract. It is proposed to define the set of interval comparisons for P1788 as a Binary Relation Algebra(BRA). This BRA is built from some basic comparison relations named the generators. These generators provide an accurate mathematical description of the interval features on which the comparisons are grounded. As an illustration such a BRA is presented. Since the number of comparisons defined that way may be very high ($\sim 2^{10}$) , it is impossible to enumerate and give a name to each comparison. It is thus suggested that a new type of data named `IntervalComparison` should be defined in 1788. A set of specific operations for `IntervalComparison` is proposed.

This paper has been written for the IEEE working group P1788 on the standardization of interval arithmetic with improvements suggested by the remarks of Nate HAYES, Vincent LEFEVRE, John PRYCE and Jürgen WOLFF von GUDENBERG.

1. Introduction

It is proposed to define the set of interval comparisons as a *B*inary *R*elation *A*lgebra(BRA) [6]. This BRA is built from some basic comparison relations named the BRA *generators*. The generators provide an accurate mathematical description of interval features on which the comparisons are grounded.

The rest of this paper is as follows. In Section 2 a rationale of the position is presented. In Section 3, a brief summary of the notions of BRA is provided. Then a sketch of a complete BRA generation is given on the example of the BRA of real numbers comparisons(Section 4). Finally as a conclusion a motion defining a framework for interval comparisons in P1788 Standard is proposed (Section 5)

2. Rationale

Defining the interval comparison relations for p1788, attention is first paid[5][2][4] to choose the most useful comparisons (the *basic comparisons*). It must be noticed that from such basic comparisons, a lot of other comparisons can be derived using the Boolean operators. However it not an easy task to determine if a given comparison can be calculated from the basic comparisons. It is even more difficult to find the most efficient expression. Thus the user must be provided with i) an easy way to determine if his favorite Interval Comparison is derivable ii) an easy way to specify such a comparison iii) an easy way to check if a comparison condition hold for a given pair of intervals. To achieve these purposes it is necessary to study the algebraic structure of the set of derivable comparisons. It is found that the BRA structure is appropriate. Thus the Interval comparison interface provided to the user can defined (Section 5) , even while the precise interval comparison BRA is not yet chosen.

When the the definition of the interval set or the choice of the basic comparisons is varied, completely different BRA can be generated. Besides a common pool of interval properties, some specific interval properties are included in each BRA. Thus a careful discussion is required. In addition no existing result in the literature [1] [3] [4] is directly compatible with the interval set definition of motion 3 [7]. Thus

TABLE 1. Operations over Binary Relations

Notation	Set definition	Logical definition
$T = R^c$	$(a,\, b) \in T \Leftrightarrow (b,\, a) \in R$	$t(x,\, y) = r(y,\, x)$
$T = \neg R$	$(a,\, b) \in T \Leftrightarrow (a,\, b) \notin R$	$t(x,\, y) = \neg r(x,\, y)$
$T = P \cap Q$	$(a,\, b) \in T \Leftrightarrow (a,\, b) \in P \wedge (a,\, b) \in Q$	$t(x,\, y) = p(x,\, y) \wedge q(x,\, y)$
$T = P \cup Q$	$(a,\, b) \in T \Leftrightarrow (a,\, b) \in P \vee (a,\, b) \in Q$	$t(x, y) = p(x,\, y) \vee q(x,\, y)$
$T = P \odot Q$	$(a,\, b) \in T \Leftrightarrow \exists z\, (a,\, z) \in P \wedge (z,\, b) \in Q$	$t(x,\, y) = \exists z\, p(x,\, z) \wedge q(z,\, y)$

a cautious adaptation is required. Int that context, it is wise to delay the choice of the BRA to a future motion devoted to this choice.

Level 2 and lower levels considerations [4] are very important to the efficiency of the implementation. However they cannot be usefully addressed at the present time. They have to be delayed to a future motion.

With the available algorithms the user knows during the design of his program what comparisons are required. Thus comparison handling could be restricted to preprocessor. This approach may appear as very interesting. It may cause a significant improvement of the execution time. However this suggestion raises some difficulties. A standard like 1788 is not oriented to any particular language and a preprocessor is not available in all languages. Thus the wording of the motion in Section 5 suggests a dynamical access to interval comparisons. As a side effect new kinds of algorithms are allowed.

3. Binary relations and comparisons

In this section some basic properties of BRA are reminded and the utility of this notion to handle comparison operations is discussed. Given a set S, a binary relation R over S is a set of ordered pairs of S. The relation R can be written:

$$R = \{(a,\, b) \,|\, a \in S \wedge b \in S \wedge r(a,\, b)\} \tag{1}$$

From Eqn. 1, it appears that a binary predicate $r(x,\, y)$ is associated to any binary relation R. This predicate can be used to compare elements of S. To emphasize that correspondence, the same name is used for the relation and the predicate. The capital initial is used in the relation name, while the initial is never written in capital in the name of the predicate.

In any non-empty set, there is a generic relation $\mathbf{1}'$ defined by $\mathbf{1}' = \{(x,x)|x \in S\}$. This relation is often named the *IdentityRelation*. It can also be viewed the *Equality* of the elements of S since an alternative definition of $\mathbf{1}'$ is $\mathbf{1}' = \{(x,y)|x \in S \wedge y \in S \wedge x = y\}$.

From a relation R a one can deduce its *converse* R^c and its *complement* $\neg R$. Similarly two relations P and Q can be combined with the *meet*, *join* and *composition* operators to produce respectively $P \cap Q$, $P \cup Q$ and $P \odot Q$ relations. The definitions of these new relations are provided in Table 1.

The operations *meet* and *join* are associative, commutative and mutually distributive. The *composition* is associative and distributive over *join*. It is neither commutative nor distributive over *meet*.

It can be checked that a large variety of comparisons can be generated that way. In fact the only restriction is that the corresponding logical expression involves at most three different variables. Except for the *composition*, these operations introduce the usual combinations of Boolean conditions and comparisons.

A Binary Relation Algebra over S is a set of binary relations over S containing at least $\mathbf{1}'$ and stable under the operations of Table 1. It can be shown that the

Empty Relation $\mathbf{0}$ is always present in a BRA over S. This relation is such that for any relation R of the BRA the condition $R \cap \mathbf{0} = \mathbf{0}$ holds. Thus no pair of elements of S is contained in $\mathbf{0}$. The main use of the empty relation is to check that two relations P and Q are disjointed. The condition to check is simply $P \cap Q = \mathbf{0}$.

It can be shown that the *ComparabilityRelation* $\mathbf{1}$ is always present in a BRA over S. This relation is such that for any relation R of the BRA the condition $R \cup \mathbf{1} = \mathbf{1}$ holds. Thus $\mathbf{1}$ contains all the pairs of elements of S which can be compared with some relation of the BRA. It can be easily shown that $\mathbf{1}$ is an equivalence relation. When the complement is defined as in Table 1 , the relation $\mathbf{1}$ is necessarily $S \times S$t he S-Universal Relation, that is the set of the pairs of elements of S. The notion of BRA can be extended to accommodate the situation where not all the elements of S can be compared. The complement $\neg R$ of some relation R is then required to satisfy the conditions $R \cup \neg R = \mathbf{1}$ and $R \cap \neg R = \mathbf{0}$. All the others BRA properties are preserved. Applying this to some partial order relation *PartialLessEqual*, the comparability relation is found to be $\mathbf{1} = PartialLessEqual \cup PartialLessEqual^c$. The complement of the *PartialLessEqual* relation is the *PartialGreater* relation. This result is similar to what would be found for a total order relation. Of course such a result would not have been found if the definition of complement of Table 1 had been used.

The presence of the equality in any BRA reinforces the interest of BRA as an algebraic structure for comparison sets.

Finally there are some *atomic* relations $A_1,\ A_2 \dots A_N$ such that i) $A_i \cap A_j = \mathbf{0}$ for any i and j. ii) any relation R in the BRA can be written *in an unique way* as the *join* of some A_i. This expression is called *the atomic decomposition* of R. It follows from these observations that a BRA is completely defined by i) the knowledge of the A_i ii) the specifications of the atomic decomposition of the relations $A_i \odot A_j$ for any i and j. Since for each relation R, a given atom A_i must be either present or absent in the atomic decomposition of R, the number of relations in a BRA is 2^N, where N is the number of atoms in the BRA.

In the next section, it is shown as an illustration how the usual comparisons between real numbers can be derived as a member of appropriately defined BRA.

4. The real numbers comparisons from a BRA point of view

The comparisons over $\mathbb{R}$ are defined from the BRA built from a single generator which is the binary relation $\leq$. This relation is well known to be an order relation for $\mathbb{R}$. The order is total, unbounded, dense. In addition the order set $(\mathbb{R}, \leq)$ is a lattice. In the following, we use a literal notation *LessEqual* instead of $\leq$ to emphasize the relation character and avoid strange formulas like $= \Leftrightarrow \leq \cap \geq$. In this section the *ComparabilityRelation* $\mathbf{1}$ refers to the $\mathbb{R}$-Universal relation. The derivation is summarized in Table 2

In this particular case, the use of *composition* is not needed. This is generally not true. Since the BRA is defined by 3 atomic relations, the total number of relations in the BRA is $2^3 = 8$. In this count, both the universal relation $\mathbf{1}$ and the empty relation $\mathbf{0}$ are included.

To achieve the description of the BRA , a composition table of the atomic relations is provided in Table 3 .

It must be noticed that the following properties of the *LessEqual* relation can be retrieved from the results of this table: i) *LessEqual* is an order relation. ii) the order is total (since $\mathbf{1} \Leftrightarrow Equal \cup LessThan \cup GreaterThan$) iii) the order is dense (since $LessThan \odot LessThan \Leftrightarrow LessThan$) iv) for each pair of real numbers there are a lower bound and a upper bound(this arises from the statements $LessThan \odot GreaterThan \Leftrightarrow \mathbf{1}$ and $GreaterThan \odot LessThan \Leftrightarrow \mathbf{1}$).

TABLE 2. Derivation of the Real comparisons BRA from the unique generator *LessEqual*. The derivations of the atomic relations *LessThan* , *Equal* and *GreaterThan* is shown in steps 1-5. In steps 6-9, it is checked that these relations are atomic. The completude of the set of atomic relations is verified at step 9. Atomic decomposition of non-atomic relations are given in steps 9 -12

Step Number	Calculated relation	Expression Used
1	*GreaterEqual*	$GreaterEqual \Leftrightarrow LessEqual^c$
2	$Equal(\equiv \mathbf{1}')$	$Equal \Leftrightarrow LessEqual \cap GreaterEqual$
3	*UnEqual*	$UnEqual \Leftrightarrow \neg Equal$
4	*LessThan*	$LessThan \Leftrightarrow LessEqual \cap UnEqual$
5	*GreaterThan*	$GreaterThan \Leftrightarrow GreaterEqual \cap UnEqual$
6	**0**	$\mathbf{0} \Leftrightarrow LessThan \cap Equal$
7	**0**	$\mathbf{0} \Leftrightarrow GreaterThan \cap Equal$
8	**0**	$\mathbf{0} \Leftrightarrow LessThan \cap GreaterThan$
9	**1**	$\mathbf{1} \Leftrightarrow LessThan \cup Equal \cup GreaterThan$
10	*LessEqual*	$LessEqual \Leftrightarrow LessThan \cup Equal$
11	*GreaterEqual*	$GreaterEqual \Leftrightarrow GreaterThan \cup Equal$
12	*UnEqual*	$UnEqual \Leftrightarrow LessThan \cup GreaterThan$

TABLE 3. Table of compositions of atomic relation for The BRA of real numbers comparison. The results are developed as an appropriate join of atomic relations.

$\odot$	*Equal*	*LessThan*	*GreaterThan*
Equal	*Equal*	*LessThan*	*GreaterThan*
LessThan	*LessThan*	*LessThan*	$Equal \cup LessThan \cup GreaterThan$
GreaterThan	*GreaterThan*	$Equal \cup LessThan \cup GreaterThan$	*GreaterThan*

Some properties cannot be expressed in this framework i) the order relation is unbounded (this arises because the BRA generated by the *LessEqual* relation would admit the same atomic composition table whether the set is $\mathbb{R}$ or $\mathbb{R}^*$. ii) The lattice property cannot be modeled within a BRA because the corresponding logical formulas necessarily involve **4** distinct variables .

Despite these limits, the use of a BRA for the set of comparisons over a set S allows to accurately state many mathematical properties on which the choice is grounded on.

As long as the number of atomic relations N is very low, the number of relations in the BRA (2^N) remains small enough so all the comparisons can be enumerated and named. This is the case for the real numbers comparisons where $N = 3$ leading to $2^N = 8$. Unfortunately it is not expected to be true with interval comparisons, since the known interval BRA [1], [4] involves 13 or even 26 atomic comparisons [3]. So that enumerating and naming all the comparisons become impossible. If all the comparisons in the BRA have to be available to the user, it is necessary then to define a new type of data so that the user may handle the comparisons he finds appropriate to his problem.

5. The motion

P1788 defines a new type type of data named `IntervalComparison.` The set of data is structured as a Binary Relation Algebra (BRA) [6]. So algebraic manipulations are possible with `IntervalComparison`.

The choice of the `IntervalComparison` BRA is left to a future motion.

5.1. Interval Comparison predefined constants.

In P1788 the following predefined `IntervalComparison` constants must be provided

- The `IntervalComparabilityRelation`
- The `EmptyRelation`
- The `IntervalEquality`
- The generators of the BRA
- All the atomic relations of the BRA

Other predefined `IntervalComparison` constants may be provided. These additional constants must be members of `IntervalComparison` BRA. From each additional constant the atomic decomposition must be provided in the documentation.

5.2. Interval Comparison Handling functions.

The following handling functions must be provided

comparisonConverse:
- `IntervalComparison`$\mapsto$`IntervalComparison`
- $comparisonConverse(A) = A^c$

comparisonComplement:
- `IntervalComparison`$\mapsto$`IntervalComparison`
- $comparisonComplement(A) = \neg A$

comparisonJoin:
- `IntervalComparison`$\times$`IntervalComparison`$\mapsto$`IntervalComparison`
- $comparisonJoin(A,\, B) = A \cup B$

comparisonMeet:
- `IntervalComparison`$\times$`IntervalComparison`$\mapsto$`IntervalComparison`
- $comparisonMeet(A,\, B) = A \cap B$

comparisonComposition:
- `IntervalComparison`$\times$`IntervalComparison`$\mapsto$`IntervalComparison`
- $comparisonComposition(A,\, B) = A \odot B$
- The composition must be calculated from the abstract BRA definition given by the atomic composition table. It must not be calculated using computer interval calculations.

comparisonEquality:
- `IntervalComparison`$\times$`IntervalComparison`$\mapsto$`Boolean`
- $comparisonEquality(A,\, B) = true$ if A and B admit the same atomic decomposition. A and B may refer to different computer objects.
- $comparisonEquality(A,\, B) = false$ otherwise

comparisonInclusion:
- `IntervalComparison`$\times$`IntervalComparison`$\mapsto$`Boolean`
- $comparisonInclusion(A,\, B) = true$ if all the atomic relations present in the atomic decomposition of A are also present in the atomic decomposition of B
- $comparisonInclusion(A,\, B) = false$ otherwise

The possibility of providing these functions as operators is left to the implementation.

5.3. **Interval Comparison Use Functions.** The following IntervalComparison use functions must be provided:

comparisonApplication:

- `IntervalComparison`$\times$`Interval` $\times$`Interval`$\mapsto$`Boolean`
- $comparisonApplication(R, A, B) = true$ if $(A, B) \in R$
- $comparisonApplication(R, A, B) = false$ otherwise

comparisonExtraction:

- `Interval` $\times$`Interval`$\mapsto$`IntervalComparison`
- $comparisonExtraction(A, B) = R$ where R is the unique atomic relation such that $(A, B) \in R$

These definitions are purely conceptual (Level 1). The translation to lower levels is left to a future motion.

References

1. James F. ALLEN and Patrick J. HAYES, *A common-sense theory of time*, Proceedings to the International Joint Conference on Artificial Intelligence, 1985, pp. 528–531.
2. Bo EINARSON, *Comparison relations*, Available at http://grouper.ieee.org/groups/1788/, Submitted as Motion 13.02 to the IEEE P1788 working group on Interval arithmetic Standardization, Seconded By D. ZURAS.
3. Itay MEIRI, *Combining qualitative and quantitative constraints in temporal reasoning*, Artificial Intelligence **87** (1996), no. 1-2, 343–385.
4. Marco NEHMEIER and Jürgen WOLFF von GUDENBERG, *The interval overlapping relation as basic tool for defining interval comparisons*, Available at http://grouper.ieee.org/groups/1788/, Submitted to the IEEE P1788 working group on Interval arithmetic Standardization.
5. Arnold NEUMAIER, *Vienna proposal-final version*, Available at http://grouper.ieee.org/groups/1788/, Submitted as a position paper to the IEEE P1788 working group on Interval arithmetic Standardization.
6. Alfred TARSKI and Steven GIVANT, *A formalization of set theory without variables*, Colloquium publications, no. 41, American Mathematical Society, 1987.
7. Jürgen WOLFF von GUDENBERG, *Set of reals*, Available at http://grouper.ieee.org/groups/1788/, Submitted as Motion 3 to the IEEE P1788 working group on Interval arithmetic Standardization, Seconded By M. HACK.

Current address: Institut Supérieur de l'Electronique et du Numérique de Lille (ISEN-Lille), 41, Bd Vauban, 59046 LILLE CEDEX, France