Computing manuscript No.
(will be inserted by the editor)

How to Tell When a Product of Two Partially Ordered Spaces Has a Certain Property

Francisco Zapata · Olga Kosheleva · Karen Villaverde

Received: date / Accepted: date

Abstract In this paper, we describe how checking whether a given property F is true for a product $A_1 \times A_2$ of partially ordered spaces can be reduced to checking several related properties of the original spaces A_i.

This result can be useful in the analysis of properties of intervals $[a, b] \stackrel{\text{def}}{=} \{x : a \le x \le b\}$ over general partially ordered spaces – such as the space of all vectors with component-wise order or the set of all functions with component-wise ordering $f \le g \Leftrightarrow \forall x\,(f(x) \le g(x))$. When we consider sets of pairs of such objects $A_1 \times A_2$, it is natural to define the order on this set in terms of orders in A_1 and A_2 – this is, e.g., how ordering and intervals are defined on the set $\mathbb{R}^2$ of all 2-D vectors.

This result can also be useful in the analysis of ordered spaces describing different degrees of certainty in expert knowledge.

Keywords interval computations · product spaces · partially ordered sets · degrees of certainty

This work was presented at the 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics SCAN'2010 (Lyon, France, September 27–30, 2010).

Francisco Zapata and Olga Kosheleva
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
E-mail: fazg74@gmail.com, olgak@utep.edu

Karen Villaverde
Department of Computer Science
New Mexico State University
Las Cruces, New Mexico 88003, USA
E-mail: kvillave@cs.nmsu.edu

1 Formulation of the Main Problem

Interval uncertainty for numbers, vectors, functions, etc. In many practical situations, we do not know the exact value x of a physical quantity, we only know the lower bound $\underline{x}$ and the upper bound $\overline{x}$. In this case, the only information that we have about the unknown value x is that x belongs to the interval $\{x : \underline{x} \le x \le \overline{x}\}$.

For example, if we have a measurement result $\widetilde{x}$ and an upper bound Δ on the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$, then we can conclude that the actual (unknown) value x belongs to the interval $[\underline{x}, \overline{x}] = [\widetilde{x} - \Delta, \widetilde{x} + \Delta]$; see, e.g., [3, 11, 17].

If we are interested in the values of two different quantities x_1 and x_2, then, to describe the actual values of these two quantities, we need a tuple $x = (x_1, x_2)$. In practice, we usually do not know the exact value of x. Instead, we have a tuple $\underline{x} = (\underline{x}_1, \underline{x}_2)$ which is a "lower bound" for the actual tuple x and a tuple $\overline{x} = (\overline{x}_1, \overline{x}_2)$ which is a "upper bound" for the actual tuple x. This informal description of bounds can be formalized if we introduce a natural component-wise ordering relation between tuples:

$$(x_1, x_2) \le (x_1', x_2') \Leftrightarrow ((x_1 \le x_1') \,\&\, (x_2 \le x_2')).$$

In terms of this ordering, the set of all possible tuples x can also be described as an interval $\{x : \underline{x} \le x \le \overline{x}\}$.

In this case, we started with the ordering relations of two different sets – the set X_1 of possible values of x_1 and the set X_2 of possible values of x_2 – and we defined the corresponding ordering relation on the set $X_1 \times X_2$ of all possible pairs (x_1, x_2). In the above case, X_1 and X_2 were sets of real numbers with usual linear order, but a same construction can be useful in more complex cases as well.

For example, when both x_1 and x_2 are vectors, the ordering relation on each set X_i is a partial order, so we need to analyze the product of partial orders.

When we are interested in the function $f(x)$ – e.g., the function that describes the dependence of one physical quantity on another one – we rarely know the exact function, we usually know some lower and upper bounds $\underline{f}(x)$ and $\overline{f}(x)$. If we consider a pair of functions, or a pair consisting of a function and a number, then we need to define an appropriate ordering relation on the set of all possible pairs.

In the above examples, we had a component-wise order, but in principle, we could have a more complex ordering relation on the product set $X_1 \times X_2$.

Need to analyze properties of products of partially ordered spaces. The above examples show that we need to consider ordering relations on the product $X_1 \times X_2$ of two partially ordered sets. It is therefore desirable to analyze when this new ordering relation satisfies certain property: e.g., when it is linearly ordered, when it is a lattice, etc.

What we do in this paper. In this paper, we provide a general algorithm that reduces the question whether a certain property is satisfied for a product to several properties of component spaces.

2 Other Situations in Which Similar Problems Emerge

Fuzzy modeling and fuzzy techniques: an area of AI where interval methods help. Interval techniques originated from situations in which the interval uncertainty comes from the bounds on measurement errors. However, the same techniques are useful in more general situations. For example, it is known that interval techniques can be very helpful in analyzing expert information, in so-called *fuzzy logic* (see, e.g., [4,15]). In this technique, to describe informal expert statements like "x is small", we assign, to each value x, a number $\mu(x)$ from the interval $[0,1]$ that describes the expert's confidence that this particular value x is small.

From the computational viewpoint, it is often convenient to describe the corresponding function $\mu(x)$ (called *membership function*) by the sets

$$\mathbf{x}(\alpha)=\{x:\mu(x)\ge\alpha\}$$

called α-*cuts*. When we increase α, the α-cut decreases: if $\alpha<\alpha'$, then $\mathbf{x}(\alpha)\supseteq\mathbf{x}(\alpha')$. Thus, in this representation, expert knowledge is described by "nested" sets $\mathbf{x}(\alpha)$ each of which is the set of all the values which are, according to the expert, possible with the degree $\ge\alpha$. In many cases, e.g., for terms like "medium", "approximately 0.3", the expert's degree of confidence first grows with x then decreases; in such situations, each α-cut is an interval.

Operations with expert knowledge can be naturally reformulated in terms of the α-cuts. In particular, we have a problem similar to interval computations: we have expert knowledge about the quantities $x_1,\ldots,x_n$, we know the relation $y=f(x_1,\ldots,x_n)$ between x_i and y, and we want to make conclusions about y. In this case, a usual fuzzy way of finding the membership function for y can be equivalently described in terms of α-cuts as

$$\mathbf{y}(\alpha)=f(\mathbf{x}_1(\alpha),\ldots,\mathbf{x}_n(\alpha))\stackrel{\text{def}}{=}\{f(x_1,\ldots,x_n):x_1\in\mathbf{x}_1(\alpha),\ldots,x_n\in\mathbf{x}_n(\alpha)\}.$$

Thus, processing fuzzy data can be reduced (and is often reduced) to processing interval data – i.e., to solving interval computation problems corresponding to several possible values of α; see, e.g., [2,4,12,13,15].

Since the value α describes the expert's degree of confidence, it is not known with any high accuracy: e.g., hardly anyone can say that his or her degree of confidence is some statement is 0.71 but not 0.72. So, it is sufficient to take only values $\alpha=0$, 0.1, ..., 0.9, 1.0.

Degrees of certainty: from $[0,1]$ to general partially ordered sets. Traditionally, fuzzy logic uses values from the interval $[0,1]$ to describe uncertainty. In this interval, the order is total (linear) in the sense that for every two elements $a,a'\in[0,1]$, either $a\le a'$ or $a'\le a$. Often, partial orders provide a more adequate description of the expert's degree of confidence. For example, since an expert cannot describe her degree of certainty by an exact number, it makes sense to describe this degree by an *interval* $[\underline{d},\overline{d}]$ of possible numbers (see, e.g., [9,14]) – and intervals are only partially ordered; e.g., the intervals $[0.5,0.5]$ and $[0,1]$ are not easy to compare.

More complex sets of possible degrees are also sometimes useful. Not to miss any new options, in this paper, we consider general partially ordered spaces.

Need for product operations. Often, two (or more) experts evaluate a statement S. Then, our certainty in S is described by a pair (a_1, a_2), where $a_i \in A_i$ is the i-th expert's degree of certainty. To compare such pairs, we must therefore define a partial order on the set $A_1 \times A_2$ of all such pairs.

First example of a product operation: Cartesian product. One example of a partial order on $A_1 \times A_2$ is a *Cartesian* product:

$$(a_1, a_2) \le (a_1', a_2') \Leftrightarrow ((a_1 \le a_1') \,\&\, (a_2 \le a_2')).$$

Logical meaning of Cartesian product. This product corresponds to a *cautious* approach, when our confidence in S' is higher than in S if and only if it is higher for both experts.

Second example of a product operation: lexicographic product. Another example is a *lexicographic* product:

$$(a_1, a_2) \le (a_1', a_2') \Leftrightarrow ((a_1 \le a_1') \,\&\, a_1 \ne a_1') \vee ((a_1 = a_1') \,\&\, (a_2 \le a_2'))).$$

Logical meaning of lexicographic product. This product corresponds to the case when we have the absolute confidence in the first expert; then, we only use the opinion of the second expert when, to the first expert, the degrees of certainty are indistinguishable.

We can have other product operations in which the relation between the pairs (a_1, a_2) and (a_1', a_2') is defined in terms of the relations between the elements $a_1, a_1' \in A_1$ and between the elements $a_2, a_2' \in A_2$.

A natural question. Once a product is defined, it is reasonable to ask when the resulting partially ordered set $A_1 \times A_2$ it satisfies a certain property: is it a total order? is it a lattice order? etc. It is desirable to have some criteria that would transform the question about the product space into questions about related properties of component spaces.

Some such criteria are known (see, e.g., [19, 20] and references therein). For example:

- A Cartesian product is a total order if and only if one of the components is a total order, and the other consists of a single element.
- A lexicographic product is a total order if and only if both components are totally ordered.

Applications beyond logic. Similar questions arise in other applications of ordered sets, e.g., in space-time geometry where the causality ordering relation $a \le b$ means that an event a can influence the event b; see, e.g., [1, 5–8, 10, 16, 18].

Applications beyond orders. Our algorithm does not use the fact that the original relations are orders (i.e., transitive antisymmetric relations). Thus, our algorithm is applicable to a general case when we have an arbitrary binary relation – equivalence, similarity, etc. Moreover, this algorithm can be applied to the case when we have a space with *several* binary relations – e.g., an order relation and a similarity relation.

3 Definitions and the Main Result

In the following text, we fix a positive integer m; this integer will be called a *number of binary relations.* Our main case is $m = 1$, when we consider a single binary relation, and this binary relation is an order. However, our result is applicable to an arbitrary finite set of binary relations.

Definition 1. *By a* space*, we mean a set A with m binary relations $P_1(a, a')$, ..., $P_m(a, a')$.*

Clarification. In this definition and in the following definitions, we only consider *crisp* relations – such as an order between the traditional fuzzy degrees of belief, i.e., between the numbers from the interval $[0, 1]$.

Terminological comment. Strictly speaking, a space is thus defined as a tuple $(A, P_1, \ldots, P_m)$. Following the usual mathematical practice, we will, however, usually simplify our notations and simply talk about a space A – implicitly meaning the relations as well.

Definition 2. *By a* first order property *(or simply* property*, for short), we mean a (closed) formula F which is obtained from formulas $P_i(x, x')$ by using logical connectives $\vee$, &, $\neg$, and $\rightarrow$, and quantifiers $\exists x$ and $\forall x$.*

Comment. Most properties in which we may be interested are first order properties. For example, the property to be a total order has the form

$$\forall a \forall a' \, ((a \leq a') \vee (a' \leq a)).$$

The property to be a lattice L means that for every two elements a and a' there is a least upper bound and a greatest lower bound: $L \Leftrightarrow \overline{L} \,\&\, \underline{L}$, where

$$\overline{L} \Leftrightarrow \forall a \forall a' \exists a^+ \, ((a \leq a^+) \,\&\, (a' \leq a^+) \,\&\, \forall a'' (((a \leq a'') \,\&\, (a' \leq a'')) \rightarrow a^+ \leq a'')),$$

and

$$\underline{L} \Leftrightarrow \forall a \forall a' \exists a^- \, ((a^- \leq a) \,\&\, (a^- \leq a') \,\&\, \forall a'' (((a'' \leq a) \,\&\, (a'' \leq a')) \rightarrow a'' \leq a^-)).$$

Notations. When a property F is true for a space X, we will denote it by $F(X)$.

Definition 3. *By a* product operation*, we mean a collection of m propositional formulas that describe the relation $P_i((a_1, a_2), (a'_1, a'_2))$ between the elements $(a_1, a_2), (a'_1, a'_2) \in A_1 \times A_2$ in terms of the relations between the components $a_1, a'_1 \in A_1$ and $a_2, a'_2 \in A_2$ of these elements, i.e., in terms of the relations $P_1(a_1, a'_1)$, ..., $P_m(a_1, a'_1)$, $P_1(a'_1, a_1)$, ..., $P_m(a'_1, a_1)$, $P_1(a_2, a'_2)$, ..., $P_m(a_2, a'_2)$, $P_1(a'_2, a_2)$, ..., $P_m(a'_2, a_2)$.*

Comment. The above formulas that define Cartesian and lexicographic products of partially ordered sets show that these two product operations are examples of product operations in the sense of Definition 3.

Notational comment. For each operation, the space of all the elements is the set of all pairs $A_1 \times A_2$; so, in line with the above terminological comment, we will simply talk about the space $A_1 \times A_2$.

Main result. *There exists an algorithm that, given a product operation and a property F, generates a finite list of properties F_{11}, F_{12}, F_{21}, F_{22}, ..., F_{p1}, F_{p2}, such that*

$$F(A_1 \times A_2) \Leftrightarrow ((F_{11}(A_1) \,\&\, F_{12}(A_2)) \vee \ldots \vee (F_{p1}(A_1) \,\&\, F_{p2}(A_2))).$$

Comment. The above examples of checking when a Cartesian or a lexicographic products are total orders are examples of such equivalences. For example, for the Cartesian product, we have $p = 2$,

- $F_{11}(A_1)$ meaning that A_1 is a total order,
- $F_{12}(A_2)$ meaning that A_2 is a one-element set,
- $F_{21}(A_1)$ meaning that A_1 is a one-element set, and
- $F_{22}(A_2)$ meaning that A_2 is a total order.

Generalizations. As we will see from the proof, a similar algorithm can be formulated for a product of three or more spaces, and for the case when we allow ternary and higher order operations in the definition of a space.

4 Proof

1°. Let us start with the desired property F. This property uses basic relations $P_i(a, a')$ between elements $a, a' \in A_1 \times A_2$ and quantifiers $\forall a$ and $\exists a$ over elements $a \in A_1 \times A_2$.

2°. Every element $a \in A_1 \times A_2$ is, by definition, a pair (a_1, a_2) in which a_1 is an element of the set A_1 and a_2 is an element of the set A_2.

Let us explicitly replace each variable with such a pair.

3°. By definition of a product operation, each relation $P_i(a, a')$ – i.e., each relation $P_i((a_1, a_2), (a'_1, a'_2))$ – can be replaced by a propositional combination of relations between elements $a_1, a'_1 \in A_1$ and between elements $a_2, a'_2 \in A_2$.

Let us perform this replacement.

4°. Each quantifier can also be replaced by two quantifiers corresponding to components:

- $\forall(a_1, a_2)$ is equivalent to $\forall a_1 \forall a_2$, and
- $\exists(a_1, a_2)$ is equivalent to $\exists a_1 \exists a_2$.

Let us perform this replacement as well.

5°. As a result, we get an equivalent reformulation of the original formula F in which elementary formulas are relations between elements of A_1 or between A_2 and quantifiers are over A_1 or over A_2.

We want to reduce this formula to the desired form

$$((F_{11}(A_1) \,\&\, F_{12}(A_2)) \vee \ldots \vee (F_{p1}(A_1) \,\&\, F_{p2}(A_2))). \tag{1}$$

We will reduce this by induction. Elementary formulas are already of the desired form – provided, of course, that we allow free variables.

We will show that if we apply a propositional connective or a quantifier to a formula of this type, then we can reduce the result again to the formula of this type.

6°. When we apply propositional connectives to formulas of type (1), we thus get a propositional combination of the formulas of the type $F_{ij}(A_j)$. It is known that an arbitrary propositional combination can be described in a Disjunctive Normal Form (DNF), i.e., as a disjunction of conjunctions. Each conjunction combines properties related to A_1 and properties related to A_2, i.e., has the form

$$G_1(A_1)\,\&\,\ldots\,\&\,G_p(A_1)\,\&\,G_{p+1}(A_2)\,\&\ldots\&\,G_q(A_2).$$

Thus, each conjunction has the from $G(A_1)\,\&\,G'(A_2)$, where

$$G(A_1) \Leftrightarrow (G_1(A_1)\,\&\,\ldots\,\&\,G_p(A_1))$$

and

$$G'(A_2) \Leftrightarrow (G_{p+1}(A_2)\,\&\ldots\&\,G_q(A_2)).$$

Thus, the disjunction of such properties has the desired form (1).

7°. When we apply an existential quantifier, e.g., $\exists a_1$, then we get a formula

$$\exists a_1\,((F_{11}(A_1)\,\&\,F_{12}(A_2)) \vee \ldots \vee (F_{p1}(A_1)\,\&\,F_{p2}(A_2))).$$

It is known that $\exists a\,(A \vee B)$ is equivalent to $\exists a\,A \vee \exists a\,B$. Thus, the above formula is equivalent to a disjunction

$$\exists a_1\,(F_{11}(A_1)\,\&\,F_{12}(A_2)) \vee \ldots \vee \exists a_1\,(F_{p1}(A_1)\,\&\,F_{p2}(A_2)).$$

If we prove that each term in this disjunction can be transformed into the desired form (1), then, by using the Part 6 of this proof, we will be able to conclude that the entire disjunction has the desired form. Thus, it is sufficient to prove that each formula

$$\exists a_1\,(F_{i1}(A_1)\,\&\,F_{i2}(A_2)) \tag{2}$$

has the desired form. The term $F_{i2}(A_2)$ does not depend on a_1 at all, it is all about elements of A_2. Thus, the formula (2) is equivalent to

$$(\exists a_1\,F_{i1}(A_1))\,\&\,F_{i2}(A_2),$$

i.e., to the formula

$$F'_{i1}(A_1)\,\&\,F_{i2}(A_2),$$

where

$$F'_{i1} \Leftrightarrow \exists a_1\,F_{i1}(A_1)$$

is a formula depending only on the space A_1.

The reduction is proven.

8°. When we apply a universal quantifier, e.g., $\forall a_1$, then we can use the fact that $\forall a_1\,F$ is equivalent to $\neg\exists a_1\,\neg F$. We have assumed that the formula F is of the desired type (1). Thus,

- by using Part 6 of this proof, we can conclude that the formula $\neg F$ can be reduced to the desired type;

– now, by applying Part 7 of this proof, we can conclude that the formula $\exists a_1\,(\neg F)$ can also be reduced to the desired type;
– finally, by using Part 6 again, we conclude that the formula $\neg(\exists a_1\,\neg F)$ can be reduced to the desired type.

9°. By induction, we can now conclude that the original formula can be reduced to the desired type. The main result is proven.

5 Example

To clarify our algorithm, let us apply it to the above simple case of checking whether a Cartesian product is totally ordered. In this case, the formula F that we want to check has the form

$$\forall a \forall a'\,((a \le a') \vee (a' \le a)).$$

According to our algorithm, we first explicitly replace each variable $a, a' \in A_1 \times A_2$ with the corresponding pair. As a result, we get the following formula:

$$\forall (a_1, a_2) \forall (a'_1, a'_2)\,(((a_1, a_2) \le (a'_1, a'_2)) \vee ((a'_1, a'_2) \le (a_1, a_2))).$$

Replacing the ordering relation on the Cartesian product with its definition, we get

$$\forall (a_1, a_2) \forall (a'_1, a'_2)\,((a_1 \le a'_1 \,\&\, a_2 \le a'_2)) \vee ((a'_1 \le a_1 \,\&\, a'_2 \le a_2))).$$

Replacing quantifiers over pairs with individual quantifiers, we get

$$\forall a_1 \forall a_2 \forall a'_1 \forall a'_2\,((a_1 \le a'_1 \,\&\, a_2 \le a'_2)) \vee ((a'_1 \le a_1 \,\&\, a'_2 \le a_2))).$$

By using the relation $\forall \Leftrightarrow \neg\exists\neg$, we get an equivalent form

$$\neg \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,\neg((a_1 \le a'_1 \,\&\, a_2 \le a'_2) \vee (a'_1 \le a_1 \,\&\, a'_2 \le a_2))).$$

Moving negation inside the propositional formula, we get

$$\neg \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,((a_1 \not\le a'_1 \vee a_2 \not\le a'_2) \,\&\, (a'_1 \not\le a_1 \vee a'_2 \le a_2))).$$

The propositional formula

$$(a_1 \not\le a'_1 \vee a_2 \not\le a'_2)) \,\&\, (a'_1 \not\le a_1 \vee a'_2 \not\le a_2)$$

must now be transformed into a DNF form. The result is

$$(a_1 \not\le a'_1 \,\&\, a'_1 \not\le a_1) \vee (a_1 \not\le a'_1 \,\&\, a'_2 \not\le a_2) \vee (a_2 \not\le a'_2 \,\&\, a'_1 \not\le a_1) \vee (a_2 \not\le a'_2 \,\&\, a'_2 \not\le a_2).$$

Thus, the formula

$$\exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,\neg((a_1 \le a'_1 \,\&\, a_2 \le a'_2) \vee (a'_1 \le a_1 \,\&\, a'_2 \le a_2)))$$

is equivalent to

$$F_1 \vee F_2 \vee F_3 \vee F_4,$$

where

$$F_1 \Leftrightarrow \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,(a_1 \not\leq a'_1 \,\&\, a'_1 \not\leq a_1), F_2 \Leftrightarrow \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,(a_1 \not\leq a'_1 \,\&\, a'_2 \not\leq a_2),$$

$$F_3 \Leftrightarrow \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,(a_2 \not\leq a'_2 \,\&\, a'_1 \not\leq a_1), F_4 \Leftrightarrow \exists a_1 \exists a_2 \exists a'_1 \exists a'_2\,(a_2 \not\leq a'_2 \,\&\, a'_2 \not\leq a_2).$$

By applying the quantifiers to the corresponding parts of the formulas, we get

$$F_1 \Leftrightarrow \exists a_1 \exists a'_1\,(a_1 \not\leq a'_1 \,\&\, a'_1 \not\leq a_1), F_2 \Leftrightarrow (\exists a_1 \exists a'_1\, a_1 \not\leq a'_1) \,\&\, (\exists a_2 \exists a'_2\, a'_2 \not\leq a_2),$$

$$F_3 \Leftrightarrow (\exists a_1 \exists a'_1\, a'_1 \not\leq a_1) \,\&\, (\exists a_2 \exists a'_2\, a_2 \not\leq a'_2), F_4 \Leftrightarrow \exists a_2 \exists a'_2\,(a_2 \not\leq a'_2 \,\&\, a'_2 \not\leq a_2).$$

Then, we again reduce

$$\neg(F_1 \vee F_2 \vee F_3 \vee F_4)$$

to DNF.

The result is more complex than the above criterion – because our algorithm does not use the fact that $\leq$ is an order relation.

Acknowledgements The authors are thankful to all the participants of the 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics SCAN'2010 (Lyon, France, September 27–30, 2010), especially to Vladik Kreinovich and Jürgen Wolff von Gudenberg, for valuable discussions.

This work was partially supported by a CONACyT scholarship to F. Zapata.

References

1. Busemann H (1967) Timelike Spaces. PWN, Warszawa
2. Dubois D, Prade H (1978) Operations on fuzzy numbers. International Journal of Systems Science 9:613–626
3. Jaulin L, Kieffer M, Didrit O, Walter E (2001) Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics Springer-Verlag, London
4. Klir G, Yuan B (1995) Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, Upper Saddle River, New Jersey
5. Kosheleva O, Vroegindeweij PG (1998) When is the product of intervals also an interval? Reliable Computing 4(2):179–190
6. Kreinovich V, Kosheleva O (2008) Computational complexity of determining which statements about causality hold in different space-time models. Theoretical Computer Science 405(1–2):50–63
7. Kronheimer EH, Penrose R (1967) On the structure of causal spaces. Proc. Cambr. Phil. Soc. 63(2):481–501
8. Levichev A, Kosheleva O (1998) Intervals in space-time. Reliable Computing 4(1):109–112
9. Mendel JM (2001) Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper Saddle River, New Jersey
10. Misner CW, Thorne KS, Wheeler JA (1973) Gravitation, W. H. Freeman, New York
11. Moore RE, Kearfott RB, Cloud MJ (2009), Introduction to Interval Analysis, SIAM Press, Philadelphia, Pennsylvania
12. Moore RE, Lodwick W (2003) Interval analysis and fuzzy set theory. Fuzzy Sets and Systems 135(1):5–9
13. Nguyen HT, Kreinovich V (1996) Nested intervals and sets: concepts, relations to fuzzy sets, and applications. In: Kearfott RB, Kreinovich V, eds., Applications of Interval Computations, Kluwer, Dordrecht, 245–290
14. Nguyen HT, Kreinovich V, Zuo Q (1997) Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS) 5(3):317–358
15. Nguyen HT, Walker EA (2006) First Course on Fuzzy Logic, CRC Press, Boca Raton, Florida

16. Pimenov RI (1970) Kinematic Spaces: Mathematical Theory of Space-Time, Consultants Bureau, New York
17. Rabinovich S (2005) Measurement Errors and Uncertainties: Theory and Practice, American Institute of Physics, New York
18. Vroegindeweij PG, Kreinovich V, Kosheleva OM (1980) From a connected, partially ordered set of events to a field of time intervals. Foundations of Physics 10(5/6):469–484
19. Zapata F, Kosheleva O, Villaverde K (2010) Product of partially ordered sets (posets) and intervals in such products, with potential applications to uncertainty logic and space-time geometry. Abstracts of the 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics SCAN'2010, Lyon, France, September 27–30, 2010, pp. 142–144
20. Zapata F, Kosheleva O, Villaverde K (2011) Francisco Zapata, Olga Kosheleva, and Karen Villaverde, How to Tell When a Product of Two Partially Ordered Spaces Has a Certain Property: General Results with Application to Fuzzy Logic. Proceedings of the 30th Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2011, El Paso, Texas, March 18–20, 2011.

Appendix: Auxiliary Result

Formulation of the auxiliary result. Let us prove that for partial orders, the only product operations that always leads to a partial order on $A_1 \times A_2$ for which

$$(a_1 \leq_1 a'_1 \,\&\, a_2 \leq_2 a'_2) \rightarrow (a_1, a_2) \leq (a'_1, a'_2)$$

are Cartesian and lexicographic products.

Proof.

1°. According to the definition, whether $(a_1, a_2) \leq (a'_1, a'_2)$ depends on the two relation: the relation between a_1 and a'_1 and on the relation between a_2 and a'_2. For each pair a_i and a'_i, we have four possible relations:

– the relation $a_i <_i a'_i$; we will denote this case by $+$;
– the relation $a'_i <_i a_i$; we will denote this case by $-$;
– the relation $a_i = a'_i$; we will denote this relation by $=$; and
– the relation $a_i \not\leq_i a'_i$ and $a'_i \not\leq_i a_i$; we will denote this relation by $\|$.

The case when we have relation R_1 for a_1 and a'_1 and relation R_2 for a_2 and a'_2 will be denoted by $R_1 R_2$. So, we have 16 possible pairs of relations: $++$, $+-$, $+=$, $+\,\|$, $-+$, $--$, etc. To describe the product, it is sufficient to describe which of these 16 pairs correspond to $(a_1, a_2) \leq (a'_1, a'_2)$.

Due to the consistency requirement, pairs $++$, $+=$, $=+$, and $==$ always result in $\leq$, so it is sufficient to classify the remaining 12 pairs. If only these four pairs result in $\leq$, then we have the Cartesian product. So, to prove our theorem, it is sufficient to prove that if at least one other pair leads to $\leq$, then we get a lexicographic product. To prove this, let us consider the remaining 12 pairs one by one.

2°. Let us first consider pairs that contain $-$.

2.1°. Let us prove that the pair $--$ cannot lead to $\leq$. Indeed, when both A_1 and A_2 are real lines $\mathbb{R}$ with the usual order, due to the fact that $++$ leads to $\leq$, we get $(0,0) \leq (1,1)$, while due to the fact that $--$ leads to $\leq$, we get $(1,1) \leq (0,0)$. Hence, we have $(0,0) \leq (1,1)$ and $(1,1) \leq (0,0)$ but $(0,0) \neq (1,1)$ – a contradiction to antisymmetry.

2.2°. Similarly, the pair $-$ $=$ cannot lead to $\leq$ because otherwise, for the same example $A_1 = A_2 = \mathbb{R}$, we would get $(0,0) \leq (1,0)$ and $(1,0) \leq (0,0)$ but $(0,0) \neq (1,0)$ – also a contradiction to antisymmetry.

2.3°. Let us now consider the pair $- \parallel$.

To prove that it cannot lead to $\leq$, we consider $A_1 = \mathbb{R}$ and $A_2 = \mathbb{R} \times \mathbb{R}$ with Cartesian order. In this case, $(0,0) \parallel_2 (1,-2)$ and $(1,-2) \parallel_2 (-1,-1)$. Thus, if $- \parallel$ leads to $\leq$, we have $(0,(0,0)) \leq (-1,(1,-2))$ and $(-1,(1,-2)) \leq (-2,(-1,-1))$. Thus, due to transitivity of $\leq$, we get $(0,(0,0)) \leq (-2,(-1,-1))$. On the other hand, due to consistency, from $-2 \leq_1 0$ and $(-1,-1) \leq_2 (0,0)$, we conclude that $(-2,(-1,-1)) \leq (0,(0,0))$ – a contradiction with antisymmetry.

2.4°. Similarly, pairs $= -$ and $\parallel -$ cannot lead to $\leq$. Thus, the only pais containing $-$ that can potentially lead to $\leq$ are pairs containing a $+$.

3°. Let us prove a similar property for pairs containing $\parallel$. We already know that pairs $\parallel -$ and $- \parallel$ cannot lead to $\leq$, so it is sufficient to consider pairs $\parallel=$, $=\parallel$, and $\parallel\parallel$.

3.1°. To prove that the pair $=\parallel$ cannot lead to $\leq$, let us consider the same case $A_1 = \mathbb{R}$ and $A_2 = \mathbb{R} \times \mathbb{R}$. In this case, due to $(0,0) \parallel_2 (1,-2)$ and $(1,-2) \parallel_2 (-1,-1)$, if $=\parallel$ leads to $\leq$, we have $(0,(0,0)) \leq (0,(1,-2))$ and $(0,(1,-2)) \leq (0,(-1,-1))$. Thus, due to transitivity of $\leq$, we get

$$(0,(0,0)) \leq (0,(-1,-1)).$$

On the other hand, due to consistency, from $0 \leq_1 0$ and $(-1,-1) \leq_2 (0,0)$, we conclude that $(0,(-1,-1)) \leq (0,(0,0))$ – a contradiction with antisymmetry.

3.2°. Similarly, it is possible to prove that the pair $\parallel=$ cannot lead to $\leq$.

3.3°. To prove that the pair $\parallel\parallel$ cannot lead to $\leq$, let us consider the case when $A_1 = A_2 = \mathbb{R} \times \mathbb{R}$. In this case, due to $(0,0) \parallel_i (1,-2)$ and $(1,-2) \parallel_i (-1,-1)$, if $\parallel\parallel$ leads to $\leq$, we have $((0,0),(0,0)) \leq ((1,-2),(1,-2))$ and $((1,-2),(1,-2)) \leq ((-1,-1),(-1,-1))$. Thus, due to transitivity of $\leq$, we get $((0,0),(0,0)) \leq ((-1,-1),(-1,-1))$. On the other hand, due to consistency, from $(-1,-1) \leq_i (0,0)$, we conclude that $((-1,-1),(-1,-1)) \leq ((0,0),(0,0))$ – a contradiction with antisymmetry.

4°. Thus, due to Part 2 and 3 of this proof, the only additional pairs that can, in principle, lead to $\leq$ are pairs containing $+$, i.e., pairs $+-$, $+\parallel$, $-+$, and $-\parallel$.

5°. Let us prove that the pair $+-$ leads to $\leq$ if and only if the pair $+\parallel$ leads to $\leq$.

5.1°. Let us first prove that if the pair $+-$ leads to $\leq$, then the pair $+\parallel$ also leads to $\leq$.

Indeed, let us consider the case when $A_1 = \mathbb{R}$ and $A_2 = \mathbb{R} \times \mathbb{R}$. If $+-$ leads to $\leq$, then $0 <_1 1$ and $(-1,-1) <_2 (0,0)$ imply $(0,(0,0)) \leq (1,(-1,-1))$. Due to consistency, $1 \leq_1 1$ and $(-1,-1) \leq_2 (-1,1)$ lead to $(1,(-1,-1)) \leq (1,(-1,1))$. Due to transitivity of $\leq$, we get $(0,(0,0)) \leq (1,(-1,1))$. In this case, $\leq$ holds for a pair for which $0 <_1 1$ and $(0,0) \parallel_2 (-1,1)$, i.e., for a pair of type $+\parallel$. By our definition of an order on the product, this means that $\leq$ must hold for all pairs of this type, i.e., that the pair $+ \parallel$ indeed leads to $\leq$.

5.2°. Let us now prove that if the pair $+ \parallel$ leads to $\leq$, then the pair $+-$ also leads to $\leq$.

Let us consider the same case $A_1 = \mathbb{R}$ and $A_2 = \mathbb{R} \times \mathbb{R}$. If $+ \parallel$ leads to $\leq$, then $0 <_1 1$ and and $(1, -2) \parallel_2 (-1, -1)$ imply $(0, (0, 0)) \leq (1, (1, -2))$, and $1 <_1 2$ and $(0, 0) \parallel_2 (1, -2)$ imply and $(1, (1, -2)) \leq (2, (-1, -1))$. Due to transitivity of $\leq$, we get $(0, (0, 0)) \leq (2, (-1, -1))$. In this case, $\leq$ holds for a pair for which $0 <_1 2$ and $(-1, -1) <_2 (0, 0)$, i.e., for a pair of type $+-$. By our definition of an order on the product, this means that $\leq$ must hold for all pairs of this type, i.e., that the pair $+-$ indeed leads to $\leq$.

6°. Similarly, we can prove that the pair $-+$ leads to $\leq$ if and only if the pair $\parallel +$ leads to $\leq$. Thus, adding $+-$ is equivalent to adding $+ \parallel$, and adding $-+$ is equivalent to adding $\parallel +$.

If we add $+-$ (and hence $+ \parallel$), we get the lexicographic product $A_1 \times A_2$. If we add $-+$ (and hence $\parallel +$), we get the lexicographic product $A_2 \times A_1$. Thus, to complete the proof, it is sufficient to show that we cannot simultaneously add $+-$ and $-+$.

7°. Let us prove that $+-$ and $-+$ cannot simultaneously lead to $\leq$.

We will prove this by contradiction. Let us assume that adding both $+-$ and $-+$ always leads to a consistent partial order. In this case, let us take $A_1 = A_2 = \mathbb{R}$. Since $+-$ leads to $\leq$, the conditions $0 <_1 1$ and $-2 <_2 0$ lead to $(0, 0) \leq (1, -2)$. Similarly, since $-+$ leads to $\leq$, from $-1 <_1 1$ and $-2 <_2 -1$,
we conclude that $(1, -2) \leq (-1, -1)$. By transitivity of $\leq$, we can now conclude that $(0, 0) \leq (-1, -1)$. However, due to consistency, $(-1, -1) \leq (0, 0)$ – a contradiction to anti-symmetry.

The statement is proven, and so is the main result of this Appendix.