Parallel Detection of Interval Overlapping

Marco Nehmeier, Stefan Siegel, and Jürgen Wolff von Gudenberg

Institute of Computer Science, University of Würzburg
Am Hubland, D 97074 Würzburg, Germany
{nehmeier,st.siegel,wolff}@informatik.uni-wuerzburg.de

Abstract. In this paper we define the interval overlapping relation and develop a parallel hardware unit for its realization. As one application we consider the interval comparisons. It is shown that a detailed classification of the interval overlapping relation leads to a reduction of floating-point comparisons in common applications.

Keywords: interval arithmetic, interval relations, hardware unit

1 Introduction

Detection of overlapping intervals is a problem that occurs in many application areas. The detection of overlapping boxes in computer graphics, overlapping time slots in scheduling problems, containment or membership tests, or enclosure tests in self-verifying scientific computing algorithms are some examples. In most of the applications, like in scheduling, the information whether 2 intervals overlap or not is not sufficient, but we also like to know how they overlap: completely contained in the interior vs. touching one bound, e.g. In this paper we, hence, define a general relation that describes the kind of overlapping between two one-dimensional intervals, and develop a hardware unit for its evaluation. As our intended first application of this unit we discuss the comparison relations in interval arithmetic.

An interval is a connected, closed, not necessarily bounded subset of the reals. It can be represented by its two bounds.

$$X := [\underline{x}\,, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\} \tag{1}$$

In this definition $\underline{x}$ can be $-\infty$, $\overline{x}$ can be $+\infty$, but the infinities never are members of an interval. The set of all intervals including the empty set is denoted as $\overline{\mathbb{IR}}$.

Preprint. The final publication is available at link.springer.com
http://dx.doi.org/10.1007/978-3-642-28145-7_13

2 Definition

The interval overlapping relation is not a boolean relation but delivers 14 different states describing all the possible situations that occur when the relative positions of 2 intervals are regarded with respect to overlapping. Table 1 illustrates the meaning of the relation. Each row represents a different state. The columns 2 through 5 contain sketches of the scene with the interval A at the bottom and B on the top. Singleton or point intervals are denoted as dots where appropriate. As usual numbers grow from left to right. Let Q be the set of the 13 cases for non-empty intervals [1] listed in Tab. 1.

Definition 1 (Interval Overlapping). *The overlapping relation for two non-empty intervals is defined by the mapping*

$$\text{⚭}_1 : (\overline{\mathbb{IR}} \setminus \emptyset) \times (\overline{\mathbb{IR}} \setminus \emptyset) \to Q \tag{2}$$

$$A \,\text{⚭}_1\, B \mapsto q_i \in Q \;, \quad i = 1 \ldots 13 \tag{3}$$

State 14, not in the table, characterises that one of the operands is empty. Then there is no overlapping at all.

Remark 1. Note that all possible situations are considered.

We represent the states by 4-bit strings resulting from specific comparisons of the bounds of the input intervals.

Definition 2 (Interval Overlapping Representation). *The interval overlapping relation*

$$\text{⚭} : (\overline{\mathbb{IR}} \setminus \emptyset) \times (\overline{\mathbb{IR}} \setminus \emptyset) \to \{0,1\}^4 \tag{4}$$

$$A \,\text{⚭}\, B \mapsto (r_1, r_2, r_3, r_4) \tag{5}$$

for two non-empty intervals $A = [\underline{a}\,,\,\overline{a}], B = [\underline{b}\,,\,\overline{b}] \in \overline{\mathbb{IR}} \setminus \emptyset$ *is defined by:*

$$\begin{aligned} r_1 := \;& ((\underline{a} \neq \underline{b}) \oplus (((\overline{a} \leq \underline{b}) \vee (\underline{a} > \overline{b})) \\ & \wedge ((\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b})))) \end{aligned} \tag{6}$$

$$\begin{aligned} r_2 := \;& ((\overline{a} \neq \overline{b}) \oplus (((\overline{a} < \underline{b}) \vee (\underline{a} \geq \overline{b})) \\ & \wedge ((\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b})))) \end{aligned} \tag{7}$$

$$r_3 := (\underline{a} \geq \underline{b}) \tag{8}$$

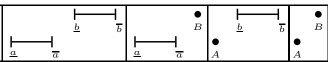
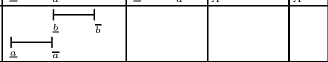
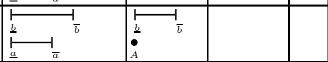
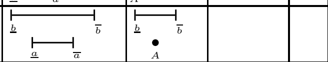
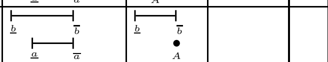
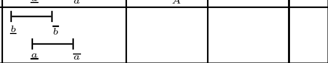
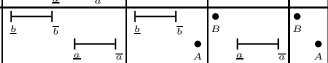
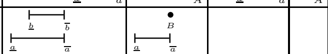
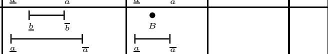
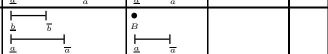
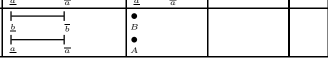
$$r_4 := (\overline{a} \leq \overline{b}) \tag{9}$$

The function ⚭ can be written as a composition of

$$\text{⚭} = \xi \circ \text{⚭}_1$$

where $\xi : Q \to \{0,1\}^4$ maps the state into a representation as defined in Tab. 1 columns 1 and 6. The 6^{th} column of the table shows the state of overlapping coded into 4 bits.

Table 1. The 13 different cases of the interval overlapping relation for non-empty intervals

$A \Cap_1 B$					$A \Cap B$	$A \subseteq B$	$A \supseteq B$	$A = B$	$A \cap B = \emptyset$
q_1					0001				•
q_2					0101				
q_3					1101				
q_4					0111	•			
q_5					1111	•			
q_6					1011	•			
q_7					1110				
q_8					1010				
q_9					0010				•
q_{10}					1001		•		
q_{11}					1100		•		
q_{12}					0110		•		
q_{13}					0011	•	•	•	

In principle, each result bit refers to one comparison of the bounds.

$$\begin{aligned}
r_1 &:= (\underline{a} \neq \underline{b}) \\
r_2 &:= (\overline{a} \neq \overline{b}) \\
r_3 &:= (\underline{a} \geq \underline{b}) \\
r_4 &:= (\overline{a} \leq \overline{b})
\end{aligned}$$

With these simple definitions we could not separate the states q_1, q_2, q_3 or q_7, q_8, q_9, respectively. Therefore we developed the comparisons (6) and (7).

Corollary 1. *For two non-empty intervals* $A = [\underline{a}\,, \overline{a}], B = [\underline{b}\,, \overline{b}] \in \overline{\mathbb{IR}} \setminus \emptyset$ *the states with bitset "0000", "0100" or "1000" do not occur as a result of the interval overlapping relation* $A \mathbin{\circ\!\!\circ} B$.

Proof.

$$\begin{aligned}
& (A \neq \emptyset) \wedge (B \neq \emptyset) \wedge (\neg r_3 \wedge \neg r_4) \\
\overset{(8,\,9)}{\Rightarrow}\; & (\underline{a} < \underline{b}) \wedge (\overline{a} > \overline{b}) \\
\Rightarrow\; & (\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b}) \wedge (\overline{a} > \underline{b}) \wedge (\underline{a} < \overline{b}) \\
\overset{(6,\,7)}{\Rightarrow}\; & r_1 \wedge r_2
\end{aligned}$$

□

3 Comparisons in Interval Arithmetic

Interval arithmetic is currently being standardized. Our definition of intervals as connected, closed, not necessarily bounded subsets of the reals in section 1 follows the presumable standard P1788 [3]. Various competing sets of interval comparisons are under discussion. Nearly every combination of operator symbol and quantifier is proposed in the Vienna proposal [7]. That includes the so called "certainly" and "possibly" operations where the relation holds for some or all members of an interval, respectively. A smaller set of comparisons is given in the book [5].

There is, however, consensus that the subset-relation, either interior or proper or equal, the membership of a point in an interval, and the test for disjointness are mandatory.

In the following propositions we show that all these comparisons can easily be obtained from the interval overlapping relation.

Proposition 1 (Set Relations). *For two non-empty intervals* $A = [\underline{a}\,, \overline{a}]$, $B = [\underline{b}\,, \overline{b}] \in \overline{\mathbb{IR}} \setminus \emptyset$ *the three relations* $=, \subseteq$ *and* $\supseteq$ *as well the test for disjointness* *are implied by the interval overlapping relation $A \circledcirc B$ as follows:*

$$A \subseteq B \Leftrightarrow r_3 \wedge r_4 \tag{10}$$
$$A \supseteq B \Leftrightarrow (r_1 \oplus r_3) \wedge (r_2 \oplus r_4) \tag{11}$$
$$A = B \Leftrightarrow \neg r_1 \wedge \neg r_2 \wedge r_3 \wedge r_4 \tag{12}$$
$$A \cap B = \emptyset \Leftrightarrow \neg r_1 \wedge \neg r_2 \wedge \neg(r_3 \wedge r_4) \tag{13}$$

Proof.
(10):

$$\begin{aligned} A \subseteq B &\Leftrightarrow (\underline{b} \leq \underline{a}) \wedge (\overline{a} \leq \overline{b}) \\ &\overset{(8,9)}{\Leftrightarrow} r_3 \wedge r_4 \end{aligned}$$

(11):

$$\begin{aligned} A \supseteq B &\Leftrightarrow (\underline{a} \leq \underline{b}) \wedge (\overline{b} \leq \overline{a}) \\ &\Leftrightarrow \neg(\underline{a} > \underline{b}) \wedge \neg(\overline{b} > \overline{a}) \\ &\Leftrightarrow ((\neg(\underline{a} \geq \underline{b}) \wedge (\underline{a} \neq \underline{b})) \vee ((\underline{a} \geq \underline{b}) \wedge \neg(\underline{a} \neq \underline{b}))) \\ &\quad \wedge ((\neg(\overline{a} \leq \overline{b}) \wedge (\overline{a} \neq \overline{b})) \vee ((\overline{a} \leq \overline{b}) \wedge \neg(\overline{a} \neq \overline{b}))) \\ &\overset{\text{Def. 2}}{\Leftrightarrow} (r_1 \oplus r_3) \wedge (r_2 \oplus r_4) \end{aligned}$$

(12):

$$\begin{aligned} A = B &\Leftrightarrow (A \subseteq B) \wedge (A \supseteq B) \\ &\overset{(10,11)}{\Leftrightarrow} r_3 \wedge r_4 \wedge (r_1 \oplus r_3) \wedge (r_2 \oplus r_4) \\ &\Leftrightarrow \neg r_1 \wedge \neg r_2 \wedge r_3 \wedge r_4 \end{aligned}$$

(13):

$$\begin{aligned} A \cap B = \emptyset &\Leftrightarrow (\overline{a} < \underline{b}) \vee (\underline{a} > \overline{b}) \\ &\Leftrightarrow (((\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b})) \wedge (\overline{a} \leq \underline{b}) \wedge (\overline{a} < \underline{b}) \\ &\quad \wedge \neg(\underline{a} \geq \underline{b}) \wedge (\overline{a} \leq \overline{b})) \\ &\quad \vee (((\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b})) \wedge (\underline{a} \geq \overline{b}) \wedge (\underline{a} > \overline{b}) \\ &\quad \wedge (\underline{a} \geq \underline{b}) \wedge \neg(\overline{a} \leq \overline{b})) \\ &\overset{\text{Def. 2}}{\Leftrightarrow} \neg r_1 \wedge \neg r_2 \wedge (r_3 \oplus r_4) \end{aligned}$$

□

Besides the formal proofs given in this section the formulas can be verified with the help of Tab. 1.

Corollary 2 (Set Membership). *The set membership* $a \in B$ *with* $a \in \mathbb{R}$, $B \in \overline{\mathbb{IR}} \setminus \emptyset$ *can be deduced to*

$$[a\,,\,a] \circledcirc B = (r_1, r_2, 1, 1) \tag{14}$$

with $r_1, r_2 \in \{0, 1\}$.

Proof.

$$a \in B \Leftrightarrow\ [a\,,\,a] \subseteq B \overset{(10)}{\Leftrightarrow}\ [a\,,\,a] \circledcirc B = (r_1, r_2, 1, 1)$$

□

Up to now we only have considered non-empty intervals. In many realizations of interval arithmetic the empty interval is represented as a pair of `NaN`s.

$$\emptyset := [\mathtt{NaN}\,,\,\mathtt{NaN}] \tag{15}$$

Corollary 3 (Empty Interval). *For two intervals* $A, B \in \overline{\mathbb{IR}}$ *where at least one of them is empty, the following equation*

$$A \circledcirc B = (0, 0, 0, 0) \tag{16}$$

holds if the empty interval is represented by two `NaN`*s.*

Proof. As defined in the IEEE standard for floating-point arithmetic [4], comparisons to `NaN` always return `false`. □

Hence, state 14 happens to be “0000” which was an unused bit combination so far.

Remark 2. With this definition of the empty set we can omit the assumption of non-empty intervals in Prop. 1 and Cor. 2.

Proposition 2 (Order Relations). *The order relations* $A \leq B, A \prec B$, $A \geq B$ *and* $A \succ B$ *with* $A = [\underline{a}\,,\,\overline{a}], B = [\underline{b}\,,\,\overline{b}] \in \overline{\mathbb{IR}} \setminus \emptyset$ *follow from the interval overlapping relation* $A \circledcirc B$ *by*

$$A \leq B :\Leftrightarrow (\underline{a} \leq \underline{b}) \wedge (\overline{a} \leq \overline{b}) \Leftrightarrow (\neg r_1 \vee \neg r_3) \wedge r_4 \tag{17}$$

$$A \prec B :\Leftrightarrow (\overline{a} < \underline{b}) \qquad \Leftrightarrow \neg r_1 \wedge \neg r_2 \wedge \neg r_3 \wedge r_4 \tag{18}$$

$$A \geq B :\Leftrightarrow (\underline{a} \geq \underline{b}) \wedge (\overline{a} \geq \overline{b}) \Leftrightarrow (\neg r_2 \vee \neg r_4) \wedge r_3 \tag{19}$$

$$A \succ B :\Leftrightarrow (\underline{a} > \overline{b}) \qquad \Leftrightarrow \neg r_1 \wedge \neg r_2 \wedge r_3 \wedge \neg r_4 \tag{20}$$

Proof.
(17):

$$\begin{aligned}(\underline{a} \leq \underline{b}) \wedge (\overline{a} \leq \overline{b}) \ &\Leftrightarrow\ ((\underline{a} = \underline{b}) \vee \neg(\underline{a} \geq \underline{b})) \wedge (\overline{a} \leq \overline{b}) \\ &\overset{(6,8,9)}{\Leftrightarrow} (\neg r_1 \vee \neg r_3) \wedge r_4\end{aligned}$$

(18):

$$(\overline{a} < \underline{b}) \;\Leftrightarrow\; (\underline{a} \neq \underline{b}) \wedge (\overline{a} \neq \overline{b}) \wedge (\overline{a} < \underline{b}) \wedge \neg(\underline{a} \geq \underline{b}) \wedge (\overline{a} \leq \overline{b}) \\ \overset{\text{Def. 2}}{\Leftrightarrow} \neg r_1 \wedge \neg r_2 \wedge \neg r_3 \wedge r_4$$

Proof of (19) and (20) analogous to (17) and (18). □

Closely related with comparisons are the lattice operations like interval hull or intersection. They also can exploit the information obtained by one computation of the overlapping relation.

Proposition 3 (Intersection). *The intersection* $A \cap B$ *with* $A = [\underline{a}\,,\,\overline{a}]$, $B = [\underline{b}\,,\,\overline{b}] \in \mathbb{IR}$ *follows from the interval overlapping relation* $A \circledcirc B$ *by*

$$A \cap B := \begin{cases} [\underline{a}\,,\,\overline{a}] & \textit{if } (r_3 \wedge r_4) \\ \emptyset & \textit{otherwise if } (\neg r_1 \wedge \neg r_2) \\ [\underline{a}\,,\,\overline{b}] & \textit{otherwise if } (r_3) \\ [\underline{b}\,,\,\overline{a}] & \textit{otherwise if } (r_4) \\ [\underline{b}\,,\,\overline{b}] & \textit{otherwise} \end{cases} \tag{21}$$

Proof.
if $(r_3 \wedge r_4)$:

$$r_3 \wedge r_4 \overset{(10)}{\Rightarrow} A \subseteq B \\ \Rightarrow A \cap B = [\underline{a}\,,\,\overline{a}]$$

otherwise if $(\neg r_1 \wedge \neg r_2)$:

$$\neg r_1 \wedge \neg r_2 \wedge \neg(r_3 \wedge r_4) \overset{(13),\text{ Cor. 3}}{\Rightarrow} A \cap B = \emptyset$$

otherwise if (r_3):

$$(r_1 \vee r_2) \wedge r_3 \wedge \neg r_4 \overset{(8,9,13)}{\Rightarrow} (\underline{a} \geq \underline{b}) \wedge (\overline{a} > \overline{b}) \wedge (A \cap B \neq \emptyset) \\ \Rightarrow A \cap B = [\underline{a}\,,\,\overline{b}]$$

otherwise if (r_4):

$$(r_1 \vee r_2) \wedge \neg r_3 \wedge r_4 \overset{(8,9,13)}{\Rightarrow} (\underline{a} < \underline{b}) \wedge (\overline{a} \leq \overline{b}) \wedge (A \cap B \neq \emptyset) \\ \Rightarrow A \cap B = [\underline{b}\,,\,\overline{a}]$$

otherwise:

$$(r_1 \vee r_2) \wedge \neg r_3 \wedge \neg r_4 \overset{\text{Cor. 1}}{\Rightarrow} r_1 \wedge r_2 \wedge \neg r_3 \wedge \neg r_4 \\ \overset{(11)}{\Rightarrow} A \supseteq B \\ \Rightarrow A \cap B = [\underline{b}\,,\,\overline{b}]$$

□

4 Hardware Unit

Remark 3. We realize that 8 independent floating-point comparisons are needed. The hardware unit given in Fig. 1 thus consist of 8 comparators. They all work in parallel followed by at most 3 gates to obtain the result.

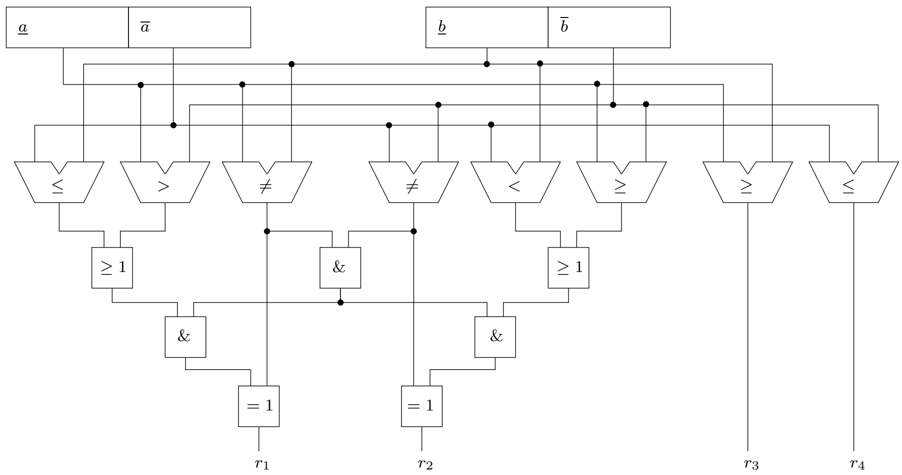


Fig. 1. Logic circuit of the interval overlapping relation

Algorithm 1: INewton (classical)

Input:
f : function
Y : interval
ϵ : epsilon
$yUnique$: flag
$Zero$: list of enclosing intervals
$Info$: flag vector
N : number
Output: $[Zero, Info, N]$
begin

```
if 0 ∉ f(Y) then
    return (Zero,Info,N)
c ← μ(Y);
/* extended division */
[Z1, Z2] ← f(c)/f'(Y);
[Z1, Z2] ← c − [Z1, Z2];
V1 ← Y ∩ Z1;
V2 ← Y ∩ Z2;
if V1 = Y then
    V1 ← [y_, c];
    V2 ← [c, ȳ];
if V1 ≠ ∅ and V2 = ∅ then
    yUnique ← yUnique or V1 ⊂ Y;
foreach i = 1, 2 do
    if Vi = ∅ then
        continue;
    if drel(Vi) < ϵ then
        N = N + 1;
        Zero[N] = Vi;
        Info[N] = yUnique;
    else
        INewton(f, Vi, ϵ, yUnique, Zero, Info, N);
return (Zero,Info,N);
```

Algorithm 2: INewtonRel (relational)

Input:
f : function
Y : interval
ϵ : epsilon
$yUnique$: flag
$Zero$: list of enclosing intervals
$Info$: flag vector
N : number
Output: $[Zero, Info, N]$
begin

```
if 0 ∉ f(Y) then
    return (Zero,Info,N)
c ← μ(Y);
/* extended division */
[Z1, Z2] ← f(c)/f'(Y);
[Z1, Z2] ← c − [Z1, Z2];
R1 ← Y ⚭ Z1;
R2 ← Y ⚭ Z2;
if R1.subseteq() then
    V1 ← [y_, c];
    V2 ← [c, ȳ];
    bisected ← true;
else if not R1.disjoint() and R2.disjoint() then
    yUnique ← yUnique or R1.state() == containedBy;
foreach i = 1, 2 do
    if not bisected then
        if Ri.disjoint() then
            continue;
        Ri ← Y ⚭ Zi;
        Vi ← Ri.intersect();
    if drel(Vi) < ϵ then
        N = N + 1;
        Zero[N] = Vi;
        Info[N] = yUnique;
    else
        INewtonRel(f, Vi, ϵ, yUnique, Zero, Info, N);
return (Zero,Info,N);
```

In this paper we represent the states with 4 bits to have a compact numbering. We then need 8 comparisons to separate all states.

One may argue that specific relations like $A \subseteq B$ or $A = B$ can already be checked by 2 (parallel) floating-point comparisons. But the benefit of our general interval overlapping relation is that, if all 8 comparisons have been computed (in one parallel step), we have enough information to perform dependent interval comparisons only by bit-operations. The same holds for intersection that usually needs 3 floating-point comparisons. See the discussion of the interval Newton method in section 5.

5 Example

As an example for the use of the interval overlapping relation we discuss the extended interval Newton method [2]. In the usual formulation in Alg. 1 up to 6 interval comparisons and 2 intersections are used. We observe, however, that the same intervals are compared several times. Hence, the bitsets R_1 and R_2 gather all information with 2 calls of the interval overlapping relation in Alg. 2.

This reduction of the use of interval comparisons and intersections is done by replacing the 2 intersections by 2 calls of the interval overlapping relation storing the precise information about the relative positions of the interval operands. Then we can deduce all the necessery interval comparisons depending on the results and operands of the replaced intersections by applying the rules of Prop. 1 to the precomputed states R_1 and R_2. That means that the interval comparisons are replaced by bitset operations.

Additionally we introduce a flag *bisected* to determine, if a bisection of the input interval was performed. Otherwise we can use the stored information R_1 and R_2 to catch up the replaced intersection for an recursive call of the algorithm by applying the rules of Prop. 3.

6 Future Work

A companion paper [6] concerning interval comparisons has been submitted to the IEEE interval standard working group P1788. In this paper we emphasize the theoretical influence of the interval overlapping relation as a foundation for interval comparisons. An abstract datatype for the specification of the interval overlapping relation has been introduced. We further want to study its interface in an object oriented environment.

We plan to explore other applications in the area of computer graphics and time scheduling.

The hardware unit will be extended and optimized. Its collaboration with other hardware units for interval arithmetic will be discussed.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26/11/1983:832–843, 1983.

2. R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *C++ Toolbox for Verified Computing I: Basic Numerical Problems.* Springer. Berlin, 1995.
3. IEEE Interval Standard Working Group - P1788. `http://grouper.ieee.org/groups/1788/`.
4. IEEE Task P754. *IEEE 754-2008, Standard for Floating-Point Arithmetic.* IEEE, New York, NY, USA, Aug. 2008.
5. U. Kulisch. *Computer Arithmetic and Validity. Theory, Implementation, and Applications.* de Gruyter Studies in Mathematics 33. Berlin, 2008.
6. M. Nehmeier and J. Wolff von Gudenberg. Interval Comparisons and Lattice Operations based on the Interval Overlapping Relation. Technical Report 476, Institute of Computer Science, University of Würzburg, Sept. 2010.
7. A. Neumaier. Vienna proposal for interval standardization, Version 3.0, Nov. 2008. in [3].